Vydavateľstvo Univerzity Mateja Bela v Banskej Bystrici
Belianum

Ľuboš Balážovič

Spracovanie a analýzy geopriestorových údajov

Banská Bystrica
2015

Učebný materiál vznikol v rámci projektu „Inovatívne kroky pre potreby vysokoškolského vzdelávania v 21. storočí“ (ITMS: 26110230109) v súlade s Operačným programom Vzdelávanie. Projekt je spolufinancovaný zo zdrojov Európskej únie. Prijímateľom nenávratného finančného príspevku je Univerzita Mateja Bela v Banskej Bystrici.

Názov: Spracovanie a analýzy geopriestorových údajov

Autor: Ľuboš Balážovič

Recenzenti: Mgr. Richard Feciskanin, PhD., Mgr. Michal Vagač, PhD.

Schválené: Vysokoškolské skriptá schválené prorektorom pre rozvoj a informatizáciu UMB 09/2015.

© Univerzita Mateja Bela v Banskej Bystrici, 2015

ISBN 978-80-557-0953-6

Obsah

Predhovor **7**

I Spracovanie údajov **9**

1 Úvod do priestorového modelovania **11**
1.1 Historický kontext 11
1.2 Prístupy k štúdiu priestorových údajov 12

2 Tvorba údajového modelu **15**
2.1 Modelovanie 15
2.2 Metodika tvorby údajového modelu 16
2.2.1 Sémantické modelovanie 17
2.2.2 Konceptuálne modelovanie 19
2.2.3 Logické modelovanie 24
2.2.4 Fyzické modelovanie 27

3 Technologické prostredia **33**
3.1 PostgreSQL a PostGIS 34
3.1.1 Inštalácia 35
3.1.2 Inicializácia databázy 36
3.1.3 Aktivácia databázového systému 38
3.2 QGIS 38
3.2.1 Inštalácia 40
3.2.2 Použitie PostGIS prostredníctvom aplikácie QGIS 40
3.3 GRASS 42
3.3.1 Inštalácia 42
3.3.2 Základné nastavnenie 42

4 Spracovanie údajov v systéme PostgreSQL s rozšírením PostGIS **45**
4.1 Tvorba a odstránenie databázy na úrovni operačného systému . 45
4.2 Konzola psql 46
4.2.1 Tvorba tabuliek 46
4.2.2 Jednoduché vkladanie údajov 47

4.2.3 Aktualizácia údajov 48
4.3 Pokročilý import údajov 49
4.3.1 Všeobecné nástroje 49
4.3.2 Nástroje na import a export údajov 50

II Analýza údajov 55

5 Zdroje údajov pre priestorové analýzy 57
5.1 Získanie priestorových údajov samplovaním 57
5.2 Priestorové interpolácie 58
5.2.1 Dirichletovo delenie 59
5.2.2 Rozklad zhlukov na bunky 60
5.2.3 Triangulácia 60
5.2.4 Metóda IDW (Inverse distance weighting) 60
5.2.5 Kriging 61
5.2.6 Metódy základných radiálnych funkcií 63

6 Filtrovanie cez atribútové dopyty 69
6.1 Jednoduché dopyty 69
6.2 Vnorené dopyty 73
6.3 Spájanie tabuliek 74

7 Kartografické modelovanie 77
7.1 Modelovanie cez tematické mapy 77
7.1.1 Metóda kartogramu 78
7.1.2 Tematická mapa podľa atribútu 79
7.1.3 Metóda kartodiagramu 80
7.2 Vzťahy medzi geometriami 80
7.2.1 Prienik 82
7.2.2 Obsahovanie 83
7.2.3 Prekrývanie 83
7.2.4 Dotýkanie 83
7.2.5 Disjunktný vzťah 84
7.2.6 Zhoda 84
7.3 Tvorba odvodených geometrií 85
7.3.1 Agregačné funkcie 85
7.3.2 Zjednotenie, prienik, rozdiel a symetrický rozdiel 86
7.3.3 Rozdeľovanie línií na menšie úseky 86
7.3.4 Úpravy s geometriami objektov 87
7.3.5 Funkcie nastavenia a prístupu 88
7.3.6 Funkcie miery 88
7.3.7 Kompozičné funkcie 89
7.3.8 Funkcie na zjednodušenie (generalizáciu) 90
7.4 Vzdialenostné analýzy v GIS 90
7.5 Geotagging 91

8 Analýzy nad rastrami **93**
8.1 Základná práca s rastrami 94
8.1.1 Dlaždice 94
8.1.2 Vytváranie rastrov v databáze 95
8.1.3 Priraďovanie hodnôt pixelom 96
8.1.4 Výrezy z rastra 96
8.1.5 Metaúdaje 96
8.2 Analýza blízkeho susedstva 97
8.2.1 Výpočet morfometrických charakteristík reliéfu 97
8.3 Lokálne operácie 99
8.3.1 Mapová algebra 99
8.3.2 Plávajúce okno a fokálna štatistika 101
8.4 Analýza vzdialeného susedstva 101
8.4.1 Analýza viditeľnosti 104

9 Sieťové analýzy **107**
9.1 Topológia 107

10 Štatistické analýzy **111**
10.1 Základné pojmy 111
10.2 Popisná štatistika 112
10.2.1 Početnosť 112
10.2.2 Zonálna štatistika 113
10.2.3 Histogram 114
10.3 Štatistická analýza lineárnej závislosti dvoch javov 115

III Komplexné príklady **117**

11 Návrh údajového modelu pre systém na správu cintorína **119**
11.1 Cieľová funkcionalita 119
11.2 Modelovanie databázy 121
11.2.1 Sémantické a konceptuálne modelovanie 121
11.2.2 Logické modelovanie 125
11.2.3 Fyzické modelovanie 129
11.3 Diskusia 140

12 Priestorové analýzy s využitím DTM – vybrané aplikácie **141**
12.1 Umiestnenie vyhliadok v národnom parku Muránska planina 141
12.2 Potenciálny výskyt nezvestného človeka 143
12.3 Potenciálny výskyt borovice horskej 143

A Schémy **147**

B Výpisy zdrojového kódu **155**
B.1 SSC 155

Zoznam bibliografických odkazov **159**

Predhovor

Rýchly vývoj v oblasti informačných technológií posúva možnosti ich využitia vo vede, výskume aj praktickom živote neustále vpred. Priestorové analýzy dokážu odhaliť skryté štruktúry a zákonitosti v množstve zhromaždených údajov, predpovedať vývoj časopriestorových fenoménov, tvoriť ich modely, experimentovať s ich vstupnými parametrami a hodnotiť odozvy. Pri použití vhodných modelov a nástrojov je možné prepájať údaje z rôznych zdrojov a zdieľať výsledky ďalej tak, aby mohli byť použité pre ďalší výskum a bádanie. Hoci problematika priestorových analýz má svoje historické korene už v geografických prácach v 50. rokoch 20. storočia, až rozvoj štandardizácie ukladania priestorových údajov a manipulácie s nimi nám umožnil komplikované metodiky riešené v rôznych technologických prostrediach výrazne unifikovať. Mnohé úlohy, v ktorých vystupujú geopriestorové údaje je možné riešiť abstrakciou na úrovni formulácie požiadavky v jazyku SQL. To nevyhnutne vedie k potrebe spracúvania geopriestorových údajov v takých údajových štruktúrach, kde je ďalšie modelovanie možné práve na úrovni takejto abstrakcie. Základným objektom výskumu sú geopriestorové informácie, t.j. informácie ktoré majú priradenú určitú polohu.

Problematika, o ktorej hovoríme, je relatívne nová a mnohé štandardy pre spracovanie OGC konzorciom nie sú doteraz úplne implementované. Na Slovensku sa problematike priestorových analýz najpodrobnejšie venoval v svojej monografii *Geografické informačné systémy – priestorové analýzy* Hlásny (2007). Žiadne ďalšie učebnice či monografie takéhoto rozsahu neboli u nás dosiaľ publikované. V svetovej literatúre patrí k najvýznamnejším publikáciám *Spatial data analysis: theory and practice* (Haining, 2003). Praktickú aplikáciu riešení geopriestorových problémov nad databázou PostgreSQL s rozšírením PostGIS rieši publikácia *PostGIS in Action* od autorov Obe, Hsu (2015). Z tejto základnej literatúry vychádzajú aj tieto učebné texty.

V Česku a na Slovensku sa väčšina učebných materiálov venuje buď spracovaniu údajov všeobecne, bez špecializácie na geopriestorové informácie, alebo sa stretneme s učebnicami zameranými na jedno technologické prostredie, nie princíp všeobecne. Práve tento nedostatok a istú medzeru v učebných materiáloch má ambíciu vyplniť táto publikácia. Teoretické poznatky sú tu doplnené praktickými príkladmi riešenými nad jazykom SQL, v prípade niektorých analýz v prostredí QGISu s využitím modulov GRASSu. Učebný text je primárne určený pre vyššie ročníky študentov odboru Geografia, regionálny

rozvoj, avšak môže byť významným pomocníkom aj pre študentov odboru Aplikovaná informatika, Geoinformatika či odborníkov z praxe. Jeho čítanie predpokladá, že čitatelia sú oboznámení na základnej úrovni s problematikou GIS a základmi kartografie, preto niektoré elementárne pojmy týchto vedných oblastí nie sú detailne popisované.

Predkladaný učebný text je členený do troch častí – prvá je zameraná na základné spracovanie geopriestorových údajov formy priestorovej abstrakcie a návrh údajového modelu. Druhá časť sa venuje priestorovým analýzam. Tretiu časť tvoria komplexné príklady doplnené náčrtom metodiky ich riešenia. Pojmy a názvy sú uvádzané *kurzívou*, kód a názvy funkcií a príkazov sú zobrazené `neproporcionálnym písmom`. Vo výpisoch kódu je pre lepšiu orientáciu zvýraznená a farebne odlíšená syntax. Údaje k príkladom sú dostupné cez e-learningový portál Univerzity Mateja Bela v Banskej Bystrici[1].

Učebný materiál vznikol v rámci projektu „*Inovatívne kroky pre potreby vysokoškolského vzdelávania v 21. storočí*“ (ITMS: 26110230109) v súlade s Operačným programom *Vzdelávanie*. Projekt je spolufinancovaný zo zdrojov Európskej únie. Prijímateľom nenávratného finančného príspevku je Univerzita Mateja Bela v Banskej Bystrici.

[1] `http://lms2.umb.sk`

Časť I

Spracovanie údajov

Kapitola 1

Úvod do priestorového modelovania

V tejto kapitole sa oboznámime v širšom kontexte s problematikou priestorových analýz, motiváciou ich vzniku, vývoju a rôznym prístupom k ich realizácii. Umožní nám to lepšie pochopiť postupy spracovania a analýzy priestorových informácií uvádzané v ďalších kapitolách.

1.1 Historický kontext

Hoci problematika priestorových analýz má svoje historické korene už v geografických prácach v 50. rokoch 20. storočia (Berry – Marble, 1968), najviac bola rozvíjaná v rámci geoinformatiky v 80. a 90. rokoch 20. storočia. V GIS softvéri boli implementované analytické metódy z oblasti matematiky a súčasne boli rozpracované rôzne metodiky, údajové štruktúry a modely postavené na priestorových analýzach aplikovateľné v oblasti kriminalistiky, zdravotníctva (šírenie chorôb), dopravy (optimalizácia dopravných systémov a navigácia), životného prostredia, demografie, urbánnej geografie, optimalizácie využívania prírodných zdrojov, skúmania biodiverzity a mnohých ďalších oblastiach. Množstvo údajov a rozmanitosť metodík podnietil v druhej polovici 90. rokov vznik viacerých združení koordinujúcich prácu s priestorovou informáciou tak, aby bola zabezpečená čo najväčšia úroveň interoperability týchto informácií. Dominantný vplyv na ďalší vývoj malo *Open Geospatial konzorcium*[1] (združenie inštitúcií komerčnej aj nekomerčnej sféry zaoberajúcich sa výskumom, vývojom a publikovaním geoinformácií), ktoré v prvom desaťročí 21. storočia vydalo množstvo štandardov a odporúčaní na prácu s priestorovými informáciami. Základnou paradigmou tohto obdobia, ktorá pretrváva dodnes, je možnosť zdieľania priestorových informácií a metód ich spracovania nezávisle od zariadenia (PC, notebook, tablet, smartfón, geodetická stanica...) či údajovej

[1] `http://www.opengeospatial.org`

štruktúry. Na základe týchto odporúčaní boli vyvinuté a upravené aj viaceré databázové systémy na správu údajov schopné štruktúrovanej úschovy priestorových údajov. Tie zároveň obsahujú aj aparát, ktorý zabezpečuje prístup k týmto údajom a štandardizované operácie vykonávané nad nimi. Systém riadi tiež simultánny paralelný prístup k údajom pre viacerých užívateľov.

V súčasnosti existuje niekoľko týchto riešení: databázový systém *Oracle* s rozšírením *Oracle Spatial*, *MSSQL* s podporou pre priestorové informácie, *MySQL* s čiastočnou podporou ukladania geopriestorových informácií a *PostgreSQL* s rozšírením *PostGIS* a *ArcGIS* s rozšírením *SDE* (Spatial Database Engine), ktoré je schopné pracovať nad databázami MSSQL alebo Oracle. Vývojovo najpokročilejšie riešenie vzhľadom na implementáciu OGC štandardov je PostgreSQL s rozšírením PostGIS. Pomocou neho je možné vykonávať priestorové analýzy prostredníctvom SQL jazyka, čoho dôsledkom sú široké možnosti skriptovateľnosti a obsluhy analýzy pomocou jednoduchých tenkých klientov v rôznych prenosných zariadeniach, kým samotné spracovanie údajov prebieha na serveri resp. môže byť tiež spracované v externých štatistických nástrojoch. Analýzy, ktoré v aplikáciách Desktop GIS vyžadovali množstvo krokov, je možné vďaka týmto technológiám vykonať pomocou jedného príkazu.

1.2 Prístupy k štúdiu priestorových údajov

Pojem priestorová analýza sa objavuje často v literatúre o geografických informačných systémoch (GIS) a oblasti geoinformačných vied (v zahraničnej literatúre *Geographic Information Sciences*). Priestorové analýzy môžeme definovať ako súbor techník a modelov skúmajúci výhradne tie údajové hodnoty alebo objekty, ktorým je priradená poloha. Metódy priestorových analýz využívajú údaje popisujúce priestorové vzťahy a vzájomné priestorové pôsobenie študovaných objektov. V prípade, že sa zmení priestorové rozloženie objektov alebo hodnôt, zmenia sa aj výsledky ľubovoľnej priestorovej analýzy študovaného systému (Haining, 1994, s. 45).

Podľa Haining (2003) pozostáva priestorová analýza z troch hlavných oblastí – kartografického modelovania, matematického modelovania a štatistických metód modelovania.

V *kartografickom modelovaní* je každý súbor údajov reprezentovaný mapou a na základe rôznych operácii nad týmito mapami vzniká nová mapa. Napríklad využitím metódy tvorby okolia (bufferu/zóny) môžeme identifikovať všetky oblasti v danej vzdialenosti od vybraného objektu (napríklad hľadáme obce vo vzdialenosti do 5 km od železnice, alebo zisťujeme počet domov do 10 km od objektu nemocnice). Inou metódou je metóda prekrývania máp (Overlay) s využitím aritmetických (+,-,*,/) a logických operátorov (AND, OR, XOR). Logický súčet (operátor OR) je ekvivalentom množinového zjednotenia – zahrnie tak do výslednej mapy oblasti, ktoré sú na jednej alebo na druhej mape. Logický súčin (operátor AND) predstavuje operáciu, ktorá vráti mapu, na ktorej sú územia na oboch zdrojových mapách súčasne. Napríklad tak môžeme

zobraziť územie, na ktorom je les a súčasne je toto územie zákonom chránené (národný park alebo chránená krajinná oblasť). Metódam kartografického modelovania sa venuje kapitola 7.

Pri *matematickom modelovaní* sú výstupy modelu závislé na forme priestorovej interakcie medzi objektami modelu alebo priestorových vzťahoch alebo geografickej polohe objektu v modeli. Napríklad rozloženie tokov a georeliéf krajiny, ktorou pretekajú bude mať vplyv na pohyb vody v rôznych oblastiach povodia. Geografické rozloženie rôznych skupín populácie a rozloženie hustoty obyvateľstva v danom regióne bude mať vplyv na rozšírenie infekcie, zatiaľ čo topografické bariéry budú mať vplyv na kolonizáciu nových druhov.

Priestorová analýza zahŕňa tiež *vývoj a aplikáciu štatistických metód* vhodných pre analýzu priestorových údajov a tiež tých ktorých výsledkom je priestorová referencia údajov.

Cieľom priestorovej analýzy je opísať variácie hodnôt atribútu v priestore záujmového územia. Ďalším krokom by malo byť vysvetlenie priestorovej zákonitosti vzhľadom na iné atribúty. Opis môže zahŕňať identifikáciu zaujímavých vlastností údajov ako je napríklad zhlukovanie v určitých miestach, hľadanie koncentrácie najväčších či najmenších hodnôt. Následne sa snažíme zistiť, pochopiť a vysvetliť, prečo sa v určitých oblastiach na mape koncentrujú vysoké alebo nízke hodnoty. V niektorých oblastiach priestorových analýz (napr. geoštatistika) je cieľom odhadnúť alebo predpovedať hodnoty atribútu na miestach, v ktorých neboli hodnoty zaznamenané alebo je cieľom vytvoriť mapu zo vzoriek údajov.

Príklad 1.1: Majme údaje o výskyte borovice horskej. Bude nás zaujímať v akom prírodnom prostredí sa nachádza a prečo jej výskyt je koncentrovaný práve na tých lokalitách, kde sa nachádza.

Riešenie: Budeme postupovať tak, že vytipujeme niekoľko vlastností prírodného prostredia, ktoré by mohli jej výskyt ovplyvňovať – napríklad nadmorská výška, zrážky, teploty, pôdny typ, príkon slnečného žiarenia. Každú z týchto vlastností bude reprezentovať jedna konkrétna mapa. Mapy následne prekryjeme s mapou reálneho výskytu a tak zistíme rozsah hodnôt jednotlivých atribútov (teploty, úhrnu zrážok, atď.) na tomto území. Výsledkom bude štatistický opis územia, kde by sa borovica horská mohla nachádzať. Následne vieme zostrojiť mapu potenciálneho výskytu borovice horskej a tú validovať terénnym výskumom. Postup k tomuto príkladu sa nachádza v časti 12.3 na str. 143.

Podobné príklady je možné nájsť aj v humánnej geografii, vieme napríklad hľadať súvislosť medzi volebným správaním obyvateľstva a hľadať priestorové zákonitosti tohto správania vzhľadom na iné atribúty.

Kapitola 2

Tvorba údajového modelu

Kapitola nás vovedie do problematiky tvorby údajového modelu – t.j. procesu, ktorý vytvára na základe pozorovanej skutočnosti zjednodušený model reality. Na tomto modeli je možné vykonávať ďalšie analytické úlohy a získať tak cenné informácie, ktoré častokrát nie je možné získať meraniami a pozorovaniami reálneho sveta.

2.1 Modelovanie

Každá udalosť sa deje v nejakom priestore a čase. V klasických experimentálnych prírodných vedách (fyzika, chémia, biológia) môžeme miesto a čas vo väčšine prípadov vynechať – mechanika pohybu funguje v laboratóriu v Bratislave, Pretórii aj na Barbadose rovnako dnes ako aj pred sto rokmi; podobne kryštalizácia chloridu sodného (NaCl) soli bude prebiehať rovnako v prírode, či v experimentálnej reprodukcii v laboratóriu v minulosti i dnes. Vo vedách o Zemi a mnohých sociálnych vedách takéto zjednodušenia s vynechaním času a miesta možné nie sú. Poloha udalosti a javu na zemskom povrchu výrazne určuje jej priebeh a výsledok (príkladom je pohyb vzdušných hmôt, na ktorý má vplyv poloha, vzhľadom na rovník aj nadmorská výška). Silnú priestorovú závislosť nachádzame aj v ekonomickej problematike (poloha zákazníkov vzhľadom na prevádzku, poloha konkurencie).

Vo všeobecnosti postupujeme pri analýze nasledovne:

1. Formulácia problému, cieľa a riešenia
2. Modelovanie (tvorba modelu)
3. Zbieranie a napĺňanie modelu údajmi
4. Návrh dopytu (dopytov)
5. Zobrazenie výsledku

Krok 1 záleží od konkrétneho vedeckého problému – vybrané problémy a ich riešenie uvádzame v tretej časti týchto učebných textov (str. 119). Druhým krokom je modelovanie, ktorým sa budeme zaoberať v nasledujúcej kapitole. Priestorové modelovanie však možno realizovať len keď máme k dispozícii údaje, ktoré obsahujú polohu alebo údaje, pri ktorých máme metódu, ktorou im vieme polohu priradiť (3. krok).

Priestorové údaje je možné získať prirodzene zo zemského povrchu. Zaznamenať možno rôzne údaje, napríklad pH pôdy, hĺbku jazera, výšku snehovej pokrývky alebo zrážky. Údaje môžu mať diskrétnu alebo spojitú povahu. Zemský povrch je spojitý a väčšina údajov na ňom, či nad ním, má preto tiež spojitý charakter. Samotná hodnota atribútu rovnako môže byť spojitá alebo diskrétna. Spojitá bude reprezentovaná číselnou hodnotou (napríklad atribút nadmorská výška bude spojitý), kým diskrétna hodnota nám vytvorí určité triedy objektov (napr. hodnotami atribútu využitie budovy budú „občianska vybavenosť“, „bývanie“, „rekreácia“...). Spojité údaje môžu byť v skutočnosti zaznamenávané pre určité diskrétne bloky – napríklad využitie krajiny bude zaznamenávané pre katastrálne parcely. Hodnota atribútu v tomto prípade reprezentuje celý blok, hoci ten by sa pri podrobnejšom rozbore nemusel javiť ako homogénny.

Priestorové údaje tiež môžeme získať z objektov na zemskom povrchu. Údaje sa tak viažu k určitému bodu alebo plošnej oblasti geografického priestoru. Príkladom takýchto údajov môže byť trvalé bydlisko osoby, pri určitej mierke môže byť bodovo reprezentovaná aj sieť maloobchodných predajní alebo dokonca celé mestá. Atribúty aj v tomto prípade môžu mať spojitú alebo diskrétnu povahu – kvalitatívnu alebo kvantitatívnu. Objekty môžu byť združované do väčších skupín (napr. obce do okresov, okresy do krajov), v tom prípade atribúty reprezentujú jednou hodnotou celú oblasť ako celok. Modelovať s využitím geopriestorových údajov možno v špecializovaných nástrojoch (OpenModeller[1]) alebo GIS aplikáciách (v QGISe nástroj *Graphical modeler* v menu *Processing* – pozri prácu Kudělka et al. 2014).

Štvrtým krokom je návrh (formulácia) dopytu, ktorý získa z nášho modelu potrebnú informáciu. Na tento krok využijeme dopytový jazyk SQL.

Posledným piatym krokom je vizualizácia výsledku vo forme mapy, 3D modelu alebo tabuľky či grafu.

2.2 Metodika tvorby údajového modelu

Pochopenie javov a procesov reálneho sveta nie je možné bez istej miery abstrakcie. Reálny svet nahrádzame jeho modelom, z obrovskej množiny vlastností vyberáme tie, o ktorých predpokladáme alebo vieme, že majú vplyv na problém, ktorý sa pomocou modelu snažíme vyriešiť, snažíme sa identifikovať základné vlastnosti modelu, kvantifikujeme ich a usporadúvame do skupín (tried).

[1] `openmodeller.org`

Modelom rozumieme abstrakciu alebo zjednodušenie reality, ktorá umožní konštruovať testovateľné hypotézy, pomocou ktorých môžeme študovať javy, skutočnosti a procesy, ktorých podstata neumožňuje ich priame štúdium (Turner et al., 2001).

Podľa Mitášovej et al. (Mitášová et al., 1990) model nikdy nedosiahne komplexnosť reálneho systému a snaha o vytvorenie perfektného modelu v zmysle takého zobrazenia, ktoré by s modelovanou realitou bolo izomorfné, spravidla vedie k vytvoreniu neprimerane zložitej modelovacej štruktúry, ktorá sa vyznačuje obťažnou formalizáciou a problematickou ovládateľnosťou.

Pequet (1990) proces abstrakcie v GIS rozdelil do nasledovných krokov:

1. Reálny svet musí byť vyjadrený vo forme údajového modelu
2. Je vybraná konkrétna údajová štruktúra, pomocou ktorej opíšeme údajový model
3. Je vybratý konkrétny údajový formát na určenie údajovej štruktúry

Tieto tri kroky abstrakcie predstavujú z hľadiska teórie údajového modelovania (Simsion – Witt, 2005) tri úrovne modelovania:

1. Konceptuálne modelovanie
2. Logické modelovanie
3. Fyzické modelovanie

Kaluža – Kalužová (2012) rozlišuje ešte nultú úroveň, tzv. sémantické modelovanie, ktoré je v inej literatúre (Simsion – Witt, 2005) súčasťou konceptuálneho modelovania. Priestorové údaje sú podmnožinou údajov, takže je na ne možné použiť koncepciu trojúrovňového modelovania údajov.

2.2.1 Sémantické modelovanie

Cieľom sémantického modelovania je nájdenie typov objektov odrážajúcich modelovanú realitu. Pri ich identifikácii používame tri základné typy abstrakcie – klasifikáciu, agregáciu a generalizáciu (v súlade s Batini et al. (1992)).

Klasifikácia predstavuje zatriedenie entít, ktoré zdieľajú podobnú charakteristiku do jednej triedy – príkladom môžu byť ľudia, ktorí pracujú v jednom podniku a môžeme z nich vytvoriť triedu zamestnanci. Podobne z priestorových entít, ktoré predstavujú cesty (spevnené, nespevnené, diaľnice, miestne komunikácie), môžeme vytvoriť triedu cesty.

Agregácia znamená spájanie entít do väčších celkov, pričom tieto budú tvoriť komponenty tohto celku. Príkladom je zlúčenie (agregácia) entít dátum pozorovania, čas pozorovania, miesto pozorovania, výsledky do väčšieho celku "pozorovanie". Posledným typom abstrakcie je generalizácia – ide o zovšeobecnenie (Pels) jednej alebo viacerých tried – napríklad objekty zamestnanci, zákazníci, partneri možno generalizovať na triedu „ľudia“.

Hodnotenie kvality údajov

Už pri sémantickom modelovaní je potrebné poznať, resp. vedieť hodnotiť kvalitu údajov. Medzi základné kritériá hodnotenia kvality geopriestorových údajov patria podľa Mičietová et al. (2008) presnosť, rozlišovacia schopnosť, konzistentnosť a kompletnosť.

Presnosť (správnosť) je opakom chyby. Pojmom chyba sa označuje rozpor medzi kódovanou a skutočnou hodnotou atribútu danej entity. Všetky údaje sú zbierané s ohľadom na použitý model ich spracovania, čo podmieňuje požadovanú úroveň abstrakcie a generalizácie. Hodnota atribútu je potom hodnota, ktorá vyplýva z úrovne generalizácie prístupu k entitám objektívnej reality. Správnosť je vždy relatívna, pretože je hodnotená vo vzťahu ku konkrétnemu opisu reality. Pri hodnotení vhodnosti alebo nevhodnosti určitých údajov je potrebné súčasne uvažovať aj o limitoch zvoleného prístupu k opisu danej entity. Priestorová presnosť je charakteristika kvality geografických údajov, ktoré vyjadruje priestorový komponent geografickej informácie. Jej hodnotenie závisí od dimenzionality entít, ktoré berieme do úvahy. Pre bodové entity je polohová (priestorová) správnosť definovaná ako vzdialenosť medzi kódovanou polohou a skutočnou polohou. Meranie chyby možno vyjadriť klasickými štatistickými mierami – strednou chybou, strednou kvadratickou chybou, hodnotením limít spoľahlivosti atď. Pre líniové a plošné (areálové) entity je hodnotenie polohovej presnosti komplexnejšie, pretože chyba polohovej presnosti sa skladá z polohovej chyby (chyba v polohe bodov generujúcich líniu) a generalizačnej chyby (chyby vo výbere bodov reprezentujúcich líniu). Chyba polohovej presnosti líniových a areálových entít sa vyjadruje tzv. Epsylon-ovým pásmom, ktoré definuje zónu neurčitosti okolo kódovanej línie, v rámci ktorej existuje skutočná línia. Časová správnosť je zhoda medzi kódovanými a skutočnými časovými súradnicami uvažovanej vlastnosti entity. Časové súradnice vyjadrujú časové ohraničenie, v ktorom je uvažovaná entita platná. Z hľadiska časovej platnosti možno geografické údaje rozdeliť na okamihové a intervalové. Časovú správnosť vyjadruje teda vhodnosť výberu časového okamihu alebo časového intervalu, v ktorom sa merajú hodnoty atribútov vybranej entity.

Rozlišovacia schopnosť údajov vyjadruje množstvo detailov, ktoré rozlíšime v priestore, čase a téme. Rozlišovacia schopnosť je vždy konečná, pretože žiadny systém hodnôt nie je nekonečne presný a modelovanie geografických javov a procesov vychádza z generalizácie pohľadu na geografickú sféru. Rozlišovacia schopnosť súvisí so správnosťou, keďže úroveň rozlíšenia podmieňuje údajovú základňu vo vzťahu, ku ktorej je správnosť hodnotená. Dve databázy s rovnakou úrovňou správnosti, ale rôznou úrovňou rozlišovacej schopnosti, nemajú rovnakú kvalitu. Priestorovú rozlišovaciu schopnosť možno vyjadriť pri rastrovom údajovom modeli ako rozmer jednej bunky rastra. Pre vektorové údaje sa vyjadruje priestorová rozlišovacia schopnosť stanovením minimálnej plochy mapovanej jednotky. Rozlišovaciu schopnosť určuje veľkosť vzorkovanej jednotky. Ak je miera vzorkovania vyššia ako rozlišovacia schopnosť, vzorkovacie jednotky sa prekrývajú. Časová rozlišovacia schopnosť vyjadruje dĺžku (časové trvanie) vzorkového intervalu. Zatiaľ čo časová

rozlišovacia schopnosť vyjadruje dĺžku vzorkovacieho intervalu, miera vzorkovania je frekvencia vzorkovania v čase. Tematická rozlišovacia schopnosť zodpovedá presnosti, resp. kategorizácii určitej témy. Pre kategoriálne údaje vyjadruje tematická schopnosť jemnosť definície kategórií, pre kvantitatívne údaje ju možno chápať analogicky s priestorovou rozlišovacou schopnosťou.

Konzistentnosť údajov môžeme hodnotiť napr. redundantnosťou atribútov, ktorým chýbajú interakcie na časovú a priestorovú dimenziu a nemôžu byť východiskom pre generovanie úplnej geografickej informácie. Priestorová konzistencia zahŕňa aj topologickú konzistenciu priestorových objektov geografickej databázy. Časová konzistentnosť platí vo vzťahu k časovej topológii, kde platí, že na danom mieste v danom čase sa môže vyskytnúť len jedna udalosť.

Kompletnosť (úplnosť) údajov možno vyjadriť dvoma formami. Údajová nekompletnosť je merateľná chyba, jej nesúlad medzi úrovňou generalizácie opisu určitého geografického javu a jej reprezentáciou v databáze. Kompletnosť údajov z hľadiska vhodnosti použitia určitých modelov vyjadruje komplexnosť údajovej základne a jej schopnosť poskytnúť všetky potrebné údaje pre danú aplikáciu. Nekompletnosť môže byť hodnotená z hľadiska atribútov priestoru času a témy.

Výsledkom sémantického modelovania bude zoznam objektov s jednoduchým popisom, ktoré identifikujeme na základe štúdia problematiky, v ktorej ideme vytvárať model. V ďalších štádiách tieto objekty môžu splynúť alebo byť rozdelené.

2.2.2 Konceptuálne modelovanie

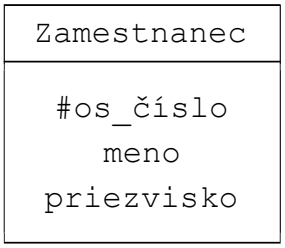
Konceptuálne modelovanie je realizované najčastejšie pomocou E-R (entitno-relačných) diagramov alebo cez diagram tried (metodika UML) (Pender, 2003). Základom je zoznam objektov vytvorený pri sémantickom modelovaní doplnený vzájomnými väzbami (vzťahmi).

Tvorba entitno-relačného diagramu predstavuje určenie

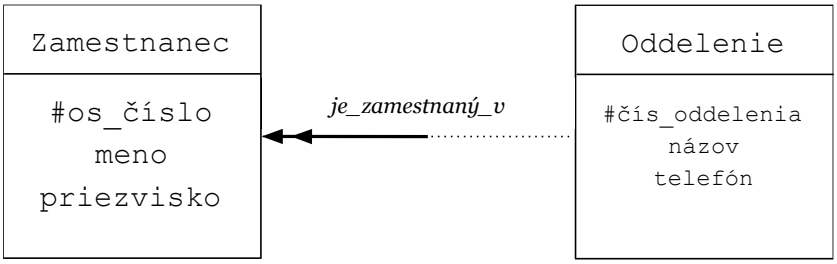
1. entít,
2. vzťahov medzi entitami,
3. atribútov entít,
4. domén hodnôt,
5. kľúčov.

Entita reprezentuje typ objektu reálneho sveta, napríklad *strom, vodojem, človek, výrobok, predajňa, park*... Graficky entitu vyjadrujeme v ER diagrame obdĺžnikom s uvedením mena v hornej časti (obr. 2.1).

Medzi entitami môže existovať vzťah. Rozoznávame dva typy vzťahov – generický a asociatívny.



Obr. 2.1: Entita v grafickom vyjadrení ER diagramu



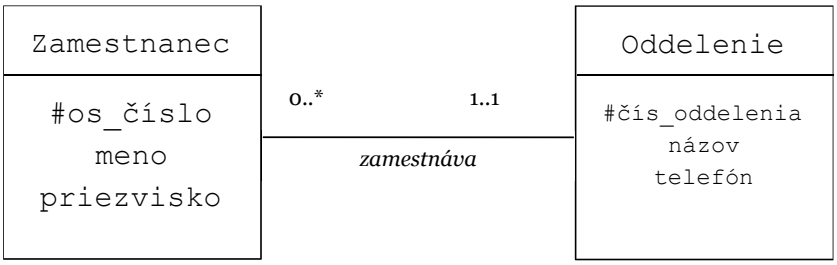
Obr. 2.2: Asociatívny vzťah medzi entitami

Asociatívny vzťah reprezentuje asociáciu jednej alebo viacerých entít, napríklad „strom je súčasťou parku“, „zamestnanec pracuje na oddelení“, „výrobok sa vyskytuje na určitej predajni“. Graficky sa vzťah vyjadruje spojnicou doplnenou slovným popisom (obr. 2.2).

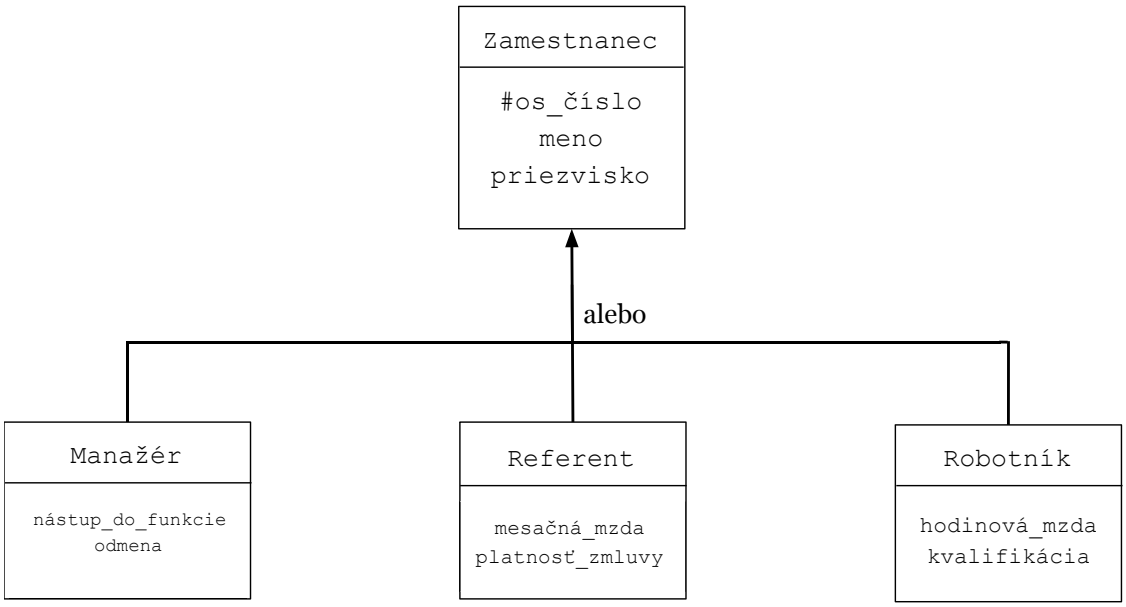
Každý asociatívny vzťah sa vyznačuje stupňom, kardinalitou a voliteľnosťou. Pod stupňom sa chápe počet entít asociovaných v jednom vzťahu. Stupeň 1 predstavuje unárny vzťah, čo znamená, že entita je asociovaná sama so sebou (napr. Zamestnanec riadi zamestnanca, cesta križuje cestu, ...). Binárny vzťah je vzťahom medzi dvoma entitami, terciárny vzťah medzi troma (napr. výrobok je produktom výrobcu a výrobok je doručený k zákazníkovi). Naproti tomu kardinalita vyjadruje počet výskytov entít zúčastnených na jednom výskyte vzťahu. Kardinalita môže nadobúdať hodnotu 1 alebo mnoho (značí sa n alebo m). Existujú tri základné možnosti: $1:1$, $1:n$, $n:m$. Prakticky sa v ER diagrame používa označenie maximálnej a minimálnej kardinality na oboch stranách vzťahu, viď obr. 2.3.

Voliteľnosť vzťahu vyjadruje, či je vzťah povinný alebo voliteľný z jednej alebo druhej strany, t.j. či každému výskytu vzťahu musí alebo iba môže zodpovedať jeden alebo niekoľko výskytov príslušnej entity. Graficky sa vyjadruje takýto vzťah prerušovanou čiarou.

Generický vzťah (generalizácia) – entita E je generalizáciou skupiny entít $E_1, E_2, ..., E_n$ ak každý objekt tejto skupiny entít je zároveň objektom entity E. Entita E tak obsahuje určité spoločné atribúty entít $E_1 - -E_n$. Subtypy môžu byť neprekrývajúcimi sa podmnožinami alebo prekrývajúcimi sa podmnožinami. Ďalšou charakteristikou generického vzťahu môže byť voliteľ-



Obr. 2.3: Kardinalita vzťahu



Obr. 2.4: Generický vzťah

nosť, ktorá určuje, či každý výskyt nadradenej entity musí byť zároveň výskytom niektorého jej subtypu. V ER diagrame vyjadrujeme generický vzťah graficky spojnicou od subtypov k supertypom.

Príkladom na generický vzťah môže byť entita *stĺp* obsahujúca subtypy *stĺp elektrického vedenia*, *stĺp trolejbusového vedenia*, *stĺp verejného osvetlenia* alebo vzťah medzi entitou *zamestnanec* a jej subtypmi *manažér*, *referent*, *robotník*.

Atribút reprezentuje elementárnu vlastnosť entity alebo vzťahu. Napr. entita *zamestnanec* môže obsahovať atribúty: *meno*, *priezvisko*, *dátum narodenia*; entita *parcela* môže obsahovať atribúty *číslo*, *rozloha*, *vlastník*, ... Graficky atribúty v ER diagrame zobrazujeme v spodnej časti entity, viď obr. 2.1. Väčšina atribútov má povahu jednoduchých atribútov, ale existujú aj zložené atribúty tvorené komponentami, napr. atribút adresa môže byť zložený z jednoduchých komponentov PSČ, mesto, ulica, číslo domu.

Doménou hodnôt rozumieme množinu všetkých prípustných hodnôt jedného alebo viacerých atribútov.

Kľúč predstavuje jeden alebo viaceré atribúty, ktoré jednoznačne identifikujú výskyt danej entity. Pokiaľ identifikuje entitu jediný atribút, nazýva sa jednoduchým kľúčom, pokiaľ sa použije na identifikáciu viac atribútov, označujeme ich ako zložený kľúč. Kandidátnym kľúčom podľa definície (Date, 2003) nazývame kľúč entity E, ak spĺňa:

1. Podmienku jedinečnosti, t.j. neexistujú dva výskyty entity E, ktoré majú rovnakú hodnotu kandidátneho kľúča.

2. Podmienku neredukovateľnosti, t.j. ak sa vypustí ktorákoľvek časť kandidátneho kľúča, podmienka 1. prestane platiť.

Primárnym kľúčom entity nazývame kľúč, ktorý sme zvolili spomedzi kandidátnych kľúčov na jednoznačnú identifikáciu výskytov entity. Napr. pri entite *zamestnanec* môžeme mať kandidátne kľúče *osobné číslo* alebo *rodné číslo*, z ktorých jeden atribút zvolíme za primárny kľúč. Grafický primárny kľúč vyznačujeme symbolom mriežka doplneným menom atribútu. Odporúča sa, aby primárny kľúč bol volený podľa nasledujúcich kritérií:

1. Kandidátny kľúč s najmenším počtom atribútov

2. Kandidátny kľúč, pri ktorom je zmena hodnôt najmenej pravdepodobná

3. Kandidátny kľúč, pri ktorom existuje najmenšia pravdepodobnosť, že by v budúcnosti stratil svoju jedinečnosť

4. Kandidátny kľúč s najmenším počtom znakov (pri textových kľúčoch)

5. Kandidátny kľúč s najmenšou maximálnou hodnotou (pri číselných kľúčoch)

6. Kandidátny kľúč, ktorý je možné najľahšie použiť z hľadiska používateľa

Priestorový atribút

Už vo fáze konceptuálneho modelovania prichádzame k otázke atribútu, ktorý by mohol vyjadriť polohu objektu. Ak by sme uvažovali o jednorozmernom konečnom priestore (predstavuje ho úsečka), je možné polohu vyjadriť jedným číslom predstavujúcim vzdialenosť od počiatku. Okrem určenia polohy ľubovoľného bodu jedným celým číslom vieme prirodzene definovať aj vzdialenosť dvoch bodov ako ich vzájomný rozdiel. Dvomi bodmi je jednoznačne určený aj úsek (úsečka patriaca nášmu priestoru). V jednorozmernom (1- dimenzionálnom) súradnicovom systéme môžu existovať prvky s dimenziou 0 (0-rozmerné) a s dimenziou 1 (1-rozmerné prvky). Hoci takáto abstrakcia môže mať svoje praktické využitie (viď napr. Blazek (2005)), napríklad pri jednej trati železnice alebo linkách mestskej hromadnej dopravy, vo všeobecnosti však na opis javov na zemskom povrchu nie je jednorozmerný priestor dostatočný.

Uvažujme o dvojrozmernom priestore. Geometricky ho predstavuje rovina daná dvoma navzájom kolmými osami s počiatkom v ich prieniku. Každému jednému bodu vieme jednoznačne priradiť polohu, ktorá bude daná usporiadanou dvojicou (súradnicou) v karteziánskom súradnicovom systéme. Podobne ako v jednorozmernom priestore, vieme tu zaviesť pojem vzdialenosť bodov definovanú ako:

$$\sqrt{x_2 - x_1, y_2 - y_1} \tag{2.1}$$

V 2D priestore existujú 0-rozmerné prvky (body), 1-rozmerné (línie) a 2-rozmerné (areály). Hoci je tento model najčastejšie využívaný, je len abstrakciou 3D modelu.

V objektívnej realite, v ktorej pozorujeme naše okolie, vnímame svet okolo nás ako 3-rozmerný. Preto je najprirodzenejšie zaznamenať polohu v 3-dimenzionálnom modeli priestoru. Každý bod vieme jednoznačne vyjadriť usporiadanou trojicou (súradnicami) x,y,z. Vzdialenosť definujeme ako

$$\sqrt{(x_2 - x_1)^2 + (y2 - y1)^2 + (z_2 - z_1)^2} \tag{2.2}$$

.

V 3D priestore existujú 0-rozmerné prvky (body), jednorozmerné (línie), dvojrozmerné (areály) a trojrozmerné (povrchy). Okrem zápisu v karteziánskom tvare (x, y, z) sa často využíva aj polárna forma súradníc (φ, λ, q). Viď obr. 2.6.

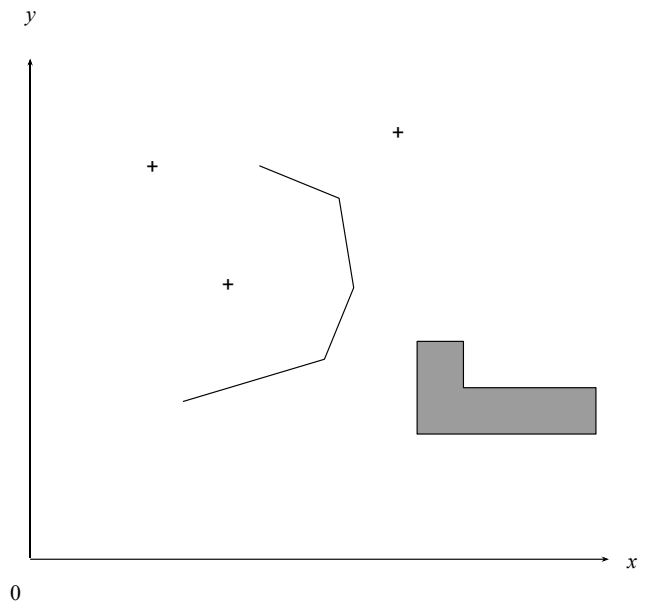
Poloha súradnicového systému vzhľadom na Zem je väčšinou daná nasledovne:

1. os x sa nachádza v rovine rovníka (rovník tvorí prienik roviny kolmej na os otáčania Zeme umiestnenej v rovnakej vzdialenosti od obidvoch pólov).

2. Os y prechádza rovinou danou zemskou osou a prechádzajúcou hvezdárňou v Greenwichi (Londýn), je kolmá na os rotácie a prechádza rovinou rovníka.

3. Os z je totožná so zemskou osou.

Takto definovaný súradnicový systém je síce matematicky korektný, ale z praktického hľadiska len veľmi ťažko použiteľný. V bežnom živote sa pohybujeme po zemskom povrchu a sme zvyknutí udávať aj polohu nie vzhľadom na zemskú os, ale vzhľadom na zemský povrch. Zemský povrch je však veľmi nepravidelný a nie je to dané len samotným georeliéfom, ale tiež rozložením hmoty v zemskom telese. Praktický dôsledok tohto je, že pokiaľ by na Zemi neexistoval reliéf a všetok povrch by bol pokrytý oceánom, získali by sme nepravidelné teleso, ktoré nazývame geoid. Určovanie polohy na geoide je matematicky veľmi zložité a preto ho v praxi nahrádzame jednoduchším telesom – rotačným elipsoidom alebo ešte jednoduchším telesom – guľou. Nepresnosť, ktorá vzniká týmto zjednodušením, sa môže čiastočne zredukovať pre určité územie na Zemi tak, že referenčné teleso posunieme vzhľadom na



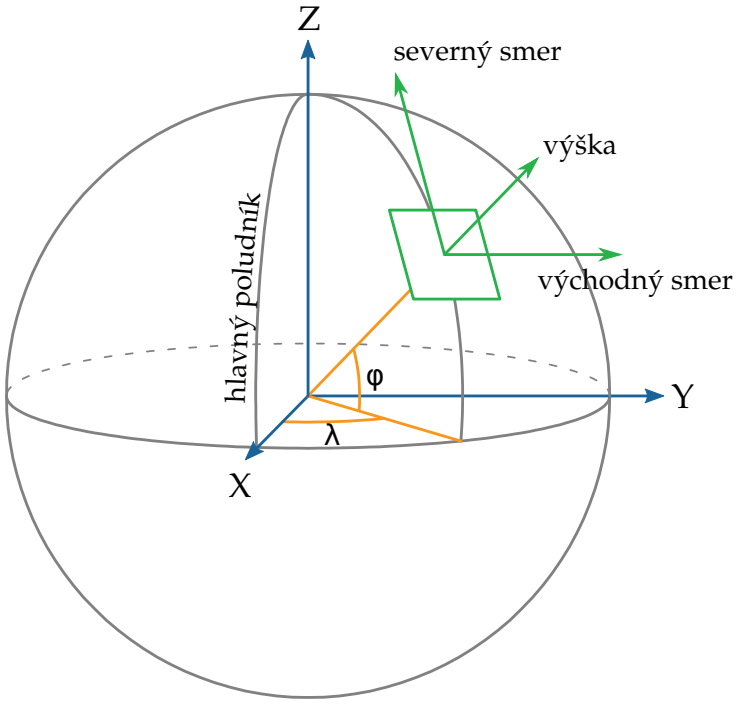
Obr. 2.5: Poloha – 2-D model

osi tak, aby lepšie priľnulo k záujmovému územiu. Objekty na zemskom povrchu premietame na referenčnú plochu danú vybraným referenčným telesom. Polohu udávame polárnymi súradnicami φ, λ, pričom tretia súradnica (veľkosť sprievodiča bodu) je daná a priamo závislá len od referenčného telesa, preto sa v zápise neudáva. Uvedený postup má zmysel pre čo najpresnejšie určenie a zaznamenanie polohy (rôzne geodetické aplikácie). Jeho negatívnou stránkou je pomerne zložité zjednocovanie údajov zaznamenaných v rôzne zvolených súradnicových systémoch (rôzne parametre a poloha referenčného telesa).

2.2.3 Logické modelovanie

Pri logickom modelovaní uprednostníme relačný údajový model. Tento model je v súčasnosti najviac rozšírený a je však potrebné podotknúť, že nie je jediným používaným modelom. Z historického hľadiska treba spomenúť hierarchickú databázovú koncepciu a tiež sieťovú koncepciu. Naproti tomu budúcnosť pravdepodobne výraznejšie ovplyvní objektovo-relačný údajový model.

Relačné údajové modelovanie nadväzuje na pojmy z konceptuálnej úrovne (atribút, doména, kľúč). Dôležitým novým pojmom je relácia, pod ktorou rozumieme dvojrozmernú údajovú štruktúru tvorenú záhlavím a telom relácie.



Obr. 2.6: Poloha – 3-D model

Záhlavie relácie je definované ako množina dvojíc (A_i, D_i), kde atribút A_i je priradený práve jednej doméne D_i pre $i = 1, 2, ..., n$; všetky A_i musia byť vzájomne rôzne. Telo relácie je tvorené množinou n-tíc, ktoré sú množinami dvojíc, kde A_i je i-tý atribút a $v_j i$ je j-tá hodnota z domény D_i pre $j = 1, 2..., m$, kde m je počet n-tíc v množine; m je kardinalitou a n je stupňom relácie. K základným vlastnostiam relácie patrí

1. Neexistencia duplikátov n-tíc
2. Ľubovoľné poradie atribútov
3. Ľubovoľné poradie n-tíc
4. Nerozložiteľnosť hodnôt atribútov

Prvé tri vlastnosti odrážajú fakt, že telo relácie je množinou v matematickom zmysle. Štvrtý prípad hovorí o atomičnosti atribútov, čo znamená, že hodnoty žiadneho atribútu nemožno ďalej rozložiť tak, aby nedošlo ku strate informácií. Atribút teda nemôže obsahovať skupiny údajov alebo opakujúce sa hodnoty. Niekedy v tomto prípade sa hovorí o viachodnotovom atribúte. Relácia, ktorá viachodnotové atribúty neobsahuje, sa nachádza v prvej normálovej forme. Takáto relácia spĺňajúca definičné vlastnosti nie je tabuľkou vo všeobecnosti a z hľadiska presnosti termínov by sa tieto pojmy nemali zamieňať. Okrem základných relácií existujú aj odvodené relácie. Najčastejšie sa v praxi

stretneme s odvodenou reláciou typu pohľad (view), čo je odvodená (virtuálna) relácia, ktorá je definovaná odkazmi na iné existujúce relácie. Pohľad neobsahuje vlastné údaje. Užívateľ, resp. aplikácia tak nie je závislá na fyzickej štruktúre uložených údajov, pristupuje len k údajom, ktoré potrebuje.

V relačnom modelovaní sa používa zápis vo výrokovej forme. Grafické znázornenie je zriedkavé. Relácia v procese modelovania bude reprezentovaná svojím záhlavím v tvare:

$$R(A_1\#, A_2\#, ..., A_m\#, A_{m+1}, ..., A_n) \tag{2.3}$$

Kde A_i je i-tý atribút relácie R, ktorá je stupňa n a ďalej atribúty A_1#, A_2#, ..., A_m# tvorí primárny kľúč relácie s m zložkami.

Tak napríklad relácia oddelenie bude vyjadrená nasledovne:

```
Oddelenie(č_oddelenia#, názov, miesto)
```

Telo relácie R sa symbolicky vyznačí ako $r(R)$. Ak je identifikácia atribútov v rôznych reláciach rovnaká, potom je možné pre rozlíšenie uplatniť kvalifikáciu menom príslušnej relácie. Napríklad atribút názov môže byť kvalifikovaný menom relácie oddelenie:

```
Oddelenie.Názov
```

N-tice t relácie R sa v tejto notácii vyznačia symbolicky takto:

$$T =< h_1, h_2, ...h_n > \tag{2.4}$$

kde h_i je hodnota zodpovedajúca atribútu A_i.

Použitím vyššie uvedenej notácie sa vykoná transformácia konceptuálneho modelu do modelu relačného (logický).

Entita alebo trieda bude transformovaná do jednej relácie vrátane svojich atribútov. Súčasne sa v zápise vyznačí primárny kľúč. Asociatívny vzťah sa modeluje taktiež jednou reláciou, ktorej primárny kľúč bude zložený z primárnych kľúčov entít alebo tried zúčastnených na vzťahu. Atribút alebo atribúty prevzaté z inej relácie do danej relácie sa nazývajú cudzím kľúčom (foreign key), ktorý sa vo formálnom zápise vyznačí ako (CK). V tomto prípade plní cudzí kľúč i úlohu (časti) primárneho kľúča.

Normálové formy

Mnohí autori (Batini, 1992, Connolly, 2008 a ďalší) formulujú dve základné podmienky pre vytvorenie relačného údajového modelu:

1. Požiadavka, aby žiadna hodnota primárneho kľúča nebola prázdna (tzv. `NULL` hodnota); táto podmienka sa nazýva entitná integrita

2. Každá entica odkazujúca sa na inú reláciu (cudzí kľúč) sa musí odkazovať na existujúcu enticu; podmienka sa nazýva referenčnou integritou.

Logický údajový model vytvorený podľa vyššie uvedených zásad nie je konečným riešením. Po priradení atribútov reláciám je potrebné vyriešiť dva nadväzujúce problémy:

1. Viachodnotové atribúty
2. Funkčná závislosť atribútov

Tieto anomálie sa riešia procesom nazývaným normalizácia údajov. Jeho podstatou je postupné rozloženie údajového modelu rozdelením atribútov do väčšieho počtu relácií, ktoré už nevykazujú vyššie uvedené nedostatky. Postupne je množina všetkých relácií prevedená do vyšších normálových foriem.

V prvej normálovej forme (1NF) sa relácia nachádza dodržaním princípu svojej definície.

V druhej normálovej forme (2NF) je relácia práve vtedy, ak je v 1NF a každý atribút, ktorý nie je kľúčom, je plne funkčné závislý na primárnom kľúči relácie.

V tretej normálovej forme je relácia práve vtedy, ak je v 2NF a každý nekľúčový atribút je netranzitívne závislý na primárnom kľúči (t.j. nezávisí na inom atribúte závisiacom na primárnom kľúči).

Okrem 3NF existuje Boyce-Coddova normálová forma (BCNF), ktorá predstavuje prísnejšiu formu tretej normálovej formy. Relácia je v BCNF práve vtedy, ak každý determinant funkčnej závislosti v relácii je zároveň kandidátnym kľúčom.

V praxi stačí transformácia údajov na úroveň BCNF, avšak pre úplnosť uvádzame aj definíciu štvrtej a piatej normálovej formy.

V štvrtej normálovej forme 4NF je relácia práve vtedy, ak je v BCNF a všetky viachodnotové závislosti obsiahnuté v relácii sú zároveň funkčnými závislosťami.

Relácia je v piatej normálovej forme (5NF) pokiaľ je v 4NF a nemôže byť ďalej bezstratovo rozložená.

Podobne by sme mohli eliminovať redundantné priestorové údaje – spoločné hranice medzi polygónmi by boli definované len raz, línia spoločná pre vodný tok a hranicu by taktiež v databáze vystupovala len raz. Tento prístup vedie k zavedeniu tzv. topologického údajového modelu (viď. kapitola 9). V praxi má význam iba pri určitých typoch objektov pri iných by zníženie redundatných údajov viedlo k prílišnej zložitosti celej údajovej štruktúry.

2.2.4 Fyzické modelovanie

Poslednou etapou tvorby údajového modelu je transformácia logického modelu (relačného údajového modelu) do fyzického modelu v konkrétnom prostrední vybraného databázového systému. V ďalšom texte budeme popisovať implementáciu logického údajového modelu v prostredí databázového systému PostgreSQL s rozšírením PostGIS. Komunikácia s databázovým systémom bude prebiehať pomocou jazyka SQL.

Prvým krokom je vytvorenie tabuliek (relácií) s atribútmi navrhnutých na úrovni logického modelu. V jazyku SQL sa na vytvorenie tabuliek používa príkaz `CREATE TABLE` v tvare:

```
CREATE TABLE nazov_tabulky (atribut1 typ_atributu1,
                            atribut2 typ_atributu2,
                            ...
                            atributN typ_atributuN);
```

Typ atribútu čiastočne vymedzuje doménu hodnôt atribútu, avšak na jej úplné vymedzenie je potrebné doplniť databázu o ďalšie obmedzenia (constraints). Dostupné typy atribútu sú:

1. Číselné typy (celočíselné, s plávajúcou desatinnou čiarkou a autoinkrementálne)
2. Peňažný typ
3. Znakové typy
4. Binárny typ
5. Dátumovo-časový typ
6. Boolean typ
7. Typ číselník
8. Ďalšie typy (sieťová adresa, bitstring, textsearch, UUID, XML, JSON, polia, kompozitné typy, rozsahy, OID) využívané v špeciálnych aplikáciách.
9. Geometria

Priestorové atribúty

PostGIS rozoznáva nasledujúce typy pre atribút obsahujúci geometriu:[2]

Geometry – predstavuje geometriu v rovine. Rozmerný karteziánsky súradnicový systém. Je to základný typ a jeho výstup môžeme stotožniť s mapou. Zemský povrch reprezentuje v tomto type jednoduchá plocha.

Geography predstavuje sférický geodetický typ. Body, línie a polygóny sú zobrazené na zakrivenom zemskom povrchu. Výpočty vzdialeností a plôch prebiehajú na neskreslenej referenčnej ploche (elipsoide alebo guli) pred ich priemetom do roviny.

Raster predstavuje mnohokanálový bunkový typ. Rastrový model vyjadruje priestor ako pravidelnú mriežku bodov, pričom každá bunka obsahuje číselné pole hodnôt. Kým predchádzajúce dva typy (`Geometry` a `Geography`)

[2] Všade v ďalšom texte budeme pod typmi geometrie uvažovať o geometrii tak, ako je implementovaná v rozšírení PostGIS. Databázový systém PostgreSQL obsahuje vlastnú implementáciu typov geometrie, tá však nie je vhodná pre ďalšie spracovanie a analýzu údajov. Používa sa len ako údajový sklad.

sú založené na vektorovej algebre, raster predstavuje maticu hodnôt. Rastrové vyjadrenie priestoru sa používa pri digitálnom zachytení obrazu fotoaparátom alebo vizualizácii obrazovej informácie prostredníctvom počítačového monitora. Bunku rastra nazývame pixel. Pixel nemá žiaden fyzický význam. Je to iba vzorka údajov. Každý pixel reprezentuje určitú jednotku plochy z reality.

Topology je relačný typ modelu. Topologický model predstavuje svet ako sieť prepojených uzlov hrán a povrchov. Objekty sú tvorené týmito prvkami a môžu ich vzájomne zdieľať. Topologický model rešpektuje priestorové vzťahy a konceptuálne najlepšie zodpovedá relačnému údajovému modelu. Pri niektorých aplikáciách však nie je nevyhnutný a zbytočne svojou komplexnou štruktúrou zaťažuje modelovanie údajov. Príkladom vhodnej aplikácie sú parcely. Uvažujme, že nastane zmena hranice jednej parcely, napr. parcelu potrebujeme zväčšiť. Zväčšenie jednej parcely automaticky implikuje zmenšenie inej parcely. Pri topologickom údajovom modeli sú jednotlivé parcely poskladané z hrán, ktoré sú zdieľané. Zmena polohy hrany sa tak prejaví na všetkých parcelách, ktoré sú z nej zostavené. Ďalším dobrým príkladom na modelovanie údajov je cestná sieť. Je praktické, aby dve križujúce sa cesty neboli nezávislými objektami, ale obsahovali bod (uzol), ktorý bude spoločný pre obe cesty. Vieme tak jednoducho zistiť napr. najkratšiu cestu alebo riešiť iné úlohy v oblasti navigácie.

Každý údajový typ môže obsahovať svoj modifikátor. Ten slúži na presnejšiu špecifikáciu daného typu. Pomocou neho môžeme vyčleniť subtyp. Príkladom je deklarácia stĺpca obsahujúceho text. Pokiaľ typ atribútu označíme ako `varchar(8)`, znamená to, že stĺpec bude môcť obsahovať text zložený zo znakov. Modifikátor vytvára subtyp textového reťazca, ktorý môže pozostávať z maximálne 8 znakov. Podobne modifikátor typu údajov číslo zapísaný ako `numeric(3,2)`, deklaruje číslo s tromi platnými číslicami s presnosťou na 2 desatinné miesta (t.j. číslo rozsahu -9,99 až 9,99). Podobne údajový typ `geometry` a `geography` obsahuje svoje subtypy, ktoré možno vymedziť typovými modifikátormi. Typový modifikátor určuje typ geometrie a číslo súradnicového systému, v ktorom je geometria vyjadrená. Vzájomné geometrické vzťahy medzi jednotlivými objektami obsahujúcimi geometriu sú štandardne možné len ak používajú jeden a ten istý súradnicový systém. V opačnom prípade je potrebné údaje priestorovo zjednotiť pomocou príslušných nástrojov. Typická deklarácia subtypu v PostGIS má tvar:

```
Geometry(POINT,4326)
```

kde `Geometry` je názov údajového typu, `POINT` je modifikátor subtypu, ktorý určuje, že údaje budú obsahovať body, a `4326` je modifikátor SRID definujúci súradnicový systém, v ktorom sú údaje zaznačené.

Subtypy údajového typu Geometry a Geography

Základnými subtypmi geometrie v type `geometry` sú:

1. Body (subtyp `POINT`, `POINTZ`, `POINTM` a `POINTZM`)

(a) Jednoduchá línia (b) Nejednoduchá línia

Obr. 2.7: Jednoduchá a nejednoduchá línia (invaldiná geometria)

2. Línie (`LINESTRING` a jeho modifikácie analogicky ako pri bodoch)
3. Polygóny (`POLYGON` a jeho modifikácie analogicky ako pri bodoch)
4. Povrchy (`POLYHEDRALSURFACE` a jeho modifikácie)
5. Nepravidelné trojuholníkové siete (`TIN` a jeho modifikácie)
6. Kolekcie geometrií (`MULTIPOLYGON`, `MULTIPOINT`, `MULTILINESTRING`, `GEOMETRYCOLLECTION`)
7. Krivkové geometrie (`CIRCULARSTRINGS`, `COMPOUNDCURVE`, `CURVEPOLYGONS`)

Bodový objekt môžeme vyjadriť nasledovnými subtypmi subtypu `POINT`:

- `POINT` – bod v 2D priestore vyjadrený súradnicami x a y.
- `POINTZ` – bod v 3D priestore vyjadrený súradnicami x, y a z; pozor, súradnica z vyjadruje výšku nad zemským povrchom v rovine, nie je to skutočný 3D priestor Zeme.
- `POINTM` – bod v 2D priestore definovaný súradnicami x, y a meranou hodnotou m, m je bezrozmerným číslom používaným hlavne pre lineárne merania. Môže mať svoje špecifické využitie avšak vo všeobecnosti je možné ho nahradiť iným atribútom.
- `POINTZM` – bod v 3D priestore definovaný súradnicami x, y, z a bezrozmernou hodnotou m.

Ďalším subtypom sú línie. Línie môžu pozostávať z jedného alebo viacerých úsekov (úsek je definovaný dvoma bodmi). Podobne ako body má aj línia štyri varianty, viď vysvetlenie pri bodoch. Línia môže byť aj uzavretá. Príkladom uzavretej línie je štátna hranica. Pokiaľ línia križuje samú seba nazývame ju *nejednoduchou* líniou. Jednoduchá línia nemôže krížiť samú seba.

Okrem základného subtypu `LINESTRING` existujú aj modifikácie `LINESTRINGZ`, `LINESTRINGM`, `LINESTRINGZM`, ktoré majú význam analogický ako pri bodovom objekte.

Subtyp polygón predstavuje uzavretú líniu, ktorej patrí aj plocha ňou zovretá. Najjednoduchší polygón je trojuholník. Ten je tvorený troma nekolineárnymi bodmi. Uzavretá línia tvoriaca hranicu polygónu sa nazýva prstenec polygónu, presnejšie vonkajší prstenec. Polygóny môžu mať aj viacero prstencov, ktoré tak vytvárajú diery. V reálnom svete má takýto model významnú úlohu. Môžeme tak napr. vyjadriť územie štátu obsahujúceho enklávu (napr. Územie Juhoafrickej republiky bez enklávneho územia Kráľovstva Lesotho). Pri polygónoch je vhodné zmieniť sa o koncepte validity. Polygón považujeme za validný, len keď sa jeho vnútorné prstence navzájom nepretínajú, nezdieľajú spoločnú hranicu a neležia mimo vonkajšieho prstenca.

Kolekcie geometrií

PostGIS okrem klasických typov objektov používaných väčšinou GIS aplikácií (bod, línia, polygón) rozoznáva aj tzv. kolekcie geometrií. V praxi sa ukázalo ako veľmi nepraktické, aby napr. Územie štátu pozostávajúce z viacerých oddelených území (napr. Spojené kráľovstvo Veľkej Británie a Severného Írska rozkladajúce sa na viacerých ostrovoch) bolo vyjadrené skupinou nezávislých polygónov. Pri údajnom modelovaní považujeme za entitu štát ako celok. Na vyjadrenie jeho geometrie je potom možné použiť práve subtyp multipolygón. Podobne môžeme definovať aj skupinu línií ako `MULTILINESTRING` a skupinu bodov ako multipoint. Okrem toho je možné vytvoriť aj kolekciu objektov, ktoré nie sú rovnakého typu prostredníctvom subtypu `GEOMETRYCOLLECTION`.

Polygóny v 3D priestore môžu vytvárať pokiaľ majú spoločné hrany polyhedrický povrch. Špeciálnym prípadom polyhedrického povrchu je sieť TIN (nepravidelná triangulačná sieť). Sieť TIN sa skladá len z polygónov, ktoré sú trojuholníkmi. Špecifikácia OGC SQL/MM časť 3 špecifikuje aj prácu s krivkovými geometriami. Vzhľadom na to, že ich implementácia v ďalších analytických nástrojoch je zatiaľ na veľmi nízkej úrovni, podrobnejšie sa s nimi zaoberať nebudeme. Ďalšie informácie o nich možno nájsť v dokumentácii PostGIS a publikácii PostGIS in action (Obe – Hsu, 2015).

Subtyp RASTER

Rastre organizujú informácie s použitím pixelov (buniek). Medzi základné vlastnosti rastra patria:

- šírka a dĺžka rastra (v pixeloch),
- počet kanálov,
- typ pixelov v kanáli (doména hodnôt – jednobitový, 2, 8, 16, 32-bitový so znamienkom/bez znamienka, s plávajúcou desatinnou čiarkou),

- SRID vyjadruje, v ktorom priestorovom referenčnom systéme sú údaje zaznamenané,
- Šírka a výška pixelu
- Mierka pixelu
- Skosenie X a Y

Subtyp Topology
Využitie subtypu `Topology` je popísané v kapitole 9.

Kapitola 3

Technologické prostredia

Koncepty modelovania priestorových údajov uvedené v predchádzajúcich kapitolách je možné previesť do praxe jedine prostredníctvom vhodných softvérových aplikácií. V tejto kapitole si priblížime hlavné úlohy softvérových aplikácií a podrobnejšie sa zoznámime s technologickým prostredím databázového systému PostgreSQL a aplikáciami QGIS a GRASS.

Softvérové aplikácie na spracovanie geografickej informácie musia zabezpečovať nasledujúce úlohy:

- Zber priestorových údajov
- Spracovanie a uloženie priestorových údajov
- Analýza priestorových údajov
- Distribúcia a vizualizácia priestorových údajov

Existuje viacero profesionálnych aplikácií GIS, ktoré sú navrhnuté tak, aby vedeli realizovať všetky štyri úrovne (napr. ESRI ArcGIS, Geomedia, QGIS,...). Množstvo veľmi kvalitných nástrojov existuje aj takých, ktoré riešia len jednu vybranú oblasť. Zber priestorových údajov možno realizovať napr. pomocou GPS zariadení vybavených aplikáciou SurveyPro, Carlson SurvCE. Problematika zberu priestorových údajov na úrovni geodézie a diaľkového prieskumu Zeme presahuje rámec tejto učebnice, preto sa jej podrobnejšie venovať nebudeme.

Spracovanie, ukladanie a manipulácia s údajmi je doménou systémov riadenia bázy dát, označovanými ako databázové systémy. Špeciálne požiadavky vyplývajúce z povahy priestorových údajov majú tieto systémy riešené zväčša prídavnými rozšíreniami. Existuje množstvo pokročilých databázových systémov, ktoré sú schopné pracovať aj s veľkými údajovými skladmi a realizovať požiadavky vkladania a výberu a transformácie údajov veľmi rýchlo. Patria medzi ne: *Oracle* s rozšírením *Oracle Spatial*, *Microsoft SQL Server*, *SpatiaLite*

(vhodný pre veľmi malé databázy), *MySQL* a *PostgreSQL* s rozšírením *PostGIS*. Nie je možné opísať podrobne všetky technológie a tak budeme v ďalších častiach pracovať s databázovým systémom *PostgreSQL* a s priestorovým rozšírením *PostGIS*. Výber softvérového nástroja bol urobený na základe dostupnej funkcionality a licenčnej politiky a spĺňania OGC štandardov. Väčšinu analýz je však možné realizovať podobným spôsobom aj v ostatných uvedených nástrojoch (podrobnosti nájde čitateľ v príslušnej dokumentácii).

Bohatý aparát na analýzu priestorových údajov poskytujú hlavne klasické desktop GIS aplikácie (Proprietárne ArcGIS, MapInfo, ErdasImagine, ENVI, SmallWorld a slobodné GRASS GIS, QGIS, SAGA GIS, gvSIG, Open JUMP). Časť tejto funkcionality bola implementovaná už na úrovni databázového systému ako vrstva manipulácie s údajmi. V budúcnosti možno predpokladať rozšírenie virtuálnych analytických funkcií založených na štandarde WPS, čo prakticky znamená, že analýza bude vykonávaná na serveroch kam cez tenkého klienta alebo DesktopGIS aplikáciu budú zaslané údaje, server ich spracuje a odošle naspäť užívateľovi. V tejto učebnici budeme využívať analytický aparát databázového rozšírenia PostGIS, nástroje aplikácie QGIS a moduly aplikácie GRASS GIS. Opäť platí, že tieto analýzy by bolo možné realizovať aj v iných softvérových prostrediach.

3.1 PostgreSQL a PostGIS

PostgreSQL[1] je objektovo relačný databázový systém využívajúci na komunikáciu jazyk SQL. Je napísaný v jazyku C. Pracuje na väčšine v súčasnosti používaných operačných systémov – systémoch Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) a Windows.

Databázový systém spĺňa vlastnosti ACID v zmysle definície podľa Graya (1993).

PostgreSQL obsahuje podporu pre cudzie kľúče, relácií medzi tabuľkami (prostredníctvom JOIN), tabuľkové pohľady, spúšťače (triggers), možnosť ukla-dania procedúr (vo viacerých jazykoch). Obsahuje väčšinu údajových typov (INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, a TIMESTAMP...) definovaných v norme ISO SQL:2008[2] (PostgreSQL Global Development Group).

PostgreSQL dokáže pracovať aj s veľkými binárnymi objektami ako sú obrázky, zvuky a videá.

Okrem toho dovoľuje PostgreSQL používateľovi definovať nové typy založené na normálnych SQL typoch, čo dovoľuje databáze používať komplexné údaje (tento znak zaraďuje databázový systém medzi „objektovo“ relačné databázové systémy). Napríklad je možné definovať typ *adresa*, ktorý pozostáva z niekoľkých reťazcov, napr. pre číslo ulice, mesto a krajinu. Z tohto pohľadu

[1] `http://www.postgres.org`
[2] ISO/IEC 9075(1-4,9-11,13,14):2008

je možné vytvoriť tabuľku obsahujúcu všetky polia potrebné pre adresu v jednom riadku (Fekiač, 2008; Wikipedia, 2015b).

K databázovému systému existujú rozhrania pre programovanie aplikácií pre jazyky C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC.

Pôvod databázového systému PostgreSQL siaha do prvej polovice osemdesiatych rokov 20. storočia. Systém vznikol ako výstup akademického výskumného projektu na univerzite v Berkey v Kalifornii (USA). Od verzie 5 prešiel projekt na slobodnú licenciu a vývoja sa tak mohla ujať komunita vývojárov z celého sveta. Od verzie 6 nesie projekt názov *PostgreSQL*.

Systém PostgreSQL je šírený pod licenciou Postgre (licencia podobná MIT).

PostGIS[3] je rozšírením objektovo relačnej databázy PostgreSQL pre ukladanie a správu priestorových (geografických) objektov. Umožňuje použitie databázového systému PostgreSQL ako nástroja pre uchovávanie údajov rôzneho GIS softvéru (podobne ako ESRI SDE alebo rozšírenie Oracle's Spatial). PostGIS spĺňa štandard OpenGIS konzorcia *Simple Features Specification for SQL* a je v súlade s jeho profilom *Types and Functions*.

Okrem funkcií podľa štandardu OGC ďalej obsahuje:

- nástroje užívateľského rozhrania,
- základnú podporu topológie,
- schopnosť validácie údajov,
- transformácie súradníc,
- API rozhranie.

V budúcnosti sa očakáva pridanie úplnej podpory pre topológiu, rastre, siete a navigáciu, trojrozmerné povrchy, krivky, splajny a ďalšie.

PostGIS vyvinula firma Refractions Research ako projekt technológie slobodnej priestorovej databázy a je šírený pod licenciou GNU/GPL (Refractions Research, 2015). V súčasnosti jej vývoj zastrešuje nadácia OSGeo[4]. PostGIS nemôže existovať izolovane bez databázového systému PostgreSQL, preto v nasledujúcom texte stručne popíšeme inštaláciu PostgreSQL a pridanie rozšírenia PostGIS.

3.1.1 Inštalácia

Prvým krokom je prevzatie najnovšej verzie (v čase písania tohto textu je to verzia 9.4.4 inštalačného balíčka PostgreSQL) – odporúčame využiť inštalátor od spoločnosti EnterpriseDB zo stránky `www.enterprisedb.com/products/pgdownload.do`. Inštalátor užívateľa prevedie celým procesom inštalácie a umožní mu nainštalovať aj rôzne rozšírenia, medzi ktorými je aj rozšírenie PostGIS. Databázový systém PostgreSQL umožňuje existenciu viacerých databáz

[3] `http://postgis.refractions.net/`
[4] `www.osgeo.org`

na jednom systéme súčasne. Rozšírenie PostGIS sa aktivuje pre každú databázu osobitne.

Po úspešnej inštalácii môžeme vytvoriť databázu cez konzolu PGSQL pomocou príkazu:

```
CREATE DATABASE moja_databaza;
```

Ďalším krokom bude aktivácia rozšírenia PostGIS. Cez konzolu PGSQL sa pripojíme k databáze `moja_databaza` cez nasledujúci príkaz:

```
postgres=# \c moja_databaza;
psql (9.4.4) Type "help" for help. You are now
connected to database "testdb" as user "postgres".
moja_databaza=#
```

Po pripojení k databáze aktivujeme rozšírenie PostGIS cez príkaz

```
CREATE EXTENSION postgis;
```

S databázovým systémom PostgreSQL sa štandardne dodávajú dva nástroje na obsluhu databázy: psql a pgAdmin. Oba nástroje je možné využiť na vytváranie databáz užívateľov a tvorbu dopytov.

PSQL je nástroj, ktorý funguje ako príkazový riadok v textovom režime. Pokiaľ pracujeme na serveri bez grafického užívateľského rozhrania, potom je PSQL jediná možnosť ako obsluhovať databázu.

Pokiaľ máme k dispozícii grafické užívateľské rozhranie je lepšie pre začiatočníka využiť nástroj pgAdmin. PgAdmin je možné nainštalovať už pri inštalácii PostgreSQL zaškrtnutí príslušnej voľby pri výbere voliteľných komponentov. Aktivácia rozšírenia PostGIS cez pgAdmin prebieha kliknutím pravým tlačidlom na položku Rozšírenia (extensions) v stromovom menu a vybratí položky Nové rozšírenie (New Extension) a vybratím položky PostGIS. Položka PostgGis by tam nemala chýbať, pokiaľ inštalácia postgreSQL a PostGIS prebehla bez chýb.

Na záver overíme nainštalovanú verziu príkazom

```
select PostGIS_full_version();

                postgis_full_version
-------------------------------------------------------
 POSTGIS="1.5.3" GEOS="3.3.3-CAPI-1.7.4" PROJ="Rel.
 4.8.0, 6 March 2012" LIBXML="2.7.8" USE_STATS
(1 row)
```

3.1.2 Inicializácia databázy

Pred vytvorením samotnej databázy je potrebné inicializovať databázový systém (len pokiaľ sme administrátor systému a nie je ho užívateľ). Týmto krokom sa v nami určenom adresári vytvoria konfiguračné súbory a pripraví sa

tak prostredie pre vytvorenie samotnej databázy. Na inicializáciu sa využíva program initdb s parametrom $-D$ (za parametrom sa uvádza cesta k adresáru/priečinku, v ktorom budú ukladané údaje a konfiguračné súbory.

Inicializácia by mala prebehnúť nasledujúcim spôsobom

```
D:\initdb.exe -D D:\studenti\pgsql\data

The files belonging to this database system will be
owned by user "student".
This user must also own the server process.

The database cluster will be initialized with locale
Slovak_Slovakia.1250.
The default database encoding has accordingly been set
to WIN1250.
initdb: could not find suitable text search
configuration for locale Slovak_Slovakia.1250
The default text search configuration will be set to
"simple".

fixing permissions on existing directory D:/studenti/
pgsql/data ... ok
creating subdirectories ... ok
selecting default max_connections ... 100
selecting default shared_buffers/max_fsm_pages ... 32MB
/204800
creating configuration files ... ok
creating template1 database in D:/studenti/pgsql/data/
base/1 ... ok
initializing pg_authid ... ok
initializing dependencies ... ok
creating system views ... ok
loading system objects' descriptions ... ok
creating conversions ... ok
creating dictionaries ... ok
setting privileges on built-in objects ... ok
creating information schema ... ok
vacuuming database template1 ... ok r
copying template1 to template0 ... ok
copying template1 to postgres ... ok

WARNING: enabling "trust" authentication for local
connections
You can change this by editing pg_hba.conf or using the
-A option the next time you run initdb.
```

```
Success. You can now start the database server using:

    "postgres" -D "D:/studenti/pgsql/data"
or
    "pg_ctl" -D "D:/studenti/pgsql/data" -l logfile start
```

3.1.3 Aktivácia databázového systému

Na to, aby sme mohli vytvoriť databázu alebo pracovať s databázou už jestvujúcou, je potrebné spustenie databázového systému. Na zapínanie, vypínanie a reštart databázového systému slúži príkaz `pg_ctl`. Cez parameter -D je potrebné uviesť cestu k adresáru určenému pre údaje databázy. Povinným parametrom je akcia, ktorú má `pg_ctl` vykonať. Povolené hodnoty sú start (zapnutie systému), stop (vypnutie), restart (reštart).

Úspešné zapnutie systému by malo vyzerať nasledovne:

```
D:\pg_ctl.exe -D D:\studenti\pgsql\data start
server starting
LOG:  database system is ready to accept connections
LOG:  autovacuum launcher started
```

3.2 QGIS

QGIS[5] (pôvodne označovaný ako Quantum GIS) je geografický informačný systém (GIS) s množstvom štandardných funkcií a vlastností GIS systémov. Je napísaný v jazyku C++ a Python a postavený na knižniciach GDAL/OGR, GEOS, PROJ4. QGIS funguje na operačných systémoch Linux, Windows a Mac. Medzi jeho základné funkcie a vlastnosti patria (kol. autorov, 2015):

- prezeranie a nakladanie vektorových a rastrových údajov v rôznych formátoch a súradnicových systémoch (bez nutnosti konverzie do nejakého interného formátu) – medzi podporované formáty patria:
  - priestorovo rozšírené tabuľky v databázach PostgreSQL (s rozšírením PostGIS) a SQLite (s rozšírením SpatiaLite),
  - väčšina vektorových formátov podporovaných knižnicou OGR (ESRI Shapefile, MapInfo, SDTS, GML . . .)
  - rastrové formáty podporované knižnicou GDAL (rôzne druhy formátov používaných ako digitálne modely reliéfu, letecké a družicové snímky)
  - lokality a mapsety GRASS GIS,
  - priestorové údaje zo služieb WMS a WFS;

[5] `http://qgis.org`

- vytváranie máp a interaktívne prezeranie priestorových údajov prostredníctvom grafického užívateľského rozhrania, medzi ktorého nástroje patria:
  - priame prevody medzi zobrazeniami,
  - skladateľ tlačových výstupov,
  - okno s prehľadom (referenčnou mapou),
  - priestorové záložky,
  - identifikácia/výber objektov,
  - úprava/prezeranie/vyhľadávanie atribútových hodnôt,
  - popisy objektov,
  - vrstva s vektorovými diagramami (napr. vo forme kartodiagramov),
  - zmena vektorovej a rastrovej symboliky,
  - možnosť generovania vrstvy so súradnicovou sieťou,
  - mapové doplnky ako smerová ružica, grafická mierka a označenie autora,
  - ukladanie a načítavanie projektov;
- vytváranie, úprava a export priestorových údajov s využitím:
  - digitalizačných nástrojov pre vektorové formáty a digitalizačných nástrojov GRASSu,
  - modulu na polohové priradenie,
  - nástroje na prácu s GPS (import a export do formátu GPX, konverzia z iných GPS formátov do formátu GPX, alebo priame stiahnutie resp. nahratie údajov z prístroja GPS);
- priestorové analýzy s použitím modulu fTools (pre súbory ESRI Shape) alebo s využitím nástrojov GRASS ako je:
  - mapová algebra,
  - terénne analýzy,
  - hydrologické modelovanie,
  - sieťové analýzy,
  - a iné
- publikáciu máp na internete s využitím exportu do súboru .map (vyžaduje to mať nainštalovaný webový server s Mapserverom)
- možnosť rozšíriť QGIS o ďalšiu funkcionalitu cez zásuvné moduly (napísané v Pythone).

Quantum GIS je v súčasnosti vyvíjaný pod záštitou nadácie OSGeo. Je je šírený pod licenciou GNU/GPL.

3.2.1 Inštalácia

QGIS nainštalujeme spustením inštalačného balíčka prevzatého zo stránky: `http://www.qgis.org/en/site/forusers/download.html`. Inštalácia je jednoduchá, zvládne ju aj začínajúci užívateľ. Pri inštalácii viacerých open source technológií odporúčame inštalovať cez inštalátor OSGEO (`https://trac.osgeo.org/osgeo4w/`).

3.2.2 Použitie PostGIS prostredníctvom aplikácie QGIS

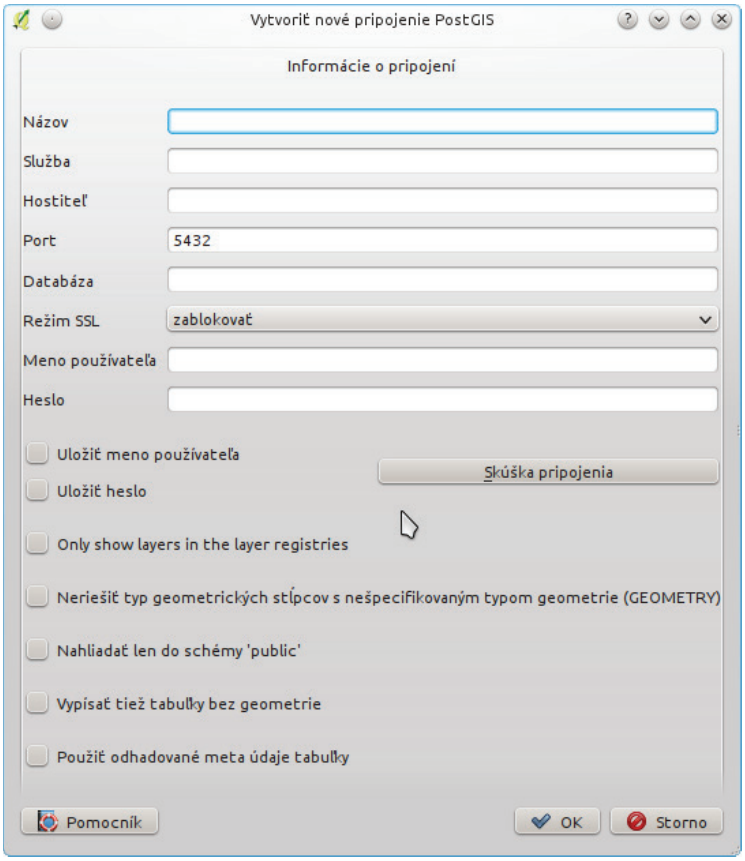
QGIS dokáže natívne pracovať s údajmi uloženými v databáze PostgreSQL s rozšírením PostGIS ak je geometria typu `geometry` (viac kapitola 2.2.4 na str. 28). Podpora geometrií typu `geography` je v súčasnosti v štádiu testovania a preto môže obsahovať aj nejaké chyby. Geometria typu `topology` je podporovaná cez dva externé zásuvné moduly: DB Manager TopoViewer a PostGIS Topology Editor. Oba sú zatiaľ v experimentálnej fáze vývoja. Zobrazovanie údajov typu geometrie raster je možné cez zásuvný modul DB Manager. Na sieťové analýzy možno využiť zásuvný modul *pgRouting*[6], ktorý spolupracuje s rozšírením PG ROUTING pre PostGIS. Všetky zásuvné moduly je možné nainštalovať jednoducho cez inštalátor zásuvných modulov v menu zásuvné moduly (PLUGINS).

Pridanie vrstvy s geometriou v QGISe je veľmi jednoduché. Cez položku v menu *Vrstva->Pridať vrstvu->Pridať vrstvu PostGIS*. V dialógovom okne je najskôr potrebné vytvoriť nové spojenie k databáze kliknutím na tlačidlo *Nové* (obr. 3.1).

Po vyplnení a uložení formulára spojenia k databáze je možné kliknutím na tlačidlo *Pripojiť* (Connect) získať zoznam tabuliek (predstavujúcich vrstvy) z databázy. Kliknutím na tlačidlo *Nastaviť filter* je možné filtrovať objekty zobrazovanej vrstvy prostredníctvom jednoduchých dopytov. Funkčne bohatším nástrojom, v ktorom možno využiť väčšinu funkcionality, ktorú poskytuje PostGIS, je modul DBManager. V ňom je možné písať priamo SQL dopyty. V novších verziách QGISU je tento modul inštalovaný automaticky. Pokiaľ ho nenájdete v menu Databáza, je možné ho doinštalovať cez menu *Zásuvné moduly->Správa zásuvných modulov*. Prostredie zásuvného modulu obsahuje naľavo zoznam všetkých spojení k databázam a všetkých tabuliek v každej jednej databáze. V pravej časti sa nachádza okno s troma záložkami: *Info* – obsahuje základné informácie o databáze, o aktívnej databáze, resp. tabuľke; *Tabuľka* – obsahuje náhľad na aktívnu tabuľku; záložka *Ukážka* obsahuje grafický náhľad/ukážku aktuálnej vrstvy z databázy. V hlavnom menu sa nachádzajú štyri ikony: prvá slúži na aktualizáciu/obnovenie spojenia/vrstvy, druhá otvára SQL okno, v ktorom je možné púšťať SQL dopyty. Pri zadávaní dopytov je potrebné dodržať nasledovné pravidlá:

- Na výstupe môže byť práve jeden stĺpec s geometriou

[6] `http://pgrouting.org/`



Obr. 3.1: Pripojenie k databáze v QGISe

- Ak chceme zobraziť údajový typ GEOGRAPHY alebo raster, je potrebné ich pretypovať na typ GEOMETRY
- Tabuľka musí obsahovať jedinečný kľúč údajového typu INTEGER
- Stĺpec s geometriou musí obsahovať jeden typ geometrie. Stĺpec, ktorý obsahuje v niektorých záznamoch body a v iných napr. polygóny, spôsobí, že vrstvu (tabuľku) nie je možné načítať.

Cez *DBManager* je možné zobrazovať aj topológiu (menu Schéma –> Topoviewer). DBManager vie zabezpečiť aj import a export údajov do databázy (menu Tabuľka). Taktiež je možné vytvárať, mazať a upravovať tabuľky.

3.3 GRASS

GRASS (skratka od Geographic Resources Analysis Support System) GIS je multiplatformný geografický informačný systém (GIS) určený pre správu geopriestorových (2D/3D rastrových a vektorových) údajov, obrazových záznamov, produkciu vysoko kvalitných grafických výstupov, priestorové modelovanie a vizualizáciu údajov.

GRASS GIS (`http://grass.osgeo.org`) je free software publikovaný pod všeobecnou licenciou GNU GPL. Knižnice systému GRASS a jeho nástroje (tzv. moduly) sú z väčšej časti implementované v programovacom jazyku ANSI C. Niekoľko málo modulov je potom implementovaných v programovacom jazyku C++, iné sú dostupné v podobe skrípt v jazyku Python.

Medzi najväčšie výhody systému QGIS patrí veľmi jednoduché a intuitívne grafické užívateľské rozhranie a integrácia modulov systému GRASS. Okrem vybraných modulov, ktoré vyžadujú špeciálne funkcie grafického užívateľského rozhrania systému GRASS je tak možné v aplikácii QGIS naplno využívať väčšinu nástrojov na priestorové analýzy, ktoré ponúka GRASS GIS. Je preto postačujúce na analýzy uvedené v týchto učebných textoch nainštalovať si iba aplikáciu QGIS.

3.3.1 Inštalácia

Pri inštalácii pod operačným systémom Windows odporúčame GRASS inštalovať cez inštalátor OSGEO `https://trac.osgeo.org/osgeo4w/`.

3.3.2 Základné nastavnenie

Na prácu s GRASSom (resp. zásuvným modulom GRASS v QGISe je potrebné porozumieť ako sú v GRASSe organizované údaje. Pred samotnou prácou je potrebné nastaviť *projekt* cez menu *Zásuvné moduly->GRASS->Nový mapset*. Význam jednotlivých nastavení je nasledovný:

GRASSDBASE je priečinok na disku v ktorom bude GRASS ukladať svoje údajové štruktúry (lokality s mapsetmi). Odporúčame na počítači vytvoriť

priečinok GRASSDATA na ktorý nasmerujeme potom GRASSDBASE nastavenie. Do tohto priečinku nijako nezasahujeme – celú jeho správu zabezpečuje samotný GRASS GIS

LOCATION (lokalita) predstavuje názov celého územia projektu. Môžeme ho pomenovať napr. Slovensko, alebo BB. Pri názvoch lokalít a mapsetvo nepoužívame diakritiku a interpunkciu.

MAPSET je „súbor“ máp jedného užívateľa. Je to akýsi priečinok v rámci lokality. V jednej *LOCATION* môže existovať niekoľko rôznych mapsetov. Okrem uživateľských mapsetov existuje v lokalite vždy mapset *PERMANENT*

REGION predstavuje pracovný výrez územia. Akékoľvek analýzy sa budú uskutočňovať len na tomto výreze. Súčasťou nastavenia regiónu je aj rozlíšenie regiónu (*RESOLUTION*).

RESOLUTION rozlíšenie – je charakteristické pre vybraný región. Vyjadruje veľkosť jednej bunky (resp. veľkosť výsledného rastra).

Kapitola 4

Spracovanie údajov v systéme PostgreSQL s rozšírením PostGIS

Po sémantickom konceptuálnom a logickom modelovaní opísanom v kapitole 2 nastupuje fyzická implementácia údajového modelu, nad ktorou budú prebiehať ďalšie priestorové analýzy. Zdrojom údajov častokrát nie sú tabuľky vo forme, ktorá je stopercentne komformná s našim údajovým modelom. Môžeme sa stretnúť s údajmi v rôznych formátoch a v rôznych tabuľkách. V tejto kapitole sa budeme venovať problematike ich importu, zjednotenia a ich prevedenia do požadovanej formy.

V tejto kapitole sa oboznámime najskôr s jednoduchým vytvorením databázy a vložením niekoľkých záznamov. Druhá časť sa zaoberá pokročilými technikami napĺňania databázy.

4.1 Tvorba a odstránenie databázy na úrovni operačného systému

Novú databázu vytvoríme príkazom `createdb`. Ako parameter treba uviesť názov novej databázy. Vytvorenie databázy test prebehne nasledovne:

```
D:\createdb.exe test
```

Databázu odstraňujeme príkazom `dropdb`. Syntax príkazu je rovnaká ako pri `createdb`.

4.2 Konzola psql

Pracovať s databázou pomocou jazyka SQL možno viacerými spôsobmi. Najjednoduchšie je použiť textovú konzolu psql. Parametrom je názov databázy, ku ktorej sa pripája. Okrem príkazov jazyka SQL možno využiť aj interné príkazy konzoly, ktoré sa zadávajú cez spätnú lomku. Ich zoznam získame cez príkaz `\h`. Nasledujúci listing zobrazuje obrazovku konzoly po prihlásení do databázy `test` a následné ukončenie práce s konzolou pomocou príkazu `\q`

```
D:\studenti\pgsql\bin>psql.exe test
Welcome to psql 8.2.6, the PostgreSQL interactive termi
nal.

Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help with psql commands
      \g or terminate with semicolon to execute query
      \q to quit

Warning: Console code page (852) differs from Windows
code page (1250) 8-bit characters might not
work correctly. See psql reference page "Notes
for Windows users" for details.
test=#
test=#\q
```

Z prostredia konzoly psql je možné aj vytvárať nové databázy (podobne ako na úrovni operačného systému). Využijeme na to SQL príkaz `CREATE DATABASE` v tvare:

```
CREATE DATABASE nazov_databazy
```

Odstrániť databázu môžeme príkazom DROP v tvare:

```
DROP DATABASE nazov_databazy
```

4.2.1 Tvorba tabuliek

Na vytvorenie novej tabuľky sa využíva v jazyku SQL príkaz CREATE TABLE. Tabuľky môžeme organizovať do akýchsi priečinkov, ktoré sa nazývajú schémy (na rozdiel od priečinkov nemajú hlbšiu hierarchickú štruktúru – t.j. v schéme sa nachádzajú iba tabuľky a pohľady, nie je možné vnoriť jednu schému do druhej). Pokiaľ nešpecifikujeme schému, v ktorej ideme vytvárať tabuľku, použije sa schéma `public`. Odkazovanie na tabuľku v inej schéme sa zapisuje v tvare `nazov_schemy.nazov_tabulky`. Odkaz na tabuľku `zamestnanci` v schéme `Agenda_firmy` zapíšeme v tvare `Agenda_firmy.zamestnanci`.

Nasledujúci príkaz vytvorí novú tabuľku `zamestnanci`, ktorá bude obsahovať stĺpce `id`, `meno`, `priezvisko`. Tabuľka bude vytvorená v schéme `public`. Za stĺpcom sa uvádza typ údajov – `int4` je 4 bytový integer (celé číslo), `varchar(cislo)` je reťazec s premennou dĺžkou nepresahujúcou dĺžku uvedenú v zátvorke (počet znakov):

```
test=# CREATE TABLE zamestnanci (id int4,
                                 meno varchar(50),
                                 priezvisko varchar(50));
CREATE TABLE
test=#
```

Príkazy jazyka SQL by sa mali písať veľkými písmenami (nie je to povinné, ale je to zaužívaná konvencia). Každý príkaz sa vykoná len ak je ukončený bodkočiarkou. V opačnom prípade sa očakáva ďalší vstup.

4.2.2 Jednoduché vkladanie údajov

Údaje do tabuliek sa vkladajú príkazom `INSERT INTO` v tvare:

```
INSERT INTO nazov_tabulky VALUES ('hodnota_atributu_1',
                                  'hodnota_atributu_2',
                                  'hodnota_atributu3')
```

Číselné hodnoty sa nemusia uzatvárať do apostrofov, všetky ostatné hodnoty však áno.

Príklad 4.1:

```
test=# INSERT INTO zamestnanci
      VALUES ('1','Adam','Šangala');
INSERT 0 1
test=# INSERT INTO zamestnanci
      VALUES ('2','Maco','Mlieč');
INSERT 0 1
test=# INSERT INTO zamestnanci
      VALUES ('3','Juraj','Jánošík');
INSERT 0 1
```

Pri vkladaní viacerých záznamov je možné ich vložiť naraz tak, že za kľúčové slovo `VALUES` umiestňujeme jednotlivé záznamy a navzájom ich oddelíme čiarkou.

Príklad 4.2:

```
INSERT INTO vedenie_krajov (kraj, predseda)
VALUES
('Banskobystrický', 'Marián Kotleba'),
('Žilinský','Juraj Blanár');
```

Tabuľku je možné napĺňať aj hodnotami získanými dopytom z inej tabuľky:

Príklad 4.3:

```
INSERT INTO dialnice (id, geom, oznacenie)
  SELECT id, geom, označenie
  FROM cesty WHERE typ='dialnica'
```

Veľmi často sa stretneme s alternatívou, pri ktorej vytvárame úplne novú tabuľku na základe dopytu z existujúcej tabuľky. Existujú dva spôsoby, ako to vytvoriť:

```
SELECT stlpec1, stlpec2
   INTO nova_tabulka FROM povodna tabulka;

CREATE TABLE nova_tabulka AS
   SELECT stlpec1, stlpec2 FROM povodna_tabulka;
```

Prvý spôsob je univerzálnejší a mal by fungovať v každom databázovom systéme.

4.2.3 Aktualizácia údajov

Na aktualizáciu záznamov v tabuľke slúži príkaz UPDATE. Používa sa v tvare:

```
UPDATE tabul'ka SET stĺpec = nová hodnota WHERE podmienka
```

Nasledujúci príklad aktualizuje stĺpec hustota obyvateľstva v tabuľke okresy, pričom vloží do neho hodnotu, ktorú získa ako pomer hodnôt zo stĺpcov obyvateľstvo a rozloha.

Príklad 4.4:

```
UPDATE okresy
  SET hustota=obyvatelstvo/rozloha
```

Pri aktualizácii záznamov s použitím príkazu UPDATE je možné použiť aj iný dopyt. Napríklad nasledujúci dopyt priradí jednotlivé záznamy bodovej mapy do príslušného okresu.

Príklad 4.5:

```
UPDATE bodova_vrstva SET okres=okres.nazov
  FROM okresy AS okres
  WHERE ST_Intersects(bodova_vrstva.GEOM,okres.GEOM);
```

Podobne je možné využiť aj vnorené dopyty. Nasledujúci príklad spočíta obce v každom kraji a túto informáciu vloží do ďalšieho stĺpca v tabuľke kraje.

```
UPDATE kraje
  SET pocet=okresy.pocet
  FROM
    (
      SELECT kraj,
             COUNT(GID) AS pocet
      FROM okresy AS okr GROUP BY okr.kraj
    )
       AS okresy
  WHERE kraje.nazov=okresy.kraj;
```

4.3 Pokročilý import údajov

4.3.1 Všeobecné nástroje

Psql je textová konzola na obsluhu databázy PostgreSQL. Pracuje v dvoch režimoch:

- Interaktívny režim – PSQL vykonáva príkazy hneď po napísaní v príkazovom riadku. Na načítanie údajov vo forme csv súboru[1] do databázy PostrgreSQL je možné použiť príkaz `\copy` (viď príklad nižšie)
- Neinteraktívny režim – je možné pripraviť skripty a vykonať ich prostredníctvom psql.

Príklad 4.6: Majme súbory o nezamestnanosti v tabuľke, ktorú uložíme vo formáte CSV. V našej databáze chceme vytvoriť tabuľku, ktorá bude obsahovať všetky údaje z našej CSV tabuľky:

```
test=#\copy zamestnanci FROM '~/zamestnanci.csv'
WITH CSV;
```

Psql je možné spustiť s prepínačmi určujúcimi názov servera (-h), užívateľa (-u) a databázy (-d).

Pgadmin III je grafické rozhranie v databáze PostgreSQL na rozdiel od `psql` nie je možné použiť príkaz `\copy` avšak je možné použiť štandardný príkaz jazyka SQL – príkaz `COPY`.

`Pg_dump` a `Pg_restore` slúži na zálohovanie databázy a obnovu databázy zo zálohy. Oba nástroje pracujú aj s databázami rozšírenými o priestorové údaje.

Z ďalších užitočných nástrojov možno zmieniť program *wget* a *7-zip*. Wget slúži na automatizované preberanie (sťahovanie) údajov z webu. Poslúži dobre najmä pri sťahovaní rozsiahlejších kolekcií údajov (obsahuje možnosti rekurzívneho sťahovania, divoké znaky atď.). 7-zip je program na archiváciu

[1] csv súbor predstavuje najjednoduchší zápis tabuľky v textovom súbore. Jednotlivé riadky predstavujú riadky tabuľky a stĺpce oddeľujeme vopred dohodnutým znakom napr. čiakrou, bodkočiarkou, tabulátorom atď. Do tohto fomátu dokáže ukladať tabuľky napr. aj MS Excel

a rozbaľovanie archívov rôznych formátov, s ktorými sa možno stretnúť pri distribúcii údajov.

4.3.2 Nástroje na import a export údajov

Rozšírenie postGIS obsahuje dva nástroje na import a export údajov z formátu esrishapefile *shp2pgsql* a *pgsql2shp* (a ich varianta s grafickým užívateľským rozhraním *shp2pgsql-gui*). Esrishapefile je netopologický vektorový formát, ktorý uviedla firma Esri začiatkom 90. rokov dvadsiateho storočia. Vďaka masovému rozšíreniu produktov firmy Esri v oblasti geoinformačných technológií sa stal tento formát nepísaným štandardom na ukladanie vektorových vrstiev. V skutočnosti to nie je formát jedného súboru, ale údaje sú rozdelené do troch povinných .shp, .shx a .dbf súborov a ďalších voliteľných, pričom sa navzájom líšia príponami súboru. Nástroj shp2psql sa používa v tvare:

```
shp2pgsql -názov_prepínača1 hodnota_prepínača1
  -názov_prepínača2 hodnota_prepínača2 ...
  > vektorova_vrstva_v_sql.sql
```

Kde môžeme použiť nasledujúce prepínače:

- `-s` určuje priestorový referenčný systém (SRS - Spatial Reference System). Hodnota srid je vždy číselná zo zoznamu všetkých srid. Na Slovensku sa stretneme najčastejšie s srid 4326 (priestorový referenčný systém WGS84) a číslom 5514 (priestorový referenčný systém s-jtsk s Křovákovým kartografickým zobrazením cez hlavný poludník Greenwich a severovýchodnou orientáciou osí). PostGIS (od verzie 2.0) dokáže tranformovať údaje z jedného súradnicového systému už počas importu. Je však nutné zadať parameter v tvare `-ssrid_zdrojové: srid_cieľové`.
- `-W` určuje kódovanie znakov v zdrojovom dBASE súbore (.dbf). Údaje budú automaticky konvertované do znakovej sady utf8.
- `-I` vytvorí priestorový index stĺpca s geometriou hneď po nahratí údajov. Tento prepínač má zmysel, pokiaľ do tabuľky plánujeme v blízkej budúcnosti vkladať ďalšie údaje. Indexovanie totiž spomaľuje proces vkladania údajov.
- `-g` určuje názov stĺpca s geometriou – štandardne je to malé geom pre subtyp geometry a geog pre subtyp geography.
- `-G` vytvorí namiesto typu geometry typ geography.

`Vektorová_vrstva_v_sql.sql` je súbor, do ktorého `shp2pgsql` zapíše reprezentáciu importovanej vektorovej vrstvy v príkazoch jazyka SQL. Z tohto súboru môžeme importovať údaje priamo do databázy postgress rozšírením postGIS nasledovným príkazom:

```
psql -h názov_servera -p 5432 -d moja_databaza
     -U postgres -vektorova_vrstva_v_sql.sql
```

Identickú funkcionalitu avšak s grafickým uživateľským rozhraním zabezpečuje nástroj `shp2pgsql-gui`. Ten dokáže navyše importovať viacero vrstiev súčasne.

Prostredníctvom nástroja `pgsql2shp` môžeme exportovať tabuľku predstavujúcu vrstvu z databázy postgresql s rozšírením postGIS do formátu esrishapefile. Nástroj používa nasledujúcu syntax:

```
pgsql2shp -f umiestnenie_databázy názov_databázy
  názov_schémy.názov_tabul'ky "upresňujúci dopyt SQL"
```

kde ako upresňujúci dopyt je možné (nie povinné) zadať dopyt, ktorý do exportu odfiltruje len údaje zodpovedajúce len požiadavke dopytu.

Príklad 4.7:

```
pgsql2shp -f gisdata/test/ slovensko administrativne.obce
          "SELECT * FROM obce WHERE okres='Banská Bystrica'"
```

Ďalším užitočným nástrojom na import a export údajov je `ogr2ogr`. Obsahuje veľké množstvo podporovaných formátov. Úplný zoznam možno nájsť na stránke `www.gdal.org/ogr/ogr_formats.html`. Najčastejšie sa stretneme s jeho využitím pri nasledujúcich formátoch (uvedieme iba príklady bez podrobnejšieho vysvetlenia. Viac o inštalácii a OGR a jeho použití nájde čitateľ dokumentácií na stránke `www.gdal.org/ogr`):

- Import Gpx súboru (získaného z GPS zariadenia):

  ```
  ogr2ogr -f "PostgreSQL" PG: "host=nazov_servera
      user=meno_uzivatela port=5432
      dbname=moja_databaza password=heslo"
  subor_z_GPS.gpx
  -overwrite
  -lco GEOMETRY_NAME=geom
  -nln "nazov_schémy.názov_ciel'ovej_tabul'ky"
  ```

- Import ESRI Personal Geodatabaz

  ```
  ogr2ogr -f "PostgreSQL" PG: "host=nazov_servera
      user=meno_užívatel'a port=5432
      dbname=moja_databaza password=heslo"
      subor_z_GPS.gpx
      -overwrite
      -lco GEOMETRY\_NAME=geom
      -nln "nazov_schémy.názov_ciel'ovej_tabul'ky"
  ```

- Import Mapinfo TAB

```
ogr2ogr -f "PostgreSQL" PG: "host=nazov_servera
    user=meno_užívatel'a port=5432
    dbname=moja_databaza password=heslo"
    priečinok_s_TAB_súborom
    -overwrite
    -lco GEOMETRY_NAME=geom
    -nln "nazov_schémy.názov_ciel'ovej_tabul'ky"
```

- Export do formátu KML

```
ogr2ogr -f "KML" /gisdata/moje.kml
PG: "host=nazov_servera user=meno_užívatel'a
port=5432 dbname=moja_databaza password=heslo"
vystup.kml NameField=nazov_stlpca_s_nazvami_objektov
```

Údaje z OpenStreetMap
Veľmi užitočným zdrojom údajov sú údaje z projektu OpenStreetMap[2]. Zachytávajú mnoho geografických fenoménov vo vysokom rozlíšení (napr. cestnú sieť, toky, parcely využitia územia, budovy atď.). Údaje sú poskytované v licencii opendatabaselicense alebo creative commons adtribution-sharealike2.0, to znamená, že je ich možné voľne využiť v komerčných a nekomerčných projektoch. Údaje získame cez stránku `www.openstreetmap.org/export`. Alebo cez REST XAPI službu s využitím nástroja `wget`. Výsledkom bude súbor vo formáte OSM. Súbor môžeme importovať do databázy nástrojom `osm2pgsql`, ktorý používa syntax v tvare:

```
osm2pgsql -d názov_databázy -H meno_DB_servera
  -U meno_uzivatela -P 5432 -S štýl.style
  zdrojovy_subor.osm
```

OSM používa na ukladanie atribútov tvar tag=hodnota, kde jeden objekt môže mať priradených aj viacero tagov. Tento spôsob ukladania nie je triviálne transformovateľný do štruktúry relačnej databázy. Ak chceme využívať tento tvar, je potrebné aktivovať v databáze postgresql rozšírenie hstore. Pri importe databázy použijeme v takomto prípade prepínač --hstore a sqldopyt na výber údajov z databázy sa používa v tvare:

```
SELECT name,
       array_to_string(akeys(tags!, '|') As keys,
       tags -> 'cycleway' As cycleway
FROM planet_osm_line
WHERE tags ? 'cycleway';
```

[2] `http://www.openstreetmap.org`

Na import rastrov do databázy postgresql s rozšírením PostGIS slúži nástroj `raster2pgsql` (je súčasťou rozšírenia PostGIS). Používa nasledovnú syntax:

```
raster2pgsql -s 4326 -C raster.tif
             schéma.tabul'ka > raster_v_sql.sql
```

Súbor raster `raster_v_sql.sql` následne importujeme do databázy analogicky ako pri vektorových súboroch (obsahuje sadu SQL príkazov s iportovanými údajmi).

Export rastrových údajov možno vykonať priamo cez SQL príkaz s využitím funkcií rozšírenie PostGIS – slúžia na to funkcie `ST_AsPNG()`, `ST_AsJPEG()`,`ST_AsTiff()`,`ST_AsRaster()`.

Časť II

Analýza údajov

Kapitola 5

Zdroje údajov pre priestorové analýzy

Analýza údajov, či napĺňanie údajového modelu závisí od toho, v akej podobe sa nachádzajú zdrojové údaje. V tejto kapitole sa oboznámime s podmienkami, ktoré je potrebné dodržiavať pri samplovaní (vzorkovaní) údajov z reálneho sveta a vysvetlíme ako pomocou rôznych interpolačných metód odhadnúť neznáme hodnoty zo skutočne nameraných hodnôt.

5.1 Získanie priestorových údajov samplovaním

Každé modelovanie priestorových údajov a odvodzovanie ďalších informácií vyžaduje určitú množinu vstupných údajov. Tie musia spĺňať kvalitatívne kritériá: presnosť, rozlišovacia schopnosť, konzistentnosť a kompletnosť. Údaje získavame samplovaním (vzorkovaním), keďže ich množstvo je nekonečné (napr. hĺbku jazera možno spočítať na nekonečne veľa miestach jeho plochy) alebo ich povaha ich nedovoľuje zaznamenať úplne (napr. pri sčítavaní obyvateľstva nesčítavame každého obyvateľa na mieste, kde sa práve v tom okamihu nachádza, ale sčítanie sa robí na väčších jednotkách plochy, ako sú sčítacie okrsky resp. základné sídelné jednotky). Vzorkovanie údajov v priestore môže byť:

- náhodné – body sú vyberané náhodným spôsobom (pomocou vybraného stochastického výberu);
- náhodne s pravidelným rozdelením – priestor je rozdelený pravidelnou mriežkou a vzorkovacie body sú vybrané náhodne jeden v každej oblasti vymedzenej mriežkou;
- systematické – vybrané body tvoria pravidelnú mriežku, prípadne iný pravidelný útvar;

Pri získaní údajov samplovaním môžu vzniknúť nasledujúce problémy:

- Hrubé chyby samplovanie – pri zbere údajov sa môže vyskytnúť hrubá chyba napr. prepisom údajov o nadmorských výškach vrstevníc bude kvôli preklepu priradená výška vrstevnice rádovo iná hodnota (napr. namiesto 960 priradíme hodnotu 690). Tieto chyby je možné objaviť resp. eliminovať viacerými štatistickými postupmi.
- Nedostatočné rozlíšenie vzorkovaných bodov (najčastejším riešením je využitie priestorovej interpolácie)

5.2 Priestorové interpolácie

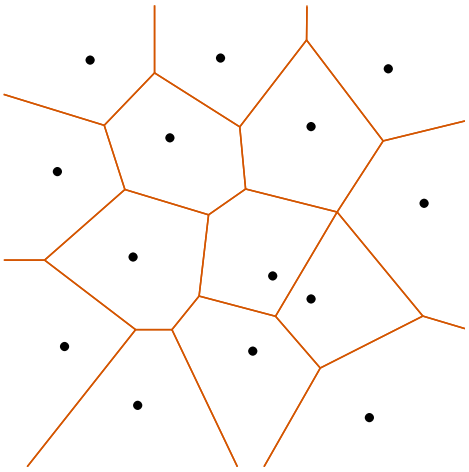
Interpolácia v chápaní technológie GIS je pracovný postup, ktorý na základe predpokladu priestorovej autokorelácie priestorovej informácie mení diskrétnu bodovú priestorovú informáciu na spojitú priestorovú informáciu.

Na odhad alebo predpovedanie hodnôt na priestorovo-spojitom povrchu pomocou interpolačných metód musia byť splnené nasledovné podmienky:

- metóda musí využívať priestorovú štruktúru na povrchu a vyuužiť pri odhade údaje s priradenou váhou podľa vzdialenosti lokality, pre ktorú sú údaje odhadované;
- pozorované hodnoty, ktoré sú priestorovo blízke, zdvojujú hodnotu, takže bez nejakej formy váhovania členov zhluku je tu nebezpečenstvo, že odhad bude príliš ovplyvnený meraním z jednej časti mapy;
- metóda by mala dávať indikáciu chyby pravdepodobnosti asociovanú s predpoveďou;
- metóda by mala rešpektovať známe vlastnosti vzorky a najmä každá metóda by mala vrátiť tým presnejšiu hodnotu, čím je bližšie k známej nameranej hodnote. Tieto požiadavky na získanie chýbajúcich údajov vychádzajú z prístupov jednorozmernej štatistiky. Podobné požiadavky kladieme aj na metódy vyhladenia údajov.

Pojem priestorová interpolácia sa niekedy vzťahuje na metódy, ktoré poskytujú odhad hodnôt údajov, kde sa nepracuje so žiadnym pravdepodobnostným modelom. Interpolačné metódy sú založené na vyťažení geometrických atribútov lokalít s nameranými hodnotami a vo všeobecnosti tak spĺňajú prvé dve kritériá. Priestorová alebo geoštatistická predpoveď stavia na pravdepodobnostnom modeli vztiahnutom na zemský povrch. Kriging zahŕňa skupinu metód na predpovedanie údajových hodnôt. Rieši tak nedostatky prístupov založených na geometrii a tiež zabezpečuje predpoveď chýb.

Pozorovania prislúchajúce oblastiam môžeme previesť na bodové pole tým, že každej oblasti priradíme bod – centroid (viac v časti 7.3.6 na str. 88). Naopak pokiaľ máme k dispozícii bodové údaje, je možné okolo nich vytvoriť



Obr. 5.1: Ukážka Dirichletovho delenia roviny medzi množinu bodov

polygóny s použitím Dirichletovho delenia (viac na str. 59). Tieto dve metódy nám umožňujú vzájomne zamieňať metódy určené pre bodové údaje a areálové údaje. Dôležité je, aby tieto metódy boli aplikované kriticky s ohľadom na ich vlastnosti a povahu údajov v príslušnej priestorovej sade údajov.

V súčasnosti existuje široké spektrum interpolačných algoritmov. Každý interpolačný algoritmus sa vyznačuje špecifickými vlastnosťami. Tieto vlastnosti určujú jeho výhody a nevýhody a taktiež oblasť jeho použitia. V nasledujúcej časti rozoberieme najpoužívanejšie metódy.

5.2.1 Dirichletovo delenie

Dirichletovo delenie (označované aj ako Voronoiove diagramy alebo Thiesennove polygóny) je najjednoduchšia metóda na priradenie rovnakej atribútovej hodnoty v akejkoľvek lokalite v jej najbližšom susedstve. Rovina je rozdelená na Dirichletove polygóny, z ktorých každý má vlastnosť, že ľubovoľný bod nachádzajúci sa v danom polygóne má najbližšiu vzdialenosť k bodu, okolo ktorého bol Dirichletový polygón vytvorený (obr. 5.1). Toto pravidlo vytvára na miestach hraničných línií body diskontinuity, t.j. body, ktoré sú na hranici medzi dvomi polygónmi a nepatria jednoznačne do žiadneho z nich. Na vyriešenie tohto problému je možné využiť ďalšie metódy využívané na zhlukovanie hodnôt.

PostGIS obsahuje funkciu `ST_DelaunayTriangles()`, ktorá vytvorí kolekciu polygónov (prípadne `MULTILINESTRING` alebo `TIN` sieť) na základe Delaunayovej triangulácie. Z trojuholníkov Delaunayovej triangulácie sa dajú vhodnou sériou SQL dopytov zostrojiť Voronoiove diagramy[1].

[1] postup nájdeme na `http://gis.stackexchange.com/questions/114764/how-to-use-st-delaunaytriangles-to-construct-a-voronoi-diagram`

5.2.2 Rozklad zhlukov na bunky

Táto metóda rozdelí plochu okolo lokality na kvadranty. Hodnota pre všetky bunky kvadrantu je získaná ako priemer hodnôt bodov v kvadrante.

5.2.3 Triangulácia

Metóda triangulácie predstavuje váženú interpoláciu. Tri body najbližšie k bodu, ktorý určujeme, sú vybraté po vykonaní delaunajovej triangulácie z množiny nameraných bodov (pozri Isaaks and Srivastava, 1989,pp.251–6). Tieto tri body, ktoré sú najbližšími susedmi v Dirichletovom delení, sú vybraté tak, že trojuholník obklopuje bod, ktorý je odhadovaný. Ďalším krokom je vyriešenie rovnice roviny s troma neznámymi koeficientmi.

$$y(j) = aE(j) + bN(j) + c \tag{5.1}$$

Rieši sa to cez substitúciu, kde $y(j)$ je pozorovaná hodnota a $E(j)$ a $N(j)$ sú hodnoty východná súradnica a severná súradnica bodu $j(j = 1, 2, 3)$ takže platí:

$$\hat{y}(i) = \hat{a}E(i) + \hat{b}N(i) + \hat{c} \tag{5.2}$$

kde a, b, c sú odhady koeficientov. Tento spôsob odhadu nám poskytuje priestorovo hladšie hodnoty ako predchádzajúce opísané metódy, ale na okrajoch trojuholníka vytvára náhle zmeny gradientu hodnôt.

5.2.4 Metóda IDW (Inverse distance weighting)

Interpolačná metóda IDW využíva priradenie rôznej váhy pozorovaniam v závislosti na ich vzdialenosti od chýbajúcej hodnoty. Vážená vzdialenosť je postavená na myšlienke, že hodnoty atribútov v blízkosti neznámej hodnoty atribútu sú podobné, avšak podobnosť slabne so zväčšujúcou sa vzdialenosťou, preto najbližším bodom je priradená najväčšia váha. Matematicky je metóda IDW váženým priemerom podľa obrátenej vzdialenosti. Interpolovanú hodnotu pre $y(I)$ vypočítame ako:

$$\hat{y}(i) = \sum_{j=1,\ldots n} \lambda(i,j)y(j) \tag{5.3}$$

kde $\lambda(i,j) = d(i,j)^{-\alpha}$, α je kladná konštanta a $d(i,j)$ je vzdialenosť medzi bodmi i a j. Inou možnosťou je $\lambda(i,j)$ definovať ako $exp(-\alpha d(i,j))$ alebo $exp(-\alpha d^2(i,j))$ a parametru $\lambda(i,j)$ priradiť nulu pokiaľ presiahne zvolenú vzdialenosť. Koeficienty sú vážené, takže $\sum_{j=1,\ldots,n} \lambda i,j) = 1$. Voľba parametra α ovplyvňuje príspevok rôznych údajov z rôznych vzdialeností a vlastnosti zhladenia povrchu (Ripley, 1981). Veľké hodnoty parametra α znamenajú, že vzdialené body majú pri výpočte len malú váhu, čoho následkom je veľmi špicatá mapa z interpolovaných hodnôt. Malé hodnoty parametra α dávajú rovnakú váhu aj dlhším vzdialenostiam a spôsobujú vo výslednej mape

isté zhladenie. V literatúre (Isaaks – Srivastava, 1989) sa uvádzajú príklady použitia rôznych hodnôt parametra α a porovnanie výsledných máp.

Metódy váženej vzdialenosti sú často ovplyvnené zhlukmi údajových bodov. Na dosiahnutie optimálneho výsledku je vhodné skombinovať niekoľko metód. Metóda rozkladu buniek môže byť použitá na zistenie, v ktorom kvadrante je priemer pridaný k centroidu pozorovaných hodnôt a potom výsledok použiť na váženú vzdialenosť priemeru podľa vzťahu (5.3).

Literatúra (Isaaks – Srivastava, 1989) porovnávajú niekoľko z vyššie uvedených metód na sérii príkladov. Výber najlepšej metódy záleží od kritéria, ktoré zvolíme ako mieru vhodnosti. Niektoré metódy z relatívne veľkým parametrom rozkladu, (t.j. $\alpha = 2.0$) sú vhodné ak je cieľom minimalizovať najväčšie chyby odhadu, kým napr. metódy triangulácie majú relatívne dobré priemerné úrovne chyby. Tvrdia, že začlenenie viacerých blízkych bodov vylepší odhad, avšak pokiaľ body tvoria zhluk, môže to byť na škodu. Najlepšie metódy, ktoré sú, používajú všetky najbližšie body a počítajú s možnosťou zhlukovania. Tieto vlastnosti budú prezentované v metóde odhadu kriging, ktorú preberieme nižšie a ktorá využíva informácie o vlastnostiach priestorovej kovariancie skúmaných údajov. Na rozdiel odhady opísané vyššie závisia len na geometrických vzťahoch medzi bodmi a odhad, ktorý z nich získame je závislý na veľkosti susedstva a zadaných váhach. Neposkytujú odhad možnej chyby asociovanej výpočtom.

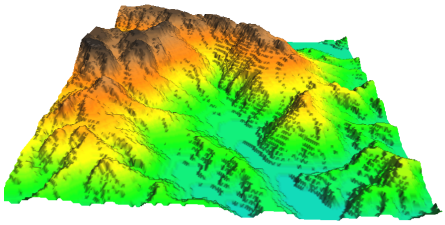
5.2.5 Kriging

Predpokladajme, že údaje $y(1), ..., y(n)$ sú realizáciou (slabo stacionárneho) stochastického modelu s priemerom $\mu(.)$ a (symetrickou) variačno-kovariačnou maticou $\sum$. Daná vzorka veľkosti n, potom BLUP (najväčšiu lineárnu nevychýlenú predpoveď) ľubovoľného nevzorkovaného bodu z povrchu môžeme získať metódou jednoduchého krigingu. Na odhad atribútovej hodnoty v bode o, (o), ktorý nie je začlenený vo vzorke výpočtu:

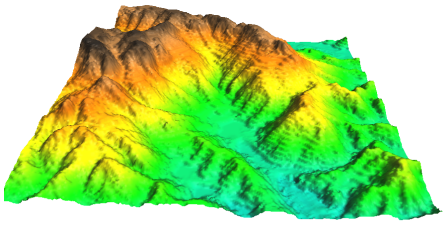
$$\tilde{y}(o) = \mu(o) + c^T \sum^{-1} (y - \mu) \tag{5.4}$$

kde $c^T = (cov(Y(0), Y(1)), ..., cov(Y(o), Y(n))$, $\sum$ predstavuje symetrickú maticu $n \times n$ s (i, j)-tým prvkom rovným $cov(Y(i), Y(j)), y = (y(1), ..., y(n))^T$, $\mu = (\mu(1), ..., \mu(n))^T$ a $\mu(o)$ je priemer odhadovaný v mieste o. Druhý člen vo vzťahu (5.4) určuje váhy jednoduchého krigingu, $c^T \sum^{-1}$ priradené ku každému údajovému bodu, ktorý vstupuje do BLUP odhadu neznámej atribútovej hodnoty.

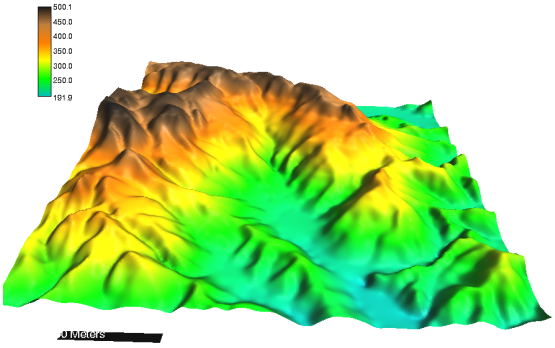
Váhy vo vzťahu (5.4) odrážajú priestorovú kovarianciu vlastností v údajoch alebo inak „*štatisticky* váženú vzdialenosť“. Vektor c uvádza do odhadu formu váženia vzdialenosti, pretože kovariancia má tendenciu klesať so stúpajúcou vzdialenosťou, takže body, ktoré sú vzdialenejšie, budú mať na odhad menší vplyv. Prítomnosť sumy zahŕňa kovarianciu medzi každou jednou pozorovanou hodnotou a ľubovoľnou inou. Pozorované hodnoty, ktoré sú vzájomne blízko, majú veľké hodnoty sumy, tie ktoré sú vzájomne ďaleko, majú



(a) Metóda Voronoiových polygónov



(b) Metóda váženej inverznej vzdialenosti (IDW



(c) Metóda regularizovanej splajnovej funkcie s tenziou

Obr. 5.2: Interpolácia digitálneho terénneho modelu

malé hodnoty. Vynásobenie vektora c sumou na mínus prvú prispôsobí surovú obrátenú štatistickú váhu vzdialenosti (vektora c) tak, aby rátala s možnými redundanciami medzi pozorovanými hodnotami (Isaaks – Srivastava, 1989). Prítomnosť sumy na mínus prvú sa postará o vplyv na odhad zhlukovania údajov pozorovaných hodnôt. To sa dosiahne znížením váhy vplyvu z vzorky bodov, ktoré sú členmi zhlukov. Pokiaľ zhlukovanie nepredstavuje problém (napr. v prípade, že údaje pochádzajú zo systematickej priestorovej vzorky alebo mriežky pixlov), nie je prekvapujúce, že jednoduchšia metóda váženej vzdialenosti alebo dokonca najbližšieho suseda, ktorá aproximuje prvky c ale neobsahuje pojem zodpovedajúci suma na mínus prvú dosiahne rovnako dobré výsledky v niektorých prípadoch.

Odhad chyby pre vzťah (5.4) je:

$$\sigma^2 = Cov(Y(o), Y(o)) - c^T \sum^{-1} c \tag{5.5}$$

Prvý člen $\sigma^2 = Cov(Y(o), Y(o))$ vyjadruje mieru variability atribútu y. Druhý člen je váženou sumou kovariancií medzi vzorkami a hodnotou, ktorá bude

predpovedaná (c), kde váhy sú dané váhami jednoduchého krigingu $c^T \sum^{-1}$. Intervaly odhadu môžu byť odvodené zo vzťahu (5.5). V prípade 95%-ných intervalov je tento interval daný $y(o) \pm 1,96\sigma$.

V praxi priemer zvyčajne nepoznáme a pre mapovanie a interpoláciu váh je zvyčajne potrebné nejaký vypočítať. Táto posledná podmienka zistí univerzálnu nezávislosť. Predpokladajme, že priemer je konštanta a kovariančná matica suma je známa minimálne do skaláru σ^2. Optimálny odhad v zmysle minimalizácie chyby odhadu dokážeme dosiahnuť cez *základný kriging*. Základný kriging zabezpečuje odhad v jednom kroku a nevyžaduje explicitnú identifikáciu priemeru. Pokiaľ je priemer neznámy, ale sleduje trend povrchu, optimálny odhad možno dosiahnuť *s univerzálnym krigingom*. Ten je takisto odhadom v jednom kroku, nevyžaduje poznať priemer avšak vyžaduje špecifikovať rád trendového povrchu. Nasledujúci popis predpokladá slabú stacionaritu. Výrazy predpokladajú iba prirodzenú nie slabú stacionaritu. Viac v Cressie (1991).

kriging sa používa v geoštatistike na priestorový odhad viac než univerzálny kriging. Priestorový odhad je založený na lokálnej informácii a tak predpoklad konštantného priemeru nie je vhodný. Existujú aj metódy ako prakticky v geoštatistike využiť univerzálny kriging.

Existujú aj iné formy krigingu aplikovateľné v situáciach, kde údaje nemajú gausovské rozdelenie (ako napr. kategórie alebo údaje o početnosti) a kde je potrebné použiť nelineárnu geoštatistiku.

S výnimkou metódy inverznej váženej vzdialenosti ($\alpha = 2$), kde váhy klesajú veľmi prudko so stúpajúcou vzdialenosťou, všetky metódy poskytujú odhad, ktorý je širší ako odhad získaný zo základného krigingu považovaného za zlatý štandard. Toto tvrdenie je pravdivé vo väčšine množín údajov avšak v niektorých špecifických prípadoch to platiť nemusí.

Jednoduchý, základný a univerzálny kriging je implementovaný v systéme *SAGA GIS*[2]. Tieto interpolačné moduly možno využiť aj pri práci v QGISe prostredníctvom *Toolboxu* (menu *Processing*).

5.2.6 Metódy základných radiálnych funkcií

Metóda radiálnych funkcií (angl. Radial basis function) pracuje v závislosti od použitia vyhladzovacieho parametra λ ako exaktná alebo neexaktná interpolačná technika. Interpolačný postup je založený na lineárnej váženej kombinácii určitého počtu základných kernelových funkcií so snahou čo najpresnejšieho preloženia zdrojových údajov. Tieto funkcie sú analogické modelom variogramov používaných pri geoštatistických odhadoch. K najčastejšie používaným patria:

- Multikvadratická metóda
- Prirodzené kubické splajny

[2] `www.saga-gis.org`

- Metóda „thin plate spline“
- Multilogaritmická metóda

Parameter h predstavuje anizotropicky korigovanú vzdialenosť medzi bodom s odhadovanou hodnotou a údajovým bodom, pre ktorý je odhadovaná váha. Sárközy – Závoty (1995) k metódam základných radiálnych funkcií zaraďuje aj regularizované splajnové funkcie s tenziou (Mitášová et al., 1995), Fourierove a Wavelet (vlnkové) analýzy.

Interpolácia pomocou týchto funkcií dáva spoľahlivé výsledky aj pri interpolácii povrchov korýt vodných tokov (Anstead, 2013).

Regularizovaný splajn s tenziou

V ďalšom texte si opíšeme použitie algoritmu RST (regularizovaný splajn s tenziou) v technologickom prostredí QGIS s využitím modulov GRASSu. V opise vychádzame z práce Burian et al. (2015).

RST je špecifická aproximačná, resp. interpolačná funkcia dvoch premenných, ktorá v miestach bodov diskrétneho vstupného bodového poľa, určeného súradnicami X a Y (uzlové body), nadobúda funkčné hodnoty blízke (aproximácia), resp. zhodné (interpolácia) so skalárnymi hodnotami priradenými k týmto bodom. V GRASS GIS slúži táto funkcia na výpočet skalárnych hodnôt v ťažiskách buniek generovaného rastra reprezentujúceho vytváraný povrch (DTM).

Algoritmus RST v prostredí GRASS GIS je nastavený tak, aby bol bez obmedzení schopný naraz pracovať so 700 vstupnými bodmi. V prípade, že vstupné bodové pole má väčší počet bodov ako 700, je potrebné pred samotnou interpoláciou rozdeliť body do segmentov tak, aby v žiadnom nebolo viac ako 700 bodov. To znamená, že v prípade 2800 bodov vstupného bodového poľa sa interpolačný algoritmus spustí 4× nad štyrmi rôznymi vstupnými bodovými poľami. Výsledkom sú 4 vygenerované rastre modelovaného povrchu, ktoré algoritmus následne zlúči do jedného rastra reprezentujúceho predmetný DTM.

Jedným z hlavných problémov segmentácie je zabezpečenie hladkého spojenia jednotlivých segmentov (pri vizualizácii výslednej rastrovej vrstvy môže byť hranica medzi segmentami jasne badateľná). Tento problém sa vyskytuje hlavne v prípadoch, keď pri čiastkovom procese interpolácie využívame len body ležiace v rámci segmentu. Tento problém rieši algoritmus RST pomocou dvoch hodnôt segmentácie:

- maximálny počet bodov v segmente (*Maximum number of points in a segment*).
- minimálny počet bodov potrebných pre aproximáciu (*Minimum number of points for approximation in a segment*).

Vždy platí, že počet bodov v segmente je nižší ako počet bodov potrebných pre aproximáciu.

Správnym nastavením hodnôt segmentácie dokážeme zabezpečiť hladšie napojenie jednotlivých segmentov. Hladkosť napojenia segmentov je vhodné overiť si napr. prostredníctvom rastrov (alebo izočiar) parciálnych derivácií vygenerovaných pomocou toho istého modulu. Niekedy je nutné predĺžiť výpočtový čas a zväčšiť hranicu oblasti aproximácie, teda zvoliť vyššiu hodnotu počtu bodov potrebných pre aproximáciu. Ideálny je výpočet len v rámci jedného segmentu, teda RST aproximácia pri počte bodov v segmente nastavenom na hodnotu počtu bodov vstupujúcich do výpočtu celého DTM. V tomto prípade každý uzlový bod RST aproximácie (body vstupného bodového poľa) sa podieľa svojou váhou na stanovení hodnoty modelovanej veličiny v danej bunke rastra. Spôsobuje to výraznejšie zhladenie modelovaného povrchu oproti povrchu získaného s využitím segmentácie.

Samotný algoritmus interpolácie pomocou RST je obsiahnutý v module `v.surf.rst`. Modul je prístupný v QGISe buď prostredníctvom nástroja *Toolboxu* (menu *Processig*) alebo cez zásuvný modul GRASSu. Vstupmi pre modul `v.surf.rst` môžu byť body alebo línie (v podobe izolínií) s priradenou hodnotou interpolovanej veličiny t.j. vstupom pre vytvorenie DTM môže byť ;krétne bodové pole bodov nadmorských výšok alebo vektorová vrstva vrstevníc.

Modul `v.surf.rst` bol vyvíjaný najmä pre tvorbu DT, preto nie všetky predvolené nastavenia sú vhodné napr. pre potreby interpolácie hodnôt teploty vzduchu, hrúbky humusového horizontu, zakyslenia pôdy, atď.

Ďalej je možné nastaviť nasledujúce parametre:

- *Name of input vector map* – názov vstupnej vektorovej vrstvy (bodová alebo líniová)
- *Output surface raster map (elevation)* – názov výstupnej interpolovanej rastrovej vrstvy
- *Output slope raster map* – názov výstupnej rastrovej vrstvy s hodnotou sklonu plochy v smere spádnice
- *Output aspect raster map* – názov výstupnej rastrovej vrstvy s hodnotami orientácie plochy voči svetovým stranám
- *Output profil curvature raster map* – názov výstupnej rastrovej vrstvy s hodnotami normálovej krivosti plochy v smere dotyčnice k spádnici.
- *Output tangential curvature raster map* – názov výstupnej rastrovej vrstvy s hodnotami normálovej krivosti plochy v smere dotyčnice k vrstevnici.
- *Output mean curvature raster map* – názov výstupnej rastrovej vrstvy s hodnotami strednej krivosti plochy[3].

[3]Mean curvature, ktorá je v GRASS GIS, ale i v iných GIS rovná polovičnej hodnote súčtu normálových krivostí v smere dotyčnice k spádnici a k vrstevnici nie je strednou krivosťou plochy, ktorá je matematicky definovaná ako polovičná hodnota súčtu hlavných, to znamená, extrémnych normálových krivostí plochy. Normálové krivosti v smere dotyčnice k spádnici a k vrstevnici nie sú vždy hlavnými krivosťami plochy (Krcho, 2001)

- *Name of raster map used as mask* – názov rastrovej vrstvy, ktorá bude použitá ako maska
- *Name of attribute column with values to be used for approximation* – názov stĺpca v tabuľke atribútov vstupnej vektorovej vrstvy (bodovej alebo líniovej), ktorý obsahuje hodnoty interpolovanej veličiny (interpolovať sa dajú len číselné hodnoty)
- *Tension parameter* – hodnotu parametra tenzie aproximačnej funkcie (RST)
- *Smoothing parameter* – hodnotu parametra zhladenia aproximačnej funkcie (RST)
- *Maximu number of points in segment* – maximálny počet bodov v segmente (počet bodov v segmente)
- *Minimum number of points for approximation in a segment* – minimálny počet bodov potrebných pre aproximáciu
- *Minimum distance between points* – minimálnu vzdialenosť medzi bodmi. V prípade, že vzdialenosť medzi dvoma susednými bodmi je menšia ako je nastavená vzdialenosť, jeden z bodov sa vylúči z aproximácie.
- *Maximum distance between points on isoline (to insert additional points)* – maximálnu vzdialenosť bodov na izolínii. Nastavenie sa využíva len v prípade vstupnej izolíniovej vrstvy. V prípade, že vzdialenosť dvoch lomových bodov na izolínii je väčšia ako zadaná vzdialenosť, algoritmus pridá medzi tieto body ďalší bod a tým spresní výpočet výslednej vrstvy.
- *Allow output files to overwrite existing files* – prepísanie vrstvy s rovnakým názvom

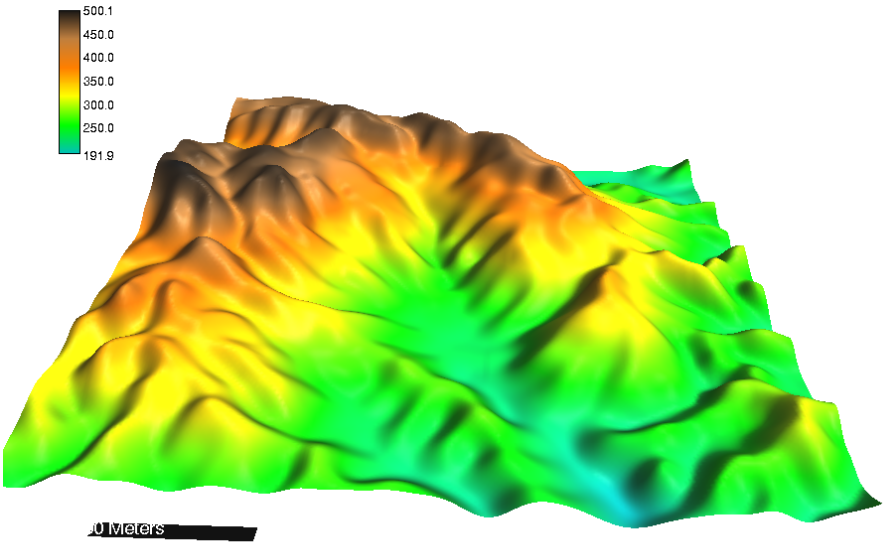
Okrem iných nastavení algoritmus RST využíva parameter tenzie a zhladenia (smoothing). Obidva uvádzané parametre menia charakter aproximačnej matematickej funkcie. Parameter tenzie hovorí o tuhosti aproximačnej funkcie. Vysoká hodnota tenzie dáva funkcii charakter elastickej membrány, v prípade použitia porovnania s gumou, charakter elastickej a mäkkej gumy. Splajnom tenkej platne modelovaný povrch s vyššími hodnotami tenzie sa v uzlových bodoch RST aproximácie stále viac približuje k ich reálnym výškam[4], avšak v ich okolí môže dôjsť k veľkým výškovým odchýlkam.

V miestach výrazných zmien sklonov reálnej plochy dochádza vďaka väčšej pružnosti funkcie často k „vydutiu“ výškového modelu, t. j. k vzniku falošných depresií alebo vrcholov (singularít). Naopak, nízka hodnota tenzie dáva aproximačnej funkcii charakter tenkej platne, v prípade použitia porovnania s gumou, charakter tuhej a tvrdej gumy. Znižovanie hodnoty tenzie sa v uzlových bodoch RST aproximácie prejaví stále väčšími odchýlkami namodelovaných výšok od výšok bodov vstupného výškového poľa, avšak v ich okolí už

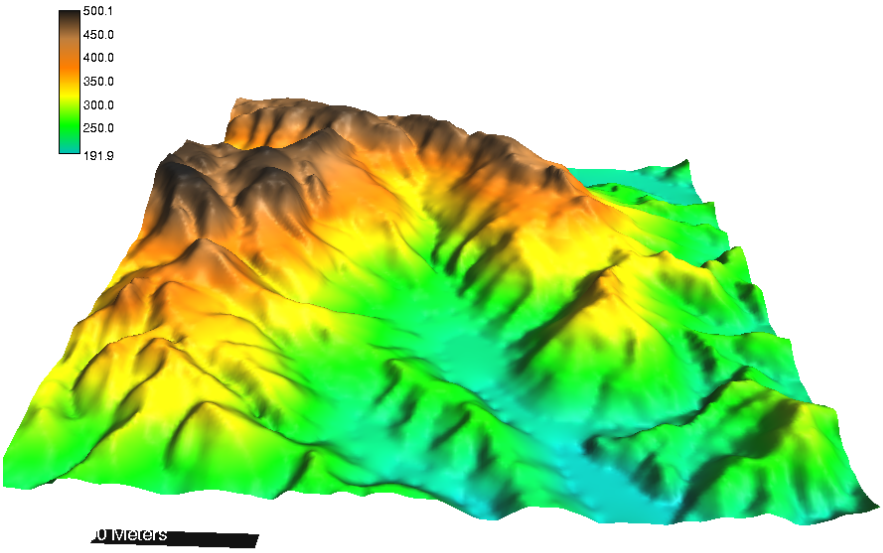
[4]Stredná odchýlka – parameter *rmsdevi* v popisnom súbore k rastrovej vrstve vygenerovanej RST aproximáciou (využijeme modul `r.info`) vykazuje minimálnu hodnotu

nedochádza k tak výrazným výškovým odchýlkam. S hodnotou parametra tenzie blížiacou sa k nule sa stále viac dostávame k modelu, síce rôzne nakloneného, ale extrémne zhladeného povrchu. V GRASS GIS je preto pre tento parameter prednastavená hodnota 40 (výsledok vidno na obr. 5.2c, str. 62). Ako sa prejaví zmena tejto hodnoty ilustruje obr. 5.3 (str. 68).

Parameter *smoothing* kontroluje výškovú odchýlku v uzlových bodoch RST aproximácie. Ak má nulovú hodnotu, aproximačná funkcia RST sa zmení na interpolačnú funkciu. Výšky vypočítané v uzlových bodoch RST aproximácie sa v tomto prípade neodchyľujú od výšok bodov vstupného bodového poľa, takže parameter *rmsdevi* v popisnom súbore rastrovej vrstvy reprezentujúcej vytvorený DTM nadobudne hodnotu rovnú nule. Nastavenie parametra smoothing na hodnotu 0 však podporuje vznik falošných singularít mimo uzlových bodov, podobne ako v prípade nastavených vyšších hodnôt tenzie. V systéme GRASS GIS je preto pre tento parameter prednastavená hodnota $0,1$.



(a) Tenzia=10



(b) Tenzia=80

Obr. 5.3: Ukážka rôzneho nastavenia parametra tenzie pri metóde RST

Kapitola 6

Filtrovanie cez atribútové dopyty

Množstvo úloh sa dá riešiť odfiltrovaním „tých správnych“ záznamov z veľkého množstva iných údajov uložených v tabuľke. V tejto kapitole sa oboznámime so základnými príkazmi jazyka na tvorbu dopytov – SQL. Ten nám umožní formulovať pravidlá na odfiltrovanie údajov spĺňajúcich požadované kritériá. Príklady odporúčame realizovať cez okno SQL v nástroji *DB Manager* (je sú časťou QGISu) nad tabuľkami umiestnenými v databáze PostgreSQL s rozšírením PostGIS.

6.1 Jednoduché dopyty

Najčastejšie realizovanou úlohou je výber údajov z databázy. Využívajú sa na to dopyty. Na tvorbu dopytov slúži príkaz `SELECT` v tvare:

```
SELECT nazov_stlpca1,nazov_stlpca2,... FROM tabulka
```

Namiesto vymenovania stĺpcov sa môže použiť znak hviezdička (*), čo znamená, že budú zobrazené všetky existujúce stĺpce tabuľky. Použitie * v dopytoch sa neodporúča, kvôli výrazne pomalšiemu spracovaniu dopytu.

Príklad 6.1:

```
SELECT * FROM kraje;

 IDN4 |       NM4        |  Shape_Leng   | VYMERA_ha
------+------------------+---------------+-----------
    1 | Bratislavský     | 305858.826365 |    205262
    2 | Trnavský         | 584344.006563 |    414642
    3 | Trenčiansky      | 488464.365558 |    450184
    4 | Nitriansky       | 571202.681983 |    634373
```

```
    5 | Žilinský         |  585592.71797 |     680850
    6 | Banskobystrický | 661268.174361 |     945402
    7 | Prešovský        | 814339.534918 |     897329
    8 | Košický          | 693753.741175 |     675449

(8 row)
```

Podmienka WHERE

Dopyt vyberie všetky záznamy z tabuľky kraje. Dopyt však väčšinou formulujeme tak, aby výsledkom boli len dopyty spĺňajúce určité kritérium, napríklad vyberiem len kraj s názvom „Banskobystrický“

```
 test=# SELECT "IDN4","NM4","VYMERA_ha" FROM kraje
                WHERE "NM4" LIKE 'Banskobystrický';

 IDN4 |        NM4        |  VYMERA_ha
------+-------------------+-----------
    6 | Banskobystrický |  945402
(1 row)
```

Podmienka sa píše za kľúčové slovo `WHERE` uvádzané na konci dopytu. Názvy stĺpcov a relácií je potrebné uzavrieť do dvojitých úvodzoviek, pokiaľ obsahujú Non-Latin1 znaky alebo znaky veľkých písmen. Operátor `LIKE` sa používa na porovnávanie s textovými (znakovými) reťazcami. Reťazce vždy uvádzame v jednoduchých úvodzovkách. Môžeme používať aj zástupné znaky % (znamená ľubovoľný počet ľubovoľných znakov), znak _ znamená jeden ľubovoľný znak.

Príklad 6.2: Vyberme všetky kraje, ktoré začínajú na písmeno T a ďalšie písmená môžu byť ľubovoľné

```
test=# SELECT "IDN4","NM4","VYMERA_ha" FROM kraje
           WHERE "NM4" LIKE 'T\%';

 IDN4 |        NM4        |  VYMERA_ha
------+---------------+--------------
    2 | Trnavský      |     414642
    3 | Trenčiansky |     450184
(2 rows)
```

Príklad 6.3: Výber všetkých krajov, ktoré začínajú na ľubovoľné písmeno, druhé písmeno je "r" a ďalšie písmená sú ľubovoľné dosiahneme nasledovným SQL dopytom:

```
test=# SELECT "IDN4","NM4","VYMERA_ha" FROM kraje WHERE "NM4"
LIKE '_r\%';
```

```
IDN4 |      NM4       | VYMERA_ha
------+--------------+-----------
    1 | Bratislavský |    205262
    2 | Trnavský     |    414642
    3 | Trenčiansky  |    450184
    7 | Prešovský    |    897329
(4 rows)
```

Podobne dokážeme filtrovať aj číselné atribúty. V podmienke môžu byť použité porovnávacie operátory: <,>,=,<>,<=,>= . Nasledujúci dopyt zobrazí kraje s rozlohou menšou ako 600000 ha.

```
SELECT "IDN4","NM4","VYMERA_ha" FROM kraje_lm
     WHERE "VYMERA_ha"<600000 ORDER BY "VYMERA_ha";
```

```
IDN4 |      NM4       | VYMERA_ha
------+--------------+-----------
    1 | Bratislavský |    205262
    2 | Trnavský     |    414642
    3 | Trenčiansky  |    450184
(3 rows)
```

Poradie výpisu ORDER BY
V predchádzajúcom dopyte sme doplnili na koniec aj klauzulu ORDER BY, ktorá zoradí záznamy podľa vybraného stĺpca (v prípade textového podľa abecedy, v prípade číselného podľa veľkosti).

Reťazenie podmienok
Podmienky môžeme reťaziť spájaním viacerých výrazov pomocou logických operátorov AND (platí súčasne podmienka A a zároveň B), OR (platí aspoň jedna z podmienok A a B), alebo možno využiť negáciu výroku využitím operátora NOT (platí negácia podmienky).

Príklad 6.4: Vyberieme všetky kraje, kde je rozloha väčšia ako 45000 ha a menšie ako 75000 ha.

```
SELECT "IDN4","NM4","VYMERA_ha" FROM kraje_lm
    WHERE "VYMERA_ha">450000 AND
          "VYMERA_ha"<750000;
```

```
IDN4 |      NM4      | VYMERA_ha
------+-------------+-----------
    3 | Trenčiansky |    450184
    4 | Nitriansky  |    634373
    5 | Žilinský    |    680850
    8 | Košický     |    675449
(4 rows)
```

Obmedzenie počtu vrátených záznamov
Výpis možno obmedziť na určitý počet záznamov cez kľúčové slovo LIMIT na konci dopytu. Nasledujúci príklad vypíše prvých desať obcí Slovenska usporiadaných podľa abecedy.

```
SELECT "NM2",okres FROM obce_okresy ORDER BY "NM2" LIMIT 10;
```

```
         NM2           |      okres
-----------------------+--------------------
 Ábelová               | Lučenec
 Abovce                | Rimavská Sobota
 Abrahám               | Galanta
 Abrahámovce           | Bardejov
 Abrahámovce           | Kežmarok
 Abramová              | Turčianske Teplice
 Abranovce             | Prešov
 Adamovské Kochanovce  | Trenčín
 Adidovce              | Humenné
 Alekšince             | Nitra
(10 rows)
```

Aliasing
V dopytoch sa často používa aliasing – t.j. nahradenie mena stĺpca alebo tabuľky dočasným názvom. Táto technika má veľký význam hlavne pri prepájaní tabuliek s rovnakými názvami (špeciálne prepájanie tabuľky so sebou samou). Tiež výrazne sprehľadňuje písanie dopytov a umožňuje využívať krátke názvy stĺpcov s pôvodne dlhými názvami.

Príklad 6.5:

```
test=# SELECT "IDN4" AS ID,
              "NM4" AS "Názov kraja",
              "vymera_ha" AS "Výmera (ha)"
        FROM kraje;
```

```
 ID |      Názov kraja    | Výmera (ha)
-----+-----------------+-----------
    1 | Bratislavský    |   205262
    2 | Trnavský        |    414642
    3 | Trenčiansky     |    450184
    4 | Nitriansky      |    634373
    5 | Žilinský        |    680850
    6 | Banskobystrický |    945402
    7 | Prešovský       |    897329
    8 | Košický         |    675449

(8 row)
```

Je možné vypísať aj hodnoty, ktoré získame výpočtom z rôznych stĺpcov. Napríklad hustota obyvateľstva počítaná ako pomer medzi počtom obyvateľov a rozlohou.

```
test=# SELECT IDN4 AS ID,
              "NM4" AS "Názov kraja",
              pocet_obyv/(vymera_ha/100) AS hustota
        FROM kraje

 id |       NAZKRAJA        |     hustota
----+-----------------------+------------------
  1 | Žilinský kraj         |  101.63730410429
  2 | Trenčiansky kraj      | 131.409609132893
  3 | Prešovský kraj        |   91.16216441848
  4 | Banskobystrický kraj  | 69.3273485799029
  5 | Košický kraj          | 117.867594531481
  6 | Bratislavský kraj     | 304.474747997709
  7 | Nitriansky kraj       | 108.010285854521
  8 | Trnavský kraj         | 134.695166546342
(8 rows)
```

6.2 Vnorené dopyty

Dopyty môžu obsahovať v sebe aj vnorené dopyty. Vnorený dopyt je dopyt v klasickom tvare `SELECT niečo FROM tabul'ka WHERE podmienka`, ktorý však tvorí súčasť iného dopytu. Vnorený dopyt sa môže nachádzať:

- v časti s `UNION`, `INTERSECT`, `EXCEPT`;
- v časti za `FROM` – výber vonkajšieho dopytu bude vztiahnutý na výber z tabuľky, ktorá je výsledkom vnútorného `SELECT`u. Vnorený `SELECT` musí mať svoj alias (pomenovanie za kľúčovým slovom `AS`);
- na mieste stĺpca t.j.:
  `SELECT stlpec,(SELECT vnutorny) FROM tabulka;` – v tomto prípade musí byť vnorený dopyt definovaný tak, že vráti len jeden stĺpec a jeden riadok;
- v podmienke `WHERE` ako súčasť iného vnoreného dopytu s použitím `IN`, `NOT IN` a `EXISTS`;
- v klauzule `WITH`;

Často sa stretávame s potrebou využitia tzv. korelovaného vnútorného dopytu. Na základe vonkajšieho dopytu je definovaný vnútorný dopyt. Ten sa potom chová ako cyklus a pre každý záznam z vonkajšieho dopytu budú vykonané všetky dopyty vnútorného (vnoreného) dopytu.

Príklad 6.6: Našou úlohou je zistiť počet okresov v každom jednom kraji. Máme k dispozícii tabuľku okresov so stĺpcom, ktorý vyjadruje príslušnosť okresu ku kraju a tabuľku krajov. Nasledujúci dopyt vypíše názov kraja a pre ďalší stĺpec použije výsledok vnoreného dopytu. Vnorený dopyt spočíta všetky záznamy okresov, ktoré zodpovedajú kraju z vonkajšieho dopytu. Týmto spôsobom zistíme pre každý jeden kraj počet okresov, ktoré sa v ňom nachádzajú. Úlohu je možné riešiť aj v prípade, že tabuľka okresov neobsahuje informáciu o tom, do ktorého kraja daný okres patrí. Ako jej náhradu môžeme využiť priestorový dopyt (viac na str. 80).

```
SELECT kraje.nazov,(
    SELECT count(okres_id) FROM okresy
    WHERE okresy.kraj=kraje.nazov) AS pocet_okresov
FROM kraje_SR AS kraje;
```

```
     nazov       | pocet_okresov
-----------------+---------------
Bratislavský     |             8
Trnavský         |             7
Trenčiansky      |             9
Nitriansky       |             7
Žilinský         |            11
Banskobystrický  |            13
Prešovský        |            13
Košický          |            11
(8 rows)
```

6.3 Spájanie tabuliek

Výsledky dopytov z rôznych tabuliek je možné kombinovať.

Príklad 6.7: Majme tabuľku zákazníkov a tabuľku zamestnancov. Obe tabuľky obsahujú aj adresu trvalého pobytu. Chceme vypísať všetky mestá, v ktorých žijú zákazníci alebo zamestnanci.

```
SELECT mesto FROM zamestnanci UNION SELECT mesto FROM zakaznici;
```

Klauzula `UNION ALL` zobrazí všetky mestá z oboch tabuliek (pokiaľ sa mesto nachádza v jednej aj druhej tabuľke, je výsledný záznam duplikovaný na rozdiel od použitia `UNION`, kedy sú vypisované len jedinečné záznamy, t.j. rovnaké mesto nachádzajúce sa v oboch tabuľkách, je vypísané len jedenkrát. Pri použití príkazu `INTERSECT` dopyt vracia len tie záznamy, ktoré sa nachádzajú vo výstupe oboch dopytov. Klauzula `EXCEPT` vracia len tie záznamy z dopytu A, ktoré sa nenachádzajú vo výstupe dopytu B.

Klauzula `JOIN` vytvára reláciu medzi dvoma tabuľkami zvyčajne s využitím primárneho a cudzieho kľúča. Existujú nasledujúce varianty:

`LEFT JOIN` – vráti všetky záznamy z prvej tabuľky (M) a práve tie záznamy z druhej tabuľky (N), ktoré zodpovedajú záznamom z prvej tabuľky. Maximálny počet záznamov, ktoré vráti dopyt s klauzulou `LEFT JOIN` bude $m \times n$, kde m je počet riadkov tabuľky M a n je počet riadkov tabuľky N. Počet stĺpcov predstavuje súčet počtov stĺpcov oboch tabuliek.

Príklad 6.8: Majme tabuľky krajov s geometriou a tabuľky krajov s počtom obyvateľov. Na ich spojenie využijeme `LEFT JOIN` nasledovne:

```
test=# SELECT "NAZKRAJA","obyvateLov" AS "obyvatelov",kraj
              FROM hranice_kraj AS hranice
              LEFT JOIN kraje_obyvatelstvo AS obyvatelstvo
              ON rtrim(hranice."NAZKRAJA",' kraj')
              LIKE "kraje_obyvatelstvo"."kraj"

       NAZKRAJA        | obyvatelov |      kraj
-----------------------+------------+-----------------
 Žilinský kraj         |     690449 | Žilinský
 Trenčiansky kraj      |     591233 | Trenčiansky
 Prešovský kraj        |     819977 | Prešovský
 Banskobystrický kraj  |     655359 | Banskobystrický
 Košický kraj          |     795565 | Košický
 Bratislavský kraj     |     625167 | Bratislavský
 Nitriansky kraj       |     684922 | Nitriansky
 Trnavský kraj         |     558677 | Trnavský

 (8 rows)
```

`INNER JOIN` vracia iba záznamy, ktoré sa nachádzajú v oboch tabuľkách. Maximálny počet záznamov, ktoré vráti dopyt s klauzulou `INNER JOIN` bude $m \times n$. Dopyt z predchádzajúceho príkladu upravený na `INNER JOIN` by vrátil iba tie záznamy, pri ktorých sa v štáte nachádza nejaký petrochemický závod.

`RIGHT JOIN` vracia všetky záznamy z tabuľky N a práve tie záznamy z tabuľky M, ktoré obsahujú zodpovedajúce záznamy z tabuľky N. Dopyt sa správa rovnako ako dopyty s klauzulou `LEFT JOIN` s navzájom vymenenými tabuľkami. Práve preto sa používa veľmi zriedkavo (je plne nahraditeľný cez `LEFT JOIN`).

`FULL JOIN` vracia všetky záznamy z tabuliek M a N aj v prípade, že záznam z tabuľky M nemá prislúchajúci záznam v tabuľke N, alebo záznam v tabuľke N nemá svoj prislúchajúci záznam v tabuľke M. V tomto tvare sa spájanie tabuliek používa veľmi zriedkavo.

`CROSS JOIN` predstavuje kartézsky súčin záznamov z tabuľky M a N. Výstupom teda bude tabuľka „každý s každým“.

Spájanie tabuliek je možné reťaziť, t.j. k tabuľke pripojiť tabuľku, ku ktorej je pripojená ďalšia tabuľka (napr. k tabuľke obce pripojím tabuľku okresy, ku ktorej pripájam tabuľku kraje).

Kapitola 7

Kartografické modelovanie

Na modelovanie budeme využívať softvér QGIS s modulmi GRASSu, databázové dopyty budeme realizovať cez modul DB Manager (sú popísané v kapitole 3). Výstupom z každej úlohy je buď mapa alebo tabuľka s objektami s požadovanými vlastnosťami.

7.1 Modelovanie cez tematické mapy

Jedným z najčastejších kartografických výstupov sú tematické mapy zobrazujúce rôzne geografické javy. Často využívanými metódami kartografickej interpretácie sú:

- *metóda kartogramu* pri ktorej sa plošné jednotky vyfarbujú rôznou intenzitou odtieňa v závislosti na relatívnej početnosti ukazovateľa vybraného javu) a
- *metóda kartodiagramu* – do mapy sú vložené grafy početnosti vybraného javu pre prislúchajúcu územnú jednotku.

V oboch prípadoch je na zostrojenie takejto mapy potrebná tabuľka s číselnými hodnotami skúmaného javu.

V GIS je pri vektorových vrstvách grafická informácia (geometria) objektov vždy prepojená s informáciou tabuľkovou, ktorá môže ku každému objektu obsahovať textové alebo číselné údaje. Napríklad, ak máme vrstvu krajov SR, tabuľka ktorá bude súčasťou tejto vektorovej vrstvy, môže mať pre každý kraj informáciu o jeho názve, počte obyvateľov, rozlohe, sídle kraja a ďalšie. Jednotlivé atribúty predstavujú stĺpce tabuľky a riadok tabuľky reprezentuje všetky údaje pre jeden kraj.

Tabuľku atribútov v aplikácii QGIS vyvoláme kliknutím na položku *Otvoriť tabuľku atribútov*, ktorú vyberieme z kontextového menu (objaví sa po kliknutí na vrstvu pravým tlačidlom na vrstvu v paneli *Vrstvy*), alebo na kliknutím na ikonu na paneli nástrojov.

V tomto prípade sa ukazuje koncepcia používaná v GIS pre vektorové vrstvy ako veľmi výhodná. Údaje potrebné k vykresleniu kartogramu či kartodiagramu sú totiž integrálnou súčasťou danej vrstvy, preto tvorba takejto tematickej mapy je naozaj veľmi jednoduchá. Po kliknutí pravým tlačidlom na danú vrstvu v paneli *Vrstvy* a otvorení dialógového okna *Vlastnosti vrstvy* je možné v tomto dialógovom okne na karte *Štýl* nastaviť niekoľko „druhov legendy".

Podľa tohto nastavenia QGIS vykresľuje mapu ako:

- „Jeden symbol", t.j. všetky objekty v danej vrstve používajú rovnakú symboliku (príklad – v prípade plošnej vrstvy „Kraje" budú všetky kraje vyfarbené rovnakou farbou, v prípade bodovej vrstvy „Obce" budú všetky obce zobrazené rovnakým symbolom)

- „Odstupňovaný symbol" – v tomto prípade je možné určiť stĺpec v tabuľke atribútov, ktorý obsahuje číselné hodnoty, podľa ktorých budú objekty v mape rozdelené do niekoľkých tried (skupín), jednotlivé triedy budú zobrazené rôznou symbolikou (príklad 1). Takáto metóda kartografickej interpretácie sa nazýva metóda kartogramu (príklad – ak máme vrstvu okresov, pričom tabuľka atribútov obsahuje pre každý okres aj údaj o počte obyvateľov na km^2, môžeme nastavením druhu legendy „Odstupňovaný symbol" vytvoriť kartogram hustoty obyvateľstva v SR podľa okresov).

- „Kategórie" – v tomto prípade sú objekty zoskupené do tried podľa vybraného stĺpca. Objekty s rovnakou hodnotou v tomto stĺpci budú zobrazené tou istou symbolikou. Napríklad, ak máme vrstvu štátov sveta a v stĺpci tabuľky atribútov je údaj o tom, či je štát vnútrozemský alebo prímorský, môžeme vykresliť mapu tak, že jedným odtieňom farby budú zobrazené štáty vnútrozemské a iným štáty prímorské.

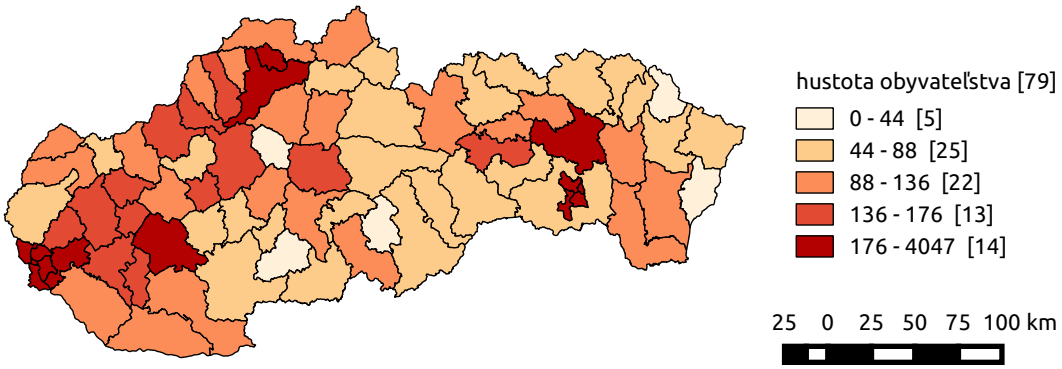
- „Pravdlami podmieňená symbolika" – vykresľovanie je realizované na základe pravidiel (pravdilo možno vytvoriť v špeciálnom editore, kde syntak je veľmi podobná jazyku SQL)

- „Inverzné polygóny" – symbolika je aplikovateľná na vrstvy typu polygón, kde sa vzťahuje na vonkajšok (exterér) polygónu

- „Heatmap" – funguje len pre bodové vrstvy; mapa je vhodná zobrazenie zhlukovania. Pokiaľ máme k dispozíci bodovú vrstvu výskytu nejakého javu, je možné práve touto symbolikou identifikovať ohniská (zhluky) tohto javu v priestore.

7.1.1 Metóda kartogramu

Tvorbu kartogramu (použitie vykresľovania symboliky v režime „Odstupňovaný symbol") ilustruje nasledujúci príklad:



Obr. 7.1: Kartogram

Príklad 7.1: Majme plošnú vrstvu `OKRESY` obsahujúcu v tabuľke atribútov v stĺpci `HUSTOTA` údaje o hustote obyvateľstva za jednotlivé okresy. Úlohou je vytvoriť kartogram hustoty obyvateľstva s 5 intervalmi.

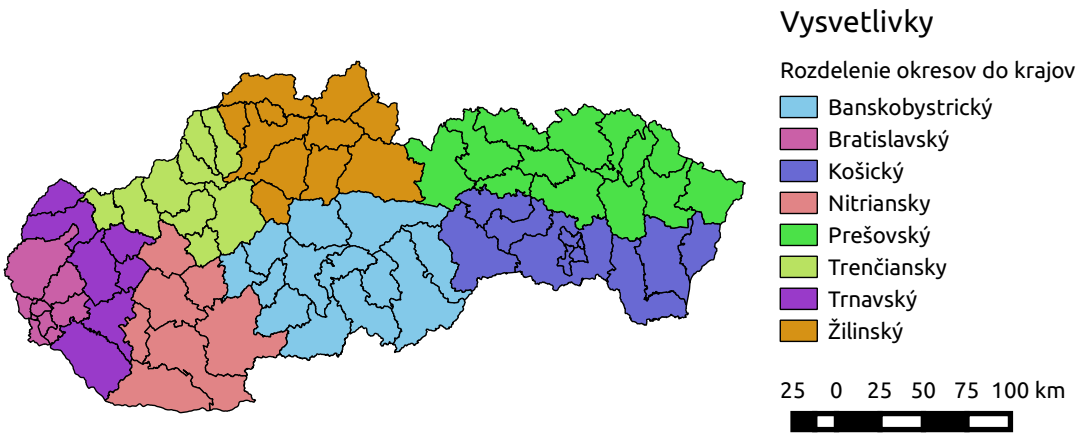
Riešenie: Na vytvorenie takejto tematickej mapy je vhodné využiť druh legendy „Odstupňovaný symbol“. Ako pole (stĺpec) určujúci triedu zvolíme stĺpec `HUSTOTAkliknutím na ikonu`.

7.1.2 Tematická mapa podľa atribútu

Tvorbu tematickej mapy s využitím vykresľovania symboliky v režime „Jedinečná hodnota“ nádejem v nasledujúcom príklade

Príklad 7.2: Majme tematickú plošnú vektorovú vrstvu `Pôdne typy`. Každá plocha tejto vrstvy predstavuje územie jedného pôdneho typu. Pôdny typ územia je zaznačený v tabuľke atribútov v dvoch stĺpcoch – v stĺpci `P_TYP_C` ako číselný kód a v stĺpci `P_TYP_TEXT` ako názov. Našim cieľom je vytvoriť tematickú mapu pôdnych typov, kde budú jednotlivé pôdne typy od seba graficky odlíšené (symbolikou).

Riešenie: Práve v takomto prípade je vhodné použiť na vykreslenie mapy druh legendy „Jedinečná hodnota“. Po jeho zadaní v záložke Symbolika v dialógovom okne *Vlastnosti vrstvy* (okno sa objaví po kliknutí na vrstvu v okne *Vrstvy*) zadáme stĺpec, podľa ktorého budú zostavené triedy objektov (tie sa objavia po kliknutí na tlačidlo „Klasifikovať“). Následne



Obr. 7.2: Tematická mapa

môžeme kliknutím na príslušnú triedu v nižšie zobrazenom zozname upraviť jej symboliku a upraviť jej popis, ktorý sa zobrazí automaticky v legende (vysvetlivkách) k vrstve.

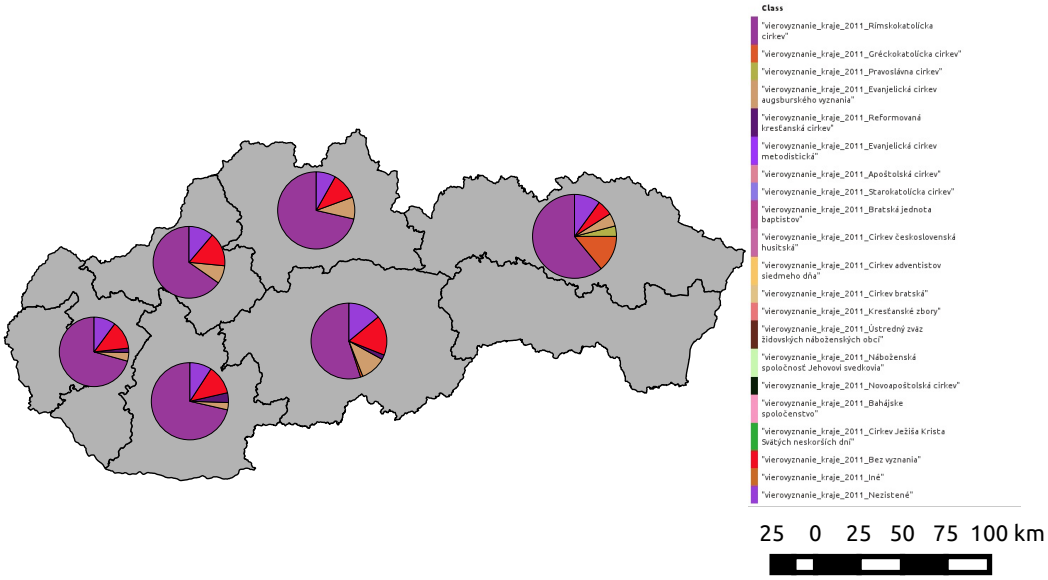
Podobne vieme zobraziť jedným odtieňom všetky okresy, ktoré prislúchajú jednému kraju. Predpokladáme, že máme pre každý jeden uvedenú túto informáciu v stĺpci tabuľky atribútov napr. v stĺpci `kraj` (obr. 7.2).

7.1.3 Metóda kartodiagramu

Kartodiagramy predstavujú technicky zložitejší spôsob vyjadrovania, pretože symbolika je na rozdiel od doteraz spomínaných metód závislá od viacerých stĺpcov. V programe QGIS existuje zásuvný modul „Diagram overlay“, ktorý slúži práve na takýto spôsob kartografickej interpretácie. Po jeho zapnutí (cez menu Zásuvné moduly->Správa zásuvných modulov) sa objaví v dialógovom okne Vlastnosti vrstvy nová záložka „Diagram overlay“. Zaškrtnutím zaškrtávacieho políčka „Zobraziť diagramy“ zapneme zobrazenie kartodiagramov. V nastaveniach tejto záložky sa nastavuje typ grafu (koláčový, stĺpcový, veľkosťou odstupňované symboly vo formáte SVG). V dialógovom okne vyberieme stĺpce pre jednoltivé zložky diagramu (grafu) a tiež stĺpec na základe ktorého hodnoty sa bude škálovať veľkosť diagramu. Výsledok pre *Religióznu štruktúru obyvateľstva podľa krajov (SOBD 2011)* je na obr. 7.3.

7.2 Vzťahy medzi geometriami

S údajmi uloženými v priestorovej databáze môžeme zobrazovať alebo filtrovať podľa určitých podmienok, avšak môžeme z nich odvodzovať aj nové



Obr. 7.3: Kartodiagram

údaje na základe ich priestorových vlastností. Je to podobné ako keď vytvárame dopyt na jednu tabuľku alebo vytvoríme prepojenie (JOIN) viacerých tabuliek. Práca s viac ako jednou geometriou prináša so sebou aj nové konceptuálne výzvy. V databázach s nepriestorovými údajmi údaje interagujú prostredníctvom matematických operácií alebo operácií s reťazcami. Hodnoty číselných atribútov možno porovnávať, sčítavať, odčítavať, násobiť, deliť. Atribúty so znakovými hodnotami môžeme spájať alebo hľadať reťazec jeden v druhom. Pri atribúte, ktorý obsahuje geometriu, je situácia zložitejšia. Predtým než opíšeme každý typ vzájomného vzťahu oddelene je potrebné sa bližšie zoznámiť s pojmom *bounding box*.

Bounding box

Každá geometria obsahuje tzv. bounding box – najmenší obdĺžnik s hranami rovnobežnými, s osami súradnicového systému, ktorý pokrýva celý objekt. Mnoho priestorových úloh možno riešiť len analýzou vzájomnej polohy Bounding boxov dvoch objektov. Napr. pokiaľ by sme chceli zistiť, ktorý štát leží severnejšie, je možné úlohu preformulovať na porovnanie bounding boxov tak, že zistíme odpoveď na pôvodnú otázku. Na porovnanie bounding boxov slúžia porovnávacie znaky uvedené v tabuľke 7.1.

Predtým než budeme skúmať vzájomnú polohu dvoch geometrií (objektov), je potrebné definovať nasledujúce pojmy:

- Vnútro – priestor vnútri geometrie nezahŕňajúci hranicu
- Vonkajšok – priestor mimo geometrie nezahŕňajúci hranicu

Výraz	Význam
&&	Ak 2D Bounding box objektu A pretína 2D Bounding box objektu B.
&&&	Ak 3D Bounding box objektu A pretína 3D Bounding box objektu B.
&<	Ak 2D Bounding box prekrýva zľava alebo sa nachádza vľavo od objektu B.
&<|	Ak 2D Bounding box prekrýva zospodu alebo sa nachádza pod objektom B.
&>	Ak 2D Bounding box prekrýva sprava alebo sa nachádza napravo od objektu B.
<<	Ak 2D Bounding box sa nachádza naľavo od objektu B.
<<|	Ak 2D Bounding box sa nachádza pod objektom B.
=	Ak 2D Bounding box objektu A je totožný s Bounding boxom objektu B.
>>	Ak 2D Bounding box sa nachádza napravo od objektu B.
|>>	Ak 2D Bounding box sa nachádza nad objektom B.
~=	Ak 2D Bounding box obsahuje Bounding box objektu B.
@	Ak 2D Bounding box objektu A je obsiahnutý v Bounding boxe objektu B.
|\&>	Ak 2D Bounding box objektu A prekrýva alebo sa nachádza nad Bounding boxom objektu B.

Tabuľka 7.1: Porovnávacie znamienka geometrie v PostGIS

- Hranica – priestor, ktorý nie je vnútri ani vonku geometrie

Každý bod v rovine s geometriou sa musí nachádzať vnútri, von alebo na hranici objektu. Nemôže ležať na dvoch miestach súčasne. Toto intuitívne platí pre polygóny. Pre otvorené línie tvoria hranicu jej koncové body. Pre uzavretú líniu a body hranica nie je definovaná.

7.2.1 Prienik

Prienikom zisťujeme pri dvoch geometriách, či majú nejaký spoločný bod, t.j. bod vo vnútri alebo na hranici objektu. Množina všetkých spoločných bodov sa nazýva prienik. Dve geometrie sa navzájom pretínajú práve vtedy, ak majú

vo vnútri nejaké spoločné body, avšak nesmú mať všetky spoločné body. Dôsledkom tejto definície je, že geometrie, ktoré sa dotýkajú, sa nemôžu pretínať. V PostGIS existujú dve funkcie na prácu s prienikom. Rozdiel je v návratovej hodnote – kým funkcia `ST_Intersects` vracia hodnotu True alebo False, funkcia `ST_Intersection` vracia geometriu objektu, ktorý tvorí prienik porovnávaných geometrií.

Príklad 7.3: Zaujíma nás, do ktorých okresov zasahuje územie CHKO Poľana. Máme k dispozícii vrstvu (tabuľku) okresov a vrstvu (tabuľku) CHKO.

Riešenie: Výstup nám poskytne nasledujúci dopyt:

```
SELECT okresy.nazov FROM okresy
  CROSS JOIN
      (SELECT the_geom FROM CHKO
          WHERE nazov='Pol'ana') AS chko_polana
  WHERE ST_Intersects(okresy.nazov,chko_polana.geom)
```

7.2.2 Obsahovanie

Pokiaľ geometria A obsahuje geometriu B, znamená to, že žiadny bod z B nemôže ležať vo vonkajšku a najmenej jeden bod z B musí ležať vo vnútri A. Ak B leží len na hranici A, znamená to, že A neobsahuje B. Ak B leží na hranici a vo vnútri A, potom A obsahuje B.

„Obsahuje“ a „je obsiahnutý“ sú navzájom inverzné vzťahy. Ak geometria A je obsiahnutá v geometrii B, potom geometria B obsahuje geometriu A. Na zistenie tejto podmienky existujú funkcie `ST_Contains` a `ST_Within`.

„Obsahuje úplne“ je silnejšou podmienkou ako obsahuje alebo prekrýva. Geometria „obsahuje úplne“ geometriu B ak všetky body z B sú vo vnútri A. Na overenie tohto vzťahu sa využíva funkcia `ST_ContainsProperly`.

7.2.3 Prekrývanie

Dve geometrie sa prekrývajú, keď majú rovnaký rozmer geometrie, pretínajú sa a jedna nie je úplne obsiahnutá v druhej. Slúži na to funkcia `ST_Overlaps`.

Okrem toho existuje aj iný vzťah prekrývnia sa – funkcia `ST_Covers` sa správa rovnako ako `ST_Contains` s výnimkou, že vracia hodnotu TRUE aj v prípade, že jedna geometria leží kompletne na hranici inej. `ST_CoveredBy` je inverzná funkcia funkcií `ST_Covers`. Preto sa správa ako `ST_Within`, ale nepripúšťa hranicu.

7.2.4 Dotýkanie

Dve geometrie sa dotýkajú, ak majú najmenej jeden bod spoločný a žiadny zo spoločných bodov neleží vo vnútri oboch geometrií. Vzťah dotýkania je komutatívny. Pokiaľ sa A dotýka B, potom sa B dotýka A. Tiež platí, že ak sa

dve geometrie dotýkajú, nemôže jedna obsahovať druhú. V rozšírení PostGIS sa na testovanie vlastností dotýkania používa funkcia `ST_Touches`.

Príklad 7.4: S ktorými obcami susedí obec Banská Bystrica?

Riešenie:

```
SELECT "NM2"  FROM hranice_obce
    WHERE
        ST_Touches(
            (SELECT the_geom FROM obce
                WHERE "NM2"='Banská Bystrica'),
            the_geom)
```

Príklad 7.5: Zistite pre každú obec na slovensku počet susedov.

Riešenie:

```
SELECT
  a."nazov",
  (SELECT count(*) FROM hranice_obce_1_mmatis as "b"
      WHERE ST_Touches((a.the_geom),b.the_geom)
  ) AS pocet
FROM obce AS "a"
ORDER BY pocet
```

7.2.5 Disjunktný vzťah

Disjunktný vzťah je negáciou vzťahu prieniku. Dve geometrie sú disjunktné práve vtedy, ak nemajú spoločné body vo vnútri a zároveň nemajú žiadne spoločné body ani na hranici. V PostGIS na testovanie tohto vzťahu existuje funkcia `ST_Disjoint`.

7.2.6 Zhoda

Pri porovnávaní dvoch priestorových objektov má pojem zhodnosť viacero rovín:

- Priestorová zhodnosť znamená, že dve geometrie zaberajú rovnaké miesto.
- Geometrická zhodnosť je silnejšia podmienka priestorovej zhodnosti a znamená, že dve geometrie zaberajú rovnaké miesto a majú rovnakú reprezentáciu.
- Zhodnosť Bounding boxov znamená, že Bounding boxy dvoch geometrií zdieľajú rovnaké miesto. Tento spôsob zhodnosti je testovaný pri použití operátora =.

Na priestorovú zhodnosť sa používa v systéme PostGIS funkcia `ST_Equals`. Funkcia považuje za zhodné geometrie napr.: bod A (subtyp `POINT`), bod A (subtyp `MULTIPOINT` s jediným bodom A), kolekcia geometrií s jediným bodom A. Rovnako bude považovať za identické línie, kde línia číslo 1 ide z bodu A do bodu B a línia číslo 2 ide z bodu B do bodu A. Podmienka geometrickej zhody by tieto prípady kvalifikovala ako rôzne (nezhodné). Na testovanie geometrickej zhodnosti sa používa PostGIS funkcia `ST_OrderingEquals`.

Všetky vyššie spomínané funkcie možno nahradiť univerzálnou funkciou `ST_Relate`, ktorá obsahuje parameter matice priestorových vzťahov. Jej použitie je pomerne zložité, preto sa ňou nebudeme podrobnejšie zaoberať – Ďalšie informácie nájde čitateľ v dokumentácii k PostGIS (The PostGIS Development Group, 2015).

7.3 Tvorba odvodených geometrií

Pri kartografickom modelovaní je možné odvodzovať nové mapy a mapové objekty ich transformáciou z existujúcich máp (geometrií objektov).

7.3.1 Agregačné funkcie

Agregácia je proces zlúčenia viacerých záznamov databázy do jedného záznamu (zlúčenie viacerých priestorových objektov do jedného). Agregácia začína výberom stĺpcov, ktoré budú použité ako kritérium na zlúčenie záznamov. Pri nepriestorových atribútoch sa ako agregačné funkcie používajú najčastejšie `count` (početnosť), `sum` (suma), `min` (minimálna hodnota), `max` (maximálna hodnota), `avg` (priemerná hodnota). Najčastejšie agregačné funkcie na geometriu sú `ST_Makeline`, `ST_Union`, `ST_Collect`, `ST_Polygonize`. Funkcie v súčasnosti pracujú len nad atribútmi typu `geometry` (údaje typu `geography` nie sú podporované).

Pomocou funkcie `ST_Union` môžeme zlúčiť viaceré objekty do jedného. V prípade, že zlučované objekty sú disjunktné (nemajú žiadny prienik), bude výsledkom funkcie kolekcia geometrií (`MULTIPOINT`, `MULTILINESTRING`, `MULTIPOLYGON`, `GEOMETRYCOLLECTION`). Ak nie sú disjunktné a sú rovnakého subtypu, výsledkom bude objekt rovnakého subtypu bez vzájomných hraníc. Napr. zlúčením všetkých obcí (subtyp `POLYGON`) jedného okresu vznikne jeden polygón obsahujúci len územie a hranicu okresu.

Funkcia `ST_MakeLine` je agregačná funkcia, ktorá vytvára línie z bodov. Predpokladajme, že máme k dispozícii vrstvu navštívených bodov a chceme z nej vytvoriť trasu cesty. Body je potrebné pospájať v poradí, v akom boli navštívené (tento údaj musí byť zaznamenaný v jednom zo stĺpcov tabuľky).

Príklad 7.6: Majme v tabuľke skupinu bodov, cez ktoré sme prechádzali. Okrem súradníc sa nachádza v tabuľke aj stĺpec s názvom lokality a časom prechodu. Našim cieľom bude vytvoriť líniu spájajúcu tieto miesta.

Riešenie:

```
SELECT 'putovanie',
        ST_MakeLine(the_geom ORDER by time) AS the_geom
FROM zaznam_cesty_body;
```

7.3.2 Zjednotenie, prienik, rozdiel a symetrický rozdiel

Vystrihnutie záujmovej časti geometrie, resp. odobratie nežiadúcich častí geometrie, sa nazýva *cliping*. Existujú dva spôsoby „vystrihnutia“: rozdiel (funkcia `ST_Difference`) a symetrický rozdiel (`ST_SymDifference`). Rozdiel objektov A a B znamená, že z objektu A odoberieme tie časti, ktoré má spoločné s objektom B ($A -$ prienik(A, B)). Symetrický rozdiel medzi A a B znamená odobratie prieniku A a B zo zjednotenia objektov A a B. Kým rozdiel je nekomutatívnou funkciou, symetrický rozdiel je komutatívny, čo znamená, že nezáleží na poradí objektov. Prienik z dvoch objektov môžeme získať pomocou funkcie `ST_intersection()`.

Príklad 7.7: Praktickým príkladom môže byť identifikácia (vystrihnutie) lesného porastu v CHKO Poľana. Vstupnými objektami budú tabuľka lesných porastov s atribútom geometrie subtyp polygón a tabuľka s geometriou obsahujúca územie chránených krajinných oblastí.

```
SELECT ST_Intersection(lesy.geom,chko_geom) FROM lesy
    CROSS JOIN chko
    WHERE ST_Intersects(lesy.the_geom,chko.the_geom)
          AND chko.name='Pol'ana';
```

Objekt môžeme rozdeliť na základe iného objektu, napr. línia križujúca polygón môže rozdeliť polygón na dva objekty (polygóny). Slúži na to funkcia `ST_Split`. Vstupom do nej môžu byť len jednoduché geometrie, nie kolekcie geometrií. Dimenzia deliaceho objektu musí byť o jeden rozmer nižšia než je dimenzia deleného objektu. Výstupom funkcie `ST_Split` je vždy kolekcia geometrií, ktoré vzniknú delením objektu A. Na delenie objektov, ktoré sú súčasťou kolekcie sa často využíva funkcia `ST_Split` v kombinácii `ST_Dump` alebo `ST_CollectionExtract`.

7.3.3 Rozdeľovanie línií na menšie úseky

Funkcia `ST_Segmentize` slúži na rozdelenie línie na menšie celky danej dĺžky.

Príklad 7.8: Chceli by sme rozdeliť línie vodného toku na úseky maximálne 25 m dlhé.

```
SELECT ST_Segmentize(geom,25) FROM tok;
```

Translácia, škálovanie a rotácie geometrií
Tieto operácie súhrnne nazývame afinné transfomácie v rovine. Je na ne možné použiť funkcie `ST_Translate`, `ST_Scale`, `ST_Rotate`, alebo všeobecnú funkciu `ST_Affine`, ktorá pracuje priamo s maticou afinnej transformácie. Hoci objekty na mape sú zvyčajne statické a neočakávame ich pohyb alebo zmenu polohy, napriek tomu majú tieto funkcie svoje využitie pri:

- Tvorba heatmaps s farebnými variáciami
- Simulácia pohybu pozdĺž cesty
- Simulácia zmeny polohy
- Oprava súradníc geometrie pri posunutých údajoch
- Tvorba rovnobežných línií alebo hrán pri úprave typu objektu cesta z línie na polygón

7.3.4 Úpravy s geometriami objektov

Na prácu s priestorovými objektami slúžia nasledujúce skupiny funkcií (členenie podľa Obe – Hsu (2015)):

- Funkcie na výstup – vrátia priestorové objekty vo vybranom formáte (WKT, WKB, GML, SVG, GeoJson)
- Konštrukčné funkcie – slúžia na vytváranie priestorových objektov v databáze zadaním údajov v požadovanej forme priamo od užívateľa (WKT WKB, GeoJson)
- Funkcie nastavenia a prístupu (pristupujú k objektu a nastavujú jeho atribúty)
- Meracie funkcie – vracajú rôzne skalárne miery priestorového objektu (rozloha, vzdialenosť, obvod...)
- Dekompozičné funkcie rozkladajú (vyťahujú) objekty z priestorového objektu
- Kompozičné funkcie spájajú objekty do väčších priestorových objektov (zložených)
- Funkcie zjednodušenia – generalizujú obsah znížením rozlíšenia priestorových objektov (odobratím redundantných bodov pri danom rozlíšení, zaokrúhlením súradníc...).

Prvé dve skupiny funkcií úzko súvisia s importom a exportom údajov (viď kapitola 4.3). Formáty na vstup a výstup údajov majú svoj význam, pokiaľ sú tieto údaje použité v externej aplikácii (napr. webová aplikácia). Toto použitie je však pre bežného užívateľa zriedkavé, vzťahuje sa na programátorov a vývojárov aplikácií, preto sa mu v ďalšom texte nebudeme venovať. Ostatné skupiny preberieme podrobnejšie.

7.3.5 Funkcie nastavenia a prístupu

Funkcia `ST_SRID(priestorový objekt)` vráti užívateľovi SRID daného priestorového objektu. Funkciou `ST_SetSRID(objekt,SRID)` nastavíme objektu daný priestorový referenčný systém. Je to deklaratívna funkcia, takže pri jej nastavení nedochádza k transformácii medzi súradnicovými systémami, iba ňou deklarujeme aký priestorový referenčný súradnicový systém používa daný objekt.

Funkcia `ST_Transform` slúži na prevod (transformáciu) objektu do iného súradnicového objektu. Používa sa v tvare

```
ST_Transform(objekt::geometry, SRID)
```

Funkcia na rozdiel od funkcie `ST_SetSRID()` vykonáva transformáciu súradníc do nového súradnicového systému. Nie je to deklaratívna funkcia.

Funkcia `ST_GeometryType` slúži na zistenie typu geometrie, čo je užitočné pokiaľ vytvárame dopyt, kde na začiatku nevieme presne aký typ geometrie budú obsahovať objekty, avšak pre vykonanie dopytu bude táto informácia potrebná. Ďalšími užitočnými popisnými funkciami sú funkcie `ST_Dimension` a `ST_CoordDim`. Prvá z nich vracia dimenziu (rozmer) objektu a druhá dimenziu súradnicového systému. Funkcie `ST_X`, `ST_Y` a `ST_Z` vracajú príslušnú časť súradnice pri bodových objektoch. Pri všetkých typoch objektov je možné použiť funkcie `ST_Xmin`, `ST_Xmax`, `ST_Ymin`, `ST_Ymax` a `ST_Zmin`, `ST_Zmax`, ktoré vracajú súradnice ohraničenia objektu v každom smere.

Funkcia `ST_IsValid` (resp. jej popisná varianta `ST_IsValid_Reason`) poskytuje informácie, či je daná geometria validná. Pre akékoľvek analýzy je validita priestorových objektov základnou podmienkou ich úspešného vykonania.

Funkcia `ST_NPoints` vracia počet bodov, ktorými je definovaná geometria objektu. Existuje aj funkcia `ST_NumPoints`, tá však vracia iba počet bodov pri líniových objektoch (počet uzlových bodov línie).

7.3.6 Funkcie miery

Funkcie merania dĺžok a rozlôh sú priamo závislé na mierke a priestore, v ktorom sú merané. Iné hodnoty získame v rovine a iné meraním na referenčnom elipsoide. Pokiaľ meriame veľké vzdialenosti, môže dĺžka v rovine (pri určitom mapovom zobrazení) obsahovať chybu skreslenia. Na meranie slúžia funkcie:

- `ST_Leght()` – vráti dĺžku línie v rovine
- `ST_Area()` – vráti rozlohu plochy objektu v rovine
- `ST_Perimeter()` – vráti dĺžku obvodu objektu v rovine
- `ST_3Dlenght()` – vráti dĺžku línie meranú v 3D priestore

- `ST_3DPerimeter()` – vráti obvod objektu v 3D priestore
- `ST_3DArea()` – vráti rozlohu plochy objektu v 3D priestore

Funkcie vracajú hodnoty v jednotkách súradnicového systému, v ktorom sa nachádzajú objekty, ktoré meriame, t.j. pokiaľ je súradnicový systém v stupňoch, jednotky budú stupne, pokiaľ je súradnicový systém v metroch, jednotky budú metre. To platí pri geometriách typu Geometry. Pri objektoch typu Geography je výsledok dĺžky na referenčnom telese prepočítaný na metre.

Funkcia `Box2D()` a `Box3D()` vracia definičné body boundingboxu.

Funkcia `ST_Boundary()` vracia ohraničenie objektu. Výstupom je objekt o jeden rozmer nižší než bol objekt pôvodný. Ohraničenie polygónu tvorí línia (subtyp `LINESTRING`), ohraničenie línie tvorí jej začiatočný a koncový bod (subtyp `MULTIPOINT`).

Pri polygónoch a líniách sa častokrát využíva akýsi reprezentačný bod daného objektu, napr. na zobrazenie názvu štátu vo vrstve polygónov sa umiestňuje názov na tento bod. Bod určíme ako ťažisko objektu daného polygónom – slúži na to funkcia `ST_Centroid()`. Pri niektorých polygónoch však tento bod leží mimo vnútra polygónu. Na riešenie tohto prípadu slúži funkcia `ST_PointOnSurface()`.

Užitočnou funkciou pri líniách je aj funkcia `ST_PointN(geometria, N)`, ktorá vráti súradnicu N-tého bodu na línii. Pomocou funkcie `ST_DumpPoints` je možné rozložiť geometriu na body, čo je zvlášť dôležité pri použití v štatistickom nástroji R[1]. Funkcia `ST_Dump()` rozkladá geometriu na základné geometrie. Funkcia `ST_GeometryN` vyťahuje z objektu zloženého z kolekcie viacerých objektov N-tý prvok geometrie. Je to obmena funkcie `ST_Dump` avšak výstupom je len jeden objekt (jedna geometria).

7.3.7 Kompozičné funkcie

Nový objekt je možné vytvoriť aj z iných objektov alebo priamo súradníc. Na vytváranie objektu bodu slúži funkcia `ST_Point` a `ST_MakePoint`. Kým `ST_Point` je funkcia len na vytváranie dvojrozmerného bodu, funkcia `ST_MakePoint` vie pracovať aj s ďalšími rozmermi. Po vytvorení bodového objektu je potrebné mu priradiť SRID cez funkciu `ST_SetSRID`.

Funkcia `ST_MakeEnvelope()` vytvorí obdĺžnikový polygón daný súradnicami dvoch bodov a SRID priestorového referenčného systému súradníc týchto bodov.

`ST_MakePolygon` vytvorí polygón z uzavretej línie reprezentujúcej jeho hranicu.

`ST_BuildArea` je rozšírením funkcie `ST_MakePolygon`. Na rozdiel od nej dokáže vytvoriť oblasť aj z geometrií typu `MULTILINESTRING`, `MULTIPOLYGON`, `GEOMETRYCOLLECTION` alebo viacerých objektov typu `POLYGON`.

[1] `www.r-project.org`

Funkcia `ST_Polygonize` funguje v dvoch formách ako agregačná funkcia databázy a tiež ako funkcia, ktorá berie na vstupe pole geometrií typu `LINESTRING` a vracia typ `GEOMETRYCOLLECTION` obsahujúci polygóny.

`ST_Multi` slúži na vytvorenie kolekcií geometrií (`MULTIPOINT`, `MULTIPOLYGON`, `MULTILINESTRING` a `GEOMETRYCOLLECTION`).

7.3.8 Funkcie na zjednodušenie (generalizáciu)

Na generalizáciu objektov slúžia tri funkcie:

- `ST_SnapToGrid` – funkcia zjednodušuje geometriu zaokrúhlením súradníc. Používa sa v tvare:

  ```
  ST_SnapToGrid(geometria, presnost_resp_hustota_mriezky)
  ```

  Príklad 7.9:

  ```
  UPDATE mytable
    SET the_geom = ST_SnapToGrid(the_geom, 0.001);
  ```

- `ST_Simplify` – funkcia generalizuje objekt odobratím uzlov využitím Douglas-Peuckerovho algoritmu. Používa syntax v tvare: `ST_Simplify(geom,tolerancia)`. Čím je väčšia hodnota parametra tolerancia, tým je výstup funkcie viac generalizovaný (väčší počet odobratých uzlov)

- `ST_SimplifyPreserveTopology` – je to obmena funkcie `ST_Simplify` s tým, že zachováva topológiu, t.j. susediace objekty zjednodušuje tak, aby medzi nimi nevzniklo prázdne miesto (podrobnejšie sa topológiou zaoberá kapitola 9).

7.4 Vzdialenostné analýzy v GIS

Pokiaľ máme geografickú databázu naplnenú vhodnými údajmi, môžeme ľahko odpovedať na otázky: ako ďaleko je z nášho domu k najbližšej diaľnici, koľko prevádzok rýchleho občerstvenia nájdeme do dvoch km jazdy autom, koľko km v priemere cestujú ľudia do práce. Všetky tieto otázky možno riešiť cez analýzu blízkosti. Tá sa zaoberá práve problematikou „ako ďaleko je niečo vzdialené od niečoho iného“. Je potrebné pripomenúť, že pri hľadaní vzdialenosti neuvažujeme o prekážokach a obmedzeniach. Ide teda o vzdialenosť vzdušnou čiarou. Hľadanie vzdialeností „po ceste“ sa zaoberá kapitola 9.

Identifikáciu objektov do určitej vzdialenosti zabezpečuje v PostGIS funkcia `ST_DWithin()`. Nasledujúci kód vráti všetky hlavné mestá, ktoré sa nachádzajú do vzdialenosti 50 km od pobrežia:

Príklad 7.10:

```
SELECT DISTINCT mesto,stat FROM hlavne_mesta
    CROSS JOIN pobrezna_ciara
    WHERE ST_DWithin(hlavne_mesta.geom,
                     pobrezna_ciara.geom,50000)
```

Predchádzajúci príklad môžeme rozšíriť tak, že zoznam miest bude usporiadaný podľa vzdialenosti a obmedzený na 5 záznamov.

Príklad 7.11:

```
SELECT DISTINCT mesto,stat FROM hlavne_mesta
    CROSS JOIN pobrezna_ciara
    WHERE ST_DWithin(hlavne_mesta.geom,
                     pobrezna_ciara.geom,50000)
    ORDER BY ST_Distance(hlavne_mesta.geom,
                         pobrezna_ciara.geom)
```

Nájdenie n najbližších miest s použitím KNN vzdialenostných bounding-box operátorov

`<#>` predstavuje KNN operátor vzdialenosti bounding-boxov. `A<#>B` vráti minimálnu vzdialenosť medzi bounding-boxmi objektu A a objektu B.

`<->` predstavuje KNN vzdialenostný operátor centroidu bounding-boxu. `A<->B` vracia vzdialenosť medzi centroidmi bounding-boxu A a B.

Tieto operátory je možné použiť iba s typom údajov `geometry`. Vrátená vzdialenosť zodpovedá skutočnosti len v prípade, že používame priestorový referenčný systém, ktorý zachováva vzdialenosti v rovine. Nasledujúci príklad zobrazuje základné použitie metódy KNN, ktoré využíva priestorový index.

Príklad 7.12: Máme lokalitu, kde by sme chceli kúpiť od mesta pozemok – chceme výpis piatich parciel najbližších k nášmu záujmovému bodu evidovaných ako stavebný pozemok, ktoré patria mestu

```
SELECT cislo,
        geom <-> ST_Point(-1200000,43000000)
FROM parcely
    WHERE typ='stavebny'
    ORDER BY geom<-> ST_Point(-1200000,43000000)
    LIMIT 5;
```

Výhoda KNN operátorov v porovnaní so skutočnou vzdialenosťou je v ich rýchlosti vďaka tomu, že na vyhodnotenie používajú indexy.

7.5 Geotagging

Geotagging patrí do triedy priestorových techník, kde sa snažíme situovať body umiestnené v kontexte inej geometrie. Existujú dve formy:

- Značkovanie regiónom – je to proces, kde označíme geometriu (napr. bod) názvom regiónu, v ktorom sa nachádza (napr. názvom štátu).
- Lineárne referencovanie – je to ďalší spôsob taggovania/označovania špeciálne pre línie, ktoré odkazujeme na bod záujmu takému, ktorý je najbližšie k línii. Značka (tag) môže byť najbližší bod na línii alebo meriame od označníka alebo zlomkom zo začiatku línie do bodu na línii najbližšie k bodu záujmu.

Geotagging a lineárne referencovanie sú bežné úlohy v GIS, pretože tvoria predprípravu pre mnohé štatistické analýzy. Napr. predpokladajme, že máme zoznam všetkých prevádzok rýchleho občerstvenia McDonalds na Slovensku a máme tabuľku všetkých okresov s geometriami. Regionálne geotaggovanie môže zistiť, ktoré okresy obsahujú ktorú prevádzku McDonalds, takže je možné jednoducho získať počet prevádzok v každom okrese. Ak namiesto toho máme zoznam všetkých hlavných ciest (diaľnice, cesty 1. triedy), môžeme lineárnym referencovaním zistiť, ktoré cesty majú prístup k McDonaldu.

Nasledujúci príklad priradí ku každej obci názov okresu, do ktorého patrí. Pracuje pritom s geometriami jednotlivých obcí a okresov.

Príklad 7.13:

```
ALTER table obce ADD COLUMN okres varchar(20);
UPDATE obce SET okres=okresy.okres
FROM okresy
WHERE ST_Intersects(obce.geom,okresy.geom)
```

Kapitola 8

Analýzy nad rastrami

Veľká časť analýz sa týka spojitých vlastností a javov, ktoré sa lepšie modelujú cez rastrový údajový model. Budeme sa im venovať podrobnejšie v tejto kapitole. Medzi základné nástroje na priestorové analýzy nad rastrovým údajovým modelom patria (Tomlin, 1990; Goodchild, 2000):

1. Nástroje na lokálne operácie – nová informácia (nová vrstva) je výsledkom matematických alebo logických operácií nad zodpovedajúcim plošným elementom jednej alebo viacerých vrstiev (rekódovanie, prekrývanie, skalárne operácie, transformácie hodnôt rôznymi funkciami – súborne nazývané mapová algebra)

2. Nástroje na operácie v lokálnom susedstve: hodnoty atribútu plošného elementu pixela novej vrstvy sú určené na základe hodnôt susedných pixelov (filtrovanie, výpočet hodnoty gradientu a smeru vektora gradientu – sklon a orientácie v prípade vrstvy digitálneho modelu reliéfu - nadmorských výšok)

3. Nástroje na operácie vo vzdialenom susedstve – výpočet vzdialeností, výpočet viditeľných plôch (viewshed) výpočet odtokových plôch (watershed)

4. Nástroje na operácie nad zónami: Identifikácia zón, výpočet plochy a obvodu zón, stanovenie vzdialenosti od okraja zóny, určenie tvaru zóny,

5. Nástroje na popis obsahu vrstiev: Výpočet štatistických ukazovateľov hodnôt atribútov jednej vrstvy (priemer, modus, medián), porovnanie hodnôt atribútov viacerých vrstiev (χ kvadrát, regresná analýza, analýza variácií), štatistické charakteristiky zón v jednej vrstve (počet zón, určenie najmenšej a najväčšej zóny, priemerná plocha zóny).

8.1 Základná práca s rastrami

V rastrovom údajovom modeli je informácia uložená v podobe pravidelnej mriežky buniek, pričom každá bunka obsahuje svoju hodnotu. V ďalšom texte sa budeme zaoberať len rastrami v 2D, t.j. rastrami definovanými v rovine. Pravidelnú mriežku buniek je možné vytvoriť aj v trojrozmernom priestore, v tomto prípade sa bunka nazýva voxel. Bunka nesie vždy iba číselnú hodnotu. Prakticky sa môžeme pri rastroch stretnúť s usporiadaním informácie do viacerých kanálov. V tomto prípade každý kanál obsahuje mriežku buniek, t.j. jedna bunka nadobúda svoju hodnotu pre každý jeden kanál (tá môže byť navzájom rôzna).

Rastrové údaje je možné uložiť priamo v databáze PostgreSQL s rozšírením PostGIS alebo je možné do databázy vložiť len referencie na dané údaje, v tom prípade je potrebné, aby sa údaje (súbory) nachádzali na dostupnom mieste a aby ich databázový systém PostgreSQL vedel čítať.

Rastrové údaje rovnako ako vektorové sa musia vzťahovať k určitému súradnicovému systému. Proces priradenia surového obrázka (snímky) do určitého geografického alebo kartografického súradnicového systému sa nazýva polohové priradenie alebo georeferencovanie rastra. Veľkosť každého pixela preto môžeme vyjadriť v jednotkách súradnicového systému (napr. V stupňoch alebo metroch).

8.1.1 Dlaždice

Jednou z veľkých nevýhod práce s rastrami je ich náročnosť na pamäť. Rastre s vyšším rozlíšením sa veľmi ťažko spracúvajú alebo ďalej upravujú. Jedným z riešení je využívanie tzv. Dlaždíc. Dlaždica (po anglicky tile) je raster malých rozmerov, ktorý je súčasťou väčšieho celku. Skupina dlaždíc tvorí celý obraz (soverage).

Už pri importe rastrov do databázy je potrebné rozložiť vstupný raster na dlaždice požadovanej veľkosti:

```
raster2pgsql -s 4326 -I -C - M *.bil -F -t 256x256 rastre
```

Ďalšou možnosťou je využiť na generovanie dlaždíc funkciu `ST_Tile` v tvare:

```
ST_Tile(raster,
x_velkost'_dlazdice,
y_vel'kost'_dlaždice,
povolit_doplnenie_nodata_udajov)
```

Príklad 8.1: Vložíme do tabuľky rastre raster vytvorený z dlaždíc 256×256 pixelov.

```
INSERT INTO rastre (rast,id)
    SELECT ST_Tile(rast,256,256,true),id
```

```
FROM rastre
WHERE rast_name='snimka_vsetko'
```

Niekedy je potrebné vnímať dlaždice ako jeden veľký raster, čo je možné dosiahnuť cez funkciu `ST_Union` (funguje rovnako aj pri typoch geometrie geometry). Jej použitie ilustruje nasledovný príklad:

```
SELECT ST_Union(raster),count(rast) AS povodnych_dlazdnic
FROM rastre;
```

8.1.2 Vytváranie rastrov v databáze

Vytvorenie rastra v databázovom systéme PostgreSQL s rozšírením PostGIS môžeme rastre ukladať v tabuľkách. Jeden záznam tabuľky reprezentuje jednu rastrovú vrstvu. Na tento prístup si je potrebné zvyknúť, keďže pri vektorových vrstvách jeden záznam tabuľky reprezentuje jeden konkrétny objekt. Pri rastrovej vrstve jeden záznam reprezentuje zvyčajne celé územie a jednotlivé triedy objektov sa odlišujú len na základe vlastností hodnoty pixela.

Rastre je možné vytvoriť nasledujúcim spôsobom:

- Konverzia z iných typov PostGIS geometrie na rastre pomocou funkcie `ST_ASraster`

  Príklad 8.2: Nahratie rastra pre ďalšie analýzy získaného z vektorovej vrstvy využitie Zeme

- Nahratím rastrov do databázy cez nástroj `raster2pgsql` (viď príklad na str. 4.3)

- Vytvorenie rastrov pomocou funkcií `ST_makeemptiraster` a `ST_AddBand` a následné priradenie hodnôt pixlom použitím rôznych rastrových funkcií

  Príklad 8.3: Prázdny raster 500x500 pixelov v súradnicovom systéme s EPSG:5514 (Krovák) s jedným kanálom a hodnotou všetkých buniek 255 vytvoríme nasledovne:

```
INSERT INTO rastre(nazov,rast)
  SELECT 'novy_raster',
          ST_AddBand(
          ST_MakeEmptyRaster(
              500,500,-400500,-1200500,
              -400000,-1200000,0,0,5514
                ),'8BUI'::text,255,0) AS rast
```

- Vytvorenie rastrov z existujúcich rastrov s použitím funkcií na zjednotenie, reklasifikáciu, prevzorkovanie, mapová algebra a ďalšie.

- Použitie funkcie `ST_FromGdalRaster` – v tomto prípade raster ostane uložený v pôvodnom rastrovom formáte.

8.1.3 Priraďovanie hodnôt pixelom

Na priradenie hodnôt slúžia funkcie `ST_SetValue` a `ST_SetValues`. `ST_SetValue` sa používa v tvare:

```
ST_SetValue(nazov_rastra, cislo_kanala, stlpec, riadok, nova_hodnota)
```

Napríklad:

```
UPDATE rastre
SET rast=STSetValue(rast,1,50,50,0)
WHERE rast_name="landuse"
```

`ST_setvalues` (dostupné od verzie PostGIS 2.1) dovoľuje nastaviť hodnoty všetkých pixlov, ktoré sú v prieniku s množinou objektov Geomval. Geomval je kompozitný údajový typ zložený z geometrie a hodnoty s plávajúcou desatinnou čiarkou.

8.1.4 Výrezy z rastra

Výrez z rastra je možné robiť podobne ako pri iných typoch geometrie prostredníctvom funkcie `ST_Clip`. Ako šablónu na vyrezanie je možné použiť aj objekt s iným typom geometrie. Napr. môžeme spraviť výrez rastra využitia Zeme podľa katastrálneho územia obce a výsledok uložiť ako nový raster v tabuľke rastre.

```
INSERT INTO rastre (Rast_Name,Rast)
SELECT "vystrihnutý_raster_využitia_Zeme",
        ST_Clip(rast,1,SelectGeomFromObceWhereName=Tajov),
        255)
FROM rastre
WHERE Rast_Name=vyuzitiezeme
```

Hodnota parametra 255 je voliteľná a reprezentuje hodnotu pre pixely bez hodnoty.

Funkcia `ST_Band` slúži na výber jedného alebo viacerých kanálov rastra na vytvorenie nového rastra s rovnakými rozmermi, súradnicami, priestorovým referenčným systémom a hodnotami pixelov. Funkcia sa dá použiť aj na preusporiadanie kanálov alebo v kombináciami s funkciou `STaddBand` na vytvorenie viackanálových rastrov z jednokanálových.

8.1.5 Metaúdaje

Z každého rastra je možné získať nasledujúce metaúdaje:

- `ST_Width()` vracia šírku rastra v pixeloch
- `ST_Height()` vracia výšku rastra v pixeloch

- `ST_Pixelvidth()` vracia veľkosť pixelu v x-ovom smere (šírka pixela) v jednotkách priestorového referenčného systému
- `ST_Pixelheight()` vracia veľkosť pixelu v y-ovom smere (výška pixela) v jednotkách priestorového referenčného systému
- `ST_SRID()` vracia identifikátor priestorového referenčného systému rastra
- `ST_BandPixelType` vracia doménu hodnôt bunky rastra

Ďalšie metaúdaje je možné získať cez funkcie `ST_MetaData` a `ST_BandMetaData`.

8.2 Analýza blízkeho susedstva

8.2.1 Výpočet morfometrických charakteristík reliéfu

Z rastrovej vrstvy reprezentujúcej DTM môžeme vypočítať morfometrické charakteristiky prvého[1] a druhého[2] rádu. Na výpočet nám poslúžia moduly GRASSu. vrstvy morfometrických parametrov môžu byť priamo čiastkovými výstupmi interpolačného modulu `v.surf.rst`. Alternatívnym riešením je využiť modul `r.slope.aspect` s nasledujúcimi parametrami[3]:

- *Name of elevation raster map* – názov vstupnej rastrovej vrstvy (DTM).
- *Name of output slope raster map* – názov výstupnej rastrovej vrstvy s hodnotou sklonu georeliéfu v smere maximálneho spádu.
- *Name of output aspect raster map* – názov výstupnej rastrovej vrstvy s hodnotou orientácie georeliéfu voči svetovým stranám[4].
- *Name of profile curvature raster map* – názov výstupnej rastrovej vrstvy s hodnotou normálovej krivosti v smere dotyčnice k spádnici.
- *Name of tangential curvature raster map* – názov výstupnej rastrovej vrstvy s hodnotou normálovej krivosti v smere dotyčnice k vrstevnici.
- *Name for output first order partial derivative dx(E-W slope)raster map* – názov výstupnej rastrovej vrstvy s hodnotami prvej smerovej derivácie dx v smere východ-západ.

[1] Sklon georeliéfu v smere maximálneho spádu a orientácia georeliéfu voči svetovým stranám

[2] Normálové krivosti, t.j. normálová krivosť plochy v smere dotyčnice k spádnici a dotyčnicová krivosť, t.j. normálová krivosť plochy v smere dotyčnice k vrstevnici

[3] prehľad parametrov je prebratý z publikácie (Burian et al., 2015)

[4] Východnej orientácii je priradená hodnota 0, resp. 360°, severnej orientácii je priradená hodnota 90 st, západnej orientácii je priradená hodnota 180°a južnej orientácii je priradená hodnota 270°. Ak bol použitý národný geodetický referenčný systém S-JTSK, vypočítané hodnoty orientácie reliéfu voči svetovým stranám je potrebné v mapovej kalkulačke opraviť o hodnotu meridiánovej konvergencie

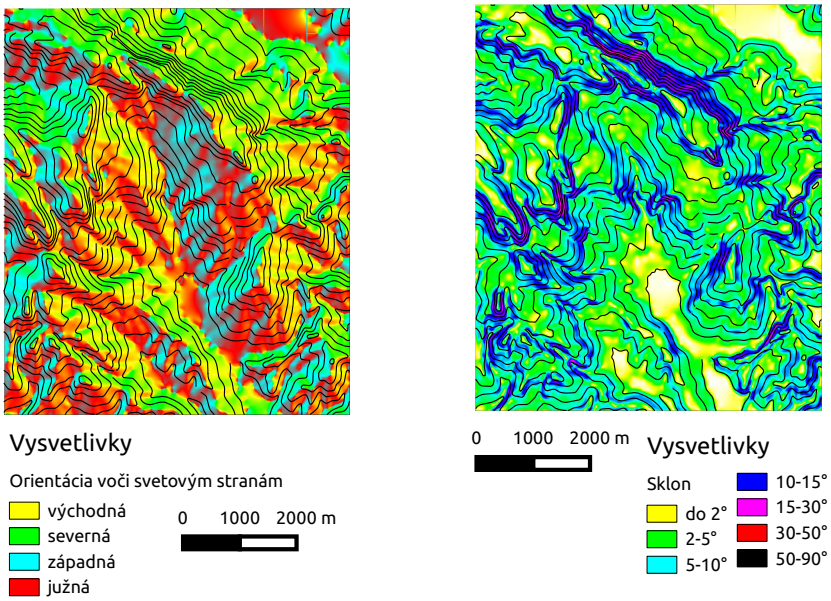
- *Name for output first order partial derivative dy(N-S slope)raster map* – názov výstupnej rastrovej vrstvy s hodnotami prvej smerovej derivácie dy v smere sever - juh.
- *Name for output second order partial derivative dxx raster map* – názov výstupnej rastrovej vrstvy s hodnotami druhej smerovej derivácie dxx.
- *Name for output second order partial derivative dyy raster map* – názov výstupnej rastrovej vrstvy s hodnotami druhej smerovej derivácie dyy.
- *Name for output second order partial derivative dxy raster map* – názov výstupnej rastrovej vrstvy s hodnotami druhej smerovej derivácie dxy.
- *Format for reporting the slope* – definuje formát vypočítanej hodnoty sklonu georeliéfu v smere maximálneho spádu - možné hodnoty sú Degrees – stupne alebo Percent – percentá
- *Type of output aspect and slope maps* – definuje spôsob zaokrúhľovania hodnôt sklonu a orientácie v ich výstupných rastrových vrstvách. Prípustné hodnoty sú Float – desatinné číslo; Double – desatinné číslo s dvomi desatinnými miestami; Integer – celé číslo
- *Minimum slope val. (in percent) for which aspect is computed* – minimálna hodnota sklonu, pre ktorú bude vypočítaná orientácia georeliéfu voči svetovým stranám. V prípade nastavenej hodnoty 1 bude algoritmus ignorovať všetky svahy so sklonom menším ako 1 % a priradí im nedefinovanú hodnotu orientácie voči svetovým stranám.
- *Allow output files to overwrite existing files* – prepísanie vrstvy s rovnakým názvom.

Niektoré morfometrické charakteristiky, ako je napr. horizontálna krivosť plochy (plan curvature)[5], sa dajú vypočítať prostredníctvom modulu `r.param.scale`. Tento modul využíva na výpočet morfometrických charakteristík aproximačnú matematickú funkciu definovanú len v malej vymedzenej oblasti. Modul pracuje s postupnosťou nasledujúcich krokov:

1. Výber bunky rastra, pre ktorý bude výpočet priestorovo platný.
2. Výber okolia bunky, v ktorom bude prebiehať aproximácia.
3. Aproximácia pomocou metódy najmenších štvorcov.
4. Výpočet zvolenej morfometrickej charakteristiky.
5. Posun na ďalšiu bunku (návrat k bodu 1.).

Ukážky vybraných parametrov sú na obr. 8.1.

[5]Horizontálna krivosť vyjadruje krivosť horizontálneho (nie normálového) rezu plochy, teda krivosť vrstevnice v mape



(a) Orientácie (b) Sklony

Obr. 8.1: Vybrané morfometrické parametre

8.3 Lokálne operácie

8.3.1 Mapová algebra

Pojmom mapová algebra sa vo všeobecnosti označujú postupy, ktoré umožňujú vykonávať logické, aritmetické a štatistické operácie s jednou alebo viacerými rastrovými vrstvami. Mapovú algebru je možné realizovať pomocou funkcie `ST_MapAlgebra()`. Operácie je možné vykonávať na jednej vrstve (napríklad zdvojnásobiť hodnotu každého pixela) na jednej vrstve medzi rôznymi kanálmi (napríklad odčítať kanály obsahujúce priemernú teplotu za rôzne mesiace) alebo medzi viacerými rôznymi rastrami. Pri práci s funkciou `ST_MapAlgebra` máme na výber medzi zadaním výrazu obsahujúceho operáciu alebo zavolaním funkcie. Pod funkciou si môžeme predstaviť vzorec podobne ako je definovaný v exceli. Túto funkciu je potrebné definovať v jazyku PL a je ju možné použiť v ľubovoľnom dopyte.

V príklade budeme vychádzať z učebného materiálu Burian et al. (2015), v ktorom bol predstavený postup na výpočet Langovho dažďového faktora s využitím mapovej algebry systému GRASS GIS. Tento postup zopakujeme s využitím údajov uložených v databáze PostgreSQL s rozšírením PostGIS.

Raster predstavuje z matematického hľadiska maticu hodnôt. Súčet dvoch rastrových vrstiev sa realizuje ako súčet dvoch matíc. Ak napr. chceme vypočítať hodnoty Langovho dažďového faktora, ktorý je rovný podielu zrážok a teploty vzduchu, potrebujeme na realizáciu výpočtu dva vstupné rastre (raster priemernej teploty vzduchu a raster príslušného úhrnu zrážok). Rastre importujeme do databázy PostgreSQL s rozšírením PostGIS (viď kapitola 4.3 na

str. 49).

Následne vytvoríme nový raster, ktorý naplníme dopytom s využitím mapovej algebry (funkcia `ST_MapAlgebra()`). Samotný postup výpočtu Langovho faktora sa skladá z nasledujúcich krokov:

1. Výber hodnoty bunky s pozíciou 1. riadok a 1. stĺpec v oboch vrstvách vstupujúcich do výpočtu.
2. Výpočet výslednej hodnoty a jej priradenie bunke nového rastra s tou istou pozíciou.
3. Výber hodnoty bunky s pozíciou 1. riadok a 2. stĺpec v oboch vrstvách vstupujúcich do výpočtu.
4. Výpočet výslednej hodnoty a jej priradenie bunke nového rastra s tou istou pozíciou.
5. Výber hodnoty bunky s pozíciou 1. riadok a 3. stĺpec v oboch vrstvách vstupujúcich do výpočtu.
6. ...atď.

Výpočet realizujeme nasledovne:
Najskôr vytvoríme tabuľku pre nový raster:

```
CREATE table lang_koef (rid integer,rast raster);
```

Pokračujeme vložením údajov:

```
INSERT INTO lang_koef (rid, rast)
SELECT rid,
  ST_MapAlgebra( ARRAY[ROW(teploty.rast, 1),
  ROW(zrazky.rast, 1), ]::rastbandarg[],
  '[zrazky.rast.val]/[teploty.rast.val]',-999)
FROM teploty CROSS JOIN zrazky ON
```

Medzi základné operátory patria:

- $+$ súčet
- $-$ rozdiel
- $*$ súčin
- $/$ podiel
- $>$ väčší ako
- $>=$ väčší alebo rovný ako
- $<$ menší ako

- $<=$ menší alebo rovný ako
- $==$ rovný
- $! =$ nerovný
- ^ mocnina

Príklady výrazov:

- $a + b$ súčet hodnôt buniek rastra a a rastra b,
- $a - b$ rozdiel hodnôt buniek rastra a a rastra b,
- $(a*a)-2*b$ rozdiel druhej mocniny hodnôt buniek rastra a a dvojnásobku hodnôt buniek rastra b,
- $\cos(a)$ kosínus hodnôt buniek rastra a,
- $if(a > 20) = 2$ podmienka, ktorá znamená, že ak hodnota bunky rastra a je väčšia ako 20, hodnota bunky výstupného rastra bude 2.

8.3.2 Plávajúce okno a fokálna štatistika

Fokálna štatistika je založená na operáciách, ktoré sa uskutočňujú nad rastrovou vrstvou, v tzv. plávajúcom okne. To znamená, že po uskutočnení operácie (výpočtu) v zadefinovanom okolí skúmanej, to znamená centrálnej bunky rastrovej vrstvy, sa centrálnou bunkou stáva susedná bunka. Algoritmus fokálnej štatistiky v GRASS GIS je založený na opakovaní nasledujúcich krokov:

1. Vytvorenie okna s rozmermi $n \times n$ buniek, pričom skúmaná bunka sa nachádza v strede tohto okna (centrálna bunka).
2. Výpočet v rámci vytvoreného okna.
3. Priradenie vypočítanej hodnoty bunke výstupného rastra ležiacej nad centrálnou bunkou.
4. Presun do nasledujúcej bunky (posun okna).
5. Nová bunka preberá úlohu centrálnej bunky.

Výpočet fokálnej štatistiky je v súčasnej verzii PostGIS príliš komplikovaný, čitateľa odporúčame na publikáciu PostGIS in Action (Haining, 2003).

8.4 Analýza vzdialeného susedstva

dĺžky. Modul r.sun funguje v dvoch režimoch:

Hydrologické modelovanie

Do prostredia GRASS GIS je implementovaných niekoľko modulov, ktoré sa využívajú pre potreby hydrologických analýz. DTM, ktorý vstupuje do výpočtov realizovaných v rámci týchto modulov, musí spĺňať podmienku sprietočnenia. Sprietočnenie modelu spočíva v úpravách hodnôt nadmorskej výšky v tých bunkách DTM, ktoré vytvárajú prekážku pre odtok zrážkovej vody. Táto situácia nastáva často, pretože vstupné bodové polia nadmorských výšok sú väčšinou získavané metódami, ktoré nerešpektujú kostru georeliéfu. Niektoré vlastnosti interpolačných funkcií taktiež podporujú vznik tvarov reliéfu, ktoré vytvárajú prekážku pre odtok zrážkovej vody, napr. vznik falošných singularít uvádzaný pri popise parametra tenzie a zhladenia (časť 5.2.6, str. 63).

Chyby môžeme podľa príčiny vzniku rozdeliť do niekoľkých kategórií:

1. Vlastnosť reálneho terénu, ktorý na základe existencie depresií, resp. bariér, nie je vždy hydrologicky korektný. Pre potreby hydrologického modelovania je nutné upraviť ho.

2. Chyby vyplývajúce z merania, ktoré vznikajú v procese získavania priestorových informácií.

3. Chyby interpolačného algoritmu.

Korekčné algoritmy slúžiace k sprietočneniu DTM sú založené na princípe vyhľadávania odtokovej bariéry alebo depresie a následnom zrezaní alebo vypĺňaní týchto morfologických porúch..

V GRASS GIS zakomponovaný modul `r.fill.dir` preferuje druhú uvádzanú možnosť. Tento modul okrem toho, že upravuje DTM na hydrologicky korektný model, dokáže pre každú bunku rastra vypočítať gravitačne podmienený smer toku látky. Táto rastrová vrstva je vyžadovanou vrstvou pre ďalšie hydrologické analýzy.

Modul `r.fill.dir` je potrebné nastaviť nasledujúce parametre:

- *Name of existing raster map containing elevation surface* – názov vstupnej rastrovej vrstvy (DTM),

- *Output elevation raster map after filling* – názov výstupnej rastrovej vrstvy po hydrologických korekciách (sprietočnený DTM),

- *Output direction raster map* – názov výstupnej rastrovej vrstvy s hodnotami smeru toku v jednotlivých bunkách.

Jedným z dôležitých parametrov pre hydrologické modelovanie alebo modelovanie erózno-akumulačných procesov je príspevková plocha. Príspevková plocha sa dá s určitou mierou zovšeobecnenia interpretovať ako plocha na svahu, z ktorej všetka voda (látka) odteká do skúmaného bodu. V rámci rastrovej vrstvy DTM, príspevkovú plochu skúmanej bunky tvoria všetky bunky, v ktorých gravitačne podmienený smer toku hmoty spôsobuje, že prenášaný materiál sa dostane do skúmanej bunky. Ľubovoľná bunka, ktorá je súčasťou

príspevkovej plochy skúmanej bunky, nemôže byť zároveň súčasťou príspevkovej plochy inej bunky, ktorá nebola súčasťou predchádzajúcej príspevkovej plochy.

Takto chápaný výpočet príspevkovej plochy umožňuje realizovať modul `r.terraflow`. Tento modul počíta aj topographic convergence index (TCI), nazývaný tiež aj topographic wetness index (TWI) alebo compound topographic index (CTI). Je definovaný nasledujúcou rovnicou:

$$TCI = \ln(\alpha \tan(\beta)) \tag{8.1}$$

kde α – príspevková plocha, β – sklon georeliéfu v smere maximálneho spádu. Je možné nastaviť nasledujúce parametre:

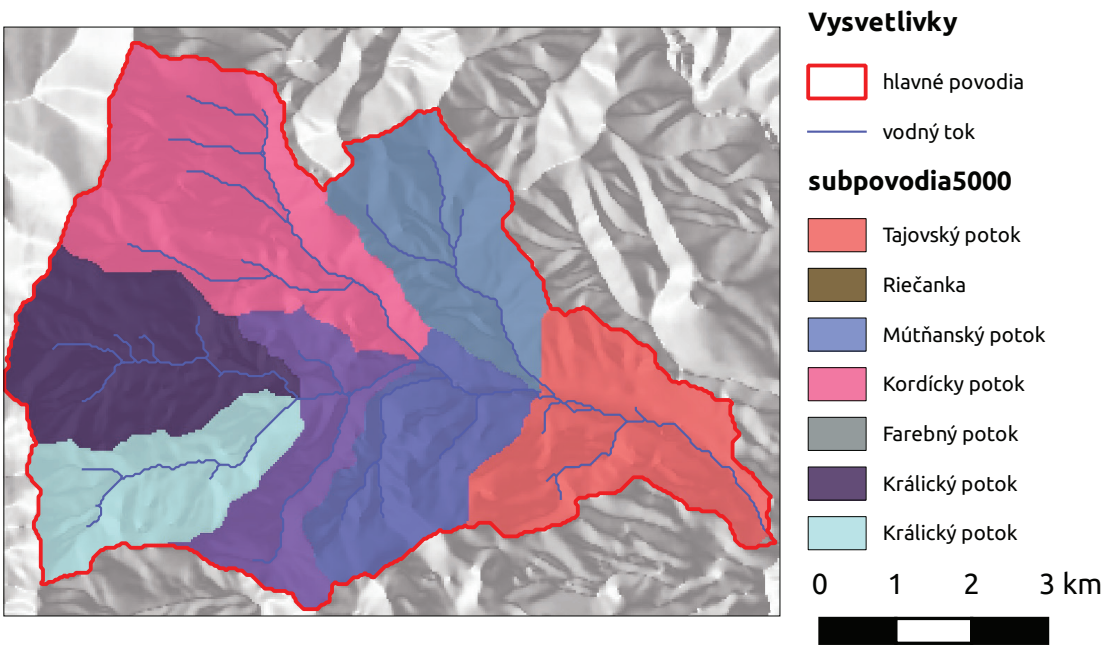
- *Name of elevation raster map* – názov vstupnej rastrovej vrstvy s hodnotami nadmorskej výšky (DTM),
- *Name for output filled (flooded) elevation raster map* – názov výstupnej rastrovej vrstvy po hydrologických korekciách,
- *Name for output flow direction raster map* – názov výstupnej rastrovej vrstvy s hodnotami smeru toku v jednotlivých bunkách,
- *Name for output sink-watersheed raster map* – názov výstupnej rastrovej vrstvy s vyčlenenými povodiami,
- *Name for output flow accumulation raster map* – názov výstupnej rastrovej vrstvy s hodnotami počtu prispievajúcich buniek (po prenásobení veľkosťou bunky ide o príspevkovú plochu),
- *Name for output topographic convergence index (tci) raster map* – názov výstupnej rastrovej vrstvy s hodnotami indexu TCI.
- *Allow output files to overwrite existing files* – prepísanie už existujúcej vrstvy v databáze GRASS GIS s rovnakým názvom,
- *Name of file containing runtime statistics* – názov súboru obsahujúceho základné štatistické údaje.

Názov je potrebné pred každým výpočtom zmeniť.

Ďalším modulom umožňujúcim realizáciu hydrologických analýz je modul `r.flow`, ktorý slúži na výpočet priebehu spádových kriviek. Na tvorbu spádových kriviek je využitý nasledujúci algoritmus vyhľadávania ich lomových bodov: Počet buniek, cez ktoré prechádza spádová krivka, definuje parameter dĺžky spádovej krivky.

Modul obsahuje nasledujúce parametre:

- *Input elevation raster map* – názov vstupnej rastrovej vrstvy s hodnotami nadmorskej výšky (DTM).



Obr. 8.2: Čiastkové povodia vygenerované cez *r.watershed*

- *Compute upslope flowlines instead of default downhill flowlines* – po aktivovaní tohto nastavenia bude modul generovať spádové krivky v smere proti spádu ("hore kopcom"),
- *Input aspect raster map* – názov vstupnej rastrovej vrstvy orientácie georeliéfu voči svetovým stranám,
- *Number of cells between flowlines* – počet buniek medzi východiskovými bodmi po sebe modelovaných spádových kriviek,
- *Output flowline vector map* – názov výstupnej vektorovej vrstvy so spádovými krivkami,
- *Output flowpath length raster map* – názov rastrovej vrstvy s hodnotami vzdialenosti bodu od začiatku teoretického toku hmoty,
- *Output flowline density raster map* – názov rastrovej vrstvy s vypočítanou hustotou spádových kriviek.

8.4.1 Analýza viditeľnosti

Analýza viditeľnosti patrí medzi analýzy vzdialeného susedstva. Slúži na zistenie dohľadnosti medzi dvomi bodmi územia. Analýza pracuje nad rastrovým údajovým modelom a neberie do úvahy zakrivenie Zeme, takže je možné ju realizovať len na plochách, kde sa zakrivenie Zeme prejavuje len minimálne.

V systéme GRASS GIS existuje na jej výpočet modul `r.los`(Line of Sight). Vstupnými údajmi modulu sú:

- Raster výšok
- Súradnice pozorovateľa
- Maximálna vzdialenosť, do ktorej bude analýza vykonávaná (prirodzená dohľadnosť),
- výška pozorovateľa nad povrchom Zeme
- binárna rastrová vrstva využiteľná ako maska

Výstupom z modulu je rastrová vrstva, ktorá v každej bunke obsahuje hodnotu uhla, pod ktorou je z miesta pozorovateľa viditeľná. Môže nadobúdať hodnoty od 0 do 180, pričom hodnota 0 znamená kolmo dole pod pozorovateľom, hodnota 90 znamená priamy pohľad kolmý na os pozorovateľa a hodnota 180 predstavuje smer priamo nad pozorovateľa.

Pomocou analýzy viditeľnosti je možné riešiť množstvo praktických problémov – je možné napr. skúmať historicko-geografický kontext mnohých slávnych bitiek a odpovedať na otázky: bolo možné nepriateľa zazrieť v dostatočnom predstihu. Je tiež možné optimalizovať rozmiestnenie vysielačov v krajine tak, aby medzi nimi nevznikali žiadne bariéry. Tiež je možné navrhnúť trasu s najlepším výhľadom na krajinu.

Kapitola 9

Sieťové analýzy

K problematike vzťahov medzi geografickými údajmi sa viažu pojmy topológia a topologické vzťahy nad ktorými je vybudovaná oblasť tzv. sieťových analýz. Kapitola vysvetľuje základné princípy a úlohy, ktoré súvisia s touto skupinou analýz.

9.1 Topológia

Koncept topológie sa týka priestorového usporiadania geografických objektov a geometrických vzťahov medzi nimi. Termín topológia pochádza z gréčtiny a znamená "Štúdium tvarov". Prvýkrát bol použitý v roku 1736 matematikom Leonardom Eulerom pri navrhovaní riešenia známej matematickej úlohy Königsberských mostov. Neskôr v roku 1970 štatistický úrad USA vniesol koncept matematickej topológie do máp s cieľom redukovať chyby pri spracovávaní rozsiahlych súborov zo sčítania obyvateľstva. Topológia predstavuje matematický prístup, ktorý vyjadruje vzťahy medzi priestorovými objektami na základe ich priľahlosti (v susedstve) a prepojenosti (konektivity).

Topologické vlastnosti objektov v euklidovskom priestore sú napr.:

- Bod je koncovým bodom línie
- Bod leží na hranici polygónu
- Línia pretína polygón
- Polygón susedí s polygónom
- Polygón neobsahuje iný polygón a pod.

Naproti tomu medzi netopologické vlastnosti patria:

- Vzdialenosť medzi bodmi
- Uhol odklonu jedného bodu od druhého

- Dĺžka línie
- Plocha a obvod polygónu

Pri použití topologického údajového modelu sa pri zmene hraníc jednej parcely automaticky mení hranica druhej parcely, keďže majú spoločnú hranicu. Podobne pri križovaní dvoch ciest vznikne na ich krížení priesečník. Netopologický údajový model (napr. špagetový) sa pozerá na objekty nezávisle, t.j. pri zmene jedného objektu sa nemení automaticky objekt s ním susediaci. Topologický údajový model nám pomáha pri udržiavaní validity údajov - nie je v ňom možné zdigitalizovať tak, aby mali nechcený presah (overlap) alebo medzeru (gap). Použitie topológie v GIS je založené na matematickej disciplíne teórií grafov a preto sa tu používa iná terminológia. Základnými komponentami topologickej údajovej štruktúry (angl. primitive) sú:

- Uzly (angl. Node) – predstavujú bodové elementy
- Hrany (angl. Edge) – predstavujú spojnice série uzlov
- Steny (angl. Face) – predstavujú plochy vymedzené hranami

Pri topologickej údajovej štruktúre musia byť dodržané nasledujúce pravidlá:

- Každá hrana má jeden počiatočný a jeden koncový uzol
- Hrany sa môžu pretínať len v uzloch
- Každá stena je ohraničená viacerými hranami
- Každá plocha musí byť pravou alebo ľavou plochou minimálne jednej hrany

V súčasnosti neexistuje jednotná softvérová implementácia topologického údajového modelu.

Na úrovni QGISu môžeme zapnúť pri digitalizácii objektov tzv. topologické editovanie (menu *Nastavenia->Možnosti Prichytávania* voľba *Zapnúť topologické editovanie*). Po zapnutí tejto voľby QGIS pri posúvaní/editácii uzlov edituje súčasne uzol susedného objektu. Avšak samotné údaje stále ostávajú uložené v netopologickej údajovej štruktúre.

PostGIS obsahuje topologické rozšírenie, ktoré je možné zapnúť príkazom:

```
CREATE EXTENSION postgis_topology;
```

PostGIS rozoznáva tri základné typy priestorových štruktúr typu topology: uzly, hrany, steny. Editácia vrstiev s topológiou je možná s využitím zásuvného modulu QGISu s názvom PostGIS Topology Editor.

Riešenie problematiky obchodného cestujúceho, hľadanie najkratšej trasy a ďalších optimalizačných úloh zabezpečuje rozšírenie PostGisu pgRouting, to však využíva vlastný topologický model.

Samostatný topologický model používa aj vektorový údajový model systému GRASS. Ten rozoznáva uzly, hrany (zložené z uzlov) a plošný element je definovaný ako hrany tvoriace uzavretý celok a uzol reprezentujúci centroid danej plochy. Využitie topológie pre sieťové analýzy s využitím údajového modelu GRASS demonštruje nasledujúci príklad:

Príklad 9.1: Predpokladajme, že máme k dispozícii líniovú vrstvu všetkých ciest v meste Banská Bystrica a tri obvodné oddelenia polície[1]. Našou úlohou je priradiť jednotlivé cesty k týmto oddeleniam tak, aby z každého bodu cesty trvala cesta najkratšie do prislúchajúceho obvodného oddelenia. Druhou úlohou je nájsť najkratšiu trasu medzi všetkými troma oddeleniami.

Riešenie:

1. Pripravíme v QGISe prostredie GRASSu (nastavenie lokality Mapsetu regiónu viď kapitola 3.3.2 na str. 42)
2. Importujeme vrstvu cesty a vrstvu stanice do vektorového údajového modelu GRASSu.
3. Vrstvy stanice a cesty zlúčime modulom v.patch do jednej mapy (uzly tvoriace stanice sa stanú uzlami siete ciest).
4. Pomocou modulu `v.net.alloc` priradíme jednotlivé hrany uzlom staníc. Výsledkom bude atribút vo vrstve stanice, ktorý určuje príslušnosť k danej stanici. Priradenie môžeme vizualizovať tak, že vrstvu zobrazíme s rôzne zafarbenými kategóriami vytvorenými z tohoto atribútu.
5. Pomocou modulu `v.net.path` nájdeme najkratšiu trasu medzi troma stanicami. Výstupom je líniová vrstva spájajúca tieto tri stanice.

[1] v skutočnosti sú len dve, jedno oddelenie je krajským oddelením

Kapitola 10

Štatistické analýzy

Mnohé geografické zákonitosti je možné opísať prostredníctvom rôznych štatistických metód. Tie môžu byť nielen popisným nástrojom, ale môžu slúžiť aj na objavenie nových zákonitostí, alebo overenie niektorých hypotéz. Táto kapitola stručne oboznamuje s princípmi geoštatistiky a podrobnejšie sa venuje základným štatistickým ukazovateľom, ktorými vieme charakterizovať aj geografické objekty či javy.

10.1 Základné pojmy

Geoštatistika študuje fenomén, ktorý sa mení v priestore a/alebo v čase (Deutsch, 2002 In. Bohling 2005). Geoštatistika je súborom numerických techník, ktoré sa snažia charakterizovať priestorové atribúty využívajúc hlavne náhodné modely spôsobom, akým sú využívané na analýzu časových radov charakterizujúcich časové údaje (Olea, 1999 In. Bohling 2005). Geoštatistika ponúka spôsob ako popísať priestorovú spojitosť prirodzeného fenoménu a využiť pritom klasické regresné techniky, ktoré využívajú spojitosť (Isaaks – Srivastava, 1989). Geoštatistika sa zaoberá údajmi, ktoré majú vlastnosť priestorovej autokorelácie. Pod autokoreláciou rozumieme koreláciu medzi prvkami radu a inými prvkami z toho istého radu oddelenými nejakým intervalom.

Okrem špeciálnej geoštatistiky sa pri analýze priestorových údajov často využíva aj základná popisná štatistika. Zaujíma nás početnosť nejakej skupiny objektov, maximálna a minimálna
hodnota vybraného atribútu, priemerná hodnota a štandardná odchýlka. V nasledujúcej časti predstavíme techniky získania týchto údajov pre vektorový aj rastrový údajový model uložený v databáze PostgreSQL s rozšírením PostGIS.

10.2 Popisná štatistika

10.2.1 Početnosť

Počet objektov určitej vlastnosti získame SQL dopytom cez agregačnú funkciu `count()`.

Príklad 10.1: Zistime počet obcí na Slovensku s počtom obyvateľov do 300.

Riešenie:

```
SELECT count(*) as pocet_obci FROM obce
    WHERE obyvatelov<300;
```

Pokiaľ chceme spočítať malé obce v jednotlivých okresoch, môžeme využiť nasledujúci dopyt:

```
SELECT DISTINCT okres,
                count(*) OVER (PARTITION BY okres)
        FROM obce<300
```

Ďalšou štatistickou úlohou je zistiť extrémy (maximálnu a minimálnu hodnotu). Slúžia na to funkcie `min()` a `max()`.

Príklad 10.2: Jednoducho zistíme obec s najväčším počtom obyvateľov:

```
SELECT max(obyvatelov) FROM obce;
```

Vzhľadom na to, že `max` je agregačná funkcia, musíme na výpis celého záznamu použiť malý trik:

```
SELECT nazov, obyvatelov FROM obce
    WHERE obyvatelov=(SELECT max(obyvatelov) FROM obce)
```

Alternatívne môžeme zistiť maximálnu hodnotu aj ďalšími spôsobmi, napríklad cez:

```
SELECT nazov,obyvatelov FROM obce ORDER BY obyvatelov
    DESC LIMIT 1;
```

Počet pixelov danej hodnoty pri rastrovom údajovom modeli získame s využitím funkcie `ST_Count()`.

Príklad 10.3: Počet buniek rastra zistíme nasledovným dopytom:

```
SELECT ST_Count() FROM raster_mapa_vyuzitia_zeme;
```

Vypočíta počet buniek rastra `raster_mapa_vyuzitia_zeme`

Funkcia `ST_Count` spočíta počet pixelov (buniek) v rastri. Používa sa v tvare `ST_Count(raster, udaj bez hodnoty)`, kde údaj bez hodnoty môže nadobúdať hodnoty True a False (pravda, nepravda) v závislosti od toho, či chceme do výpočtu zahrnúť aj pixely, ktoré neobsahujú hodnotu.

Funkcia `ST_ValueCount` vráti počet pixelov v danom kanáli rastra alebo sústavy dlaždíc rastrov, pričom je možné zahrnúť do tohto výpisu aj početnosť buniek s neznámou hodnotou. Používa sa v tvare:

`ST_ValueCount(raster,kanál,aj_nezname_hodnoty)`

Poznámka: Raster z definície tvorí pravidelnú mriežku buniek. Každá jedna bunka musí obsahovať nejakú hodnotu. V praxi sa však často stáva, že pre určité bunky táto hodnota nie je definovaná (napr. sa tam daný meraný jav vôbec nevyskytuje – napr. pH pôdy na miestach, kde sa pôda nenachádza). Týmto neznámym hodnotám sa priradí nejaká vopred dohodnutá hodnota. Slúži na to funkcia
`ST_SetBandNoDataValue(raster,hodnota_pre_nezname_hodnoty)`.

10.2.2 Zonálna štatistika

Zonálna štatistika je založená na operáciách, ktoré sa uskutočňujú nad rastrovou vrstvou vo vopred definovaných oblastiach (zónach). Medzi často využívané operácie patrí napríklad hľadanie maximálnej alebo minimálnej hodnoty v rámci jednotlivých oblastí.

Ďalším príkladom využitia zonálnej štatistiky je výpočet priemerných nadmorských výšok povodí, kde vrstvu oblastí predstavujú jednotlivé povodia a rastrová vrstva vstupujúca do výpočtu je rastrovou vrstvou nadmorských výšok (DTM). V tomto prípade bude vypočítaná priemerná nadmorská výška daného povodia priradená k všetkým bunkám výstupného rastra ležiacim nad daným povodím.

Príklad 10.4: Vypočítajme základné početnosti jednotlivých kategórií využitia zeme pre každú obec.

Riešenie: Bolo publikovaných niekoľko metód, my budeme vychádzať z riešenia Lopez (2012), kde výslednú štatistiku získame nasledujúcim dopytom:

```
CREATE TABLE vyzuzitie_zeme_podla_obce AS
  WITH
   obec AS (SELECT gid, geom FROM obce AS b ),
   b_stats AS
 (SELECT  gid, (stats).*
 FROM (
  SELECT gid, ST_SummaryStats(ST_Clip(rast,1,geom))
    AS stats
  FROM vyuzitie_zeme
```

```
      INNER JOIN obec
      ON ST_Intersects(obec.geom,rast) ) AS foo )
    SELECT gid, SUM(count) AS pocet_buniek
     FROM b_stats
     WHERE pocet > 0
     GROUP BY gid
     ORDER BY gid;
```

10.2.3 Histogram

V deskriptívnej štatistike na opis štatistického súboru často používame histogram. Ten vyjadruje početnosť jednotlivých tried štatistického súboru vytvorených intervalovým delením. Intervaly je možné vytvoriť tak, aby rozpätie každého intervalu bolo totožné alebo aby v každom intervale bol rovnaký počet prvkov.

Príklad 10.5: Chceme štatisticky popísať kyslosť pôdy na danom území. Histogram rozdelíme do 3 tried podľa hodnoty pH.

```
SELECT ST_histogram(rast,1,7) FROM rastre WHERE rast_name=PH
```

Podobná funkcia existuje na kvantilové delenia. Funkcia `ST_Quantile` vypíše hodnotu tej bunky, ktorá sa nachádza v štvrtine (desatine, n-tine) štatistického súboru.

Príklad 10.6: Do akej nadmorskej výšky sa nachádza jedna štvrtina všetkého územia?

Riešenie: Výsledok zistím z rastra výšok dtm nasledovne:

```
SELECT ST_Quantile(rast,0.25) FROM rastre
    WHERE rast_name='dtm'
```

Sumárne štatistiky je možné získať cez funkcie `ST_SummaryStats`.

Hodnotu konkrétneho pixela môžeme zistiť cez funkciu `ST_Value`. Jej použitie ilustruje nasledujúci príklad.

Príklad 10.7: Aká teplota je v Martine? K máme dispozícii raster teplôt a bodovú vrstvu mestá.

```
SELECT
    rid,
    ST_Value(rast, 1, mesta.geom) As b3pval
FROM rastre,mesta
WHERE
    mesta.nazov='Martin'
    AND ST_Intersects(rast,sometable.geom);
```

Pre rastrové vrstvy môžeme získať maximálnu, minimálnu hodnotu, sumu všetkých buniek, priemernú hodnotu, počet buniek, štandardnú odchýlku cez funkciu `ST_SummaryStats()`.

10.3 Štatistická analýza lineárnej závislosti dvoch javov

V praxi si často kladieme otázku závislosti medzi dvoma javmi (resp. dvoma vlastnosťami jedného javu). Pri práci s priestorovými údajmi môže byť jedna z týchto vlastností práve nejaký atribút vyjadrujúci polohu. Uvažujme o priemernej ročnej teplote meranej na rôznych miestach Zeme.

Závisia tieto teploty od zemepisnej šírky, v ktorej boli zmerané? Pokiaľ medzi nimi lineárna závislosť existuje, môžeme matematicky vyjadriť, že čím severnejšie (južnejšie) od rovníka sa dané miesto nachádzalo, tým nižšia bola jeho priemerná teplota. Ide tu o nepriamu úmeru, t.j. lineárny vzťah v tvare $T = K.\varphi + Q$, kde parameter K je neznáme záporné číslo a parameter Q je neznáme číslo. Na získanie týchto dvoch koeficientov (parametrov) využijeme metódu regresnej analýzy. Zo štatistického súboru je možné vypočítať aj tzv. Pearsonov korelačný koeficient, ktorý sa pohybuje v rozmedzí od -1 po +1. Hodnota -1 znamená nepriamu úmeru, hodnota +1 znamená priamu úmeru, hodnota 0 znamená, že premenné X a Y nie sú lineárne závislé.

Príklad 10.8: Overme lineárnu závislosť medzi zemepisnou šírkou a teplotami.

```
SELECT corr(ST_Y(the_geom),teplota) FROM mesta_a_teploty

        corr
--------------------
-0,8149755053
(1 row)
```

Štatisticky sme dokázali lineárnu závislosť medzi zemepisnou šírkou a priemernou ročnou teplotou. Seriózna štatistická analýza by však vyžadovala preskúmanie štatistickej vzorky (vybraných miest), či je dostatočne reprezentatívna, t.j. aký je priestorový rozptyl medzi vybranými mestami a bolo by potrebné zohľadniť mnohé ďalšie vlastnosti. V našom príklade vychádzame z predpokladu, že daná vzorka je reprezentatívna. Korelačný koeficient -0,81 je blízky hodnote -1, a preto ho možno interpretovať ako silnú nepriamu úmeru, t.j. čím je hodnota zemepisnej šírky väčšia, tým je hodnota priemernej teploty nižšia.

Príklad 10.9: Máme hypotézu, ktorá hovorí, že veľkosť obcí na Slovensku podľa plochy klesá v západo-východnom smere, t.j. čím viac smerujeme na východ, tým sú katastrálne územia obcí menšie.

Riešenie: Na overenie tejto hypotézy využijeme opäť Pearsonov korelačný koeficient, ktorý vypočítame pomocou agregačnej funkcie `corr()`. Nasledujúci dopyt zisťuje korelačnú závislosť medzi polohou geometrického stredu územia obce a jej rozlohou. Oba atribúty sú vypočítané pomocou priestorových funkcií `ST_Centroid` a `ST_Area`.

```
SELECT
    corr(
        ST_X(ST_Centroid(the_geom)),
        ST_Area(the_geom)
        )
FROM obce;
```

Časť III

Komplexné príklady

Kapitola 11

Návrh údajového modelu pre systém na správu cintorína

Príklad je praktickou aplikáciou teórie z kapitoly 2 (str. 15). Cieľom bolo navrhnúť údajový model vhodný pre systém na správu cintorína, ktorý bude slúžiť na evidenciu hrobových miest, zosnulých, nájomcov hrobových miest, zmluvných vzťahov medzi nájomcami a hrobovými miestami. Príklad vychádza z prípadovej štúdie v (Balážovič, 2010).

11.1 Cieľová funkcionalita

Výslednou aplikáciou by mal byť systém umožňujúci evidovať hierarchicky usporiadané hrobové miesta ich nájomcov a poplatky za ne. Systém by mal zohľadňovať priestorovú povahu spracúvaných informácií a vedieť výstupy vykresliť aj v podobe mapy. Systém by mal byť schopný vyhľadávať údaje podľa rôznych kritérií (mena, hrobového miesta, dátumu zaplatenia, dátumu narodenia, úmrtia atď.). Cieľová funkcionalita bude zahŕňať nasledovné operácie:

- zobrazovanie údajov, vkladanie údajov a zmena údajov o:
  - ľuďoch
  - hrobových miestach (formou tabuľky aj mapy)
  - zmluvách
- vyhľadávanie
  - hrobového miesta
    - podľa mena zomrelého
    - podľa mena nájomcu
    - podľa dátumu úmrtia

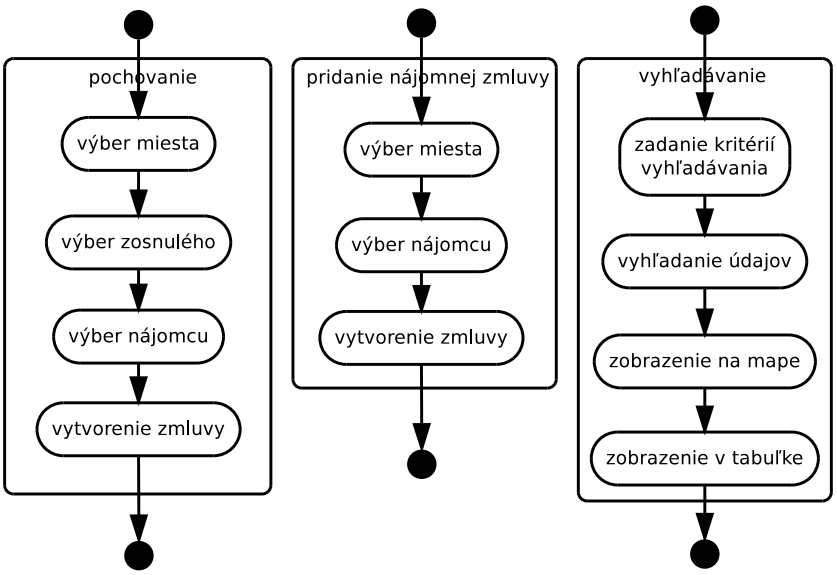
        * podľa dátumu pochovania
        * podľa dátumu narodenia
        * podľa dátumu zaplatenia (vypršané nájomné zmluvy)
        * podľa lokality (parcely, radu, miesta)
        * podľa blízkosti iných hrobových miest (hrobových miest iných zomrelých)
        * podľa vzdialenosti od chodníka
        * podľa vzdialenosti od stromov
        * na okraji radu
        * v strede radu
        * podľa obsadenosti (obsadené/voľné)
        * podľa funkčnosti (funkčné/nefunkčné)
    - človeka
        * podľa mena a priezviska
        * podľa vzťahu k cintorínu (nájomca/pochovaný)
        * podľa dátumu narodenia
    - zmluvy
        * podľa dátumu platnosti
        * podľa nájomcu
        * podľa hrobového miesta

- tlač
    - zmluvy
    - karty hrobového miesta
    - výročnej štatistickej správy
- Iné
    - správa slávnych osobností
    - správa exhumácií
    - správa fotodokumentácie

Cieľové procesy sú reťazcom vybraných základných operácií a zodpovedajú činnostiam potrebným pri správe cintorína. Z hľadiska cieľovej funkcionality boli navrhnuté nasledujúce procesy:

- pohreb
- pridanie nájomnej zmluvy
- vyhľadávanie

Graficky sú procesy znázornené formou UML Activity diagramu na obr. 11.1.



Obr. 11.1: Zoskupenie operácií do procesov (*Systém na správu cintorína*)

11.2 Modelovanie databázy

11.2.1 Sémantické a konceptuálne modelovanie

Prvým krokom pri sémantickom modelovaní databázy bolo určenie objektov a identifikácia entít. Na základe popisu cieľovej funkcionality sme definovali nasledujúce entity:

- cintorín – predstavuje celé územie cintorína
- hrobové miesto – predstavuje časť cintorína, kde je pochovaná jedna alebo viac osôb
- rad – predstavuje niekoľko hrobových miest tvoriacich spolu rad
- parcela – predstavuje spojité územie obsahujúce niekoľko radov
- trávny porast – predstavuje územie porastené trávnym porastom
- chodník – predstavuje chodník prechádzajúci cintorínom
- plot – predstavuje plot okolo cintorína
- strom – predstavuje strom rastúci na území cintorína
- vodojem – predstavuje vodojem na území cintorína
- úložisko smetí – predstavuje miesto na ukladanie smetí
- dom – predstavuje objekt domu na území cintorína
- kostol – predstavuje objekt kostola na území cintorína

- hrobka – predstavuje objekt hrobky na území cintorína
- pochovaný – predstavuje osobu pochovanú na cintoríne na určitom hrobovom mieste
- nájomca – predstavuje osobu, ktorá si prenajíma hrobové miesto
- zmluva – predstavuje zmluvu o prenájme hrobového miesta na určitý čas uzatvorenú medzi správcom cintorína a nájomcom

Ďalším krokom bola definícia foriem priestorovej reprezentácie entít a zoskupenie podobných entít medzi typy entít. Pri všetkých entitách, ktoré majú priestorovú povahu budeme uvažovať o forme vektorovej reprezentácie. Tie možno rozdeliť na typ *bodovej* entity, *čiarovej* entity a *plošnej* entity. Medzi entity bez priestorovej povahy možno zaradiť entity:

- človek,
- pochovaný,
- nájomca,
- zmluva.

Ostatné entity majú priestorovú povahu:

- bodovú:
  - strom,
  - vodojem;
- čiarovú:
  - chodník,
  - plot;
- plošnú:
  - cintorín,
  - hrobové miesto,
  - rad,
  - parcela,
  - trávny porast,
  - dom,
  - hrobka,
  - kostol,
  - úložisko smetí,

  - a chodník[1].

Tretím krokom bola definícia atribútov jednotlivých entít. Pre jednotlivé entity sme definovali nasledujúce atribúty:

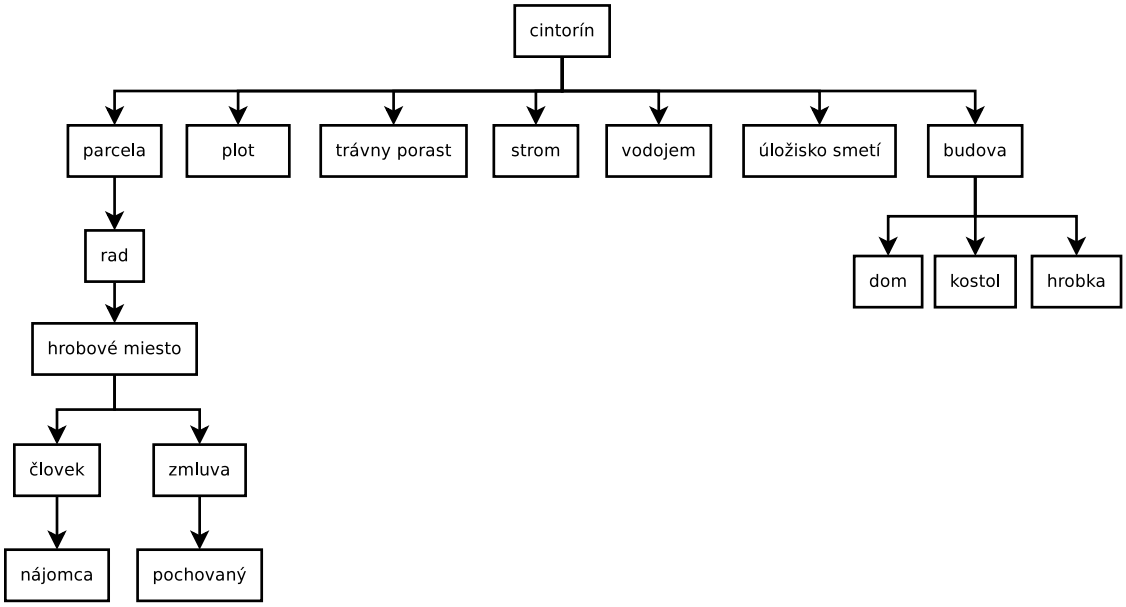
- cintorín, parcela, trávn porast, plot, chodník, vodojem, úložisko smetí, hrobka, kostol, dom
  - id – číselný identifikátor
  - geometria – priestorová reprezentácia
- hrobové miesto
  - id – číselný identifikátor miesta
  - geometria – priestorová reprezentácia územia hrobového miesta
  - parcela – id parcely, na ktorej sa miesto nachádza
  - rad – číslo radu, v ktorom sa miesto nachádza
  - miesto – poradové číslo miesta v rade
  - typ – určuje typ hrobového miesta (jednohrob, dvojhrob, trojhrob, detský hrob, urnový hrob, iné)
  - ÚKZP – číslo v Ústrednom zozname kultúrnych pamiatok (pokiaľ je do neho hrobové miesto zaradené)
- rad
  - id – číselný identifikátor radu
  - geometria – priestorová reprezentácia územia radu
  - parcela – id parcely, na ktorej sa rad nachádza
- strom
  - id – číselný identifikátor stromu
  - geometria – priestorová reprezentácia stromu
  - druh – druh stromu
  - výška – výška stromu
  - šírka – šírka koruny stromu
- pochovaný
  - id – číselný identifikátor človeka
  - meno
  - priezvisko

[1]chodník má povahu čiarovej entity, avšak pri použitej mierke je vhodné na vykresľovanie použiť chodník ako plošnú entitu

  - rodné meno
  - dátum narodenia
  - dátum úmrtia
  - dátum pohrebu
  - pohlavie
  - rodné číslo
  - hrobové miesto – miesto, na ktorom je zosnulý pochovaný
  - stručný životopis (pokiaľ sa jedná o významnú osobnosť)
  - adresa – posledná adresa pochovaného
- nájomca
  - id – číselný identifikátor človeka
  - meno
  - priezvisko
  - rodné meno
  - dátum narodenia
  - pohlavie
  - hrobové miesto – miesto, ktoré si nájomca prenajíma
  - zmluva – číslo zmluvy, ktorú má nájomca uzavretú
  - adresa – posledná platná adresa nájomcu
- zmluva
  - id – číselný identifikátor zmluvy
  - hrobové miesto – miesto, ktorého predmetom je táto zmluva
  - nájom od – dátum začiatku platnosti nájmu
  - nájom do – dátum ukončenia platnosti nájmu
  - dátum platby
  - číslo účtu
  - výška uhradeného poplatku

Štvrtým krokom bolo usporiadanie entít do hierarchickej štruktúry a zoskupenie entít do vrstiev. Najvyššiu úroveň tvorí *cintorín*. Úroveň ostatných objektov zachytáva stromová štruktúra na obr. 11.2.

Objekty *dom*, *hrobka* a *kostol* boli zoskupené do vrstvy *budova*. Vrstva *budova* bude obsahovať všetky tie atribúty, ktoré obsahovali objekty, z ktorých vznikla a navyše bude obsahovať atribút *typ*, pomocou ktorého bude možné identifikovať o aký typ budovy ide (hrobka, dom alebo kostol).



Obr. 11.2: Stromová štruktúra vrstiev (*Systém na správu cintorína*)

Objekty nájomca a pochovaný boli zoskupené do entity *človek*. Nájomcovia aj pochovaní sa líšia od seba len postavením, pričom jeden človek môže vystupovať istý čas ako nájomca a neskôr môže byť na cintoríne aj pochovaný.

Piatym krokom konceptuálneho modelovania bolo definovanie vzťahov medzi entitami. Na modelovanie vzťahov bol použitý entinto-relačný (ER) diagram (obr. A.1 v prílohe).

Posledným krokom konceptuálneho modelovania bolo definovanie štruktúry entít a ich atribútov. ER diagram bol v tomto kroku doplnený o atribúty (obr. A.2 v prílohe).

11.2.2 Logické modelovanie

V ďalšej fáze bolo potrebné zobraziť entity na objekty, ktoré obsahuje databázové prostredie. Základnou údajovou štruktúrou v relačnej databáze sú tabuľky. Tie sú pomocou relácií vzájomne prepojené. V tomto kroku na základe entít definujeme triedy objektov, ktoré budú v databázovom prostredí vystupovať ako tabuľky (obr. A.3 v prílohe).

Druhým krokom je definícia vrstiev ako súborov príbuzných objektov v jazyku databázového systému. Triedy objektov `kostol`, `dom` a `hrobka` budú nahradené triedou objektov `budova`. Triedy `pochovany` a `najomca` boli zlúčené do triedy `clovek`.

Tretí krok pri tvorbe logického modelu tvorí definícia množiny atribútov ako štruktúrovaného súboru vlastností k jednotlivým entitám v jazyku databázového systému. Trieda objektov `chodník` bude mať dva atribúty s geometriou: `the_geom_line` predstavuje líniovú reprezentáciu chodníka, `the_geom_poly` polygónovú. Tým, že je geometria objektu uložená v dvoch

rôznych atribútoch tej istej triedy, nedochádza k zbytočnej redundancii údajov a každý chodník je v databáze zastúpený práve raz a to len v jednej tabuľke.

Výsledok je zobrazený pomocou UML diagramu tried na obr. A.5 v prílohe (zatiaľ bez relačných väzieb).

Štvrtý krok tvorí definícia prepojení databázovej reprezentácie objektov a atribútov a vrstiev v jazyku databázového systému. V tomto kroku je potrebné definovať primárne kľúče, ktoré môžu tvoriť cudzie kľúče pre iné tabuľky. Primárny kľúč v tabuľke je stĺpec (alebo niekoľko stĺpcov), z hodnoty ktorého možno jednoznačne určiť ktorýkoľvek prvok (riadok) tabuľky.

Ako primárne kľúče vo všetkých tabuľkách vystupujú atribúty `id`. Primárny kľúč je obmedzenie, pri ktorom hodnoty v stĺpci musia byť jedinečné a nesmú byť prázdne.

Nasledujúce vzťahy boli vytvorené prostredníctvom cudzích kľúčov (cudzí kľúč je vždy primárnym kľúčom nejakej inej tabuľky):

- 1 parcela môže obsahovať n radov (relácia `parcela` – `rad`)
- parcela obsahuje hrobové miesta (relácia `parcela` – `hrobove_miesto`)
- 1 rad môže obsahovať n hrobových miest (relácia `rad` – – `hrobove_miesto`)
- 1 hrobové miesto môže slúžiť na odpočinok pre n ľudí (relácia `hrobove_miesto` – `clovek`)
- na 1 hrobové miesto sa môže viazať n zmlúv (relácia `hrobove_miesto` – `zmluva`)
- 1 človek môže uzatvoriť n zmlúv (relácia `clovek` – `hrobove_miesto`)

Piatym krokom je definícia a štrukturovanie vrstiev príbuzných priestorových objektov v grafickom prostredí geografickej databázy. Náš databázový model neobsahuje objekty, ktoré by bolo potrebné previesť týmto krokom.

Šiestym krokom je definícia prepojení vrstiev medzi grafickým a databázovým prostredím v jazyku databázového systému. Prakticky to znamená pridať zoznam tabuliek s geometriou do tabuľky `geometry_column` (interná tabuľka rozšírenia PostGIS). Tabuľka obsahuje informáciu o názve tabuľky obsahujúcej geometriu, názve stĺpca, obsahujúceho geometriu, typ priestorových objektov, dimenziu priestoru a súradnicový systém, v ktorom sa objekty nachádzajú.

Všetky navrhnuté tabuľky s geometriou majú stĺpec s geometriou nazvaný `the_geom`. Výnimkou je tabuľka `chodnik`, ktorá obsahuje dva stĺpce s geometriou – `the_geom_line` (pre líniovú reprezentáciu chodníka) a `the_geom_poly` (pre polygónovú reprezentáciu chodníka). Viď tabuľka:

Posledným krokom tvorby logického modelu je vytvorenie kompletnej a koherentnej štruktúry tak, aby žiadny prvok databázy nebol redundantný a teda všetky prvky databázy generovali úplnú geografickú informáciu. Naplnenie týchto cieľov budeme realizovať normalizáciou databázy.

Prvú normálovú formu (1NF) spĺňajú tabuľky databázy vtedy, ak každý atribút obsahuje len atomické hodnoty, to znamená hodnoty, ktoré sú ďalej nedeliteľné (Wikipedia, 2015a).

Prvú normálovú formu spĺňajú tabuľky okrem tabuľky `clovek`. Tabuľka `clovek` obsahuje atribút `adresa`, ktorý možno rozdeliť na ulicu, číslo ulice, obec, okres, štát. Tabuľka `clovek`bude upravená, tak aby obsahovala všetky tieto atribúty (obr. A.7 v prílohe).

Druhú normálovú formu (2NF) spĺňajú tabuľky práve vtedy, keď spĺňajú 1NF a zároveň každý nekľúčový atribút bol na primárnom úplne funkčne závislý (na celom kľúči, nie len na jeho podmnožine). V prípade, že tabuľky neobsahujú viacatribútové (kompozitné) kľúče a vyhovujú podmienkam 1NF, vyhovujú aj 2NF. Všetky tabuľky navrhovanej databázy túto podmienku spĺňajú.

Tabuľky sú v tretej normálnej forme, pokiaľ sú v 2NF a žiaden nekľúčový atribút nie je tranzitívne funkčne závislý od primárneho kľúča. To znamená, že všetky nekľúčové atribúty musia byť navzájom nezávislé.

3NF nespĺňa tabuľka `hrobove_miesto` a tabuľka `clovek`. Z tabuľky `hrobove_miesto` je potrebné vypustiť stĺpec `parcela`. Ten je funkčne závislý od stĺpca `rad` (každý rad je na určitej parcele, preto je z čísla radu možné zistiť číslo parcely). Ďalej boli vypustené stĺpce `povodne_cislo` a `uzkp`. Z oboch vznikli samostatné tabuľky `povodne_cislo_hm` a `uzkp` previazané na stĺpec `id` tabuľky `hrobove_miesto`. Výsledkom je schéma na obr. A.8 v prílohe.

V tabuľke `clovek` sa nachádza viacero atribútov, ktoré sú funkčne závislé od iných atribútov. Stĺpec `hrobove_miesto` má zmysel len v prípade, že je človek zomrelý pochovaný (takže je funkčne závislý na hodnote týchto dvoch stĺpcov). Riešením je dokompozícia do dvoch tabuliek – `clovek` a `pochovany` (tabuľka `pochovany` bude obsahovať atribúty `datum_pochovania`, `clovek` (cudzí kľúč prepojený na id v tabuľke clovek) a `hrobove_miesto`. Atribúty `hrobove_miesto` a `datum_pochovania` budú z tabuľky `clovek` vypustené. Atribút `datum_umrtia` bude ponechaný, čo umožní mať v systéme aj ľudí, ktorí síce zomreli, ale nie sú na cintoríne pochovaní (môžu predstavovať zosnulých nájomcov). Ďalším atribútom, ktorý nie je funkčne závislý len na atribúte `id` je atribút `zivotopis`. Ten sa týka len pochovaných osobností. Preto bola vytvorená ďalšia tabuľka `osobnosti`, ktorá bude obsahovať atribúty `clovek` (cudzí kľúč na tabuľku verb|clovek|) a `zivotopis`. V dekompozícii by bolo možné pokračovať aj v tabuľke `adresa`, avšak nie je to cieľom tejto práce, preto tento stupeň normalizácie nebude prekročený.

Podmienku kompletnej a koherentnej štruktúry tak, aby žiadny prvok databázy nebol redundantný a teda všetky prvky databázy generovali úplnú geografickú informáciu v prostredí databázy PostgreSQL s rozšírením PostGIS nie je možné zachovať pri prvkoch geometrie. Databázový systém používa na ukladanie geometrií štandard *Simple Features*. To neumožňuje normalizáciu na topologickej úrovni, t.j., že by každý bod, línia alebo polygón boli zapísané v databáze len raz. Nie je tak možné mať v databáze napríklad líniu, ktorá bude zároveň hranicou radu, parcely a chodníka.

Kvôli zachovaniu referenčnej integrity boli atribúty doplnené o ďalšie obmedzenia.

Obmedzenie *NOT NULL* (neprázdna hodnota) bolo pridané:

- u všetkých tabuliek s geometriou na stĺpec `the_geom` (prípadne na `the_geom_line`, `the_geom_poly`), kde zabezpečí, že nebude možné do tabuliek vložiť objekt bez jeho geometrickej interpetácie
- pre tabuľku `rad`
  - na stĺpec `cislo` – každý rad musí mať poradové číslo v rámci svojej parcely
- pre tabuľku `hrobove_miesto` na stĺpce
  - stĺpec `miesto` – každé miesto musí mať poradové číslo (v rámci radu)
  - stĺpec `funkcnost` – o každom mieste musí existovať informácia, či je funkčné (t.j. či je ho možné použiť)

Obmedzenie *UNIQUE* bolo pridané na stĺpec `the_geom` (prípadne na stĺpce `the_geom_line` a `the_geom_poly`) u všetkých tabuliek s geometriou. Obmedzenie zabezpečí, že nebude možné do tabuliek vložiť dva identické objekty (v skutočnosti tiež nie je možné, aby boli na tom istom mieste dve rôzne entity).

Obmedzenie *CHECK* bolo pridané na dátumové atribúty. Zabezpečuje, že nie je možné vložiť atribút `datum_umrtia` skorší ako `datum_narodenia`. Podobne nie je možné vložiť dátum do atribútu `datum_pochovania` skorší ako sú hodnoty atribútov `datum_narodenia` a `datum_umrtia`.

Databáza PostgreSQL nepozná typ ENUM, preto pre každý atribút tohto typu boli vytvorené pomocné tabuľky a atribút pomocou cudzieho kľúča bol previazaný s nimi. Takto vznikli tabuľky:

- `pohlavie`,
- `typ_budovy`,
- `typ_hroboveho_miesta`.

Pri návrhu užívateľského rozhrania vznikla potreba vyčlenenia zoznamu starých zmlúv do zvláštnej tabuľky. Dôvodom je uplatňovanie inej validačnej politiky pri starých a nových zmluvách. Preto bola pridaná tabuľka `stare_zmluvy` s atribútmi `id_zmluvy` (previazaná na `id` v tabuľke `zmluvy`) a `cislo_starej_zmluvy` (označenie zmluvy v starej evidencii zmlúv). Z rovnakých dôvodov bola vytvorená tabuľka `fiktivne_zmluvy` (budú v nich figurovať staré zmluvy, ktoré neobsahujú všetky potrebné náležitosti, ale z historických dôvodov je ich potrebné tiež zaevidovať).

Vzhľadom na to, že typ `date`používaný pre ukladanie dátumov je nedeliteľný, nie je možné ako dátum zadať iba rok alebo iba mesiac a rok. Keďže sa

dá predpokladať, že pre niektoré záznamy bude táto informácia chýbať (dátum pochovania/úmrtia bude daný len rokom), je potrebné zaviesť pri každom dátume atribút „nepresny“. Tento atribút poslúži na odlíšenia dátumu 1.1.1910 od dátumu, kde nebol známy mesiac a deň, ale v databáze vystupuje tiež ako dátum 1.1.1910.

Pridané tak boli tri nové atribúty:

- `datum_narodenia_nepresny`,
- `datum_umrtia_nepresny`,
- `datum_pochovania_nepresny`.

Ďalším krokom je čiastočná denormalizácia databázy návrhom databázových pohľadov. Pohľady spájajú viaceré tabuľky tak, aby z výslednej tabuľky bolo možné získať všetky potrebné informácie. Preto boli vytvorené nasledujúce pohľady:

- `clovek_adresa`– vznikne spojením tabuliek `clovek` a `adresa` – pohľad bude obsahovať kompletnú informáciu o všetkých ľuďoch v systéme
- `najomca_adresa`– vznikne spojením tabuľky `zmluva` a pohľadu `clovek_adresa` – pohľad bude obsahovať kompletnú informáciu o každom nájomcovi
- `pochovany_adresa`– vznikne spojením tabuľky `pochovany` a pohľadu `clovek_adresa` – pohľad bude obsahovať kompletnú informáciu o každom pochovanom
- `zmluvy_clovek`– vznikne spojením tabuľky `zmluvy` a pohľadu `clovek_adresa` – pohľad bude obsahovať ku každej zmluve aj kompletnú informáciu o nájomcovi
- `zmluvy_najnovsie`– vznikne výberom len najnovších zmlúv pre každé hrobové miesto
- `hrobove_miesto_rad`– vznikne spojením tabuľky `hrobove_miesto` a `rad`– pohľad bude obsahovať kompletnú informáciu o hrobovom mieste (vrátane miesta v hierachii miesto-rad-parcela)

11.2.3 Fyzické modelovanie

Poslednou etapou tvorby databázy je transformácia logického modelu do fyzického v konkrétnom databázovom prostredí. Komunikácia s databázovým systémom PostgreSQL sa uskutočňovala pomocou jazyka SQL. Transformáciou logického modelu do reálnych údajových súborov databázového systému znamenala vytvorenie tabuliek cez príkaz `CREATE TABLE` (tabuľky vytvárame

bez stĺpcov s geometriou). Všade tam, kde je potrebný inkrementovaný identifikátor sa namiesto údajového typu *integer* použije typ *serial* (systém PostgreSQL vytvorí postupnosť celých čísel, ktoré postupne bude priraďovať objektom – výhodou je, že čísla nebude potrebné zadávať explicitne). Postupnosť SQL príkazov, ktorá zabezpečí vytvorenie tabuliek, je v prílohe B.1.

Po vytvorení základnej štruktúry tabuliek pridáme pomocou funkcie `AddGeometryColumn()` stĺpce s geometriou (príloha B.1).

Ďalším krokom je transformácia logického modelu do reálnych údajových súborov databázového systému, pričom sa uplatňujú metódy syntaktickej kontroly a metódy kontroly koherentnosti databázových štruktúr. Prakticky to znamená pridať k existujúcim tabuľkám navrhnuté obmedzenia (primárne a cudzie kľúče, *CHECK* obmedzenia, *NOT NULL* a *UNIQUE*). Kód v jazyku SQL je prílohe B.1.

Posledným krokom fyzického modelovania je vytvorenie (implementácia) systému riadenia bázy dát na vykonávanie štandardných dopytov na databázu a interaktívnu komunikáciu s jej grafickým a databázovým prostredím. Znamená to navrhnúť základné dopyty v jazyku SQL a grafické užívateľské rozhranie, prostredníctvom ktorého budú spúšťané. Návrhom dopytov sa zaoberá nasledujúca časť.

Návrh dopytov

Podľa operácií a procesov, ktoré by mala aplikácia obsahovať (sú definované v časti 11.1) boli zostavené nasledujúce dopyty. V dopytoch sú kvôli prehľadnosti použité konkrétne hodnoty. Vo výslednej aplikácii budú tieto nahradené premennými napĺňanými podľa požiadaviek užívateľa (zadávanými cez grafické užívateľské rozhranie).

1. Zoznam všetkých parciel

```
SELECT id FROM parcely;
```

2. Zoznam všetkých radov z parcely číslo 2 (môže byť ľubovoľná iná)

   - s využitím atribútového dopytu:

```
SELECT DISTINCT rad,cislo FROM rady
  WHERE parcela=2
```

   - s využitím priestorového dopytu:

```
SELECT id,cislo FROM rady CROSS JOIN parcely
  WHERE
    parcela=2 AND
    ST_Contains(parcely.the_geom,rady.the_geom)
```

3. Zoznam všetkých miest z radu číslo 25 (môže byť ľubovoľný iný)

```
SELECT id FROM hrobove_miesto CROSS JOIN rady
  WHERE
    rady.id=25 AND
    ST_Contains(rady.the_geom,hrobove_miesto.the_geom)
```

4. Id miesta z parcely 2, radu 25, miesta 3 (môže byť ľubovoľné iné)

```
SELECT id FROM hrobove_miesto CROSS JOIN rady
  WHERE
    miesto=3 AND
    rady.id=25 AND
    ST_Contains(rady.the_geom,hrobove_miesto.the_geom)
```

5. Zoznam všetkých ľudí, ktorí nie sú ešte pochovaní na žiadnom mieste

```
SELECT id,
       clovek(meno,priezvisko),
       meno,
       priezvisko
  FROM clovek
  LEFT JOIN pochovany ON id_clovek=id
  WHERE id_hm isnull
```

6. Podrobná informácia o konkrétnom človeku (napr. s `id` č. 22)

```
SELECT id,
       meno,
       priezvisko,
       rodne_meno,
       datum_narodenia,
       datum_umrtia,
       pohlavie,
       tituly,
       ml_st,
       datum_narodenia_nepresny,
       datum_umrtia_nepresny,
       ulica,
       cislo,
       stat,
       okres
    FROM clovek_adresa
    WHERE id=22
```

7. Zoznam všetkých pochovaných na vybranom hrobovom (napr. mieste č. 1)

```
SELECT meno,
       priezvisko
  FROM pochovany_adresa
  WHERE id_hm=1
```

8. Zoznam nájomcov hrobového miesta č. 1 (prípadne iného)

```
SELECT meno,
       priezvisko,
       najom_od as najomod,
       najom_do as najomdo
  FROM zmluva_clovek
  WHERE id_hm=1 ORDER BY najom_do;
```

9. Zoznam ľudí

   - pochovaných

```
SELECT id,
       meno,
       priezvisko,
       rodne_meno,
       datum_narodenia,
       datum_narodenia_nepresny,
       datum_umrtia,
       datum_umrtia_nepresny,
       tituly,
       ml_st,
       adresa,
       pohlavie,
       ulica,
       cislo,
       obec,
       okres,
       stat,
       id_hm
    FROM pochovany_adresa
```

   - nájomcov

```
SELECT DISTINCT id,
       meno,
       priezvisko,
       rodne_meno,
       datum_narodenia,
       datum_narodenia_nepresny,
       datum_umrtia,
       datum_umrtia_nepresny,
       tituly,
```

```
        ml_st,
        adresa,
        pohlavie,
        ulica,
        cislo,
        obec,
        okres,
        stat
    FROM clovek_adresa
    LEFT JOIN najomca
      ON najomca.id_clovek=clovek_adresa.id
    WHERE id_clovek IS NOT NULL
```

- všetkých

```
SELECT id,
       meno,
       priezvisko,
       rodne_meno,
       datum_narodenia,
       datum_narodenia_nepresny,
       datum_umrtia,
       datum_umrtia_nepresny,
       tituly,
       ml_st,
       adresa,
       pohlavie,
       ulica,
       cislo,
       obec,
       okres,
       stat
  FROM clovek_adresa
```

- výber podľa mena, priezviska, (...) možno získať doplnením klauzuly

10. Zoznam hrobových miest aj s pochovanými a nájomcami

```
SELECT pochovany_adresa.id,
       meno,
       priezvisko,
       rodne_meno,
       datum_narodenia,
       datum_narodenia_nepresny,
       datum_umrtia,
       datum_umrtia_nepresny,
```

```
        tituly,
        ml_st,
        adresa,
        pohlavie,
        ulica,
        cislo,
        obec,
        okres,
        stat,
        id_hm,
        hrobove_miesto.id AS hmid
  FROM  chodnik_line,
        multistrom,
        (SELECT hrobove_miesto.the_geom,
                hrobove_miesto.id,
                funkcnost,
                najom_do
           FROM hrobove_miesto
           LEFT JOIN zmluvy_najnovsie
             ON hrobove_miesto.id=zmluvy_najnovsie.id_hm
         ) AS hrobove_miesto
         LEFT JOIN pochovany_adresa
           ON hrobove_miesto.id=pochovany_adresa.id_hm
```

11. Zoznam zmlúv aj s nájomcami

```
SELECT * FROM zmluvy_clovek
```

12. Hrobové miesta (geometria),

    - na ktorých sa nachádzajú pochovaní ľudia

```
SELECT the_geom FROM
    (SELECT DISTINCT ON (hrobove_miesto.id)
            pochovany_adresa.id,
            meno,
            priezvisko,
            rodne_meno,
            datum_narodenia,
            datum_narodenia_nepresny,
            datum_umrtia,
            datum_umrtia_nepresny,
            tituly,
            ml_st,
            adresa,
            pohlavie,
            ulica,
```

```
            cislo,
            obec,
            okres,
            stat,
            id_hm,
            hrobove_miesto.the_geom
        FROM chodnik_line,
             multistrom,
             (SELECT hrobove_miesto.the_geom,
                     hrobove_miesto.id,
                     funkcnost,
                     najom_do FROM hrobove_miesto
              LEFT JOIN zmluvy_najnovsie
                ON hrobove_miesto.id
                   =zmluvy_najnovsie.id_hm
              ) AS hrobove_miesto
              LEFT JOIN pochovany_adresa
                ON hrobove_miesto.id
                   =pochovany_adresa.id_hm
        ) AS nova USING UNIQUE id USING SRID=-1
```

- ku ktorým sa viažu nájomcovia

```
SELECT the_geom FROM
  (SELECT DISTINCT id,
                   meno,
                   priezvisko,
                   rodne_meno,
                   datum_narodenia,
                   datum_narodenia_nepresny,
                   datum_umrtia,
                   datum_umrtia_nepresny,
                   tituly,
                   ml_st,
                   adresa,
                   pohlavie,
                   ulica,
                   cislo,
                   obec,
                   okres,
                   stat
              FROM clovek_adresa
                LEFT JOIN najomca
                  ON najomca.id_clovek=clovek_adresa.id
      ) AS nova USING UNIQUE id USING SRID=-1
```

- ku ktorým sa viažu nájomcovia a pochovaní: *UNION* predchádzajúcich dopytov; tieto dopyty možno pomocou klauzuly *WHERE* up-

resniť (použiť vzdialenosť od stromov, chodníka, alebo špecifikovať niektorý z použitých atribútov)

13. Hrobové miesta (geometria),

- na ktorých sa nachádzajú pochovaní ľudia

```
SELECT the_geom FROM
    (SELECT DISTINCT ON (id_hm)
            pochovany_adresa.id,
            meno,
            priezvisko,
            rodne_meno,
            datum_narodenia,
            datum_narodenia_nepresny,
            datum_umrtia,
            datum_umrtia_nepresny,
            tituly,
            ml_st,
            adresa,
            pohlavie,
            ulica,
            cislo,
            obec,
            okres,
            stat,
            id_hm,
            the_geom
      FROM pochovany_adresa
      LEFT JOIN hrobove_miesto
        ON pochovany_adresa.id_hm=hrobove_miesto.id
      ) AS nova USING UNIQUE id USING SRID=-1
```

- ku ktorým sa viažu nájomcovia

```
SELECT the_geom FROM
  (SELECT DISTINCT ON (id_hm)
          meno,
          priezvisko,
          rodne_meno,
          datum_narodenia,
          datum_narodenia_nepresny,
          datum_umrtia,
          datum_umrtia_nepresny,
          tituly,
          ml_st,
          pohlavie,
          id_hm,
```

```
            the_geom
        FROM najomca_adresa_hm
          LEFT JOIN hrobove_miesto ON
          najomca_adresa_hm.id_hm=hrobove_miesto.id
    ) AS nova USING UNIQUE id USING SRID=-1
```

- ku ktorým sa viažu nájomcovia a pochovaní

```
SELECT the_geom FROM
  (SELECT DISTINCT ON (id_hm)
          meno,
          priezvisko,
          id_hm,
          the_geom
       FROM pochovany_adresa
         LEFT JOIN hrobove_miesto ON
         pochovany_adresa.id_hm=hrobove_miesto.id
     UNION
     SELECT DISTINCT ON (id_hm)
            meno,
            priezvisko,
            id_hm,
            the_geom FROM najomca_adresa_hm
         LEFT JOIN hrobove_miesto ON
         najomca_adresa_hm.id_hm=hrobove_miesto.id)
     AS nova USING UNIQUE id USING SRID=-1
```

14. Hrobové miesta (geometria), ku ktorým sa viažu vybrané zmluvy (špecifikované cez klauzulu WHERE)

```
SELECT the_geom FROM
  (SELECT pochovany_adresa.id,
          meno,
          priezvisko,
          rodne_meno,
          datum_narodenia,
          datum_narodenia_nepresny,
          datum_umrtia,
          datum_umrtia_nepresny,
          tituly,
          ml_st,
          adresa,
          pohlavie,
          ulica,
          cislo,
          obec,
          okres,
          stat,
```

```
            id_hm,
            hrobove_miesto.id AS hmid
        FROM zmluvy
          LEFT JOIN hrobove_miesto
            ON hrobove_miesto.id=zmluvy.id_hm
      ) AS hrobove_miesto
          LEFT JOIN pochovany_adresa
            ON hrobove_miesto.id=pochovany_adresa.id_hm
    ) AS nova USING UNIQUE id USING SRID=-1
```

15. Všetky hrobové miesta (geometria)

```
SELECT the_geom FROM hrobove_miesto
```

16. Vklad do tabuliek `clovek`, `adresa`, `poznamka` (pri vkladaní nového človeka)

```
INSERT INTO clovek (meno,
                    priezvisko,
                    ml_st,
                    rodne_meno,
                    pohlavie,
                    tituly,
                    datum_narodenia,
                    datum_narodenia_nepresny,
                    datum_umrtia,
                    datum_umrtia_nepresny
                    )
          VALUES ('Cyril',
                  'Zomrel',
                  'st.',
                  '',
                  'M',
                  'Ing.',
                  '1905-01-01',
                  'F',
                  '1985-01-01',
                  'F'
                 )
INSERT INTO adresa (ulica,
                    cislo,
                    obec,
                    okres,
                    stat)
               VALUES ('Gaštanová',
                       32,
                       'Tajov',
```

```
                        'Banská Bystrica',
                        'Slovensko')
INSERT INTO poznamka (autor,
                      datum_vlozenia,
                      id_zaznam,
                      tabulka,
                      obsah)
             VALUES ('spravca',
                     '2010-01-01',
                     1,
                     'ludia',
                     'nepresný záznam')
```

17. Vklad do tabuliek `zmluvy`, `pochovani`, `najomca`, `fiktivne_zmluvy`, `poznamka` (pri pohrebe)

```
INSERT INTO zmluvy (id_hm,
                    najom_od,
                    najom_do,
                    zaplatene,
                    datum_platby,
                    cu)
            VALUES (1,
                    2010-01-01,
                    2020-01-01,
                    30,
                    2009-12-30,
                    122345
            );

INSERT INTO pochovani (id_pochovany,
                       idhm)
               VALUES (1,
                       1);

INSERT INTO najomca (id_najomca,
                     id_zmluva)
             VALUES (1,
                     1);

INSERT INTO fiktivne_zmluvy (id_zmluva)
                     VALUES (2);

INSERT INTO stare_zmluvy (id_zmluva,
                          csz)
                  VALUES (1,
                          2);
```

```
INSERT INTO poznamka (autor,
                      datum_vlozenia,
                      id_zaznam,
                      tabulka,
                      obsah)
                VALUES ('uzivatel',
                        NOW(),
                        1,
                        'ludia',
                        'nepresný záznam')
                        );
```

11.3 Diskusia

Typy geometrie
Z hľadiska navrhovania geografickej databázy a priestorových dopytov boli zistené viaceré zaujímavé skutočnosti. Jedna z otázok, ktorá sa objavila, je voľba typu geometrie. Vo všeobecnosti existujú tri typy geometrií – geometria bodová, čiarová a plošná. Avšak v PostGISe existuje viac typov geometrií viď kapitola 2.2.4 na str. 31.

Užitočnosť tzv. MULTI geometrií sa ukázala práve pri návrhu priestorových dopytov. Napríklad v prípade, že nás zaujímajú všetky hrobové miesta vzdialené od stromov viac ako 2 metre, je veľmi problematické vytvoriť funkčný dopyt v prípade, že majú stromy geometriu POINT. Vzdialenosť sa totiž skúma pri všetkých usporiadaných dvojiciach strom-hrobové miesto. Preto pokiaľ existuje aspoň jeden taký strom, ktorý je od hrobového miesta vzdialený viac ako 2 metre (nezávisle od toho, že existuje aj strom, ktorý je v blízkosti do dvoch metrov), dopyt bude úspešný. Nás však zaujímajú len také miesta, kde žiadny strom nie je bližšie ako dva metre. Výsledkom je veľmi komplikovaný viacnásobne vnorený dopyt. Druhé riešenie je pozerať sa na všetky stromy ako na jeden objekt (to je možné s použitím geometrie `MULTIPOINT`). Výsledkom je jednoduchý dopyt v tvare

```
NOT ST_DWithin(hrobove_miesto.the_geom,
           multistrom.the_geom,
           2)
```

.

Ďalšou vlastnosťou, ktorá sa objavila pri návrhu databázy ako veľmi užitočná je možnosť použitia viacerých stĺpcov (atribútov) s geometriou pre jeden objekt. Tak je možné mať jeden objekt reprezentovaný niekoľkými typmi geometrie súčasne a pri dopytoch využívať ten v danú chvíľu vhodnejší. Napríklad pre chodník pri vykresľovaní mapy v mierke 1:200 je vhodnejšie, aby bol chodník reprezentovaný ako polygón, hoci entita chodník má prirodzene líniový charakter. Pri výpočtoch (napríklad hľadanie trasy k danému miestu sa ale využije typ geometrie línia).

Kapitola 12

Priestorové analýzy s využitím DTM – vybrané aplikácie

Kapitola obsahuje niekoľko riešených príkladov ilustrujúcich použitie viacerých techník opísaných v tomto učebnom texte. Na riešenie úloh bol okrem softvéru opísaného v kapitole 3 využitý aj nástroj *OpenModeller*[1] (verzia 1.5.0). Na tvorbu DTM boli využité lomové body vrstevníc získané zo Spojitej vektorovej mapy 1:50 000 (SVM50). Ostatné údaje boli čerpané z Atlasu krajiny SR (2001) a s využitím podkladu vo forme webovej mapovej služby (WMS) Slovenskej agentúry životného prostredia. Kapitola je rozšírením článku Balážovič (2009).

12.1 Umiestnenie vyhliadok v národnom parku Muránska planina

Umiestnenie vyhliadok v NP Muránska planina

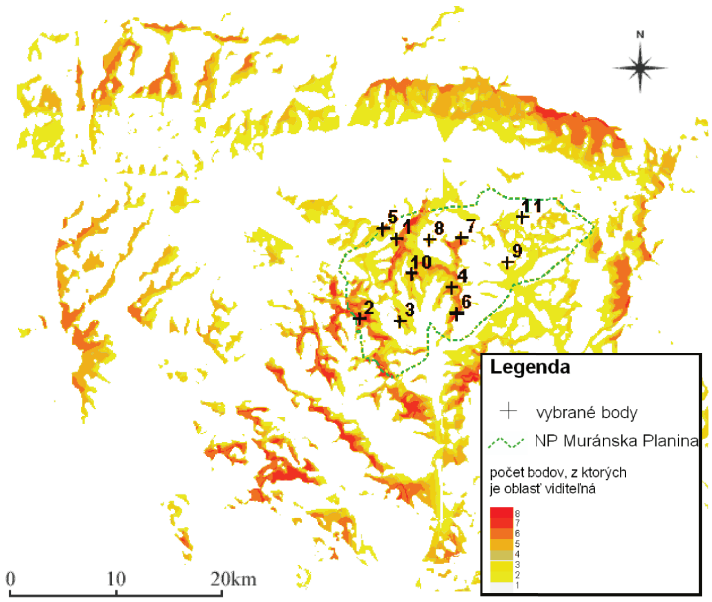
Cieľom bude vybrať miesta s najlepším rozhľadom na území Národného parku Muránska planina.

Na vyriešenie úlohy budeme potrebovať rastrovú vrstvu digitálneho terénneho modelu (DTM) a vrstvu vymedzujúcu územie NP Muránska planina. Pokiaľ nemáme k dispozícii DTM, môžeme ho vytvoriť interpolačnými metódami z nepravidleného bodového poľa alebo siete vrstevníc (viac v časti 5.2.6 na str. 5.2.6). Na spracovanie údajov využijeme QGIS s modulom GRASSu `r.los` určeným na analýzu viditeľnosti. Zakrivenie Zeme, ovplyvňuje viditeľnosť vzdialených oblastí preto ho pri riešení úlohy zanedbáme.

[1] `http://openmodeller.org`

bod	x	y	z	viditeľnosť (km^2)	meno
1	-363273	-1228025	1421	353,56	Fabova Hoľa
2	-366811	-1235540	1018	176,65	Kostolný Vrch
3	-362944	-1235675	952	151,65	Pacherka
4	-358087	-1232548	1184	165,16	Ostrica
5	-364527	-1227079	1430	464,46	Javorinka
6	-357675	-1235009	1161	309,82	Šarkanica
7	-357291	-1227925	1400	309,46	Kľak
8	-360242	-1228071	1191	93,84	Malá Stožka
9	-352952	-1230175	863	6,79	Poludnica
10	-361962	-1231238	1141	179,99	Kľak
11	-351602	-1225951	1109	99,47	Lopušná
súčet				712,72	

Tabuľka 12.1: Tabuľka vybraných bodov v NP Muránska planina



Obr. 12.1: Mapa viditeľnosti (NP Muránska planina)

Najskôr bolo intuitívne vybraných 11 bodov z oblasti národného parku s pravdepodobne vysokou viditeľnosťou. Body tvorili vrcholy, resp. okraje krasovej planiny. Modulom `r.los` bola pre každý z vybraných bodov vytvorená mapa s oblasťami viditeľnými z daného bodu. Vychádzalo sa z ideálneho prípadu, kde terén je odlesnený a výška pozorovateľa je 1,6 m. Ich súčtom vznikla výsledná mapa lokalizácie miest s najlepším rozhľadom (obr. 12.1).

Ako bod s najlepším rozhľadom v národnom parku Muránska planina bol z tejto analýzy určený vrch Javorinka, ktorý je zároveň najvyšším vrchom v oblasti. Jeho ďalšou výhodou je to, že leží na severovýchodnom okraji planiny, odkiaľ je výhľad najmä na chrbát Nízkych Tatier a Poľanu. Pri ostatných vybraných bodoch je zrejmá závislosť viditeľnosti od nadmorskej výšky. Z väčšiny vybraných bodov je dobrý výhľad na vrchy v okolí Muránskej Planiny (Tŕstie, Ostrá, Kráľova hoľa, ...).

12.2 Potenciálny výskyt nezvestného človeka

Úloha simulovala stratu dieťaťa predškolského veku na modelovom území (v heterogénnom teréne). Cieľom bolo vymedziť územie, v ktorom po ňom treba pátrať po 1 až 5 hodinách. Úlohu budeme riešiť na podklade vstupných údajov z územia okresu Púchov. Zadanie projektu si vyžiadalo použiť modul `r.walk` technologického prostredia GRASS. Vstupom do modulu bol DTM územia v rastrovom formáte a raster nákladovosti pohybu (tzv. frikčný raster). Výsledný raster obsahuje najnižšiu kumulatívnu hodnotu časovej nákladnosti pohybu medzi bunkami vstupných rastrov od zadaného počiatočného bodu, určený podľa vzťahu:

$$N_c = N_p + \lambda * N_f \tag{12.1}$$

kde N_c je výsledná časová nákladnosť, N_p – pohybová časová nákladnosť, N_f – frikčná časová nákladovosť, λ – koeficient váhy N_f na N_c.

Člen N_p je závislý na hodnote sklonu georeliéfu v smere spádových kriviek (ďalej len sklon) a kladnej alebo zápornej hodnote výškového gradientu v smere pohybu. Vypočíta sa ako súčet

$$N_p = X + Y + Z + W \tag{12.2}$$

kde: $X = a * \Delta S$, $Y = b * \Delta H$, $Z = c * \Delta H$, $W = d * \Delta H$ a ΔS predstavuje vzdialenosť, ΔH prevýšenie medzi susednými bodmi.

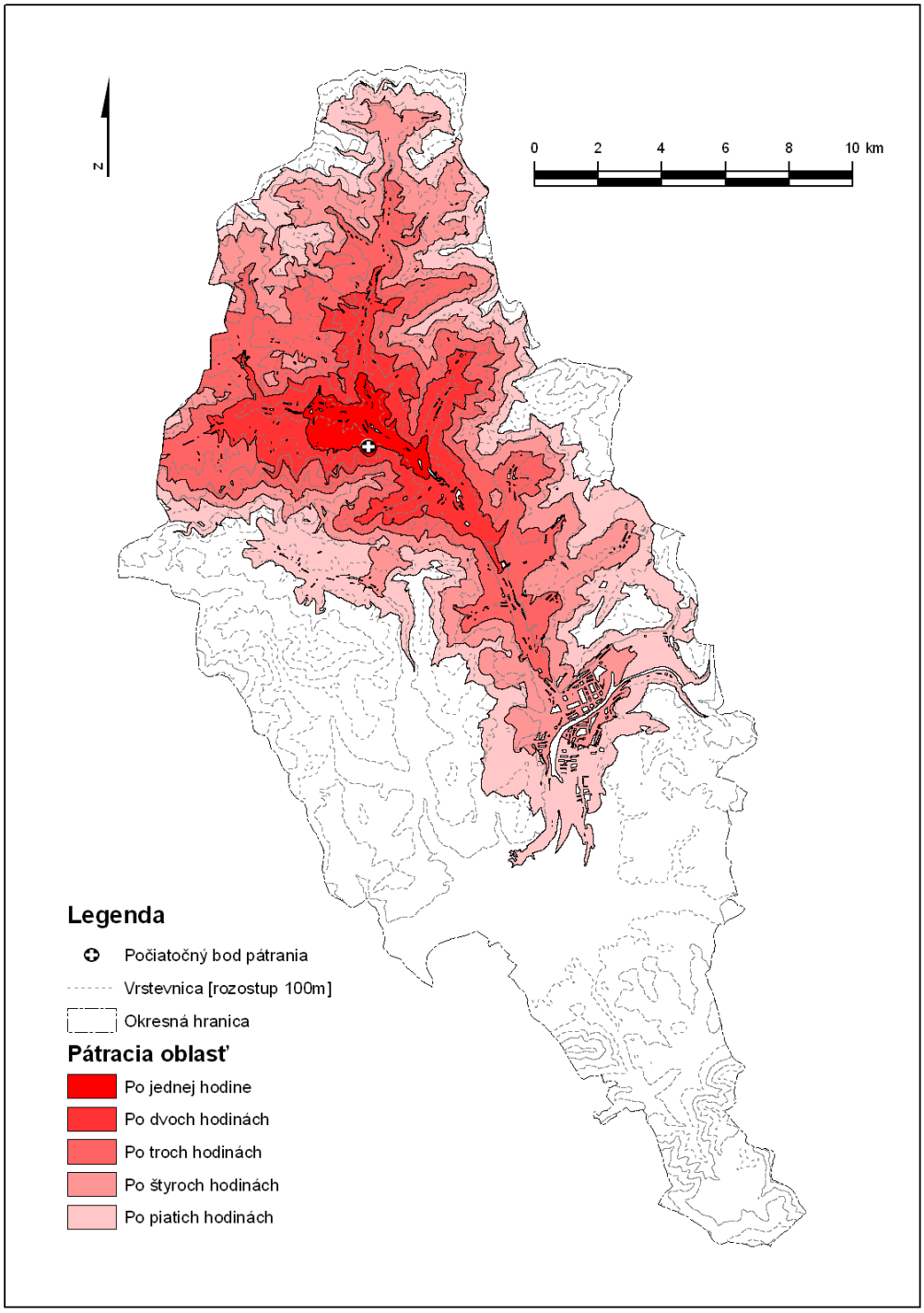
Koeficient a je prevrátenou hodnotou rýchlosti pohybu ($a = 0,931$), koeficienty b, c, d vyjadrujú zmenu rýchlosti pohybu pri rôznych podmienkach ako sú chôdza do kopca ($b = 6,0$), z kopca ($c = 1,9998$), dole strmým kopcom ($d = -1,9998$). Hodnoty boli prevzaté z manuálu programu GRASS 2015. Podľa charakteru georeliéfu (hodnoty sklonu a gradientu) do výpočtu N_p vstupujú tieto kombinácie členov rovnice:

$N_p = X + Y$, ak je sklon 0 až 90°a $\Delta H > 0$,
$N_p = X$, ak je sklon 0 až 5°a $\Delta H <= 0$,
$N_p = X + Z$, ak je sklon 5 až 12°a $\Delta H < 0$,
$N_p = X + W$, ak je sklon 12 až 90°a $\Delta H < 0$.

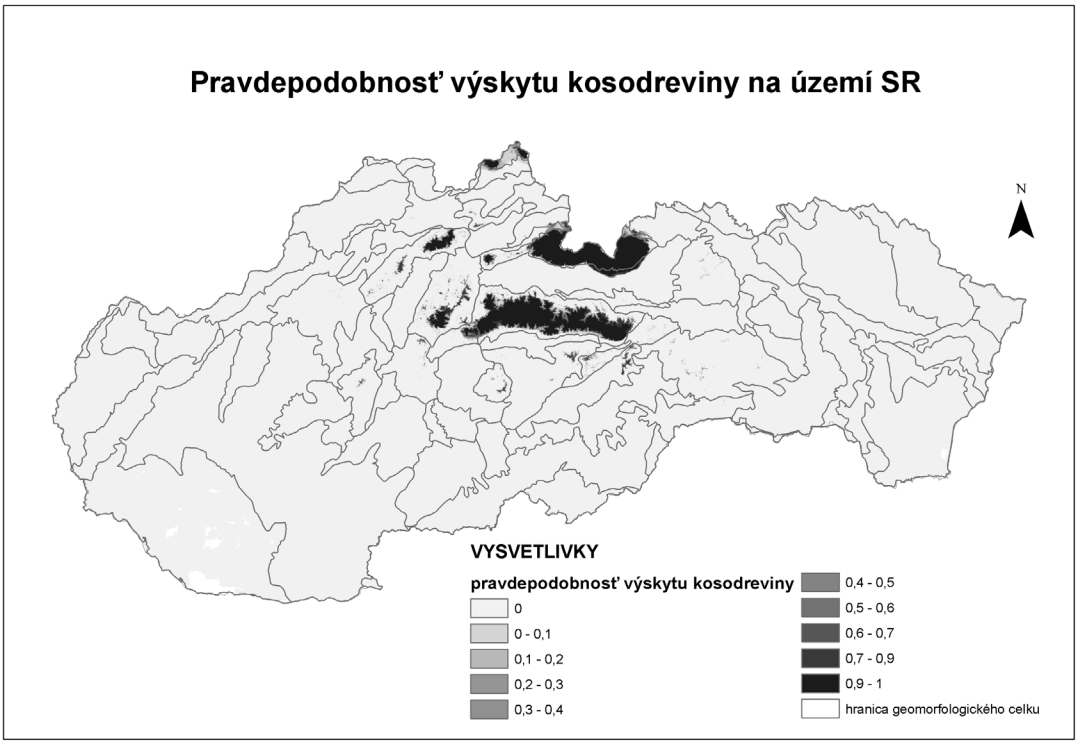
Člen N_f bol vytvorený reklasifikáciou rastrov krajinnej pokrývky, vodstva, cestnej siete a maloplošných geomorfologických foriem. Podľa náročnosti pohybu im boli priradené hodnoty zvýšenej časovej nákladovosti v sekundách. Výsledná mapa je uvedená na obr. 12.2.

12.3 Potenciálny výskyt borovice horskej

Cieľom je zistiť potenciálny výskyt borovice horskej (Pinus Mugo) na území Slovenskej republiky s využitím DTM a faktormi limitujúcimi jej výskyt.
Vstupné údaje tvorili lomové body vrstevníc získané z SVM50, izolínie priemerných ročných teplôt a zrážok a mapy pôdnych druhov, pôdnych typov



Obr. 12.2: Potenciálny výskyt kosodreviny na území Slovenskej republiky



Obr. 12.3: Potencionálny výskyt kosodreviny na území Slovenskej republiky

a mapa využitia zeme z Atlasu krajiny SR (2001). Základom práce bolo vytvorenie DTM celého územia Slovenskej republiky. Z lomových bodov vrstevníc SVM50 sme vytvorili rastre nadmorských výšok, sklonu reliéfu v smere spádovej krivky a orientácie reliéfu voči svetovým stranám v priestorovom rozlíšení 50 metrov.

Z izolínií priemerných ročných teplôt a zrážok sme vytvorili modulom `v.surf.rst` rastre priemerných ročných zrážok a teplôt.

Mapy pôdnych druhov a typov boli pôvodne vo vektorovej forme a pre konečný výpočet ich bolo potrebné previesť na rastre.

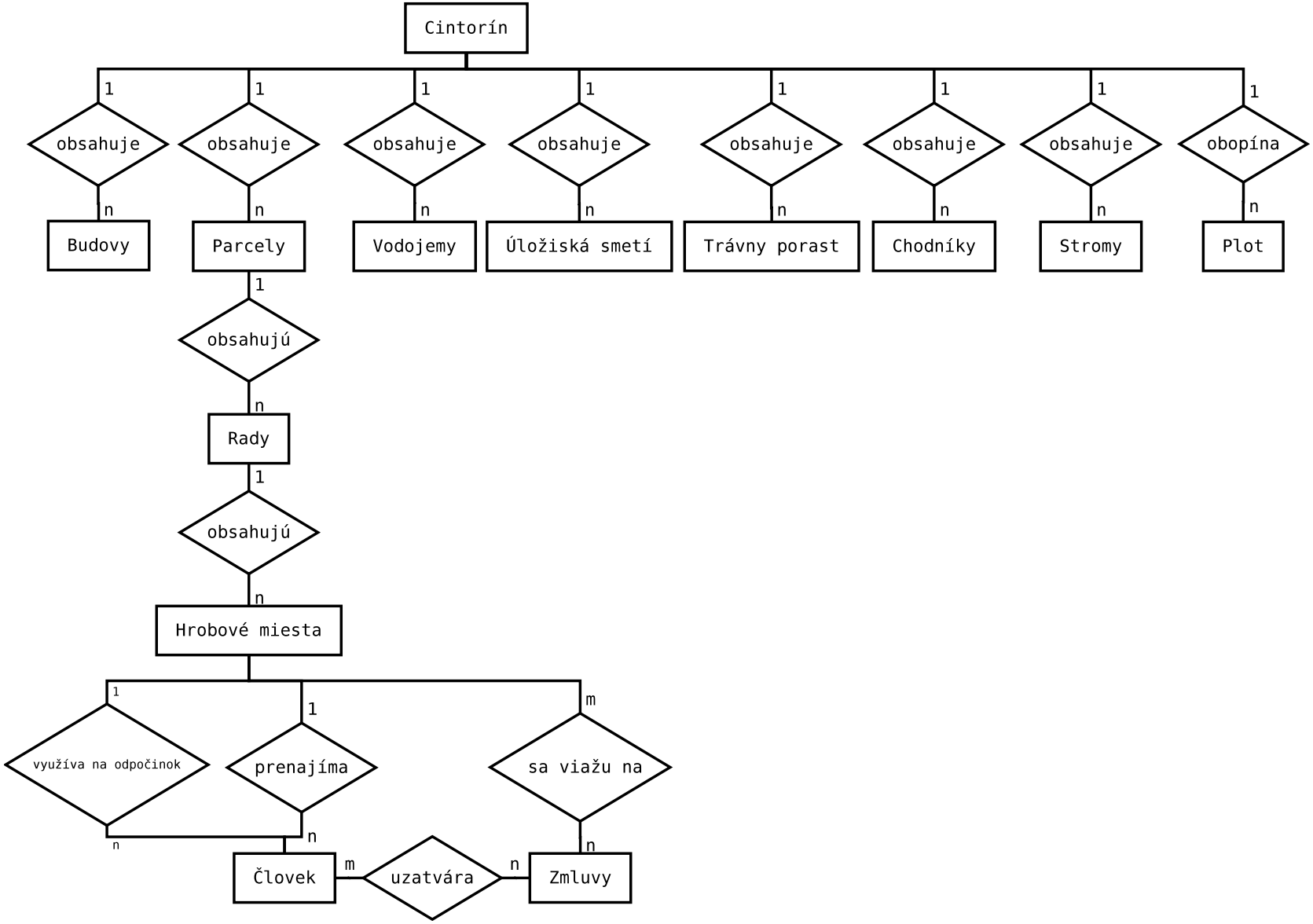
Výpočet potencionálneho výskytu borovice horskej prebiehal v prostredí programu OpenModeler. Do výpočtu vstupovali všetky spomínané rastre limitujúcich faktorov a bodová vrstva reálneho výskytu kosodreviny odvodená z mapy využitia zeme v Atlase krajiny SR (Miklós, 2001). OpenModeller obsahuje viacero algoritmov na výpočet priestorového rozšírenia druhu alebo javov v geografickej sfére. Na naše výpočty bol využitý algoritmus GARP (Genetic Algorithm for Rule Set Production), ktorý je presne stavaný na výpočet priestorového rozšírenia druhu na základe jeho limitujúcich faktorov. Výsledkom výpočtu bol raster potenciálneho výskytu borovice horskej Nadobúda hodnoty od 0 až po 1, kde 0 znamená najmenšiu pravdepodobnosť a 1 najvyššiu pravdepodobnosť výskytu.

Pre lepšie znázornenie a hodnotenie sme na raster naložili hranice geomor-

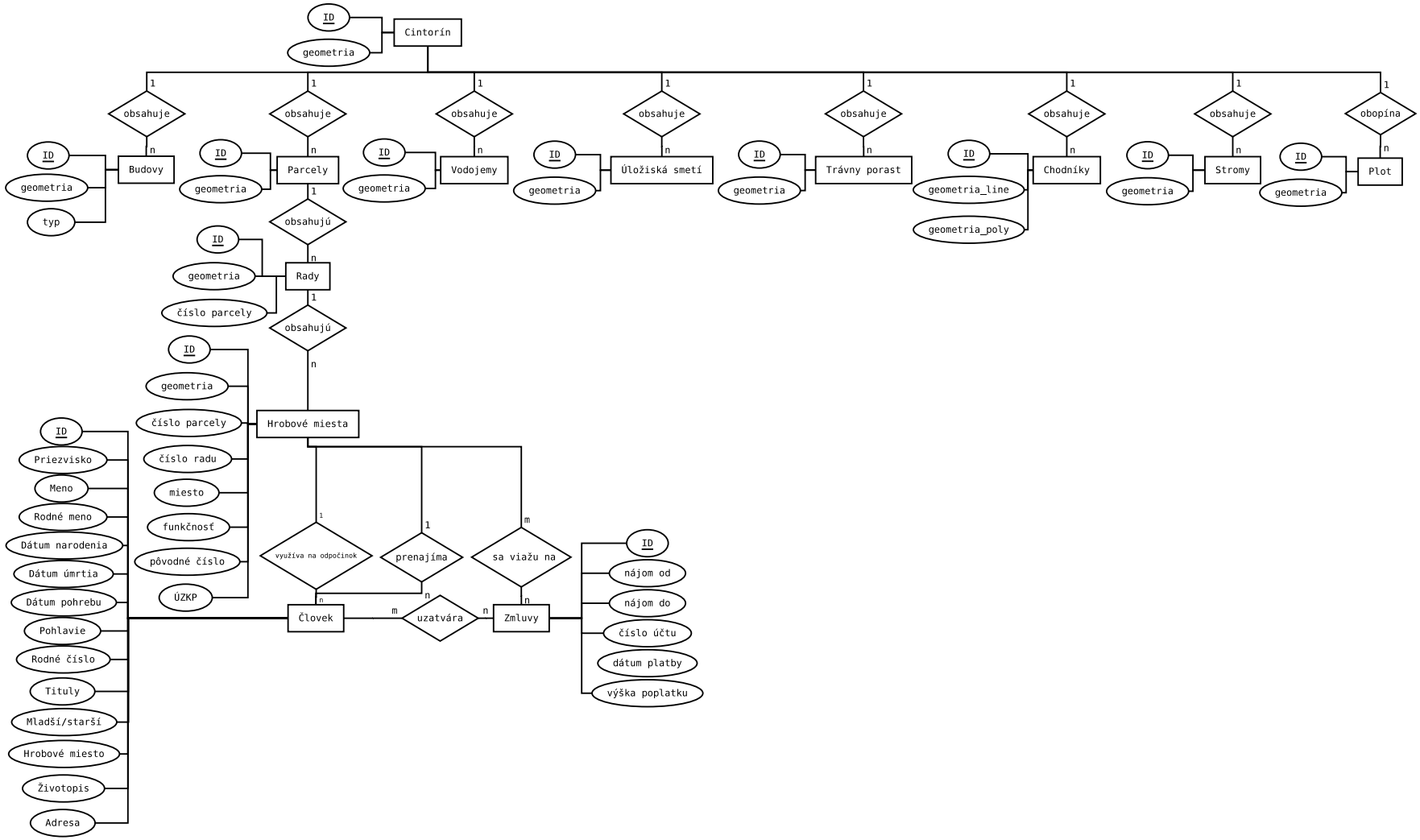
fologických celkov Slovenskej republiky (obr. 12.3). Najväčšia pravdepodobnosť výskytu sa objavila v pohoriach Tatry, Nízke Tatry, Malá a Veľká Fatra, v Oravských Beskydách (hlavne Babia Hora), v Chočských vrchoch. Malé ostrovy výskytu sa nachádzajú v najvyššie položených oblastiach pohorí Vtáčnik, Kremnické vrchy, Starohorské vrchy, Poľana, Veporské vrchy, Stolické vrchy, Volovské vrchy, Spišsko-gemerský kras, Levočské vrchy a Bukovské vrchy. Tieto výsledky sme porovnali s mapou Potenciálnej prirodzenej vegetácie SR z Atlasu krajiny 2001. Naše výsledky sa zhodovali v plnej miere.

Príloha A

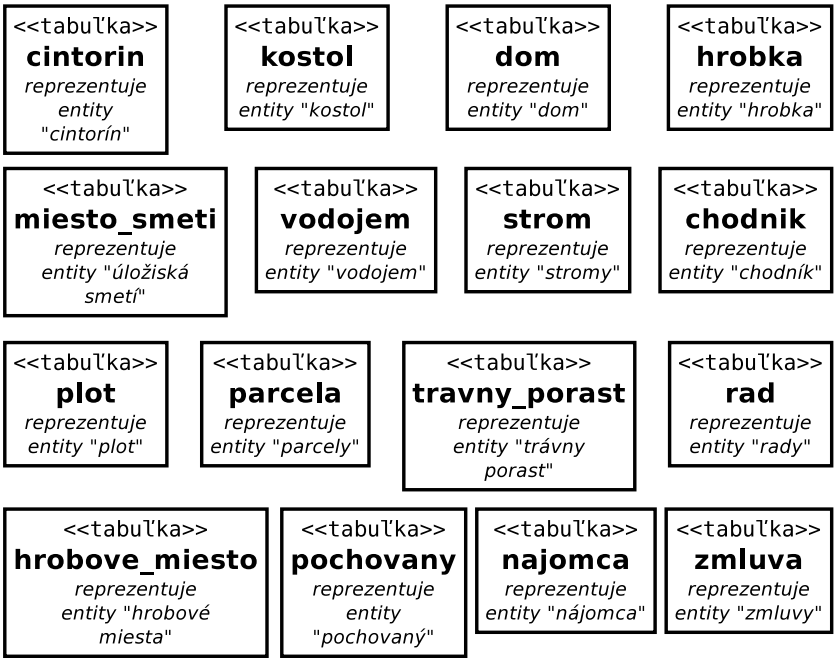
Schémy



Obr. A.1: ER diagram údajového modelu pre *Systém na správu cintorína*

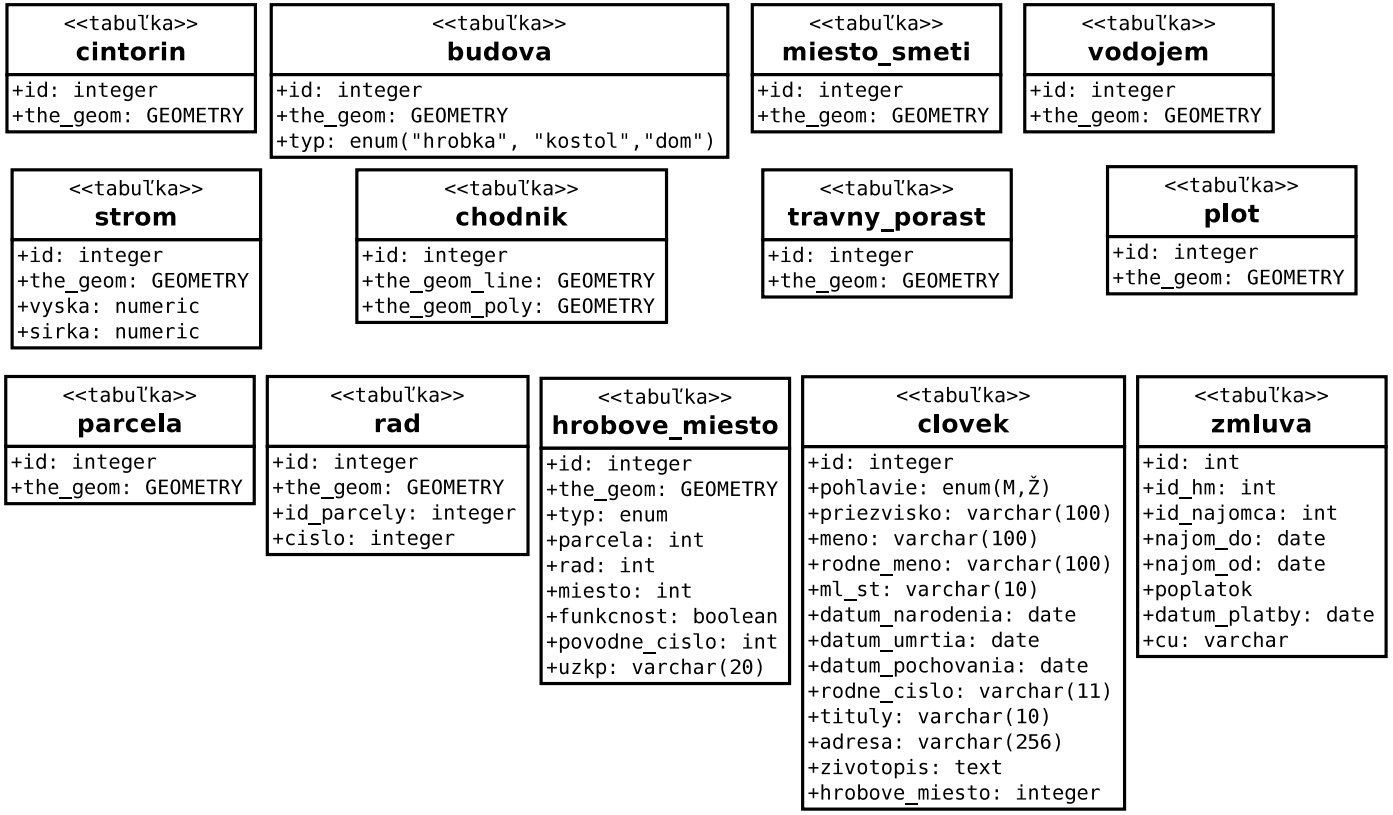


Obr. A.2: Rozšírený ER diagram údajového modelu pre *Systém na správu cintorína*

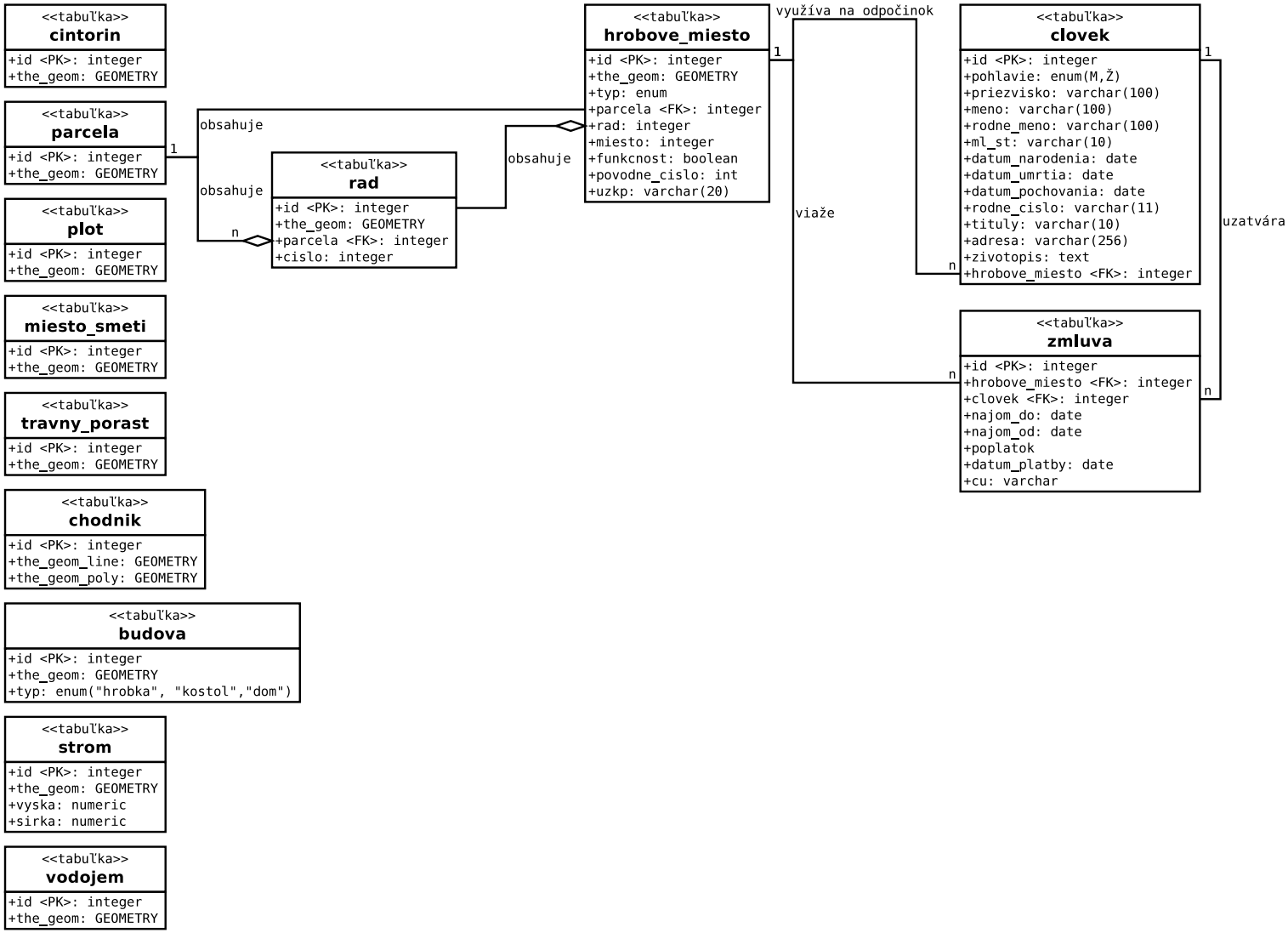


Obr. A.3: Mapovanie entít na triedy objektov (*Systém na správu cintorína*)

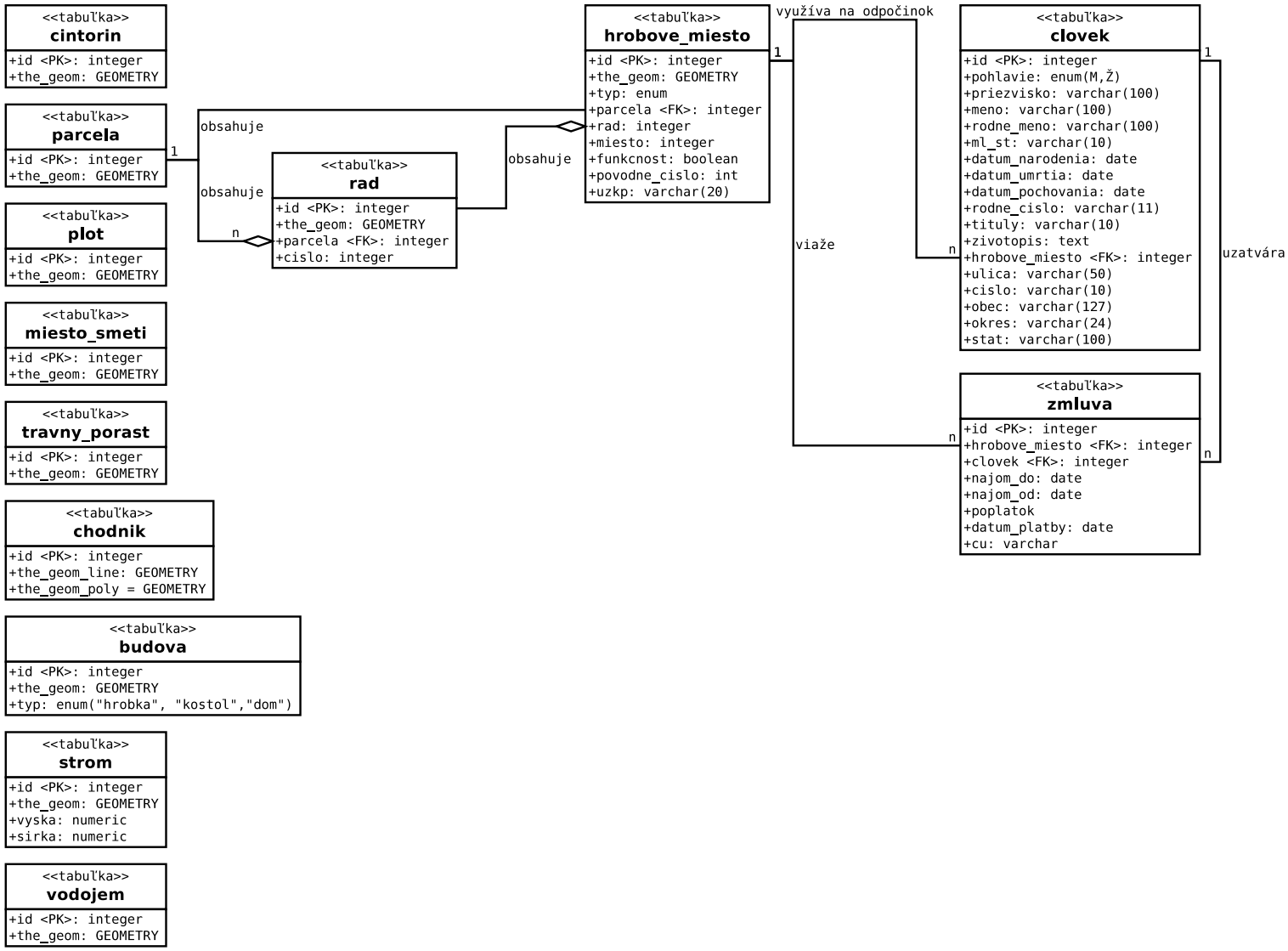
Obr. A.4: Usporiadnie triedy objektov do vrstiev (*Systém na správu cintorína*)



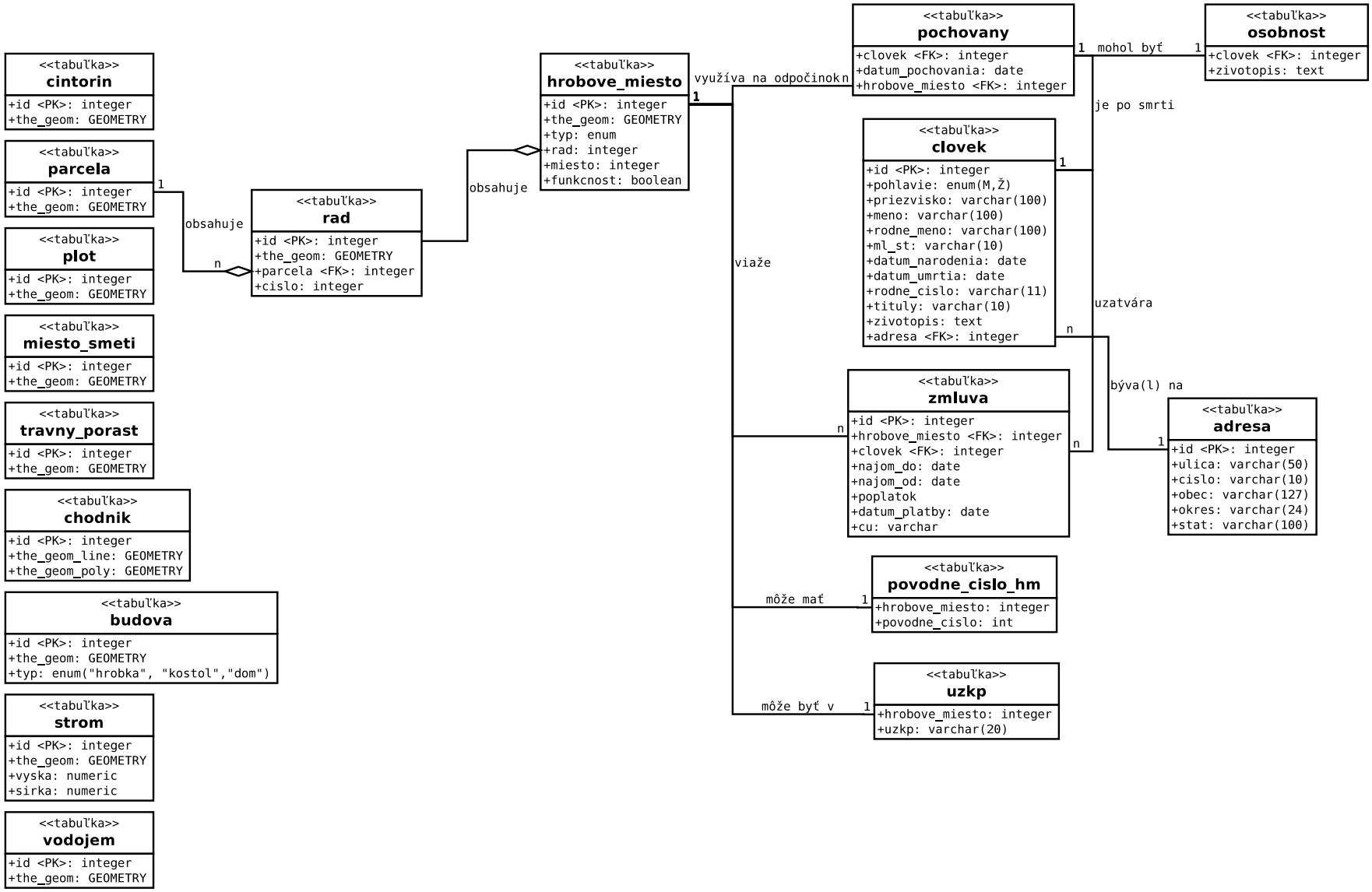
Obr. A.5: Triedy objektov s atribútmi (*Systém na správu cintorína*)



Obr. A.6: Tvorba relácií (*Systém na správu cintorína*)



Obr. A.7: Normalizovaný údajový model (diagram tried v 1NF)



Obr. A.8: Normalizovaný údajový model (diagram tried v 3NF)

Príloha B

Výpisy zdrojového kódu

B.1 Systém na správu cintorína (SSC)

Vytvorenie tabuliek

```
CREATE TABLE hrobove_miesto (id serial,
                             typ varchar(20),
                             rad int,
                             funkcnost boolean);
```

Priestorové rozšírenie tabuliek

```
SELECT AddGeometryColumn ('public',
                          'hrobove_miesto',
                          'the_geom',-1,'POLYGON',2);
```

Obmedzenia

```
ALTER TABLE hrobove_miesto ADD PRIMARY KEY (id);

ALTER TABLE hrobove_miesto ADD FOREIGN KEY (rad)
  REFERENCES rad(id);
ALTER TABLE hrobove_miesto ADD FOREIGN KEY (typ)
  REFERENCES tyb_hroboveho_miesta(typ);

ALTER TABLE hrobove_miesto
  ALTER COLUMN funkcnost SET NOT NULL;

ALTER TABLE hrobove_miesto ADD UNIQUE (the_geom);
```

Zoznam bibliografických odkazov

ANSTEAD, L. Using radial basis function with a multi-quadric option for observing changes in river channel morphology. 2013, 13, 1, s. 14. ISSN 1337-6799.

BALÁŽOVIČ, Ľ. Geografické informačné služby vo vzdelávaní spoločnosti, 2010.

BALÁŽOVIČ, Ľ. Aplikované priestorové analýzy v geografických informačných systémoch s využitím digitálneho terénneho modelu. *Kartografické listy*. 2009, 17, s. 123–126. ISSN ISSN 1336-5274.

BATINI, C. – CERI, S. – NAVATHE, S. *Conceptual Database Design, An Entity Relationship Approach*. Redwood City, CA : The Benjamin/CUmmings Publishing Company, Inc., 1992.

BERRY, B. – MARBLE, D. *Spatial Analysis*. Englewood Cliffs, NJ : PrenticeHall, 1968.

BLAZEK, R. Introducing the Linear Reference System in GRASS. *International Journal of Geoinformatics*. 2005, 3, 1, s. 95–100.

BOHLING, G. Introduction to geostatistics and variogram analysis. 2005. Dostupné na: `http://www.people.ku.edu/~gbohling/cpe940/Variograms.pdf`.

BURIAN, L. – JENČO, M. – RUSNÁK, M. *GRASS GIS Geovedné aplikácie*. Bratislava : Univerzita Komenského v Bratislave, Prírodovedecká fakulta a Geografický ústav SAV, 2015. ISBN 978-80-223-3804-2.

CRESSIE, N. *Statistics for Spatial Data*. New York : Wiley, 1991.

DATE, C. J. *An introduction to Database Systems*. : Addison Wesley Publishing, 8th ed vydanie, 2003.

FEKIAČ, M. Tvorba interaktívneho atlasu: geografická databáza. diplomová práca, Prírodovedecká fakulta UK, 2008.

GOODCHILD, M. F. Unit 055 - Rasters. 2000. Dostupné na: http://www.escholarship.org/uc/item/7s2326gs.

GRASS Development Team. *Geographic Resources Analysis Support System (GRASS GIS) Software*. ITC-irst, Trento, Italy, 2015. Dostupné na: http://grass.itc.it.

GRAY, J. – REUTER, A. *Transaction processing*. Morgan Kaufíann Publishers, 1993.

HAINING, R. P. Designing spatial data analysis modules for GIS. In FOTHERINGHAM, S. – ROGERSON, P. (Ed.) *Spatial analysis and GIS*. London: Taylor&Francis, 1994.

HAINING, R. P. *Spatial data analysis: theory and practice*. New York : Cambridge University Press, 2003. ISBN 978-0-521-77319-5 978-0-521-77437-6.

HLÁSNY, T. *Geografické informačné systém – Priestorové analýzy*. Zvolen : Zephyros a Národné lesnícke centrum – výskumný lesnícky ústav, 2007. ISBN 978-80-8093-029-5.

ISAAKS, E. H. – SRIVASTAVA, R. M. *Applied geostatistics*. New York : Oxford University Press, 1989. ISBN 978-0-19-505012-7 978-0-19-505013-4.

KALUŽA, J. – KALUŽOVÁ, L. *Modelování dat v informačních systémech*. Praha : Ekopress, 2012. ISBN 978-80-86929-81-1.

KOL. AUTOROV. *QGIS Features* [online]. 2015. [cit. 10. 21. 2015]. Dostupné na: http://docs.qgis.org/2.8/en/docs/user_manual/preamble/features.html.

KRCHO, J. *Modelling of georelief and its geometrical structure using DTM: Positional and numerical accuracy ? Modelovanie georeliéfu s jeho geometrickej štruktúry pomocou DTM: Polohová a numerická presnosť*. Bratislava : Q111, 2001.

KUDĚLKA, V. – HRIC, F. – DOBEŠOVÁ, Z. *QGIS 2.4.0 ? Processing modeler ? manuál* [online]. 2014. Dostupné na: http://freegis.fsv.cvut.cz/wiki/images/7/7e/Navod_QGIS_ModelerKomplet.pdf.

LOPEZ, A. *Moving Spatial: PostGIS 2.0 - Intersect Raster and Polygon* [online]. 2012. [cit. 13. 10. 2015]. Dostupné na: http://movingspatial.blogspot.sk/2012/07/postgis-20-intersect-raster-and-polygon.html.

MIČIETOVÁ, E. et al. *Špecializované informačné technológie v prírodovednom výskume: Geoinformačné technológie*. Bratislava : ELITA, 2008. ISBN 978-80-8044-081-7.

MIKLÓS, L. *Atlas krajiny SR*. Banská Štiavnica : Esprit, 2001.

MITÁŠOVÁ, H. – VEVERKA, B. – PEZLAR, Z. *Základy teórie systémov a kybernetiky s aplikáciami v geodézii a kartografii*. Bratislava : Alfa, 1990.

MITÁŠOVÁ, H. et al. Multidimensional Interpolation, ANalysis and Visualization for Environmental Modeling. In *GIS/LIS '93 Annual Conference, November 2-4, 1993*, s. 550–556, 1995.

OBE, R. O. – HSU, L. S. *PostGIS in action*. Shelter Island : Manning, second edition vydanie, 2015. ISBN 978-1-61729-139-5.

PELS, H. J. Classification hierarchies for product data modelling. 17, 4, s. 367–377. ISSN 0953-7287, 1366-5871. doi: 10.1080/09537280600608997. Dostupné na: `http://www.tandfonline.com/doi/abs/10.1080/09537280600608997`.

PENDER, T. *UML Bible*. Indianapolis : Taylor and Francis, 2003.

PEQUET. *Introductory Reading in GIS*. New York : Taylor and Francis, 1990.

POSTGRESQL GLOBAL DEVELOPMENT GROUP. *PostgreSQL: About* [online]. [cit. 20. 10. 2015]. Dostupné na: `http://www.postgresql.org/about/`.

REFRACTIONS RESEARCH. *PostGIS : Home* [online]. 2015. [cit. 21. 10. 2015]. Dostupné na: `http://postgis.refractions.net/`.

RIPLEY, B. *Spatial Statistics*. New York : Wiley, 1981.

SÁRKÖZY, F. – ZÁVOTY, J. Conceptional Data Model for Modeling of Scalar Fields and One Compression Method Usable for its Implementation. In *Fourth International Symposium of Liesmars Towards three Dimensional Temporal and Dynamic Spatial Data Modelling and Analysis*, 1995.

SIMSION, G. C. – WITT, G. C. *Data modeling essentials*. Amsterdam ; Boston : Morgan Kaufmann Publishers, 3rd ed vydanie, 2005. ISBN 978-0-12-644551-0.

The PostGIS Development Group. *PostGIS : Home* [online]. 2015. [cit. 4. 10. 2015]. Dostupné na: `http://postgis.refractions.net/`.

TOMLIN, D. C. *Geographic Information Systems and Cartographic Modelling*. Engelwood Cliffs, NY : PrenticeHall, 1990.

TURNER, M. – GARDNER – O'NEILL. *Landscape Ecology in Theory and Practice, Patterns and Process*. New York : Springer, 2001.

WIKIPEDIA. *First normal form — Wikipedia, The Free Encyclopedia* [online]. 2015a. [cit. 21. 10. 2015]. Dostupné na: `https://en.wikipedia.org/w/index.php?title=First_normal_form&oldid=685517332`.

WIKIPEDIA. *PostgreSQL — Wikipedia, The Free Encyclopedia* [online]. 2015b. [cit. 21. 10. 2015]. Dostupné na: `https://en.wikipedia.org/w/index.php?title=PostgreSQL&oldid=686798760`.

Názov diela: Spracovanie a analýza geopriestorových údajov
Autor: Ľuboš Balážovič
Vydavateľ: Vydavateľstvo Univerzity Mateja Bela v Banskej Bystrici Belianum
Počet strán: 160 strán
Náklad: 100 kusov
Vydanie: prvé
Formát: A4
Rok vydania: 2015
Tlač: EQUILIBRIA, s.r.o., Košice

ISBN 978-80-557-0953-6

©Univerzita Mateja Bela v Banskej Bystrici, 2015
ISBN 978-80-557-0953-6