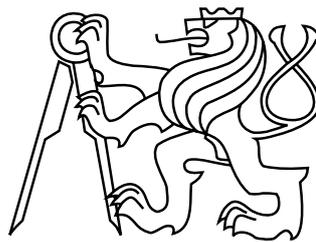


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STAVEBNÍ

DIPLOMOVÁ PRÁCE

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STAVEBNÍ
OBOR GEOINFORMATIKA



DIPLOMOVÁ PRÁCE
TVORBA DATABÁZE TÍHOVÝCH BODŮ, GRAVIMETRŮ A
ZÁZNAMŮ GRAVIMETRICKÝCH MĚŘENÍ S WEBOVÝM
UŽIVATELSKÝM ROZHRAŇÍM

Vedoucí práce: Ing. Jiří Cajthaml, Ph.D.
Katedra mapování a kartografie

leden 2011

Jan GÜRTLER

ZDE VLOŽIT LIST ZADÁNÍ

Z důvodu správného číslování stránek

ABSTRAKT

Tato diplomová práce se zabývá návrhem, tvorbou a naplněním jednotné databáze PostgreSQL, která obsahuje záznamy gravimetrických měření, gravimetrů, kterými se měření prováděla, a tíhových bodů, na nichž se měřilo. Předmětem diplomové práce je dále webového rozhraní PHP pro správu a analýzu databáze, grafické znázornění tíhových bodů na mapě (Google maps) a generování místopisu tíhového bodu. Diplomová práce popisuje také vytvořený program pro hromadný převod dat do databáze, vytváření XML výstupů pro vyrovnávací software, tvorbu výstupů pro program Surfer a ukázkou databázové statistiky.

KLÍČOVÁ SLOVA

PostgreSQL, databáze, dotaz, XML výstup, SQL, PHP, webové rozhraní, Java SE, extrakce dat, Google mapy

ABSTRACT

This thesis deals with proposing, creation and filling of a PostgreSQL database consisting of information about gravity meters, gravity observations and observation places. Another point is a web interface for a user-friendly administration and an analysis, a display of observation places in a map (Google maps) and generating a description of observation places. This thesis also deals with software for a mass data copy into the database, XML outputs for adjustment calculus software, database outputs for Surfer software and an example of the statistics creation.

KEYWORDS

PostgreSQL, database, query, XML output, SQL, PHP, web interface, Java SE, data extraction, Google maps

PROHLÁŠENÍ

Prohlašuji, že diplomovou práci na téma „Tvorba databáze tíhových bodů, gravimetrů a záznamů gravimetrických měření s webovým uživatelským rozhraním“ jsem vypracoval samostatně. Použitou literaturu a podkladové materiály uvádím v seznamu zdrojů.

V Praze dne

.....

(podpis autora)

PODĚKOVÁNÍ

Chtěl bych poděkovat vedoucímu bakalářské práce Ing. Jiřímu Cajthamlovi, Ph.D. a vedoucímu oddělení gravimetrie ZÚ Ing. Martinu Ledererovi, Ph.D. za připomínky a pomoc při zpracování této práce. Dále bych chtěl poděkovat Ing. Otakaru Nesvadbovi, Ph.D. za nápadité řešení problému při zpracování zdrojových dat a za pomoc s nastavením databáze a PHP. Děkuji Karolíně za trpělivost, kterou se mnou měla po celou dobu psaní diplomové práce.

Obsah

Úvod	8
1 Databáze	9
1.1 Zdroje dat	9
1.1.1 FoxPro databáze tíhových bodů	9
1.1.2 Textové soubory zápisníků a rovnic oprav	10
1.1.3 Katalogy programu Gravimzu	15
1.1.4 Ostatní zdroje	17
1.2 Struktura databáze	17
1.2.1 Struktura uložení tíhových bodů	18
1.2.2 Struktura uložení denních úseků měření	26
1.2.3 Struktura uložení údajů o gravimetrech	29
1.3 Naplnění databáze	32
1.3.1 Hromadné kopírování CSV souborů	32
1.3.2 Program RegisterImport	34
2 Správa databáze	42
2.1 PHP webové rozhraní	42
2.1.1 Obsluha rozhraní	42
2.1.2 Soubory PHP skriptů	45
2.2 SQL úpravy s psql	50
3 Výstupy z databáze	55
3.1 XML výstup pro Gravimzung	55
3.2 Výstupy pro program Surfer	56
3.3 Tvorba statistik	58
3.4 Příprava dat před měřením	59
Závěr	61
Reference	63
Seznam symbolů, veličin a zkratk	65
Seznam příloh	66
A Zdroje dat	68
A.1 Textové soubory adjustovaných zápisníků	68
A.1.1 Typ 4, GS12 a GS11	68

A.1.2	Typ 5, GAG	68
A.1.3	Typ 6, Sharpe a Worden	69
A.1.4	Typ 7, LaCoste & Romberg G CPI	69
A.1.5	Typ 8, LaCoste & Romberg G a Scintrex	70
A.1.6	Typ 9, LaCoste & Romberg D	71
A.1.7	Soubory rovnic oprav	71
A.2	Katalogy gravimzu	73
A.2.1	Katalog měřických skupin	73
A.2.2	Katalog gravimetrů	75
A.2.3	Katalog tíhových bodů	76
B	Struktura databáze	77
B.1	Struktura uložení tíhových bodů	77
B.2	Struktura uložení denních úseků	80
B.3	Struktura uložení gravimetrů	81
C	Naplnění databáze	83
C.1	Skript pro program RegisterImport	83
D	PHP webové rozhraní	84
D.1	Snímky obrazovky	84
E	Výstupy z databáze	90
E.1	XML výstupy pro Gravimzung	90
E.1.1	query_point.sql	90
E.1.2	query_section.sql	91
E.1.3	query_calibration.sql	95
E.1.4	query_observation.sql	95
E.2	Výstupy pro program Surfer	96
E.2.1	Funkce get_points(point_id)	96
E.2.2	Funkce get_polygons(usek_id)	97
E.3	Tvorba statistik	98
E.3.1	Skript get_index	98
E.3.2	Četnost měření na bodech	99
E.3.3	Počet změřených denních úseků s daným počtem bodů	99
E.4	Grafické znázornění v programu Surfer	100

Úvod

V dalších odstavcích naleznete návrh struktury PostgreSQL relační databáze gravimetrických měření, gravimetrů a tíhových bodů společně se stručnými informacemi o zdrojích dat pro tuto databázi. Značnou část první kapitoly věnuji naplnění databáze, které probíhalo v závislosti na vstupním formátu buď hromadným nakopírováním nebo pomocí vytvořeného programu v jazyce Java a Bash skriptů.

V kapitole Správa databáze naleznete detailní popis webového rozhraní pro správu uložených tíhových bodů a tíhových měření. Součástí rozhraní je také generování místopisu tíhového bodu se zobrazením pozice na mapách Google.

Kapitola Výstupy z databáze se zabývá databázovými dotazy v jazyce SQL a PL/pgSQL funkcemi, které vytvářejí XML výstupy pro nový vyrovnávací program neznámých tíží a výstupy pro grafické znázornění měřených denních úseků a tíhových bodů v programu Surfer. Poslední částí této kapitoly je soubor ukázkových SQL dotazů pro vytváření databázových statistik a výběrů potřebných dat před měřením.

Přílohy obsahují ukázky vstupních dat, grafické znázornění struktury databáze, rozsáhlejší skripty, snímky obrazovky webového rozhraní a SQL dotazy pro tvorbu výstupů XML, výstupů pro program Surfer a pro tvorbu statistik.

1 Databáze

1.1 Zdroje dat

Oddělení gravimetrie Zeměměřického úřadu do dnešní doby používalo ke správě údajů o tíhovém bodovém poli databázi vedenou v programu FoxPro. Tato databáze byla zdrojem dat pro část nové databáze. Popis dat z databáze FoxPro naleznete v kapitole 1.1.1. Nejobsáhlejší část databáze tvoří naměřené a vypočtené údaje, které byly získány z výstupů vyrovnávacího programu Gravimzu, z textových souborů adjustovaných zápisníků a souborů rovnic oprav. Popisem těchto textových souborů se zabývá kapitola 1.1.2. Kapitola 1.1.3 popisuje údaje uložené v katalogích gravimetrů, měřických skupin a tíhových bodů, které načítá program Gravimzu při spuštění. Poslední kapitola 1.1.4 obsahuje všechny ostatní doplňkové zdroje dat.

1.1.1 FoxPro databáze tíhových bodů

FoxPro databáze tíhových bodů obsahuje pět základních souborů.

- Bod_zakl.dbf
- Bod_stav.dbf
- Bod_udaj.dbf
- Bod_tize.dbf
- Bod_mist.dbf

Bod_zakl.dbf obsahuje základní údaje o zařazení tíhového bodu do různých místních i mezinárodních sítí. Dále je zde popsáno umístění bodu od státu až po obec a určení mapového listu, na kterém se bod nachází. Bod_zakl.dbf obsahuje většinou informace zadané kódy. Popisy kódů jsou uloženy v těchto pomocných databázových souborech:

- bod_puv.dbf
- bod_stat.dbf
- cis_obce.dbf
- cis_okr.dbf
- bod_list.dbf
- druhbodu.dbf

- sit_m.dbf
- sit_s.dbf
- sit_v.dbf
- sit_57.dbf
- sit_64.dbf
- sit_95.dbf
- sit_05.dbf
- zakladna.dbf

Bod_stav.dbf obsahuje údaje o současném stavu tíhového bodu spolu s údaji o jeho stabilizaci a stavu místopisu. Dále obsahuje návrh opatření pro údržbu tíhových bodů. Informace jsou v souboru uloženy pomocí kódů. Informace vztahující se k příslušnému kódu jsou uloženy v pomocných souborech:

- kod_stav.dbf
- kod_mist.dbf

Bod_udaj.dbf obsahuje základní polohové a další měřené údaje o tíhovém bodě. Pomocné kódové soubory jsou následující:

- kod_sour.dbf
- kod_vys.dbf

Bod_tize.dbf obsahuje hodnoty tíže pro jednotlivé body v platných tíhových systémech.

Bod_mist.dbf obsahuje odkazy na uložené místopisné náčrty pro jednotlivé body a další doplňující informace pro tvorbu místopisu.

1.1.2 Textové soubory zápisníků a rovnic oprav

Dosud používaný vyrovnávací program Gravimzu po zadání naměřených údajů provede jejich kontrolu a uloží je do textových souborů adjustovaných zápisníků (např. AZ090101.031). Tyto soubory pak program dále zpracovává a počítá relativní tíže. Při výpočtu relativních tíží uživatel programu zadává stupeň chodu gravimetru, může vybraná měření vyloučit z vyrovnání nebo zadat skok v průběhu měření. Tyto tři informace (stupeň chodu, vyloučená měření a skok) se ukládají do textového souboru rovnic oprav (např. R1090101.031).

Pro ukládání naměřených dat nabízí Gravimzu šest různých typů adjustovaných zápisníků v závislosti na typu gravimetru:

typ 4 pro data naměřená gravimetrem Gs12 a Gs11

typ 5 pro ukládání dat naměřených gravimetrem GAG

typ 6 pro gravimetr Sharpe a Worden

typ 7 pro gravimetr LaCoste & Romberg G CPI

typ 8 pro gravimetry LaCoste & Romberg G a Scintrex

typ 9 pro gravimetr LaCoste & Romberg D

Všechny zápisníky mají stejnou strukturu názvu (např. AZ090102.006). Název začíná písmeny AZ, pokračuje rokem měření 09, číslem měřické skupiny 01, číslem gravimetru v rámci skupiny 02 a končí pořadovým číslem denního úseku 006. Obsah zápisníků se skládá z hlavičky obsahující celkové údaje o denním úseku měření, ze záznamů kalibrace (podle typu zápisníku) a záznamů měření. Hlavička je pro všechny zápisníky stejná:

```

8 2 1
31 'Ivancice - Cerna Hora'
25 125 1 67.0 14 7 2009 2.0 'Trakal' 1 'VWC4A74983'
  
```

Na prvním řádku číslo 8 značí použitý typ zápisníku. Ve druhém řádku číslo 31 značí pořadové číslo denního úseku během roku. Za číslem denního úseku následuje jeho název *Ivančice - Černá Hora*. Třetí řádek začíná údaji o gravimetru. Číslo 25 značí typ gravimetru, 125 je číslo gravimetru, účel čísla 1 je neznámý a 67.0 je kontrolní odečtení teploty gravimetru. Dále ve třetím řádku následuje datum 14 7 2009 a časový posun 2.0 = 1 hodina + 1 hodina letního času. Za datem je udané jméno měřiče *Trakal*, kód dopravy 1 a popis dopravy *VWC4A74983*.

Adjustovaný zápisník typu 4

Adjustovaný zápisník typu 4 (příloha A.1.1) pro gravimetr Gs12 nebo Gs11 obsahuje po údajích v hlavičce údaje o kalibraci:

```

2
07 10 45 12 -20.7 45 2 11.1 45 12 -20.4
12 34 39 91 -15.7 39 81 16.6 39 91 -15.8
  
```

Na prvním řádku je udán počet následujících řádků kalibrace 2. Každý další řádek začíná časovým údajem 07 10, za kterým následuje trojice odečtení hrubého měření 45, odečtení mikrometru 12 a odečtení galvanometru -20.7. Po údajích kalibrace následují záznamy měření:

```

34 2
2003.0000 1 50 2 9.0 100.9 6 28 ' 15 ' 40.37 .4 .6 -.8 88.0 0
2003.0000 1 50 2 9.0 100.9 6 32 ' 5 ' 35.18 -.3 -1.0 -.8 88.0 0
2005.0000 1 50 2 14.0 100.8 7 5 ' 25 ' 44.83 .8 1.3 -.7 88.0 0
...

```

Na prvním řádku je opět udán počet řádků měření 34 a navíc počet odečtení 2. Každý další řádek je řádkem měření začínajícím číslem bodu 2003.0000, na kterém se měřilo. Následují údaje o počasí: 1 meteo kód, 50 oblačnost v procentech, 2 stupeň síly větru Beaufortovy stupnice, 9.0 teplota vzduchu ve stupních Celsia a 100.9 tlak v kilopascalech. Po meteorologických údajích je zaznamenán čas 6 28. Další data udávají nastavení pomocných hmot ' 15 ' , odečtení měřítka 40.37, odečtení s_i jemných čtení .4 .6 a odečtení s_0 - .8. Na závěr je udána výška stroje nad měřícím bodem v centimetrech 88.0.

Adjustovaný zápisník typu 5

Zápisník typu 5 (příloha A.1.2) používaný pro gravimetry GAG nemá kalibraci. Obsahuje údaje o měření na jednotlivých bodech v následující podobě:

```

12 1
40.0000 0 0 0 .0 .0 9 15 150 7 21.0 .0 147 9 6.2 .0 .0 0
40.0000 0 0 0 .0 .0 9 17 210 27 24.2 .0 207 29 10.0 .0 .0 0
40.0000 0 0 0 .0 .0 9 19 270 47 23.8 .0 267 49 11.6 .0 .0 0
41.0000 0 0 0 .0 .0 12 25 270 52 54.8 .0 267 51 16.2 .0 .0 0
...

```

Na prvním řádku je zadán počet řádků měření 12. Každý další řádek začíná číslem bodu 40.0000. Údaje o počasí nebyly pro měření GAG zaznamenány nebo se je nepodařilo dochovat, proto další informace pokračují až časem měření 9 15, čtyřmi ciframi odečtení t_1 150 7 21.0 .0 a čtyřmi ciframi odečtení t_2 147 9 6.2 .0. Výška stroje nad bodem .0 nebyla taktéž zaznamenána nebo nebyla dochována.

Adjustovaný zápisník typu 6

Tento zápisník (příloha A.1.3) se používal pro data naměřená gravimetrem Sharpe a Worden. Zápisník neobsahuje kalibraci a má následující záznamy měření:

```

17 3
2014.0100 1 10 2 16.0 96.4 8 33 .00 1919.00 1918.80 1918.90 41.00 3
2015.0000 1 10 1 16.0 94.4 8 53 .00 1524.80 1524.80 1524.80 41.00 3
...

```

První řádek začíná opět počtem měření 17, po kterém následuje počet odečtení jemného čtení 3. Další řádky začínají číslem bodu, pokračují údaji o počasí a časem měření stejně jako u zápisníku typu 4. Po časovém údaji následuje hrubé čtení .00, opakované jemné čtení 1919.00 1918.80 1918.90 a výška stroje v centimetrech 41.00.

Adjustovaný zápisník typu 7

Zápisník typu 7 (příloha A.1.4) slouží k záznamu dat naměřených gravimetrem La Coste & Romberg s odečtením CPI. Tento zápisník má po údajích z hlavičky zapsané údaje o kalibraci:

```

2 4 .00000000
9 50 4428 306 -1 4428 406 -265 4428 206 +287 4428 306 +8
14 43 4428 232 +1 4428 332 -266 4428 132 +272 4428 232 -8
  
```

Na prvním řádku je udán nejprve počet řádků kalibrace 2. Počet odečtení hrubého a jemného čtení s CPI udává druhé číslo 4 a poslední číslo .00000000 je nepoužívaná konstanta ϵ pro případ vynechání kalibrace. Následující řádky začínají časovým údajem odečtení prostředního čtení 9 50 (zde čas hodin mezi 2. a 3. čtením). Po časovém údaji následuje série odečtení hrubého čtení 4428, jemného čtení 306 a odečtení CPI -1. V ukázkovém případě se provádí čtení o 100 jednotek jemného čtení větší 406, o 100 jednotek menší 206 a znovu stejné jemné čtení jako na začátku 306. K těmto čtením se odečítá příslušné CPI -265 +287 +8. Po kalibraci následují záznamy měření:

```

9 3
272.0100 1 0 2 18.0 97.0 9 48
          4428. 296.0 304.0 306.0 26.0 0
                21.00 2.00 -1.00
2011.0200 1 0 1 16.0 96.0 10 21
          4405. 921.0 921.0 920.0 26.0 0
                -4.00 -5.00 -1.00
...
  
```

Na prvním řádku v záznamech měření nalezneme opět počet měření 9, za kterým následuje počet odečtení CPI a jemného čtení 3. Po prvním řádku následují záznamy měření na jednotlivých bodech, které zabírají tři řádky. První z každé trojice začíná číslem bodu, pokračuje údaji o počasí a končí záznamem času stejně jako u zápisníku typu 4. Druhý řádek obsahuje na začátku hrubé čtení 4428. a dále sérii jemných čtení 296.0 304.0 306.0 zakončenou výškou stroje nad úrovní bodu v centimetrech 26.0. Na třetím řádku, posledním z trojice, se nalézá pouze série odečtení CPI 21.00 2.00 -1.00

Adjustovaný zápisník typu 8 a 9

Do zápisníku typu 8 a 9 (přílohy A.1.5 a A.1.6) se ukládají data naměřená gravimetrem La Coste & Romberg G a La Coste & Romberg D, které měří bez odečtení CPI a tím pádem nemají kalibraci. Záznamy měření následují hned po hlavičce a mají následující podobu:

```

13 2
3346.0200 1 10 2 19.0 97.5 9 18 4403.00 5.0 4.0 46.80 0
3346.7100 1 10 1 22.9 97.8 10 12 4421.00 331.0 332.0 46.80 0
...
  
```

Struktura uložení naměřených dat je shodná se zápisníkem typu 6 pro gravimetry Sharpe a Worden.

Adjustované zápisníky typu 7, 8 a 9 umožňují na konec souboru uložit poznámku, která začíná symbolem *.

Soubor rovnic oprav

Struktura souboru rovnic oprav (příloha A.1.7) je shodná pro všechny typy zápisníků. Název souboru (např. R3930102.004) začíná písmenem R, následuje stupeň chodu gravimetru 3, rok měření 93, číslo skupiny 01, číslo gravimetru v rámci skupiny 02 a za tečkou číslo denního úseku v daném roce 004. Soubor začíná přehledem údajů o denním úseku a neznámých použitých při výpočtu:

```

ROV. OPRAV DENNIHO USEKU: 930102.004          DATUM=27.12.1999  CAS=10:05

GRAVIMETR      =   22  961
ROVNIC OPRAV   =   16
NEZNAMYCH TIZI =    5
  1  2014.0100  VOTICE
  2  2015.0000  MILICIN
  3  2016.0000  PLANA N LUZ
  4  2017.0000  RIPEC
  5  2019.0000  CHOTYCANY
POCET NEZN. CHODU =    4
  6  a'
  7  b
  8  c
  9  d
POCET SKOKU =    1
 10
MER. FAKTOR
 11  Y1'
 12  Y2'
ROVNICE OPRAV:
...
  
```

Na prvním řádku se nachází identifikační číslo denního úseku 930102.004, datum 27.12.1999 a čas výpočtu 10:05. Na dalších řádcích je uveden typ 22 a číslo gravimetru 961, počet rovnic oprav (počet použitých měření k vyrovnání) 16, počet vyrovnávaných tíží (počet bodů) 5 a výčet bodů, na kterých se opakovaně měřilo

```

1 2014.0100 VOTICE
2 2015.0000 MILICIN
3 2016.0000 PLANA N LUZ
4 2017.0000 RIPEC
5 2019.0000 CHOTYCANY
  
```

Dále je uveden počet neznámých chodu 4 a výčet neznámých použitých pro výpočet chodu.

```

6 a'
  
```

7 b

8 c

9 d

Předposledním údajem je počet skoků 1. Posledním údajem jsou neznámé použité pro výpočet měřítkového faktoru.

11 Y1'

12 Y2'

Soubor pokračuje maticí plánu vyrovnání neznámých. Zde je ukázka zjištění místa, od kterého je zaznamenán skok (3. a 4. vyrovnávané měření) v ukázkovém souboru rovnic oprav (příloha A.1.7):

```

...
      .00000      .00000      1.00000      .00000      .00000
    -1.00000      -.95000      -.90250      -.85737      .00000
      .00000      198.47400
      1.00000      198.47400

      .00000      .00000      .00000      1.00000      .00000
    -1.00000      -1.83333      -3.36111      -6.16204      1.00000
      .00000      418.73200
      1.00000      418.73200
...

```

Matice je zapsána po řádcích, které jsou ale po pěti hodnotách zalomeny. Každý řádek matice potom tvoří $n = \lceil \frac{n_t}{5} \rceil$ řádků v textovém souboru. Pokud je v měření uveden skok, přibude do vyrovnání další neznámá. V ukázkovém souboru rovnic oprav je skok desátá neznámá. Proto v desátém sloupci v každém řádku matice, na konci druhého řádku v souboru, nalezneme od určitého místa 1.00000 místo .00000. Toto místo značí, od kterého měření už bylo počítáno se skokem v měření. V ukázkovém souboru mezi 3. a 4. vyrovnávaným měřením.

Prvních n (kde n je počet vyrovnávaných tíží) sloupců matice značí, jestli se bude počítat daná neznámá tíže. Pokud je v n -tém sloupci daného řádku matice 1.00000 (na ostatních .00000), pak se bude vyrovnávat n -tá neznámá tíže. Pokud je počet vyrovnávaných tíží v rovnicích oprav roven počtu měření vstupujících do vyrovnání, byla použita všechna měření. Pokud tomu tak není, můžeme pomocí hodnot z prvních n sloupců odhalit vynechání měření při vyrovnání a získat tak informaci o vyloučených bodech.

1.1.3 Katalogy programu Gravimzu

Program gravimzu používá dvou textových souborů pro načtení konstant gravimetrů. Jsou to Kat_ms.txt a Kat_grav.txt (přílohy A.2.1 a A.2.2). Údaje o tíhových bodech načítá Gravimzu z katalogu tíhových bodů KAT_BODY.txt (příloha A.2.3).

Katalog měřických skupin

V souboru Kat_ms.txt jsou uloženy údaje o gravimetrech používaných příslušnou měřickou skupinou daný rok. Tady je ukázka části záznamu pro měřickou skupinu číslo 1 v roce 2010.

```

MERICKA SKUPINA 10 1 ZEMEMERICKY URAD PRAHA + VUGIK ZDIBY
SEZNAM GRAVIMETRU
  TYP CÍSLO  PK  NAZEV A^UZIVATEL GRAVIMETRU
1  25   125   1  SCINTREX    ZU PRAHA
2   6   1068  1  LA COSTE G  ZU PRAHA
  
```

Každá měřická skupina je určena rokem 10 (2010) a číslem 1. Pro každý použitý gravimetr je určeno číslo v rámci skupiny 1, které je také napsané v názvu adjustovaného zápisníku, typ gravimetru 25, číslo gravimetru 125, pořadové číslo různého nastavení stejného typu a čísla gravimetru v katalogu gravimetrů 1, název a uživatel gravimetru SCINTREX ZU PRAHA.

Katalog gravimetrů

V ukázce záznamu (příloha A.2.2) pro gravimetr SCINTREX v katalogu gravimetrů jsou v části začínající V E T A S O U B O R U K T G R P uloženy základní údaje o gravimetru: typ 25, číslo gravimetru 0125, pořadové číslo v rámci stejného typu a čísla gravimetru 01, název gravimetru SCINTREX a majitel gravimetru ZU. Dále je uvedeno datum uložení 050617, datum určení 050617, datum začátku platnosti 050617, datum konce platnosti 100617, kód způsobu určení 1, kód vzoru zápisníku 2, kód výpočtu 5, dolní 0 a horní 7000 mez stupnice číslo 1, dolní 0 a horní 1100 mez stupnice číslo 2, dolní 0 a horní 0 mez stupnice číslo 3, kód termostatu 1, teplota gravimetru 67.0 a svislá vzdálenost měřícího zařízení od přístrojové desky v mm -202.

Část záznamu začínající textem V E T Y S O U B O R U K T G K S T udává konstanty vnitřních vlivů gravimetru. Gravimetry Gs11 a Gs12 obsahují v matici čísel v 1. řádku v 1. až 5. sloupci koeficienty kalibračního polynomu jemného čtení a v 6. sloupci runovou chybu. Ve 2. řádku v 1. až 5. sloupci jsou zaznamenány konstanty pro výpočet přídavných hmot. Gravimetry GAG 2 a GAG 3 mají v této části záznamu pouze jeden řádek obsahující 6 konstant.

Ostatní gravimetry mají strukturu uložených konstant vnitřních vlivů v matici o pěti nebo devíti řádcích. V 1. řádku v prvním sloupci je udána konstanta galvanomeru $\epsilon = 0$, ve 2. sloupci je potom uložena standardní teplota gravimetru $t_0 = 67$, 3. sloupec udává hodnotu měřítka šroubu $\zeta = 1000$. Ve 2. řádku jsou v 1. až 5. sloupci uvedeny konstanty kalibračního polynomu jemného čtení. Ve 3. řádku jsou uvedeny amplitudy periodických chyb, ve 4. řádku jsou uvedeny fáze a v 5. řádku periody periodických chyb. Pokud má záznam vnitřních vlivů o čtyři řádky více, obsahuje

záznam v 6. řádku 5 konstant kalibračního polynomu hrubého čtení a v 7. až 9. řádku amplitudy, fáze a periody příslušných periodických chyb.

Katalog tíhových bodů

KAT_BODY.txt (příloha A.2.3) obsahuje informace o tíhových bodech nutné k výpočtu vyrovnání v programu Gravimzu. Z tohoto katalogu byly převzaty souřadnice 48 36 40.00 14 24 26.00, výška bodů 661.930 a tíže 9807840.10 (pokud již nebyly souřadnice, výška nebo tíže získány z databáze Foxpro, kapitola 1.1.1), vertikální gradient -3245 a jeho kód 1, horizontální gradient 100 a azimut horizontálního gradientu 346. Ukázka řádku katalogu tíhových bodů:

```
2026      H DVORISTE      880603   48 36 40.00   14 24 26.00   661.930   100 346
      -3245      1 64 9807840.10  68      1.40
```

1.1.4 Ostatní zdroje

Mezi další zdroje dat vytvořené databáze patří seznam identifikačních čísel všech obcí a jejich názvů, identifikátorů okresů a jejich názvů a identifikátorů krajů a jejich názvů na území ČR. Seznamy identifikačních čísel jsou propojeny v relační tabulce. Všechna tato data z roku 2009 jsem získal ze stránek ČSÚ (Český statistický úřad) [1].

Zdrojem souřadnic tíhových bodů sítě 1995 byla data získaná z revize provedené oddělením gravimetrie ZÚ v roce 2009 a 2010. Tyto souřadnice jsou v systému ETRS89 a vznikly výpočtem z dvojího určení. Prvním určením bylo odečtení z mapy Google [3] a druhým výsledek geodetických výpočtů z podkladů katastrální mapy geoportálu ČÚZK [2] a místopisných náčrtů.

1.2 Struktura databáze

Struktura databáze se skládá ze tří schémat. První schéma je obecné **public**, které obsahuje všechny tabulky týkající se tíhových bodů. Toto schéma obsahuje další dvě schémata **useky** a **gravimetry**. Všechny informace o schématu **public** naleznete v kapitole 1.2.1. Schéma **useky** zahrnuje všechny tabulky obsahující uložená data denních úseků měření. O tomto schématu pojednává kapitola 1.2.2. Posledním schématem se zabývá kapitola 1.2.3, ve které naleznete strukturu uložení údajů o gravimetrech. Všechna schémata a tabulky byly vytvořeny pomocí dotazů v jazyce SQL. Zde je ukázka vytvoření tabulky, tabulky s popisem kódů a propojení těchto tabulek pomocí cizího klíče.

```
CREATE TABLE body_vyska (
    bod_id numeric(6,2),
```

```

    vyska numeric(6,3) DEFAULT NULL,
    vyska_kod integer DEFAULT NULL,
    PRIMARY KEY (bod_id)
);

CREATE TABLE kody_vyska(
    kod integer,
    typ varchar(50) DEFAULT NULL,
    PRIMARY KEY (kod)
);

ALTER TABLE body_vyska ADD FOREIGN KEY (vyska_kod) REFERENCES kody_vyska (kod);

```

1.2.1 Struktura uložení tíhových bodů

Základem pro strukturu uložení tíhových bodů byl systém uložení zdrojových dat v databázi FoxPro (kapitola 1.1.1). Pět základních souborů s koncovkou .dbf bylo rozděleno do následujících 10 tabulek:

Bod_zakl.dbf byl rozdělen na tabulky **body_zakladni** a **body_site**

Bod_stav.dbf byl základem k vytvoření tabulek **body_stabilizace** a **body_stav**

Bod_udaj.dbf jsem rozdělil na **body_souradnice**, **body_vyska**, **body_gradient** a **body_korekce**

Bod_tize.dbf byl předlohou pro **body_tize**

Bod_mist.dbf byl předlohou pro **body_mistopis**

Zde je popis všech tabulek struktury uložení tíhových bodů:

body_zakladni

Grafické znázornění je na obrázku v příloze B.3.

bod_id je primárním klíčem tabulky. Toto identifikační číslo bodu má čtyři místa před desetinou tečkou a dvě místa za desetinnou tečkou (např. 2014.01).

bod_nazev je název tíhového bodu. Může se skládat maximálně ze 40-ti znaků.

cislo_puvodni je původní číslo bodu. Obsahuje max. 20 znaků.

okres_id je zastaralé identifikační číslo okresu. Skládá se z 9 číslic.

obec_id je zastaralé identifikační číslo obce. Skládá se z 9 číslic.

druh_kod určuje druh bodu pomocí kódu. S tímto kódem je cizím klíčem provázána tabulka **kody_druh_bodu**.

stat_id určuje stát, ve kterém se tíhový bod nachází. Id je provázáno cizím klíčem s tabulkou **staty**.

list100_kod je kód listů mapy měřítka 1:100 000 určený v tabulce **kody_list100**. Tato tabulka je opět propojena cizím klíčem.

obec_2009_id je identifikátorem obce, ve které se daný tíhový bod nachází. Cizím klíčem je tento identifikátor propojen s tabulkou obcí, okresů a krajů **struktura_obce_2009**.

lists52 je kód určení mapového listu mapy v systému S-52. Tento kód je propojen cizím klíčem s tabulkou **kody_lists52**.

polygony

bod je identifikačním číslem tíhového bodu, pomocí kterého je cizím klíčem tabulky **polygony** propojena s tabulkou **body_zakladni**.

polygon_kod je kód příslušného polygonu, přes který je cizím klíčem připojena tabulka **kody_polygon**.

kody_polygon

kod je primární klíč tabulky. Cizím klíčem je k této tabulce připojena tabulka **polygony**.

nazev je název polygonu. Obsahuje maximálně 50 znaků.

velikost je bližší určení velikosti polygonu. Obsahuje max. 10 znaků.

kody_druh_bodu

kod je primární klíč tabulky. Cizím klíčem je k této tabulce připojena tabulka **body_zakladni**.

popis je popis polygonu (např. „Karpatský polygon“). Obsahuje max. 50 znaků.

staty

stat_id je primární klíč tabulky. Cizím klíčem je k této tabulce připojena tabulka **body_zakladni**.

stat_nazev je název státu, ve kterém se tíhový bod nachází (např. „Česká republika“). Obsahuje max. 40 znaků.

kody_list100

kod je primární klíč tabulky. Cizím klíčem je k této tabulce připojena tabulka **body_zakladni**.

cislo je číslo mapového listu mapy 1:100 000.

list_nazev je název mapového listu.

nomenklatury_list100

nomenklatura_id je primární klíč tabulky.

nomenklatura_nazev

list_kod je název mapového listu, ke kterému přísluší daná nomenklatura.

kody_lists52

kod je primární klíč tabulky. Cizím klíčem je k této tabulce připojena tabulka **body_zakladni**.

list_nazev je název mapového listu v systému S-52.

struktura_obce_2009

obec_id je primární klíč tabulky. Cizím klíčem je k této tabulce připojena tabulka **body_zakladni** a tabulka **obce_2009**.

okres_id je identifikátor okresů sloužící k propojení s tabulkou **okresy_2009**.

kraj_id je identifikátor krajů sloužící k propojení s tabulkou **kraje_2009**.

obce_2009

obec_id je primární klíč tabulky. Cizím klíčem je k této tabulce připojena tabulka **struktura_obce_2009**.

obec_nazev

okresy_2009

okres_id je primární klíč tabulky. Cizím klíčem je k této tabulce připojena tabulka **struktura_obce_2009**.

okres_nazev

kraje_2009

kraj_id je primární klíč tabulky. Cizím klíčem je k této tabulce připojena tabulka **struktura_obce_2009**.

kraj_nazev

body_souradnice

Grafické znázornění je na obrázku B.1.

bod_id je primární klíč tabulky. Formát primárního klíče *bod_id* je stejný jako v tabulce **body_zakladni**, která je k této tabulce připojena cizím klíčem.

sirka_st udává stupně zeměpisné šířky daného tíhového bodu v systému S-52.

sirka_min udává minuty zeměpisné šířky daného tíhového bodu v systému S-52.

sirka_vt udává vteřiny zeměpisné šířky daného tíhového bodu v systému S-52.

delka_st udává stupně zeměpisné délky daného tíhového bodu v systému S-52.

delka_min udává minuty zeměpisné délky daného tíhového bodu v systému S-52.

delka_vt udává vteřiny zeměpisné délky daného tíhového bodu v systému S-52.

sirka je zeměpisná šířka v systému S-52 přepočtená na desetitisíciny stupně.

delka je zeměpisná délka v systému S-52 přepočtená na desetitisíciny stupně.

souradnice_kod kód typu souřadnic v systému S-52.

sirka_etr89 je zeměpisná šířka v systému ETRS-89 získaná revizí souřadnic (kapitola 1.1.4) nebo transformací souřadnic ze systému S-52 uvedená na desetitisíciny stupně.

delka_etr89 je zeměpisná délka v systému ETRS-89 získaná revizí souřadnic nebo transformací souřadnic ze systému S-52 uvedená na desetitisíciny stupně.

nadbytecna_mereni_etr89 je počet nadbytečných měření při určování souřadnic v systému ETRS-89 získaných revizí souřadnic.

chyba_etr89_m je směrodatná odchylka určení souřadnic v systému ETRS-89, které byly získány revizí v případě počtu nadbytečných měření větším než nula.

souradnice_etr89_kod je kód určení souřadnic v systému ETRS89, který slouží k propojení s tabulkou **kody_souradnice_etr89**.

sirka_etr89_st udává stupně zeměpisné šířky daného tíhového bodu v systému ETRS-89.

sirka_etr89_min udává minuty zeměpisné šířky daného tíhového bodu v systému ETRS-89.

sirka_etr89_vt udává vteřiny zeměpisné šířky daného tíhového bodu v systému ETRS-89.

delka_etr89_st udává stupně zeměpisné délky daného tíhového bodu v systému ETRS-89.

delka_etr89_min udává minuty zeměpisné délky daného tíhového bodu v systému ETRS-89.

delka_etr89_vt udává vteřiny zeměpisné délky daného tíhového bodu v systému ETRS-89.

chyba_transformace89 je chyba z transformace souřadnic do systému ETRS-89

kody_souradnice_etr89

kod je primární klíč tabulky. Cizím klíčem je k této tabulce připojena tabulka **body_souradnice**.

popis je popis původu souřadnic v systému ETRS-89.

body_vyska

bod_id je primární klíč tabulky. Má stejný formát jako primární klíč *bod_id* v tabulce **body_zakladni**, která je k této tabulce připojena cizím klíčem.

vyska je nadmořská výška bodu v metrech.

vyska_kod je kód určení nadmořské výšky. Cizím klíčem je tento sloupec propojen s tabulkou **kody_vyska**.

kody_vyska

kod je primární klíč tabulky. Cizím klíčem je k této tabulce připojena tabulka **body_vyska**.

typ je popis určení výšky (např. „Jadran“). Obsahuje maximálně 50 znaků.

body_gradient

bod_id je primární klíč tabulky. Má stejný formát jako primární klíč *bod_id* v tabulce **body_zakladni**, která je k této tabulce připojena cizím klíčem.

vertikalni je hodnota vertikálního gradientu v jednotkách E (Eötvös).

vertikalni_kod je kód určení vertikálního gradientu. Cizím klíčem je tento sloupec propojen s tabulkou **kody_vertikalni_gradient**.

horizontalni je hodnota horizontálního gradientu v jednotkách E (Eötvös).

azimut je azimut horizontálního gradientu ve stupních.

kody_vertikalni_gradient

kod je primární klíč tabulky. Cizím klíčem je k této tabulce připojena tabulka **body_gradient**.

typ určuje typ vertikálního gradientu (např. „normální“). Obsahuje maximálně 50 znaků.

body_korekce

bod_id je primární klíč tabulky. Má stejný formát jako primární klíč *bod_id* v tabulce **body_zakladni**, která je k této tabulce připojena cizím klíčem.

topo je hodnota topografické korekce v jednotkách E (Eötvös).

stavby je hodnota vlivu stavby v jednotkách E (Eötvös).

body_tize

bod_id je primární klíč tabulky. Má stejný formát jako primární klíč *bod_id* v tabulce **body_zakladni**, která je k této tabulce připojena cizím klíčem.

tize_1957 udává hodnotu vyrovnané tíže v systému 1957 v $\mu m s^{-2}$

tize_1964 udává hodnotu vyrovnané tíže v systému 1964 v $\mu m s^{-2}$

tize_1968 udává hodnotu vyrovnané tíže v systému 1968 v $\mu m s^{-2}$

tize_1995 udává hodnotu vyrovnané tíže v systému 1995 v $\mu m s^{-2}$

chyba_1995 udává střední chybu tíže v systému 1995 v $\mu m s^{-2}$

priblizna udává hodnotu přibližné tíže v $\mu m s^{-2}$.

tize_2010 připravený sloupec pro vyrovnanou tíži v systému 2010 v μms^{-2}

chyba_2010 připravený sloupec pro chybu tíže v systému 2010 v μms^{-2}

body_stabilizace

bod_id je primární klíč tabulky. Má stejný formát jako primární klíč *bod_id* v tabulce **body_zakladni**, která je k této tabulce připojena cizím klíčem. Dále je k této tabulce přes *bod_id* navázána tabulka **navstevy**.

stabilizoval je jméno osoby, která bod stabilizovala. Obsahuje max. 30 znaků.

rok_stabilizace

rok_zruseni

typ je popis stabilizace.

navstevy

bod_id má stejný formát jako *bod_id* v tabulce **body_stabilizace**, která je k této tabulce připojena cizím klíčem.

datum je datum poslední „návštěvy“, poslední návštěvy při zjišťování stavu bodu.

body_site

bod_id je primární klíč tabulky. Má stejný formát jako primární klíč *bod_id* v tabulce **body_zakladni**, která je k této tabulce připojena cizím klíčem.

sit57.kod je kód zařazení bodu do řádu v systému 1957. Cizím klíčem je na tento kód navázána tabulka **kody_sit57**.

sit64.kod je kód zařazení bodu do řádu v systému 1964. Cizím klíčem je na tento kód navázána tabulka **kody_sit64**.

sit95.kod je kód zařazení bodu do řádu v systému 1995. Cizím klíčem je na tento kód navázána tabulka **kody_sit95**.

zakladna.kod je kód zařazení bodu do řádu na měřické základně.

sit2010.kod je připravený kód zařazení bodu do řádu v systému 2010. Cizím klíčem je na tento kód navázána tabulka **kody_zakladna**.

body_stav

bod_id je primární klíč tabulky. Má stejný formát jako primární klíč *bod_id* v tabulce **body_zakladni**, která je k této tabulce připojena cizím klíčem.

stav_celkovy_kod je kód určující celkový stav tíhového bodu. Cizím klíčem je přes tento kód připojena tabulka **kody_stav_celkovy**.

nalezeni_kod je kód hodnotící vyhledatelnost bodu. Cizím klíčem je přes tento kód připojena tabulka **kody_stav**.

prijezd_kod je kód hodnotící místopisný popis příjezdu k tíhovému bodu. Cizím klíčem je přes tento kód připojena tabulka **kody_stav_oz**.

mistopis_kod je kód hodnotící místopisný popis tíhového bodu. Cizím klíčem je přes tento kód připojena tabulka **kody_stav**.

způsob_prijezdu_kod je kód hodnotící možnost příjezdu k tíhovému bodu. Cizím klíčem je přes tento kód připojena tabulka **kody_stav**.

umisteni_kod je kód hodnotící umístění tíhového bodu. Cizím klíčem je přes tento kód připojena tabulka **kody_stav**.

stabilizace_kod je kód hodnotící stabilizaci tíhového bodu. Cizím klíčem je přes tento kód připojena tabulka **kody_stav**.

ochrana_kod je kód hodnotící ochranné prostředky tíhového bodu. Cizím klíčem je přes tento kód připojena tabulka **kody_stav_oz**.

stabilizace_nova určuje jestli má bod novou stabilizaci.

tabulka_nova určuje jestli má bod novou ochranou tabulku.

tyc_nova určuje jestli má bod novou ochranou tyč.

poznamka

body_mistopis

bod_id je primární klíč tabulky. Má stejný formát jako primární klíč *bod_id* v tabulce **body_zakladni**, která je k této tabulce připojena cizím klíčem.

nacrt je cesta k uloženému souboru s místopisným náčrtem.

foto je cesta k uloženému souboru s fotografií bodu.

pripojeni je popis výškového připojení tíhového bodu.

prijezd je popis příjezdu k tíhovému bodu.

popis je popis umístění tíhového bodu.

poznámka

1.2.2 Struktura uložení denních úseků měření

Struktura uložení denních úseků je tvořena všemi tabulkami schématu **useky**. Grafické znázornění se nachází v příloze B.2. Zdrojem dat a zároveň podkladem pro návrh struktury uložení denních úseků byly soubory zápisníků a rovnic oprav (kapitola 1.1.2). Tabulka **denni_useky** obsahuje hlavičkové údaje z adjustovaných zápisníků resp. všechny údaje, jejichž hodnoty se v daném úseku měření zaznamenávají pouze jednou. Jeden záznam tabulky **mereni** obsahuje data jednoho řádku měření adjustovaného zápisníku, která se pro dané měření zaznamenávají pouze jednou. Neobsahuje jemné čtení. Jeden záznam tabulky **kalibrace** obsahuje jeden záznam kalibračního čtení. Záznam tabulky **cteni** obsahuje jedno odečtení jemného čtení a pokud je, tak i CPI. Tabulka **kody_doprava** a **kody_meteo** jsou pomocnými popisnými tabulkami pro kód dopravy a kód stavu počasí.

denni_useky

usek_id je identifikační číslo a primární klíč tabulky (např. 20100101.001). První čtyři cifry 2010 udávají rok měření, další dvě 01 číslo měřické skupiny a poslední dvě 01 před desetinnou tečkou číslo gravimetru v rámci skupiny.¹ Za desetinnou tečkou následují tři číslice pořadového čísla denního úseku v rámci roku 001. Pomocí cizího klíče jsou tímto číslem připojeny tabulky **kalibrace** a **mereni**.

cislo je pořadové číslo denního úseku v rámci roku. To samé, co za desetinnou tečkou hodnoty ve sloupci *usek_id*.

nazev je název denního úseku. Obsahuje maximálně 50 znaků.

gravimetr_typ udává typ gravimetru např. 25

gravimetr_cislo udává číslo gravimetru např. 125

gravimetr_teploata je kontrolní odečet teploty gravimetru ve stupních Celsia.

datum udává datum měření

¹Číslo skupiny a číslo gravimetru v rámci skupiny jsou pozůstatkem starého systému ukládání naměřených dat. Lze je u nových denních úseků změnit na cokoliv užitečnějšího.

posun_casu je celkový časový posun od Greenwichského času a posun letního času, pokud je, v hodinách.

observator je jméno měřiče. Obsahuje maximálně 20 znaků.

doprava_kod je kód dopravního prostředku. Cizím klíčem je tímto kódem připojena tabulka **kody_doprava**.

doprava_popis je popis dopravního prostředku. Obsahuje maximálně 20 znaků.

stufen_chodu udává stupeň chodu použitý při zpracování programem Gravimzu, pokud se dochovaly rovnice oprav.

cas_skoku je udán v případě pokud skok nastal jako datový typ *Time without timezone*.

vynechane_body udává počet bodů, které byly při zpracování programem Gravimzu vynechány, pokud se dochovaly rovnice oprav.

poznamka

gravimetr_id jednotné identifikační číslo gravimetru v tabulce **gravimetry**, která je cizím klíčem s touto tabulkou propojena (kapitola 1.2.3)

epsilon je konstanta pro výpočet čtení při odečtení CPI při vynechání kalibrace.

chod0 je absolutní člen polynomu chodu gravimetru

chod0_skok je absolutní člen polynomu chodu gravimetru od okamžiku skoku

chod1 je lineární člen polynomu chodu gravimetru

chod2 je kvadratický člen polynomu chodu gravimetru

chod3 je kubický člen polynomu chodu gravimetru

cas_skoku_h udává tu samou hodnotu jako *cas_skoku* v desetitisícinách hodiny.

mereni

mereni_id je identifikační číslo měření a primární klíč tabulky. Cizím klíčem je pomocí tohoto čísla připojena tabulka **cteni**. Identifikační číslo je přirozené číslo. Začíná číslem 1, narůstá po jedné až do čísla posledního záznamu měření.

usek_id je identifikační číslo denního úseku. Cizím klíčem je přes toto číslo připojena tabulka **denni_useky**

bod_id je identifikační číslo bodu. Cizím klíčem je přes toto číslo připojena tabulka **body_zakladni**

meteo_kod je kód uvedených meteorologických údajů. Pokud je roven nule, jsou všechny údaje rovné nule. Cizím klíčem je na tento kód připojena tabulka **kody_meteo**.

oblacnost je procentuální zatažení oblohy oblačností v desítkách procent.

vittr_kod je kód Beaufortovy stupnice větru.

teplota je okolní teplota na bodě ve stupních Celsia.

tlak je tlak na bodě v kilopascalech (převážně).

cas je čas měření v hodinách (např. 14.3833).

cteni_hrube je odečtení šroubu hrubého čtení.

vyska_stroje je výška stroje nad bodem.

vaha je váha měření. Standartně nastavena na 1. V případě vynechání bodu je váha 0.

poznamka

cteni_jednotne je jednotné vypočtené čtení pro všechny druhy gravimetrůn.

gs12_phm je odečtení pomocných hmot při měření gravimetrem Gs12 nebo Gs11.

gs12_meritko je odečtení měřítka při měření gravimetrem Gs12 nebo Gs11.

gs12_s0 je odečtení S_0 při měření gravimetrem Gs12 nebo Gs11.

gag_t1_cifra1-4 je odečtení při měření gravimetrem GAG.

gag_t2_cifra1-4 je odečtení při měření gravimetrem GAG.

kalibrace

usek_id je identifikační číslo úseku. Pomocí tohoto čísla je cizím klíčem připojena tabulka **denni_useky**.

cas je čas jednoho odečtení kalibrace. (Více odečtení tj. více záznamů se stejným časem znamená jeden řádek kalibrace v zápisníku.)

cteni_hrube je odečtení šroubu hrubého čtení.

cteni_jemne je odečtení šroubu jemného čtení.

cpi je odečtení CPI.

gs12_mer je odečtení měřítka při měření gravimetrem Gs12 nebo Gs11.

gs12_mikr je odečtení mikrometru při měření gravimetrem Gs12 nebo Gs11.

gs12_galv je odečtení galvanometru při měření gravimetrem Gs12 nebo Gs11.

cas_hodiny je hodnota *cas* v hodinách.

cteni

mereni_id je identifikační číslo měření. Cizím klíčem je tímto číslem připojena tabulka **mereni**.

cteni_jemne je odečtení šroubu jemného čtení.

cpi je odečtení CPI.

gs12_si je odečtení S_i při měření gravimetrem Gs12 nebo Gs11.

kody_doprava

kod je primárním klíčem tabulky a kódem dopravního prostředku. Cizím klíčem je přes tento kód připojena tabulka **denni_useky**.

nazev je název dopravního prostředku.

kody_meteo

kod je primárním klíčem tabulky a kódem údajů o počasí. Cizím klíčem je přes tento kód připojena tabulka **mereni**.

nazev je popis stavu počasí (např. „Krupobití“).

1.2.3 Struktura uložení údajů o gravimetrech

Veškeré údaje o gravimetrech pochází z katalogu gravimetrů (kapitola 1.1.3) a jsou uloženy v jediné tabulce **gravimetry** ve schématu **gravimetry** (příloha B.3). Jeden záznam v tabulce **gravimetry** se dá chápat jako jedno nastavení konstant gravimetru.

gravimetry

gravimetr_id je jedenáctimístné identifikační číslo jednoho záznamu a primární klíč tabulky (např. 12500125001). První cifra je vždy 1.² 2. a 3. číslice jsou vyhrazené pro typ gravimetru 25, 4.-8. číslice zaznamenávají číslo gravimetru 00125 a poslední tři cifry ukazují pořadové číslo záznamu při stejném typu a čísle gravimetru 001.

typ je typ gravimetru např. 25.

cislo je číslo gravimetru např. 125

tabulka_cislo je původně pořadové číslo v katalogu gravimetrů v rámci stejného typu a čísla gravimetru.

nazev je název gravimetru např. SCINTREX.

uzivatel je vlastník nebo dříve uživatel gravimetru (např. ZU PRAHA).

datum_ulozeni je datum uložení záznamu. Původně datum uložení konstant v katalogu gravimetrů.

datum_urceni je datum určení konstant záznamu.

zacatek_platnosti je datum, od kterého jsou určené konstanty záznamu platné.

konec_platnosti je datum, do kterého jsou určené konstanty záznamu platné.

způsob_urceni je kód určení konstant.

vzor_zapisniku je kód vzoru zápisníku z programu Gravimzu.

kod_vypoctu je kód výpočtu konstant.

stupnice1_dolni je dolní mez stupnice 1.

stupnice1_horni je horní mez stupnice 1.

stupnice2_dolni je dolní mez stupnice 2.

stupnice2_horni je horní mez stupnice 2.

stupnice3_dolni je dolní mez stupnice 3.

stupnice3_horni je horní mez stupnice 3.

²Pozůstatek problému odstraňování počátečních nul při kopírování.

termostat je kód existence termostatu. 1 značí přítomnost, 0 nebo NULL nepřítomnost.

teplota je teplota gravimetru ve stupních Celsia.

svisla_vzdalenost je svislá vzdálenost měřicího zařízení od přístrojové desky v mm.

epsilon je konstanta galvanometru ϵ .

t0 je standartní teplota gravimetru t_0 .

zeta je hodnota měřítka šroubu ζ .

gag_konstanta1-6 jsou konstanty gravimetru GAG.

gs12_runova_chyba

gs12_hmota1-5 jsou konstanty pro výpočet přídavných hmot.

vaha je váha gravimetru.

a-e_jemne jsou konstanty kalibračního polynomu jemného čtení.

amplituda1-6_jemne jsou amplitudy periodických chyb jemného čtení.

faze1-6_jemne jsou fáze periodických chyb jemného čtení.

perioda1-6_jemne jsou periody periodických chyb jemného čtení.

a-e_hrube jsou konstanty kalibračního polynomu hrubého čtení.

amplituda1-6_hrube jsou amplitudy periodických chyb hrubého čtení.

faze1-6_hrube jsou fáze periodických chyb hrubého čtení.

perioda1-6_hrube jsou periody periodických chyb hrubého čtení.

amp_sin1-6_jemne jsou amplitudy sinových vln periodických chyb jemného čtení ve tvaru součtu sin. a kosin. vln.

amp_cos1-6_jemne jsou amplitudy cosinových vln periodických chyb jemného čtení ve tvaru součtu sin. a kosin. vln.

freq_cos1-6_jemne jsou frekvence sinových a cosinových vln periodických chyb jemného čtení ve tvaru součtu sin. a kosin. vln.

amp_sin1-6_hrube jsou amplitudy sinových vln periodických chyb hrubého čtení ve tvaru součtu sin. a kosin. vln.

amp_cos1-6_hrube jsou amplitudy cosinových vln periodických chyb hrubého čtení ve tvaru součtu sin. a kosin. vln.

freq_cos1-6_hrube jsou frekvence sinových a cosinových vln periodických chyb hrubého čtení ve tvaru součtu sin. a kosin. vln.

1.3 Naplnění databáze

Naplnění databáze probíhalo ve třech fázích. V první fázi (kapitola 1.3.1) byla převedena data z databáze FoxPro (kapitola 1.1.1) do nové struktury tíhových bodů (kapitola 1.2.1). Tato struktura byla dále doplněna o aktuální údaje o obcích okresech a krajích. Poté byl vyvíjen program RegisterImport (kapitola 1.3.2) v jazyce Java, který pomocí skriptu v Bash interpretu naplnil strukturu denních úseků a samostatně spuštěný i strukturu gravimetrů.

1.3.1 Hromadné kopírování CSV souborů

Nejprve byla data z databáze FoxPro (kapitola 1.1.1) exportována do textových souborů formátu CSV. Potom byly tyto soubory převedeny do kódování UTF-8 pomocí následujícího skriptu, který zároveň ze souboru odstraní prázdné řetězce.

```
#!/bin/bash
for file in *.TXT; do
  if [ -f "$file" ]; then
    iconv -f WINDOWS-1250 -t UTF-8//IGNORE $file > temp1
    vyskyt=$(grep ',' temp1 | wc -l)
    while [ $vyskyt != 0 ]; do
      sed 's/,",/,/' < temp1 > temp2
      mv temp2 temp1
      vyskyt=$(grep ',' temp1 | wc -l)
    done
    sed '/^$/d' < temp1 > $file
  fi
done
```

Při manipulaci s daty v CSV souborech byly použity různé příkazy v interpretu Bash. Zde jsou jejich ukázky.

```
# odstrani posledni radek
sed '$d'
# vezme 1 az 8 sloupec oddeleny carkami
cut -d ',' -f 1-8 soubor
# obecne najit a nahradit
sed 's/najit/nahradit' < soubor
#zmena kodovani
iconv -f encoding -t encoding -o vystupnisoubor vstupnisoubor
#inverzni grep
grep -v
#vytahne z bod_stav.txt cislo bodu a poznamku, která je okomentovana " ale zaroven
v ni jsou carky, které slouží jako oddelovace
```

```
cut -d ',' -f 1,5- BOD.STAV.TXT| sort | grep ''' | cut -d ''' -f 1-2 | sed 's/"/"/'
| less
```

Samotné kopírování do struktury databáze probíhalo pomocí dotazů jazyka SQL. Používal jsem přímé kopírování do již vytvořených tabulek nebo nakopírování do pomocné tabulky, ze které pak bylo možné data kontrolovaně přidávat i do částečně zaplněných tabulek. Během kopírování bylo použito mnoho různých SQL dotazů. Tady je ukázka přímého kopírování, kontrolovaného kopírování a připojení nového systému obcí, okresů a krajů pomocí názvů.

—nakopirovani dat do relace body_vyska

```
COPY body_vyska (bod_id, vyska, vyska_kod)
```

```
FROM '/home/shares/allusers/smazat/body_vyska.csv'
```

```
WITH CSV;
```

—nakopirovani tize z pomocne relace z katalogu tihovych bodu do priblizne tize relace body_tize

```
UPDATE body_tize U
```

```
SET priblizna = tize_konecna
```

```
FROM (
```

```
  SELECT KAT.tize AS tize_konecna, T.bod_id AS id
```

```
  FROM body_tize T
```

```
  LEFT JOIN pom_kat.body KAT
```

```
  ON T.bod_id = KAT.cislo_bod
```

```
  LEFT JOIN body_zakladni ZAKL
```

```
  ON T.bod_id = ZAKL.bod_id
```

```
  WHERE (tize_1995, tize_1968, tize_1964, tize_1957) IS NULL
```

```
) V
```

```
WHERE V.id = U.bod_id;
```

—pripojeni noveho cislovani obci

```
UPDATE body_zakladni AS zaklad
```

```
SET obec2009_id = nove_id
```

```
FROM
```

```
(SELECT Z.bod_id AS cislo_bodu, F.fid, F.nid AS nove_id
```

```
FROM body_zakladni Z
```

```
LEFT JOIN
```

```
(SELECT DISTINCT S.id AS fid, N.obec_id AS nid, S.obec, N.obec_nazev, S.okres,
  N.okres_nazev
```

```
FROM
```

```
(SELECT OB.obec_id AS id, OB.obec_nazev AS obec, OK.okres_nazev AS okres
```

```
FROM obce OB
```

```
JOIN okresy OK ON SUBSTRING(CAST(OK.okres_id AS varchar(10)) FROM '^.{4}')
```

```
= SUBSTRING(CAST(OB.obec_id AS varchar(10)) FROM '^.{4}')
```

```
) AS S
```

```
LEFT JOIN
```

```
(SELECT obec_id, obec_nazev, okres_id, okres_nazev, kraj_id, kraj_nazev
```

```
FROM pom_struktura_obce
```

```
) AS N
```

```
ON S.obec = UPPER(N.obec_nazev) AND S.okres = UPPER(N.okres_nazev)
```

```
) AS F
```

```
ON Z.obec_id = F.fid
```

```
WHERE stat_id = 1 AND Z.obec_id IS NOT NULL
```

```
) AS konecne
```

```
WHERE zaklad.bod_id = konecne.cislo_bodu;
```

1.3.2 Program RegisterImport

Program RegisterImport jsem začal vyvíjet v jazyce Java společně s Kristýnou Kitzbergerovou a Lindou Křikavovou v rámci předmětu Projekt informatika 2 v letním semestru v roce 2010. Výsledkem projektu byl program pro příkazovou řádku systému GNU/Linux, kterému se zadala cesta k textovému souboru adjustovaného zápisníku a program z tohoto souboru načel a uložil všechna data do databáze. Tento program jsem rozšířil o možnost načíst rovnice oprav a zjistit tak další informace o zpracování zápisníku v programu Gravimzu. Dalším rozšířením bylo načtení katalogu měřických skupin pro dodání čísla tabulky, specifického pořadového čísla nastavení gravimetru v katalogu gravimetrů při stejném typu a čísle gravimetru. Toto specifické číslo bylo potřeba jako součást identifikátoru gravimetru, kterým se měřil daný denní úsek. Posledním rozšířením programu bylo načtení katalogu gravimetrů a uložení údajů z katalogu do databáze, do struktury pro gravimetry. Program jsem vytvářel a rozšiřoval v JDK - Java Development Kit verze 1.6.0_14.

Použití programu

Program RegisterImport.jar je zabalený zkompileovaný spustitelný soubor. Pro jeho spuštění je potřeba nainstalovat Java SE Runtime Environment³. Program se spouští z příkazového řádku následujícím příkazem

```
java -jar RegisterImport.jar
```

Za tímto příkazem následují parametry:

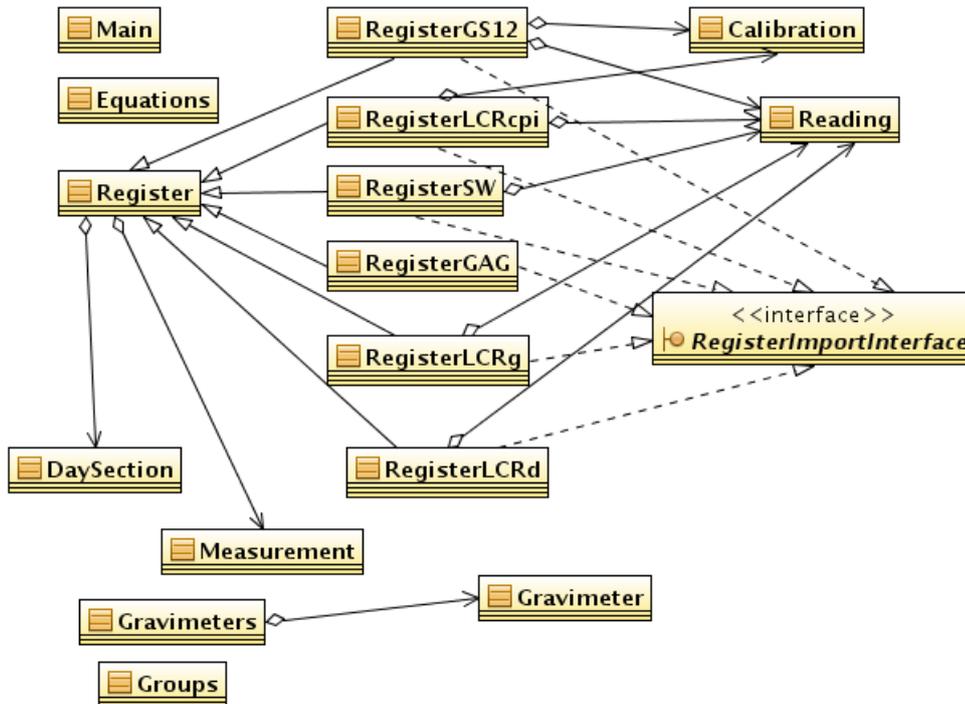
- r souborZapisniku kde souborZapisniku značí cestu k souboru s adjustovaným zápisníkem. Pokud není zadaná, program nezačne nic načítat.
- e souborRovnicOprav kde souborRovnicOprav značí cestu k souboru s rovnicemi oprav. Pokud není zadaná a je zadaná cesta k adjustovanému zápisníku, nebudou se zjišťovat data o zpracování programem Gravimzu. Adjustovaný zápisník se uloží do databáze bez údajů o stupni chodu, vynechaných bodech či času skoku.
- gp katalogMerickychSkupin kde katalogMerickychSkupin značí cestu k souboru s katalogem měřických skupin. Pokud není cesta zadaná a je zadaná cesta k souboru adjustovaného zápisníku, načtou se data adjustovaného zápisníku bez nastaveného identifikačního čísla gravimetru.
- gm katalogGravimetru kde katalogGravimetru značí cestu k souboru s katalogem gravimetrů. Tento parametr je nezávislý na předchozích parametrech. Slouží pro jednorázové načtení katalogu gravimetrů do databáze.

³Pro jednorázové nakopírování do databáze byla použita verze build 1.6.0_14-b08

Pro hromadné nakopírování byl použit skript v interpretu Bash (příloha C.1).

Třídy programu RegisterImport

V jazyce Java bylo vytvořeno 16 tříd a jedno rozhraní, které mají níže zobrazenou strukturu.



Obrázek 1.1: Struktura programu Registerimport

Třída main

Třída Main obsahuje následující metody:

`void main (String[] args)` nejdříve spustí program, potom zkontroluje argumenty zadané na příkazové řádce a podle jejich popisu přiřadí příslušné cesty ke zpracovávaným souborům. Dále zkontroluje, jestli jsou cesty k souborům platné a jestli jsou soubory, na které ukazují, určené ke čtení.

Pokud je zadána cesta k souboru adjustovaných zápisníků, vytvoří se z tohoto souboru nový objekt `Register` a nastaví se jeho buffer, ze kterého se určí typ zápisníku pomocí metody `getRegisterType()`. Dále se podle příslušného typu vytvoří daný objekt třídy `Registernazev`, kde `nazev` je název gravimetru, pro který je daný typ zápisníku určen. Metoda `extractSectionID` získá z cesty k souboru zápisníku identifikační číslo denního úseku `sectionID`, které vloží do atributu `sectionID` objektu `DaySection` daného objektu zápisníku třídy

Registernazev. Objekt třídy *Registernazev* zavolá metody `extractData()`. Pokud existují cesty k souborům s rovnicemi oprav a katalogem měřických skupin, vytvoří se nové objekty `Equations` a `Groups`. Metody těchto objektů `setAdditionalVariables(measurementList, daySection)` nastaví příslušné atributy (stupeň chodu, vynechané body, čas skoku, váhy měření a id gravimetru) objektů `DaySection` a `Measurement`, které jsou atributy objektu *Registernazev*. Nakonec se provedou metody `printExtractedData()`, `printSpecificData()`, které vytisknou ukládaná data na standartní výstup. Metody `saveData(connection)`, `saveSpecificData(connection)` uloží data do databáze.

Pokud je zadána cesta k souboru katalogu gravimetrů, vytvoří se nový objekt třídy `Gravimeters` a zavolají se metody `extractData()`, `saveData(connection)`, které načtou data z katalogu gravimetrů a uloží je do databáze.

`Connection openConnection ()` vytváří připojení do databáze.

`void closeConnection ()` ukončuje připojení do databáze.

`BigDecimal extractSectionID (String sectionID)` vytváří typ `BigDecimal` ze zadané cesty k souboru s adjustovaným zápisníkem.

`int getLastMeasurementID ()` zjišťuje poslední identifikační číslo záznamu měření v tabulce **mereni**.

`boolean testEquationSourcePath (String registerSourcePath, String equationSourcePath)` kontroluje, zda zadaný název rovnic oprav odpovídá názvu adjustovaného zápisníku.

`boolean validFile (String sourcePath)` pokusí se vytvořit soubor ze zadané cesty a vyzkouší, jestli je daný soubor čitelný.

Třída `DaySection`

Objekt třídy `DaySection` reprezentuje hlavičku textového souboru adjustovaného zápisníku. Zároveň jsou jeho atributy shodné s atributy jednoho záznamu v tabulce **denni_useky**. Vedle metod nastavujících a získávajících proměnné obsahuje jedinou důležitou metodu `void sendToDatabase(Connection connection)`, která vloží jedním dotazem v jazyce SQL všechny atributy do jednoho záznamu tabulky **denni_useky**.

Třída Measurement

Objekt třídy `Measurement` představuje jeden záznam tabulky **mereni**, jeden řádek adjustovaného zápisníku obsahující záznam měření. Neobsahuje ale údaje o jemném čtení ani o CPI. Vedle metod nastavujících a získavajících proměnné obsahuje jedinou důležitou metodu `void sendToDatabase(Connection connection)`, která vloží jedním dotazem v jazyce SQL všechny atributy do jednoho záznamu tabulky **mereni**.

Třída Calibration

Objekt třídy `Calibration` představuje jeden záznam tabulky **kalibrace**, jedno odečtení kalibračního čtení. Obsahuje také metodu `void sendToDatabase(Connection connection)` pro uložení záznamu do databáze.

Třída Reading

Objekt třídy `Reading` představuje jeden záznam tabulky **cteni**, jedno zapsané jemné čtení. Obsahuje metodu `void sendToDatabase(Connection connection)` pro uložení záznamu do databáze.

Třída Equations

Objekt třídy `Equations` reprezentuje soubor rovnic oprav. Obsahuje jediný atribut `BufferedReader buffer`, do kterého se načítá textový soubor rovnic oprav a následující seznam konstruktorů a metod:

`Equations(String sourcePath)` je konstruktor, který vytvoří nový objekt třídy `Equations` ze zadané cesty k souboru s rovnicemi oprav.

`void setAdditionalVariables(ArrayList<Measurement> measurementList, DaySection daySection)` načte nezbytná data z bufferu rovnic oprav a vytvoří seznam identifikačních čísel bodů a časů měření, který potom porovnává se zadanými čísly a časy měření v seznamu `measurementList`. V objektu `daySection` nastaví počet vynechaných bodů a pokud nastal skok, nastaví čas skoku. V objektech seznam měření `measurementList` nastaví u jednotlivých měření správnou váhu. Pokud byl bod vynechán 0, pokud ne 1.

`int findPointIndex(BigDecimal number, int from, int skip, ArrayList<BigDecimal> list)` vyhledává ve vloženém seznamu identifikačních čísel bodů (ze zápisníku) `list` `skip + první výskyt zadaného identifikačního čísla` bodu `number` od zadané pozice `from`. Při neúspěšném hledání vrátí -1, jinak pozici daného výskytu.

`int findReference (int init_skip, ArrayList<BigDecimal> equationPoint-
 IdList, ArrayList<GregorianCalendar> equationTimesList, ArrayLi-
 st<BigDecimal> registerPointIdList, ArrayList<GregorianCalendar>
 registerTimes List, ArrayList<Integer> indexReference)` vezme první
 identifikační číslo bodu ze seznamu získaného z rovnic oprav `equationPoint-
 IdList` a pokusí se k němu vyhledat od začátku `init_skip`-tý výskyt v se-
 znamu získaného ze souboru adjustovaného zápisníku `registerPointIdList`
 metodou `findPointIndex`. Pokud ho nenalezne je nutné manuální zpracování.
 Pokud ano, přidá jeho index do seznamu referencí `indexReference` a načte
 do lokální proměnné čas ze seznamu `registerTimesList` o dané pozici. Po-
 tom pro všechny následující identifikační čísla bodů v seznamu z rovnic oprav
`equationPointIDList` provede metoda znovu vyhledávání `findPointIndex` v se-
 znamu `registerPointIdList` od pozice následující poslední nalezenou. Pokud
 nějaký bod nenalezne, zahlásí metoda chybu. Pokud nalezne, spočítá se rozdíl
 času od času nastaveného na začátku do lokální proměnné a času na nalezené
 pozici v seznamu `registerTimesList`. Tento rozdíl času porovná s časem
 v `equationTimesList` a pokud se rovná, přidá příslušný index nalezeného
 čísla bodu do seznamu referencí `indexReference`. Pokud se rozdíl času ne-
 rovná času ze seznamu `equationTimesList` a je tento rozdíl menší, provede
 se hledání ještě jednou a hledá se další výskyt čísla bodu.

Třída Register

Objekt třídy `Register` popisuje data textových souborů adjustovaných zápisníků,
 která jsou pro všechny typy stejná. Jednotlivé typy zápisníků jsou popsány dále ve
 třídách `RegisterGS12`, `RegisterGAG`, `RegisterSW`, `RegisterLCRcpi`, `RegisterLCRg`
 a `RegisterLCRd`, které jsou rozšířením třídy `Register`. Nejdůležitější atributy třídy
`Register` jsou:

`BufferedReader buffer` je buffer, do kterého se načte textový soubor zápisníku
 a dále se z něj po řádcích načítají data.

`DaySection daySection` je nový objekt třídy `DaySection`, reprezentující hlavičku
 zápisníku.

`ArrayList<Measurement>` je seznam měření, seznam objektů `Measurement`.

`int registerType` je typ načteného zápisníku.

`int lastMeasurementID` je poslední identifikační číslo záznamu v databázi v ta-
 bulce **mereni**.

Konstruktory a metody objektu třídy `Register` jsou v tomto seznamu:

`Register(String sourcePath)` je konstruktor, který vytvoří nový objekt třídy `Register` ze zadané cesty k souboru s adjustovaným zápisníkem.

`Register(BufferedReader buffer, int registerType, int lastMeasurementID)` je konstruktor, který vytvoří nový objekt třídy `Register` ze zadaného existujícího bufferu, typu zápisníku a identifikačního čísla posledního záznamu v tabulce **mereni**.

`void extractRegisterType()` načte první řádek z bufferu a zjistí z něho typ zápisníku.

`void printExtractedData()` pošle na standartní výstup všechny hodnoty atributů objektu `daySection`.

`void closeBuffer()` zavře buffer.

`void saveData(Connection connection)` pokud je platné připojení do databáze tak zavolá metodu `sendToDatabase(connection)` objektu `daySection`, která se pokusí uložit své atributy do databáze.

RegisterGS12, RegisterGAG, RegisterSW, RegisterLCRcpi, RegisterLCRg a RegisterLCRd

Všechny tyto třídy rozšiřují třídu `Register`. Objekty těchto tříd představují načtené adjustované zápisníky příslušného typu podle gravimetru a obsahují následující atributy:

`ArrayList<Calibration> calibrationList` je seznam objektů `Calibration`, seznam všech záznamů kalibrace. Je definován pouze u objektů `RegisterGS12` a `RegisterLCRcpi`.

`ArrayList<Reading> readingList` je seznam objektů `Reading`, seznam všech záznamů jemného čtení.

Konstruktory a metody tříd jsou následující:

`Registernazev(Register register)` je konstruktor, pomocí kterého nový objekt převezme data z objektu `Register`. *nazev* je název gravimetru, pro který se daný zápisník nejčastěji používal.

`void extractData()` postupně načítá řádky bufferu souboru adjustovaného zápisníku a získává z něj údaje o měření, která zapisuje do příslušných atributů objektů `DaySection`, `Measurement`, `Reading` a u některých zápisníků i `Calibration`.

`void printSpecificData()` vytiskne na standartní výstup získaná data o měření příslušející danému typu zápisníku.

`void saveSpecificData(Connection connection)` uloží do databáze seznam je-mného čtení `readingList` a u objektů `RegisterGS12` a `RegisterLCRcpi` se-znam odečtení kalibrace `calibrationList`.

Gravimeter

Objekt třídy `Gravimeter` představuje část souboru katalogu gravimetrů popisující nastavení gravimetru. Obsahuje asi 80 atributů a následující nejdůležitější metody:

`void sendToDatabase(Connection connection)` odešle všechna data jedním pří-kazem v jazyce SQL do databáze.

`void printData()` odešle všechny hodnoty uložených atributů na standartní výstup.

Gravimeters

Objekt třídy `Gravimeters` představuje všechna data uložená v katalogu gravimetrů a obsahuje následující atributy.

`BufferedReader buffer` je buffer, do kterého se načte textový soubor katalogu gravimetrů.

`ArrayList<Gravimeter> gravimeterList` je seznam objektů `Gravimeter`

Zde je konstruktor a metody třídy `Gravimeters`:

`Gravimeters(String sourcePath)` je konstruktor vytvářející objekt s bufferem vytvořeným ze souboru s katalogem gravimetrů, ke kterému je zadaná cesta.

`int setProperYear int year` je statická metoda nastavující celý název roku.

`void extractData()` načítá postupně řádky bufferu a získává z něj data o gravi-metrech, která ukládá do příslušných atributů objektů `Gravimetr` v seznamu `gravimeterList`.

`void printExtractedData()` zavolá metodu `printData` všech objektů `Gravimeter` uložených v seznamu `gravimeterList`.

`void saveData(Connection connection)` uloží postupně všechny objekty ze se-znamu `gravimeterList` do databáze pomocí jejich metod `sendToDatabase (connection)`, odešle všechna načtená data z katalogu gravimetrů do databáze do tabulky `gravimetry`.

Groups

Objekt třídy `Groups` představuje načtený záznam katalogu měřických skupin. Jediným atributem objektu je `BufferedReader` `buffer`. Objekt této třídy obsahuje následující nejdůležitější konstruktor a metodu:

`Groups(String sourcePath)` je konstruktor vytvářející objekt třídy `Groups` s buferem vytvořeným ze souboru katalogu měřických skupin, ke kterému je zadaná cesta.

`setAdditionalVariables(DaySection daySection)` získá z objektu `daySection` z atributu `sectionID` rok měření, číslo skupiny a pořadové číslo gravimetru v rámci skupiny. Pomocí těchto dat dohledá v `bufferu` příslušné pořadové číslo gravimetru (při stejném typu a čísle gravimetru) v katalogu gravimetrů a uloží do atributu `gravimeterID` objektu `daySection` identifikační číslo gravimetru skládající se z čísla 1, typu gravimetru, čísla gravimetru a získaného pořadového čísla gravimetru z katalogu gravimetrů.⁴

⁴Kdo se přihlásí, že se pročetl až sem, toho zvu na pivo. Autor

2 Správa databáze

Pro správu uložených bodů a denních úseků měření bylo naprogramováno PHP webové rozhraní. Jak toto rozhraní použít a jaké soubory PHP používá, o tom se dočtete v kapitole 2.1. O možnostech upravovat databázi pomocí jazyka SQL a příkazu `psql` pojednává kapitola 2.2.

2.1 PHP webové rozhraní

2.1.1 Obsluha rozhraní

Webové rozhraní je umístěno na adrese `<http://database.nesvadba.eu/modify/>`. Před samotným zobrazením je nutné zadat přihlašovací jméno a heslo. Odhlášení z celého rozhraní proběhne buď po dvaceti minutách nečinnosti nebo po zavření celého prohlížeče. Po přihlášení se zobrazí na obrazovce nadpis, vlevo nabídka dalších odkazů a vpravo vyhledávání bodů a denních úseků (příloha D.1). Celé rozhraní funguje tak, že se nejdříve vyhledá bod nebo denní úsek a potom se k tomuto bodu nebo úseku pomocí odkazů v nabídce v levé části obrazovky zobrazí požadované informace v pravé části obrazovky.

Tíhové body se dají vyhledat podle kompletního identifikačního čísla nebo podle jeho celé části, např. *2016.70* nebo *2016*. Pokud je zadána celá část, jsou k ní doplněny nuly. Dále se dají tíhové body vyhledat pomocí části nebo celého názvu bodu. Například *rum* nalezne body s názvem Brumov a Čes Krumlov (Obrázek 2.1). Denní úseky se dají vyhledávat pomocí roku měření, celé části nebo kompletního identifikačního čísla denního úseku měření. Například *2010* nalezne všechny denní úseky změřené v tomto roce, *20100102* nalezne všechny denní úseky změřené v roce 2010, skupinou 1 a gravimetrem číslo 2. Kompletní zadání *20100102.001* nalezne denní úsek změřený v roce 2010, skupinou 1, gravimetrem číslo 2, který má pořadové číslo 1.

Vyhledávání bodů

Číslo: Název:

Podle názvu nebo jeho části byly nalezeny tyto body:

Číslo bodu	Název bodu
3637.01	Brumov
3637.00	Brumov
2027.01	Čes Krumlov
2027.00	Čes Krumlov

Obrázek 2.1: Vyhledávání podle části názvu

Nabídka odkazů je rozdělena do dvou částí. První část **Body** obsahuje následující odkazy:

Základní údaje zobrazí identifikační číslo bodu, pole s názvem bodu, pole s původním číslem bodu, roletové menu s určením druhu bodu, roletové menu výběru státu, roletové menu s výběrem mapového listu mapy 1:100 000 se zobrazením nomenklatury, roletové menu mapového listu S-52, roletové menu s více možnostmi výběru určující zařazení bodu do polygonů, roletové menu s určením názvu obce, okresu a kraje, roletová menu se zařazením tíhového bodu do sítě 1957, 1964 a 1995 a roletové menu zařazení bodu k příslušné základně (příloha D.2).

Stav zobrazí pole s názvem osoby, která bod stabilizovala. Dále se zobrazí pole s rokem stabilizace, pole s rokem zrušení, pole s datem poslední návštěvy, textové pole s popisem stabilizace bodu, roletová menu s určením celkového stavu bodu, s klasifikací vyhledatelnosti bodu, popisu příjezdu k bodu, místopisu bodu, příjezdu k bodu, umístění bodu, kvality stabilizace a roletové menu s klasifikací ochranného zařízení. Dále jsou zde zaškrťovací políčka pro novou stabilizaci, tabulku a ochranou tyč a nakonec pole s poznámkou (přílohy D.3 a D.4)

Údaje zobrazí pole pro zadávání stupňů, minut a vteřin souřadnic v systému ETRS-89, pole pro zadání střední chyby určených souřadnic, počet nadbytečných měření a chyby transformace souřadnic. Dále se zobrazuje roletové menu s popisem souřadnic, souřadnice v systému S-52, pole pro zadání výšky bodu, roletové menu určující druh výšky, pole s vertikálním gradientem, roletové menu s typem vertikálního gradientu, pole s horizontálním gradientem, pole s azimutem horizontálního gradientu, pole s topografickou korekcí a pole s vlivem stavby.

Tíže zobrazí pole s vypočtenou tíží v systémech 1957, 1964, 1968 a 1995. U tíže systému 1995 zobrazí její střední chybu (Obrázek 2.2).

Obrázek 2.2: Tíže

Místopis zobrazí vygenerovaný místopis bodu s fotografií, náčrtem a umístěním bodu v mapách Google (příloha D.5).

Vložit nový nabídne zadání nového čísla bodu. Do všech tabulek **body_*** je vložen nový záznam (Obrázek 2.3).

Obrázek 2.3: Vložit nový bod

Druhá část nabídky odkazů s nadpisem **Denní úseky** obsahuje následující odkazy:

Zobrazit údaje pokud je vyhledán denní úsek, zobrazí všechna data zapsaná v adjustovaném zápisníku daného typu. Zobrazí údaje z hlavičky: identifikační číslo denního úseku, pole s názvem denního úseku, roletová menu s nastavením identifikačního čísla, typu a čísla gravimetru, pole s teplotou gravimetru, datem měření, časovým posunem a jménem měřiče, roletové menu s kódem a názvem dopravního prostředku a pole s popisem dopravního prostředku. Následují údaje o kalibraci, pokud je daný typ zápisníku obsahuje, a dále údaje o měření (příloha D.7).

Vložit nový zobrazí nastavitelné údaje před vytvořením vlastního zápisníku: Identifikační číslo denního úseku, roletové menu s výběrem identifikačního čísla

gravimetru, zaškrtačací políčko odečtení CPI, počet řádků kalibrace a počet odečtení kalibrace, počet řádků měření a počet odečtení měření. Po vyplnění a kliknutí na tlačítko **Vytvořit zázpisník** se vygeneruje daný zázpisník pro vyplnění.

2.1.2 Soubory PHP skriptů

Celé rozhraní se skládá z následujících souborů:

modify/index.php se automaticky spustí pokud se zadá do internetového prohlížeče adresa `<http://database.nesvadba.eu/modify/>`. Nejdříve otestuje, jestli je někdo přihlášen. Na začátku nikdo přihlášen není a tak přesměruje zobrazení na skript `auth.php`, který zprostředkuje přihlášení a přesměruje zobrazení zpět na `index.php`. Po přihlášení rozdělí skript zobrazení do třech částí. V první části se odkazuje na soubor `header.php`, v druhé části se volá funkce `getMain()` a ve třetí části se odkazuje na soubor `menu.php`. Funkce `getMain()`, která je jako většina funkcí uložena v souboru `functions.php`, načte informace o názvu zobrazovaného souboru z proměnné `$_REQUEST['show']` získané metodou GET z URL adresy a pokusí se nejdříve zobrazit html soubor a následně php soubor s tímto názvem. Například adresa `.../modify/index.php?show=pointCondition` odešle do proměnné název zobrazovaného souboru `pointCondition`. Funkce `getMain()` pak bude nejdříve hledat soubor `pointCondition.html` a následně `pointCondition.php`, který opravdu najde a zobrazí ve třetí části.

modify/header.php obsahuje pouze to, co se má zobrazit v části nadpisu.

modify/menu.php obsahuje dva seznamy odkazů. První seznam obsahuje odkazy, které metodou GET odkazují na všechny PHP soubory zobrazující informace o tíhovém bodě (`pointBase.php`, `pointCondition.php`, `pointData.php`, `pointGravity.php` a `pointPlaceDescription.php`) a odkaz na soubor pro zadání nového záznamu bodu `pointInsert.php`. Druhý seznam obsahuje odkaz na PHP soubor `sectionShow.php` zobrazující data uložená ve vyhledaném denním úseku a odkaz na PHP soubor `sectionInsert.php` pro vytvoření nového zázpisníku.

modify/auth.php zobrazí formulář pro zadání přihlašovacích údajů a po jejich odeslání zavolá funkci `getUserID($username, $password)`, která odešle uživatelské jméno a otisk md5 do databáze. Pokud se z databáze vrátí záznam, nastaví se id záznamu do proměnné `$_SESSION['id']`. Existence této proměnné

pak dále zaručuje přístup na všechny další stránky. Pokud přihlášení proběhlo úspěšně, odkáže se tento skript na `index.php`

modify/connections.php obsahuje konstanty nutné k připojení do databáze.

modify/functions.php obsahuje většinu funkcí použitých v různých PHP souborech. Vedle funkce `getMain()` jsou zde následující funkce:

`getDateCzech($date)` převádí datum z iso formátu do českého.

`getDateISO($date)` převádí datum z českého formátu do iso formátu.

`checkDateCzech($date)` kontroluje datum v českém formátu.

`getDegMinSec($coordinate, $prec)` přepočítává desetinné stupně na celé stupně, minuty a vteřiny.

`getDegrees($deg, $min, $sec)` převede stupně, minuty a vteřiny zpět na desetinné stupně.

`getPointName($number, $connection)` získá název bodu z databáze k zadanému identifikačnímu číslu bodu.

`getFirstPointID($connection)` získá první identifikační číslo tíhového bodu ze seřazeného seznamu z databáze.

`getLastPointID($connection)` získá poslední identifikační číslo tíhového bodu ze seřazeného seznamu z databáze.

`getNextPointID($pointNumber, $connection)` získá následující identifikační číslo bodu z databáze.

`getPreviousPointID($pointNumber, $connection)` získá předchozí identifikační číslo bodu z databáze.

`getFirstSectionID($connection)` získá první identifikační číslo denního úseku ze seřazeného seznamu z databáze.

`getLastSectionID($connection)` získá poslední identifikační číslo denního úseku ze seřazeného seznamu z databáze.

`getNextSectionID($sectionNumber, $connection)` získá následující identifikační číslo denního úseku z databáze.

`getPreviousSectionID($sectionNumber, $connection)` získá předchozí identifikační číslo denního úseku z databáze.

`getPointNumber()` nastaví id bodu z URL adresy do proměnné `$_SESSION['pointID']`.

`getSectionNumber()` nastaví id denního úseku z URL adresy do proměnné `$_SESSION['sectionID']`.

`addZeros($number)` přidá za celé čtyřciferné číslo `.00`.

`createRollMenu($items, $name, $is_multiple, $variable)` vytvoří roletové menu pro zadané pole položek `$items`, zadaný název menu `$name`, zadaný parametr `$is_multiple` určující menu s více možnostmi a parametr `$variable` definující přednastavnou položku menu.

`getSelection($key, $names, $table, $connection)` vrací pole pro roletové menu ze zadaného názvu sloupce s klíči `$key`, názvů dalších sloupců tabulky `$names`, názvu tabulky `$table` a připojení k databázi `$connection`.

`getSelectionQuery($key, $name, $query, $connection)` vrací také pole pro roletové menu. Místo názvu tabulky je třeba zadat celý dotaz do databáze `$query`.

`getRows($query, $connection)` odešle databázový dotaz `$query` a vrátí řetězec, ve kterém jsou jednotlivé řádky oddělené oddělovačem `|`.

`getInputRow ($rowID, $cssName, $arrayInput)` vytváří jeden řádek html tabulky s jednotlivými vstupními poli v jejích buňkách, kde `$rowID` je název pro daný řádek (část názvu formuláře pro identifikaci při ukládání), `$cssName` je název třídy kaskádového stylu pro tento řádek a `$arrayInput` je vícerozměrné pole, které obsahuje pro každou buňku, každý formulář parametry: `type, name, value, size, maxlength`.

`getReadingCalCount($sectionID, $connection)` zjistí maximální počet čtení pro kalibraci pro denní úsek.

`getReadingMesCount($name, $sectionID, $connection)` získá maximální počet čtení zadaného názvu sloupce `$name` při měření v daném úseku `$sectionID`.

`getRoughMesCount($sectionID, $connection)` zjistí jestli zadaný denní úsek měl měření hrubé čtení.

`getMaxLength($name, $table, $condition, $connection)` získá maximální počet znaků dané veličiny z dané tabulky, kde `$name` je název sloupce, `$table` je název tabulky nebo více spojených tabulek, `$condition` je podmínka jako pokračování SQL dotazu od `WHERE`.

modify/notfound.php zobrazí informace, které se mají zobrazit, když není možné nálezt požadovanou stránku. Tento soubor používá funkce `getMain()`.

modify/pointBase.php zobrazí základní údaje o tíhovém bodu. Na začátku načítá soubor `selectPoint.php`, dále se nachází kontroly dat, která byla již odeslána

a je nutné je před uložením zkontrolovat. Kontroly dat využívají skript `validity.php`. Pokud už byla data z formuláře odeslána a zkontrolována, jsou uložena do databáze pomocí příkazu v SQL. Dále jsou v souboru příkazy SQL pro získání příslušných údajů o tíhovém bodě a na závěr samotný formulář s textovými poli a roletovými nabídkami.

modify/pointCondition.php zobrazí údaje o stavu tíhového bodu. Má podobnou strukturu jako soubor `pointBase.php`.

modify/pointData.php zobrazí další údaje jako souřadnice, výšku a gradient tíhového bodu. Má podobnou strukturu jako soubor `pointBase.php`.

modify/pointGravity.php zobrazí vypočtené tíže na daném bodě v různých systémech. Má podobnou strukturu jako soubor `pointBase.php`

pointPlaceDescription.php vytváří místopis bodu na samostatné stránce. Místopis obsahuje kromě popisných informací také fotografii, náčrt bodu a zobrazení bodu na mapách Google. Nejdříve se jedním dotazem do databáze získají všechny potřebné údaje, které se dále zobrazí v jednotlivých částech (div). Umístění částí, jejich velikost a ohraničení je definováno v souboru `stylePlace.css`. Pro zobrazení tíhového bodu na mapě se nejdříve vytvoří nový objekt třídy `GoogleMapApi` (ze souboru `GoogleMap.php`), nastaví se pomocí členských metod standartní zoom, typ mapy, velikost ovládacích tlačítek, šířka a délka mapového výřezu a na závěr se přidá značka se souřadnicemi.

```
$map = new GoogleMapAPI('map');
$map->_minify_js = isset($_REQUEST['min']) ? FALSE : TRUE;
$map->setZoomLevel('14');
$map->setMapType('roadmap');
$map->setControlSize('small');
$map->setWidth('397');
$map->setHeight('398');
$map->addMarkerByCoords($lon89, $lat89, $title);
```

Potom už stačí v úseku html kódu v hlavičce zadat metody pro tisk skriptu pro hlavičku

```
$map->printHeaderJS();
$map->printMapJS();
```

a nakonec v těle html kódu zadat metodu pro zobrazení mapy.

```
$map->printOnLoad();
$map->printMap();
```

modify/pointInsert.php zobrazí formulář pro zadání nového identifikačního čísla tíhového bodu. Pokud už bylo číslo zadáno zkontroluje se jeho správnost, jestli

už není v databázi a nakonec se s tímto identifikačním číslem bodu vloží do všech tabulek **body_*** prázdný záznam.

modify/sectionInsert.php zobrazí formulář pro vygenerování nového zápisníku a provede kontrolu zadaných dat. Dál zobrazí vygenerovaný zápisník, zkontroluje vyplněná naměřená data a odešle je do databáze.

modify/sectionShow.php zobrazí vygenerovaný zápisník podle vyhledaného identifikačního čísla denního úseku. Po odeslání formuláře zápisníku data zkontroluje a uloží do databáze. Má podobnou strukturu jako **pointBase.php**. Data ve formulářích jsou odesílána ve vícerozměrných polích.

modify/selectPoint.php zobrazí formulář pro vyhledávání tíhového bodu a načte soubor **shiftPoint.php**. Po odeslání vyhledávaného identifikačního čísla tíhového bodu nebo názvu tíhového bodu nebo jeho části, odeslaná data zkontroluje a pokud jsou správně, provede vyhledávací dotaz do databáze.

modify/selectSection.php zobrazí formulář pro vyhledávání denního úseku a načte soubor **shiftSection.php**. Po odeslání vyhledávaného identifikačního čísla nebo jeho části nebo názvu denního úseku nebo jeho části, odeslaná data zkontroluje a pokud jsou správně, provede vyhledávací dotaz do databáze.

modify/shiftPoint.php zobrazí tlačítka pro pohyb o jeden tíhový bod vpřed nebo vzad.

modify/shiftSection.php zobrazí tlačítka pro pohyb o jeden denní úsek vpřed nebo vzad.

modify/validity.php obsahuje funkci pro testování správnosti zadávaných údajů do formulářů pomocí regulárních výrazů.

modify/style.css definuje třídy kaskádových stylů. Používá ho soubor **index.php**

stylePlace.css definuje třídy kaskádových stylů pro vygenerovaný místopis souborem **pointPlaceDescription**.

googleMap/GoogleMap.php obsahuje třídu **GoogleMapAPI**, která umožňuje snadné vytvoření jednoduchého zobrazení souřadnic na mapách Google.

googleMap/JSTMin.php obsahuje třídu **JSTMin**, která umožňuje minimalizaci kódu v JavaScriptu.

2.2 SQL úpravy s psql

Pro správu uložených dat denních úseků nebo tíhových bodů lze použít vytvořené webové rozhraní. Pro správu uložených dat gravimetrů žádné rozhraní vytvořené nebylo, a proto je potřeba použít jiného rozhraní. Nejuniverzálnější, nejrychlejší ale také asi nejsložitější je správa pomocí dotazů jazyka SQL. Není možné abych zde obsáhl všechny použité SQL dotazy při tvorbě a správě databáze, ale popíši zde alespoň základní příkazy při správě uložených dat gravimetrů.

psql

Jedním z mnoha nástrojů, které odesílají SQL dotazy do databáze je interaktivní terminál `psql` fungující přímo na serveru, kde je databáze v provozu. Tento terminál se dá spustit jen pro sérii příkazů načtených ze souboru nebo v interaktivním módu. Zde je ukázka použití pro sérii příkazů uložených v souboru `gravimetry_vyber.sql`:

```
psql -d testovacldb -U honza -f gravimetry_vyber.sql
```

Parametr `-d testovacldb` zadává název databáze, ke které se má `psql` připojit. Parametr `-U honza` zadává jméno uživatele databáze¹. Parametr `-f gravimetry_vyber.sql` zadává název vstupního souboru s příkazy. Pokud se příkaz spustí bez vstupního souboru s SQL dotazy, objeví se příkazový řádek interaktivního terminálu:

```
psql -d testovacldb
Welcome to psql 8.3.12, the PostgreSQL interactive terminal.
```

```
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
```

Zde jsou některé užitečné příkazy `psql` pro zobrazení seznamu databází, schémat, tabulek a názvů sloupců tabulek.

`\l` vypíše seznam všech databází

`\dn` vypíše seznam všech schémat.

`\d` vypíše seznam všech tabulek, ke kterým je nastavena prohledávací cesta. Nastavení prohledávací cesty se dá změnit SQL příkazem `SET search_path TO gravimetry, useky, public` na všechna zatím vytvořená schémata.

`\d useky.denni_useky` vypíše názvy sloupců v tabulce `useky.denni_useky`

`\?` zobrazí nápovědu k příkazům v `psql`

¹Nemusí být nutně zadáváno pokud je databázový uživatel stejně pojmenován jako uživatel serveru.

`\h select` zobrazí nápovědu k dotazu SELECT v sql

Nejvíce informací o použití psql jsou k nalezení na manuálových stránkách. Stačí na serveru zadat `man psql` v příkazovém interpretu Bash.

Dotazy SELECT

Následující SQL dotazy SELECT pouze vybírají a zobrazují požadovaná data z databáze. Všechny dotazy jsou oddělené středníkem.

```
SELECT * FROM gravimetry.gravimetry
WHERE gravimetr_id = 12500125001;
```

```
SELECT typ, cislo, nazev FROM gravimetry.gravimetry
WHERE gravimetr_id = 12500125001;
```

```
SELECT gravimetr_id, nazev, uzivatel, konec_platnosti
FROM gravimetry.gravimetry
WHERE uzivatel LIKE '%ZU%';
--WHERE konec_platnosti > date '2009-01-01';
--WHERE typ IN (2,22) AND uzivatel LIKE '%PRAHA%';
```

První dotaz zobrazí všechny atributy záznamu tabulky `gravimetry` ve schématu `gravimetry`, u kterého je primární klíč `gravimetr_id` roven 12500125001. Druhý dotaz zobrazí `typ`, `číslo` a `název` gravimetru s identifikačním číslem 12500125001. Třetí dotaz zobrazí identifikační číslo gravimetru, jeho `název`, `jméno vlastníka` a `datum konce platnosti` záznamu pro všechny záznamy, které obsahují v názvu vlastníka ZU. Pokud by se před řádkem začínajícím WHERE přidala dvojice znaků “–”, bude tento řádek ignorován. Dvojice znaků “–” značí komentář v SQL. Pokud by se potom postupně zanechaly bez komentáře druhý a třetí řádek začínající WHERE, provedl by dotaz výběr záznamů následujícím způsobem.

```
WHERE konec_platnosti > date '2009-01-01';
```

Zobrazí všechny záznamy, které ještě platí po 1.1.2009.

```
WHERE typ IN (2,22) AND uzivatel LIKE '%PRAHA%';
```

Zobrazí všechny záznamy, které mají `typ` gravimetru rovný 2 nebo 22 a `název` vlastníka u nich obsahuje slovo PRAHA. Výsledek druhého a třetího dotazu je následující:

```
typ | cislo | nazev
----+-----+-----
 25 |  125 | SCINTREX
(1 row)
```

```
gravimetr_id | nazev      | uzivatel | konec_platnosti
-----+-----+-----+-----
12400253004 | SODIN      | ZU PRAHA | 1999-12-31
12500125001 | SCINTREX   | ZU PRAHA | 2010-06-17
10600176001 | LA COSTE G | ZU PRAHA | 2001-07-12
10600176003 | LA COSTE G | ZU PRAHA | 2001-07-12
```

```

10600176002 | LA COSTE G | ZU PRAHA | 2001-07-12
10601068001 | LA COSTE G | ZU PRAHA | 2001-07-12
(6 rows)

```

Příkazy INSERT

Všechny příkazy INSERT vkládají jeden nebo více řádků do tabulky databáze. Zde jsou ukázky dvou příkazů.

```

INSERT INTO gravimetry.gravimetry (gravimetr_id)
VALUES (12500125002);

```

```

INSERT INTO gravimetry.gravimetry (gravimetr_id, typ, cislo, tabulka_cislo,
    nazev,
    uzivatel)
SELECT 12500125003, typ, cislo, 3, nazev, uzivatel
FROM gravimetry.gravimetry
WHERE gravimetr_id = 12500125001;

```

První příkaz vloží do tabulky `gravimetry` jeden záznam s identifikačním číslem 12500125002. Druhý příkaz vloží do té samé tabulky záznam, který má id gravimetru rovné 12500125003 a atribut `tabulka_cislo` roven 3. Atributy `typ`, `cislo`, `nazev`, `uzivatel` se zkopírují ze záznamu, které má id gravimetru rovno 12500125001. Všechny ostatní atributy jsou nastaveny na hodnoty DEFAULT. Zde je výstup z databáze:

```

INSERT 0 1
INSERT 0 1

```

Příkazy UPDATE

Všechny příkazy UPDATE aktualizují údaje již vložených záznamů. Zde je ukázka dvou příkazů.

```

UPDATE ONLY gravimetry.gravimetry
SET datum_ulozeni = '2010-12-09'
WHERE gravimetr_id = 12500125002;

```

```

UPDATE ONLY gravimetry.gravimetry
SET
    zpusob_urceni = NOVE.zpusob_urceni,
    vzor_zapisniku = NOVE.vzor_zapisniku,
    kod_vypoctu = NOVE.kod_vypoctu
FROM
    (
        SELECT zpusob_urceni, vzor_zapisniku, kod_vypoctu
        FROM gravimetry.gravimetry
        WHERE gravimetr_id = 12500125001
    ) NOVE
WHERE gravimetr_id = 12500125002;

```

První příkaz nastaví atribut `datum_ulozeni` záznamu s identifikačním číslem gravimetru 12500125002 na 9.12.2010, druhý příkaz nastaví atributy `zpusob_urceni`,

vzor_zapisniku a kod_vypoctu na hodnoty stejné jako má záznam s id gravimetru 12500125001. Zde je výstup z databáze:

```
UPDATE 1
UPDATE 1
```

Příkaz DELETE

Příkaz DELETE vymaže jeden nebo více řádků z tabulky databáze.

```
DELETE FROM ONLY gravimetry.gravimetry
WHERE gravimetr_id IN (12500125002, 12500125003);
```

Tento příkaz vymaže záznamy, jejichž identifikační čísla jsou rovna 12500125002 nebo 12500125003. Zde je výstup z databáze:

```
DELETE 2
```

Příkaz ALTER

Příkazy ALTER upravují strukturu tabulek, strukturu uložených dat. Zde je ukázka použití při práci s novým sloupcem tabulky.

```
ALTER TABLE gravimetry.gravimetry ADD COLUMN novy_sloupec real DEFAULT NULL;
ALTER TABLE gravimetry.gravimetry ALTER COLUMN novy_sloupec SET DEFAULT 1;
ALTER TABLE ONLY gravimetry.gravimetry ALTER COLUMN novy_sloupec TYPE integer;
ALTER TABLE ONLY gravimetry.gravimetry DROP COLUMN novy_sloupec;
```

První příkaz přidá do tabulky gravimetry ve schématu gravimetry jeden sloupec s názvem novy_sloupec, datovým typem real a nastavenou standardní hodnotou NULL. Druhý příkaz změní hodnotu DEFAULT na 1, třetí příkaz změní datový typ sloupce na integer a čtvrtý příkaz tento sloupec odstraní. Zde je výstup z databáze:

```
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
```

Složitější příkazy, JOIN

Zde je ukázka dvou složitějších příkazů propojujících více tabulek.

```
SELECT U.datum, U.observator, G.nazev, G.gravimetr_id, M.cteni_hrube,
       round(CAST(SUM(C.cteni_jemne)/COUNT(C.cteni_jemne) AS numeric), 2) AS prumer
FROM useky.mereni M
JOIN useky.denni_useky U
ON U.usek_id = M.usek_id
JOIN gravimetry.gravimetry G
ON G.gravimetr_id = U.gravimetr_id
JOIN useky.cteni C
ON C.mereni_id = M.mereni_id
JOIN body_zakladni B
ON B.bod_id = M.bod_id
```

```

WHERE B.bod_id = 3402.02
GROUP BY U.datum, U.observator, G.nazev, G.gravimetr_id, M.cteni_hrube
ORDER BY U.datum;

SELECT U.datum, U.usek_id, G.gravimetr_id, G.nazev
FROM gravimetry.gravimetry G
JOIN useky.denni-useky U
ON G.gravimetr_id = U.gravimetr_id
WHERE typ IN (1,11)
ORDER BY U.datum DESC
LIMIT 10;

```

První příkaz vyhledá všechna měření na bodě s identifikačním číslem 3402.02. Tato měření seskupí podle data měření, jména měřiče, názvu gravimetru, id gravimetru a hrubého čtení. Všechny tyto atributy také zobrazí společně s vypočteným průměrným jemným čtením. Druhý příkaz vyhledá posledních 10 denních úseků naměřených gravimetry typu 1 nebo 11, gravimetry Gs11 nebo Gs12.

datum	observator	nazev	gravimetr_id	cteni_hrube	prumer
1980-05-06	DIVIS	WORDEN	12200961013	0	1646.29
1980-05-06	TRAKAL	SHARPE	10200174023	763	528.41
1990-07-11	DIVIS	WORDEN	12200961014	0	1830.18
1990-07-11	TRAKAL	SHARPE	10200174024	788	739.94
2001-07-18	LEDERER	LA COSTE G	10601068001	4538	734.56
2001-07-18	TRAKAL	LA COSTE G	10600176001	4471	631.44

(6 rows)

datum	usek_id	gravimetr_id	nazev
1988-07-21	19880704.008	10100194002	GS 12
1988-07-21	19880703.008	10100181011	GS 12
1988-07-20	19880704.007	10100194002	GS 12
1988-07-20	19880703.007	10100181011	GS 12
1988-07-19	19880704.006	10100194002	GS 12
1988-07-19	19880703.006	10100181011	GS 12
1988-07-18	19880703.005	10100181011	GS 12
1988-07-18	19880704.005	10100194002	GS 12
1988-07-15	19880703.004	10100181011	GS 12
1988-07-15	19880704.004	10100194002	GS 12

(10 rows)

3 Výstupy z databáze

Tato kapitola popisuje různé využití databáze. Nejdůležitějším úkolem databáze je poskytovat data o měření novému programu pro zpracování tíhových měření Gravimzung. O výstupech ve formátu XML pro program Gravimzung pojednává kapitola 3.1. Přípravu dat pro grafické znázornění v programu Surfer popisuje kapitola 3.2. Kapitola 3.3 pojednává o SQL dotazech, které vytváří souhrnné statistiky naměřených dat. V poslední kapitole 3.4 se zabývám zjišťováním informací z databáze, pomocí kterých se určují místa budoucích měření.

3.1 XML výstup pro Gravimzung

Současně s obnovením systému ukládání naměřených dat vyvíjí Otakar Nesvadba nový software pro vyrovnání tíhových měření Gravimzung. Protože software načítá všechna data z XML formátu, bylo nutné vytvořit systém databázových dotazů SQL a funkcí PL/pgSQL, který by data z databáze převedl do příslušné struktury v XML. Protože se Gravimzung stále vyvíjí, mění se i systém dotazů a funkcí. Databázové dotazy používají již předdefinované následující PostgreSQL funkce:

xmlelement je funkce pro vytváření XML tagů s různými atributy. V následující ukázce jsou **nazev** a **nazev_atributu** řetězce znaků. **obsah** a **hodnota** mohou být názvy sloupců tabulky, za které se dosadí příslušné atributy záznamů. Volání funkce **xmlelement** lze řetězit v obsahu a vytvářet tak vnořené XML tagy.

```
SELECT xmlelement(name nazev , xmlattributes('hodnota' as nazev\_atributu), '
      obsah');
```

Vytvoří XML tag, který má následující podobu:

```
<nazev nazev_atributu='hodnota'>obsah</nazev>
```

xmlconcat pouze spojí dvě XML proměnné k sobě.

```
SELECT xmlconcat('<ema/>', '<mele>maso</mele>');
```

Vytvoří XML tag, který má následující podobu:

```
<ema/><mele>maso</mele>
```

xmlagg je funkce, která dokáže seskupit do jedné XML proměnné (jednoho XML tagu) atributy více záznamů.

```
SELECT XMLAGG(XMLELEMENT(name 'mereni' ,mereni_id ))
FROM useky.mereni
WHERE usek_id = 20090102.002
GROUP BY usek_id;
```

Vytvoří XML tag, který má následující podobu:

```
<mereni>24542</mereni><mereni>24543</mereni><mereni>24544</mereni><mereni>24545</mereni><mereni>24546</mereni><mereni>24547</mereni><mereni>24548</mereni><mereni>24549</mereni><mereni>24550</mereni><mereni>24551</mereni><mereni>24552</mereni>
```

Zde je popis souborů obsahujících dotazy a funkce aktuální k datu 10. 12. 2010.

query_point.sql vytváří XML tagy `station` pro všechny body, na kterých se daný denní úsek měřil (příloha E.1.1).

query_section.sql vytváří XML tag `du` s vnořeným tagem `instrument`, který obsahuje údaje o použitém gravimetru. Do tagu `du` se doplňují tagy `obs` s údaji o kalibraci a tagy `obs` s údaji o měření na tíhových bodech (příloha E.1.2).

query_calibration.sql vytváří XML tagy `obs` s údaji o kalibračním měření pro daný denní úsek (příloha E.1.3).

query_observation.sql vytváří XML tagy `obs` s údaji o měření na tíhovém bodě (příloha E.1.4).

3.2 Výstupy pro program Surfer

Pro grafické znázornění bodů a linií potřebuje program Surfer dva odlišné textové soubory. Textový soubor, podle kterého se zobrazují body, obsahuje pro každý bod číslo bodu, zeměpisnou šířku, zeměpisnou délku a kód bodu. Pro vypsání jednoho řádku tohoto souboru jsem vytvořil funkci `get_points(point_id)` v jazyce PL/pgSQL, která zadanému identifikačnímu číslu bodu vedle souřadnic přiřadí i kód `sit95.kod` z tabulky `body_site`. Pokud záznamy souřadnic obsahují hodnotu NULL, objeví se místo nich text `not found`. Pokud hodnotu NULL obsahuje kód sítě, zobrazí se kód 99 (příloha E.2.1). Jednoduchý SQL dotaz volající tuto funkci pro všechny body, na kterých se měřilo v rámci čtyř zadaných denních úseků, vypadá následovně:

```
SELECT get_points(M.bod_id)
FROM useky.denni-useky U
JOIN useky.mereni M
ON M.usek_id = U.usek_id
WHERE U.usek_id IN (
    20080102.022,
    20080102.023,
    20080102.024,
    20080102.027
)
GROUP BY M.bod_id
ORDER BY M.bod_id;
```

Výsledek dotazu vypadá takto:

2027.01	14.316444	48.812814	4
2029.00	14.28357	48.865113	4
2031.00	14.267799	48.877443	4
2032.00	14.285008	48.874653	4
2033.00	14.295999	48.876382	4
2035.00	14.309223	48.845718	4
2036.00	14.29097	48.851057	4
2037.00	14.274747	48.863892	4
3346.02	15.993524	49.842899	4
3427.01	16.102858	49.763745	4
3428.01	16.265504	49.711642	4
3429.00	16.4707	49.7524	99
3429.01	16.459921	49.757426	3

Textový soubor zadávaný Surferu pro zobrazení linií udává celkový počet bodů v linii následovaný seznamem souřadnic těchto bodů. Pro vypsání jedné linie jsem vytvořil funkci `get_polygons(usek_id)` v PL/pgSQL. Tato funkce pro zadané id denního úseku nejdříve spočítá počet bodů, které mají v denním úseku rozdílné souřadnice,¹ a potom vytvoří seznam souřadnic těchto bodů (příloha E.2.2). Použití funkce `get_polygons(usek_id)` ukazuje následující SQL dotaz:

```

SELECT get_polygons(U.usek_id)
FROM useky.denni-useky U
WHERE U.usek_id IN (
    20070102.034,
    20070102.035,
    20070102.036
);

```

Výsledek dotazu vypadá takto:

4		
17.647609	49.025848	
17.7904	49.0338	
17.9639	49.0695	
18.05573	49.186167	
3		
14.148765	49.309801	
14.468176	49.295237	
14.700721	49.366238	
4		
15.593266	49.394931	
15.587219	49.281491	
15.7899	49.0418	
15.808131	49.051206	

Ze souborů obsahující body a linie ve správném formátu doplněné o linie státní hranice a tíhové body v zahraničí vykreslí program Surfer obrázek (příloha E.1).

¹Počet různých identifikačních čísel bodů nelze použít, protože některé blízké excentrické body mají zadané stejné souřadnice.

3.3 Tvorba statistik

Uložená data v databázi nabízí mnoho možností ke statistickému zpracování. Jednou z nich je vytváření statistik pro denní úseky měření, které byly použity k výpočtu tíhového systému v roce 1995. Hromadné vyrovnání a s ním spojený výběr souborů adjustovaných zápisníků resp. rovnic oprav se dříve prováděl nakopírováním textových souborů do příslušné adresářové struktury. Pro získání seznamu identifikačních čísel denních úseků, který se dále používá v SQL dotazech k vymezení výběru, byl napsán krátký Bash skript `get_index.sh` (příloha E.3.1). Skriptu mohou být zadány tři různé argumenty.

-r adresar Po argumentu `-r` následuje cesta ke kořenovému adresáři ve struktuře uložení souborů rovnic oprav.

-a adresar Po argumentu `-a` následuje cesta ke kořenovému adresáři ve struktuře uložení souborů adjustovaných zápisníků.

-h zobrazí nápovědu.

Zde je ukázka spuštění skriptu:

```
./get_index.sh -r cesta_k_adresary/Rovopr/Sit2/
```

Výstupem je seznam identifikačních čísel denních úseků oddělených čárkami:

```
19720101.001,
19720101.002,
19720101.003,
19720101.004,
19720101.005,
...
20100102.059,
```

Tento seznam poslouží k omezení výběru denních úseků v následujících dotazech. Nejdříve bylo třeba zjistit počet denních úseků, které nebyly měřeny gravimetrem Gs12 nebo Gs11 a které byly změřeny do roku 1994 včetně.

```
SELECT COUNT(*)
FROM useky.denni_useky U
JOIN gravimetry.gravimetry G
ON G.gravimetr_id = U.gravimetr_id
WHERE G.typ NOT IN (1,11) AND EXTRACT (YEAR FROM(U.datum)) < 1995 AND U.usek_id IN
(
  19720101.001,
  ...
  20100102.059
);

count
-----
  3703
(1 row)
```

Dále byla zjišťována četnost měření na jednotlivých bodech pro celou etapu (příloha E.3.2), počet změřených denních úseků s určitým počtem bodů (příloha E.3.3) a počet měření naměřených určitým typem gravimetru:

```

SELECT G.typ, COUNT(G.typ)
FROM gravimetry.gravimetry G
JOIN useky.denni_useky U
ON U.gravimetr_id = G.gravimetr_id
WHERE G.typ NOT IN (1,11) AND EXTRACT (YEAR FROM(U.datum)) < 1995 AND U.usek_id IN
(
  19720101.001,
  ...
  20100102.059
)
GROUP BY G.typ
ORDER BY G.typ;

```

typ	count
2	1846
6	135
21	14
22	1599
24	56
61	53

(6 rows)

Počet měření, která byla zpracována s daným stupněm chodu gravimetru, ukazuje následující dotaz odeslaný do databáze.

```

SELECT COALESCE (U.stupen_chodu, -1) AS stupen , COUNT(COALESCE(U.stupen_chodu, -1)
) AS pocet
FROM gravimetry.gravimetry G
JOIN useky.denni_useky U
ON U.gravimetr_id = G.gravimetr_id
WHERE G.typ NOT IN (1,11) AND EXTRACT (YEAR FROM(U.datum)) < 1995 AND U.usek_id
IN (
  19720101.001,
  ...
  20100102.059
)
GROUP BY stupen
ORDER BY stupen;

```

stupen	pocet
1	1560
2	896
3	1247

(3 rows)

3.4 Příprava dat před měřením

Před měřením nebo kontrolou stavu bodů je dobré vědět, jak dlouho se na bodě neměřilo resp. nekontroloval jeho stav. Zde je ukázka jednoho složitějšího SQL

dotazu, který zjišťuje číslo a název bodu, název okresu a kraje, datum poslední návštěvy a datum posledního měření na bodech, které jsou na území České republiky, nejsou zničeny a jsou součástí sítě systému 1995.

— *vyber bodu, které byly naposledy navštívěny před rokem 2000 a zároveň se na nich naposledy měřilo před rokem 2000, nejsou zrušeny, jsou z CR a jsou to body site 95*

```

SELECT BZ.bod_id , bod_nazev , KR.kraj_nazev , OK.okres_nazev , N.dat AS navsteva , M.
    dat AS mereni
FROM body_zakladni BZ
JOIN
    (
        SELECT MER.bod_id AS bod , MAX(DU.datum) AS dat
        FROM useky.mereni MER
        JOIN useky.denni_useky DU
        ON MER.usek_id = DU.usek_id
        GROUP BY MER.bod_id
    ) M
ON M.bod = BZ.bod_id
JOIN
    (
        SELECT N.bod_id AS bod , MAX(N.datum) AS dat
        FROM navstevy N
        GROUP BY N.bod_id
    ) N
ON N.bod = BZ.bod_id
JOIN body_stav BS
ON BZ.bod_id = BS.bod_id
JOIN body_site BSI
ON BZ.bod_id = BSI.bod_id
LEFT JOIN struktura_obce_2009 STR
ON STR.obec_id = BZ.obec2009_id
LEFT JOIN obce_2009 OB
ON OB.obec_id = STR.obec_id
LEFT JOIN okresy_2009 OK
ON OK.okres_id = STR.okres_id
LEFT JOIN kraje_2009 KR
ON KR.kraj_id = STR.kraj_id
WHERE stat_id = 1 AND BS.stav_celkovy_kod != 5 AND (( ((EXTRACT (YEAR FROM (N.dat)
    )) < 2000 ) OR N.dat IS NULL) AND ( ((EXTRACT (YEAR FROM (M.dat))) < 2000 )
    OR M.dat IS NULL)) AND BSI.sit95_kod IS NOT NULL
ORDER BY BZ.bod_id , KR.kraj_nazev , OK.okres_nazev ;

```

Ukázka jednoho řádku výsledku dotazu:

1737.00 | Kuřim | Jihomoravský kraj | Brno-venkov | 1999-08-25 | 1997-08-28

Závěr

Hlavním výsledkem diplomové práce je relační databáze gravimetrů, tíhových bodů a záznamů měření prováděných oddělením gravimetrie ZÚ od roku 1959. Databáze v systému PostgreSQL verze 8.3.12 obsahuje přes 2400 záznamů tíhových bodů, 148 záznamů nastavení 17 různých typů gravimetrů, 12728 záznamů denních úseků, 147177 záznamů měření na tíhových bodech a 407230 záznamů odečtení jemného čtení. Strukturu databáze tvoří 49 tabulek, ze kterých 42 obsahuje záznamy dat tíhových bodů a pomocné záznamy, 6 tabulek data záznamů tíhových měření a 1 tabulka obsahuje nastavení gravimetrů. Struktury uložených gravimetrů, tíhových bodů a záznamů měření navzájem propojují cizí klíče tak, že je možné klást dotazy na data z celé databáze. Při návrhu struktury databáze jsem se snažil dodržovat 12 pravidel E. F. Codda [7]. Jen ve výjimečných případech vytvořená struktura databáze tyto pravidla nedodržuje. Jedním případem je existence integritních omezení přednostně ve webovém rozhraní, druhým jsou chybějící primární klíče tabulek s uloženým jemným čtením a záznamy kalibrace. Integritní omezení se dají doplnit podle webového rozhraní. Vynechané primární klíče tabulek s jemným čtením a záznamy kalibrací nejsou podle mého názoru potřeba, protože při výběru, aktualizaci, mazání nebo vkládání záznamů do těchto tabulek uživatel pracuje vždy s n -tíci záznamů příslušející danému id měření resp. id úseku a stejnému času.

Druhým důležitým výsledkem diplomové práce je webové rozhraní pro správu databáze napsané v PHP, které umí vytvořit místopis bodu s fotografií, náčrtem a zobrazením bodu na mapách Google. Rozhraní používá 30 souborů skriptů PHP a poskytuje přístup téměř ke všem údajům o tíhových bodech a záznamech měření. Umožňuje data vkládat, zobrazovat a upravovat, neumožňuje zatím záznamy z bezpečnostních důvodů smazat. Mazání záznamů ze všech tabulek a správu údajů o gravimetrech a měření zaznamenaných zastaralými gravimetry GAG, Gs12 a Gs11 je třeba provádět pomocí SQL dotazů nebo jiného prostředí.

Mezi další výsledky práce patří program RegisterImport a skript v interpretu Bash umožňující jednorázové nakopírování nebo doplnění dat do databáze. Program RegisterImport se skládá z 16 souborů zdrojových kódů, které obsahují komentáře pro vytvoření dokumentace javadoc a které usnadní provádění případných změn v tomto programu. Vytvořené série SQL dotazů vytvářející XML výstup z databáze jsou základem pro další vývoj a optimalizaci přípravy naměřených dat ke zpracování novým programem Gravimzung, který je stále vyvíjen. Je možné, že se celá příprava dat bude provádět v rámci jediné databázové funkce v jazyce PL/pgSQL, která vytvoří pro zadané parametry vyrovnání část XML výstupu celého denního úseku. SQL dotazy generující výstupy pro grafické znázornění tíhových bodů a linií denních úseků v programu Surfer a ukázková série SQL dotazů pro tvorbu statistiky jsou jen

zlomkem možností využití SQL dotazů na data nové databáze. Důležitým výsledkem je i text diplomové práce, který slouží jako dokumentace databáze, webového rozhraní a dalších pomocných postupů, skriptů a ukázkových SQL dotazů.

Databáze slouží především k poskytování a přípravě dat tíhových měření a jejich doprovodných údajů společně s údaji o nastavení gravimetrů k dalšímu zpracování ve vyrovnávacím programu. Databáze dále poskytuje základní údaje o tíhových bodech, údaje o jejich stavu, tíži, zařazení do tíhových systémů, jejich polohové a výškové určení a další místopisné informace. Vytvořené webové rozhraní umožňuje údaje o tíhových bodech a tíhových měření nejen prohlížet, ale také snadno spravovat. Dalším praktickým využitím webového rozhraní je vytváření mísopisu z aktuálních dat, se zobrazením bodu na mapách Google, fotografií a místopisným náčrtem. Další využití může databáze nalézt například při exportu souřadnic do formátu vhodného pro nakopírování do GPS navigace usnadňující vyhledávání tíhových bodů v terénu nebo při výpočtu některých údajů před vyrovnáním pomocí PL/pgSQL funkcí.

Veškeré úpravy textových souborů, skriptů, XML souborů, tex souborů a souborů obsahující SQL jsem prováděl v textovém editoru Emacs. Pro vývoj webového rozhraní a programu RegisterImport jsem použil prostředí NetBeans IDE 6.7. Grafická znázornění struktur uložených dat jsem vytvořil pomocí programu postgresql-autodoc a upravil v programu Dia. Diplomová práce byla vytvořena v sázecím prostředí L^AT_EX na základě šablony z předmětu Kartografická polygrafie a reprografie, kterou vytvořil Ing. Petr Soukup Ph.D.

Jan Gürtler

Reference

- [1] ČSÚ [online], [cit. 2010-11-25]. Poslední aktualizace 24. 11. 2010. Dostupné z URL: <<http://www.czso.cz/>>.
- [2] *Geoportál ČÚZK* [online], [cit. 2010-11-24]. Poslední aktualizace 6. 10. 2010. Dostupné z URL: <<http://geoportal.cuzk.cz/>>.
- [3] *Mapy Google* [online], [cit. 2010-11-24]. Poslední aktualizace 24. 11. 2010. Dostupné z URL: <<http://maps.google.cz/>>.
- [4] *PostgreSQL* [online], [cit. 2010-12-16]. Poslední aktualizace 9. 12. 2010. Dostupné z URL: <<http://www.postgresql.org/>>.
- [5] *php* [online], [cit. 2010-12-16]. Poslední aktualizace 16. 12. 2010. Dostupné z URL: <<http://php.net/>>.
- [6] *Linuxsoft.cz, Seriál PHP* [online], [cit. 2010-12-16]. Poslední aktualizace 13. 11. 2009. Dostupné z URL: <<http://www.linuxsoft.cz/php/>>.
- [7] *Codd's 12 rules* [online], [cit. 2010-12-20]. Poslední aktualizace 18. 12. 2009. Dostupné z URL: <http://en.wikipedia.org/wiki/Codd's_12_rules>.
- [8] *php-google-map-api* [online], [cit. 2010-12-16]. Poslední aktualizace 23. 10. 2010. Dostupné z URL: <<http://code.google.com/p/php-google-map-api/>>.
- [9] *Java SE Technical Documentation* [online], [cit. 2010-12-16]. Poslední aktualizace 2010. Dostupné z URL: <<http://download.oracle.com/javase/>>.
- [10] *WIKIBOOKS, LaTeX* [online], [cit. 2010-12-16]. Poslední aktualizace 23. 11. 2010. Dostupné z URL: <<http://en.wikibooks.org/wiki/LaTeX>>.
- [11] SHARON ZAKHOUR, SCOTT HOMMEL, JACOB ROYALL, ISAAC RABINOVITCH, TOM RISSER, MARK HOEBER *Java 6 Výukový kurz*. Brno: Computer Press, 2007, 536 s., ISBN 978-80-251-1575-6
- [12] JAMES KEOGH *Java bez předchozích znalostí, Průvodce pro samouky*. Brno: Computer Press, 2005, 280 s., ISBN 80-251-0839-2
- [13] NESVADBA OTAKAR *Plošná revize údajů o poloze bodů ČSGS, Studie pro bod 2019 Chotýčany*, Technická zpráva. Praha: Zeměměřický úřad, 2009
- [14] LEDERER MARTIN, NESVADBA OTAKAR, LUKAVEC PETR, TRAKAL JAROSLAV, GÜRTLER JAN *Nový tíhový systém ČR; Tíhový systém 2010 - SGR10*, Rozpracovaná technická zpráva. Praha: Zeměměřický úřad, 2011?

- [15] JAN KOSTELECKÝ *Zpracování gravimetrických měření, vyrovnání denního úseku a vyrovnání gravimetrické sítě středního rozsahu v prostředí IBM-PC (verze 1.3)*, Výzkumná zpráva č. 995/99. Zdiby: Výzkumný ústav geodetický, topografický a kartografický, 1999, 22 s.

Seznam symbolů, veličin a zkratek

PL/pgSQL Procedural Language/PostgreSQL Structured Query Language

ETRS89 European Terrestrial Reference System 1989

SQL Structured Query Language - programovací jazyk pro relační databáze

Bash Bourne again shell - unixový (Linux/Unix/BSD) příkazový interpret shellu naprogramovaný v rámci projektu GNU

XML Extensible Markup Language

HTML HyperText Markup Language

PHP Hypertext Preprocessor

CPI Capacitive Positioning Indicator

JDK Java Development Kit

ČSÚ Český statistický úřad

ZÚ Zeměměřický úřad

ČÚZK Český úřad zeměměřický a katastrální

CSV Comma Separated Values - hodnoty oddělené čárkami

JDK Java Development Kit

Seznam příloh

A	Zdroje dat	68
A.1	Textové soubory adjustovaných zápisníků	68
A.1.1	Typ 4, GS12 a GS11	68
A.1.2	Typ 5, GAG	68
A.1.3	Typ 6, Sharpe a Worden	69
A.1.4	Typ 7, LaCoste & Romberg G CPI	69
A.1.5	Typ 8, LaCoste & Romberg G a Scintrex	70
A.1.6	Typ 9, LaCoste & Romberg D	71
A.1.7	Soubory rovnic oprav	71
A.2	Katalogy gravimzu	73
A.2.1	Katalog měřických skupin	73
A.2.2	Katalog gravimetrů	75
A.2.3	Katalog tíhových bodů	76
B	Struktura databáze	77
B.1	Struktura uložení tíhových bodů	77
B.2	Struktura uložení denních úseků	80
B.3	Struktura uložení gravimetrů	81
C	Naplnění databáze	83
C.1	Skript pro program RegisterImport	83
D	PHP webové rozhraní	84
D.1	Snímky obrazovky	84
E	Výstupy z databáze	90
E.1	XML výstupy pro Gravimzung	90
E.1.1	query_point.sql	90
E.1.2	query_section.sql	91
E.1.3	query_calibration.sql	95
E.1.4	query_observation.sql	95
E.2	Výstupy pro program Surfer	96
E.2.1	Funkce get_points(point_id)	96
E.2.2	Funkce get_polygons(usek_id)	97
E.3	Tvorba statistik	98
E.3.1	Skript get_index	98
E.3.2	Četnost měření na bodech	99

E.3.3	Počet změřených denních úseků s daným počtem bodů	99
E.4	Grafické znázornění v programu Surfer	100

A Zdroje dat

A.1 Textové soubory adjustovaných zápisníků

A.1.1 Typ 4, GS12 a GS11

AZ610101.001

```

4 2 1
1 ' II HRENSKO - VI STRASKOV '
1 129 3 40.0 18 5 1961 1.0 'TRAGER ' 1 '
2
06 23 40 42 -14.3 40 32 15.3 40 42 -14.0
17 42 38 32 -13.4 38 22 15.9 38 32 -13.5
34 2
2003.0000 1 50 2 9.0 100.9 6 28 ' 15 ' 40.37 .4 .6 -8 88.0 0
2003.0000 1 50 2 9.0 100.9 6 32 ' 5 ' 35.18 -.3 -1.0 -8 88.0 0
2005.0000 1 50 2 14.0 100.8 7 5 ' 25 ' 44.83 .8 1.3 -7 88.0 0
2005.0000 1 50 2 14.0 100.8 7 11 ' 15 ' 39.66 2.0 2.2 -7 88.0 0
2006.0000 1 50 3 15.0 100.4 8 5 ' 25 ' 44.19 -.3 .7 -6 88.0 0
2006.0000 1 50 3 15.0 100.4 8 11 ' 15 ' 39.02 2.2 2.7 -6 88.0 0
2008.0000 1 25 4 14.0 100.4 8 59 ' 25 ' 44.77 2.0 2.0 -5 88.0 0
2008.0000 1 25 4 14.0 100.4 9 2 ' 15 ' 39.60 3.2 4.0 -5 88.0 0
2009.0000 1 25 3 13.0 99.7 9 30 ' 25 ' 43.44 2.5 3.0 -4 88.0 0
2009.0000 1 25 3 13.0 99.7 9 33 ' 15 ' 38.27 3.8 4.8 -4 88.0 0
271.0000 1 25 2 15.0 99.4 10 6 ' 25 ' 42.56 3.9 3.4 -4 88.0 0
271.0000 1 25 2 15.0 99.4 10 11 ' 15 ' 37.39 4.1 3.7 -4 88.0 0
2009.0000 1 25 3 14.0 101.0 10 41 ' 25 ' 43.44 2.0 1.5 -5 88.0 0
2009.0000 1 25 3 14.0 101.0 10 44 ' 15 ' 38.27 3.1 2.3 -5 88.0 0
2008.0000 1 25 4 16.0 100.3 11 14 ' 25 ' 44.77 -.4 -1.0 -5 88.0 0
2008.0000 1 25 4 16.0 100.3 11 17 ' 15 ' 39.60 1.0 1.0 -5 88.0 0
2006.0000 1 25 2 18.0 100.2 12 19 ' 25 ' 44.19 -2.1 -2.9 -5 88.0 0
2006.0000 1 25 2 18.0 100.2 12 22 ' 15 ' 39.02 -1.8 -1.8 -5 88.0 0
2005.0000 1 50 3 17.0 100.5 13 3 ' 25 ' 44.83 -7.4 -6.8 -6 88.0 0
2005.0000 1 50 3 17.0 100.5 13 6 ' 15 ' 39.66 -5.0 -5.8 -6 88.0 0
2003.0000 1 50 2 20.0 100.5 14 0 ' 15 ' 40.37 -8.9 -8.0 -6 88.0 0
2003.0000 1 50 2 20.0 100.5 14 3 ' 5 ' 35.18 -9.0 -9.1 -6 88.0 0
2002.0000 1 50 3 20.0 100.4 14 13 ' 15 ' 40.51 -8.8 -8.9 -7 88.0 0
2002.0000 1 50 3 20.0 100.4 14 16 ' 5 ' 35.32 -9.2 -9.2 -7 88.0 0
2003.0000 1 50 2 18.0 100.5 14 25 ' 15 ' 40.37 -8.6 -8.9 -7 88.0 0
2003.0000 1 50 2 18.0 100.4 14 28 ' 5 ' 35.18 -9.2 -9.2 -7 88.0 0
2005.0000 1 50 3 21.0 100.4 14 59 ' 25 ' 44.83 -8.0 -7.8 -8 88.0 0
2005.0000 1 50 3 21.0 100.4 15 2 ' 15 ' 39.66 -6.0 -6.2 -8 88.0 0
2006.0000 1 100 2 17.0 100.0 15 55 ' 25 ' 44.19 -4.4 -4.0 -8 88.0 0
2006.0000 1 100 2 17.0 100.0 15 59 ' 15 ' 39.02 -3.6 -3.0 -8 88.0 0
2008.0000 1 75 2 18.0 100.1 17 3 ' 25 ' 44.77 -3.5 -3.5 -8 88.0 0
2008.0000 1 75 2 18.0 100.1 17 7 ' 15 ' 39.60 -2.2 -1.6 -8 88.0 0
2009.0000 1 60 3 16.0 99.3 17 34 ' 25 ' 43.44 -.7 -1.0 -8 88.0 0
2009.0000 1 60 3 16.0 99.3 17 37 ' 15 ' 38.27 .9 .2 -8 88.0 0

```

A.1.2 Typ 5, GAG

AZ680309.007

5 2 1

7 ' GDANSK BARTH																	
3	3	1	35.0	24	6	1968	3.0	'RUKAVISNIKOV				2	'AN 24	46207			
12	1																
40.0000	0	0	0	.0	.0	9	15	150	7	21.0	.0	147	9	6.2	.0	.0	0
40.0000	0	0	0	.0	.0	9	17	210	27	24.2	.0	207	29	10.0	.0	.0	0
40.0000	0	0	0	.0	.0	9	19	270	47	23.8	.0	267	49	11.6	.0	.0	0
41.0000	0	0	0	.0	.0	12	25	270	52	54.8	.0	267	51	16.2	.0	.0	0
41.0000	0	0	0	.0	.0	12	28	331	12	51.8	.0	328	11	16.0	.0	.0	0
41.0000	0	0	0	.0	.0	12	30	31	32	35.8	.0	28	30	59.0	.0	.0	0
40.0000	0	0	0	.0	.0	15	18	90	51	20.1	.0	87	52	50.2	.0	.0	0
40.0000	0	0	0	.0	.0	15	20	151	11	26.8	.0	148	12	56.8	.0	.0	0
40.0000	0	0	0	.0	.0	15	22	211	31	12.6	.0	208	32	43.2	.0	.0	0
41.0000	0	0	0	.0	.0	18	47	240	8	49.6	.0	237	6	58.2	.0	.0	0
41.0000	0	0	0	.0	.0	18	50	300	28	39.6	.0	297	26	47.4	.0	.0	0
41.0000	0	0	0	.0	.0	18	52	10	48	41.2	.0	7	46	51.0	.0	.0	0

A.1.3 Typ 6, Sharpe a Worden

AZ670102.020

6 2 1														
4 ' VOTICE CHOITYCANY														
22	961	14	32.0	28	4	1993	2.0	'DIVIS				1	'SF AEC	6341
17	3													
2014.0100	1	10	2	16.0	96.4	8	33	.00	1919.00	1918.80	1918.90	41.00	3	
2015.0000	1	10	1	16.0	94.4	8	53	.00	1524.80	1524.80	1524.80	41.00	3	
2016.0000	1	10	2	20.7	97.2	9	30	.00	1734.00	1734.00	1734.10	41.00	3	
2017.0000	1	10	2	22.0	97.2	10	23	.00	1529.10	1529.00	1529.00	41.00	0	
2019.0000	1	0	3	20.3	95.8	10	56	.00	1162.40	1162.50	1162.50	41.00	0	
2017.0000	1	0	3	24.0	97.2	11	56	.00	1529.80	1530.00	1529.80	41.00	0	
2016.0000	1	10	3	24.0	97.2	12	22	.00	1735.70	1735.70	1735.90	41.00	0	
2015.0000	1	20	3	24.6	94.3	13	4	.00	1526.20	1526.20	1526.40	41.00	0	
2014.0100	1	30	3	23.0	96.3	13	22	.00	1919.50	1919.60	1919.50	41.00	0	
2015.0000	1	30	3	26.3	94.3	13	41	.00	1526.20	1526.00	1526.20	41.00	0	
2016.0000	1	40	3	23.5	97.1	14	16	.00	1736.50	1736.70	1736.60	41.00	0	
2017.0000	1	40	2	26.4	97.1	14	47	.00	1531.60	1531.50	1531.60	41.00	0	
2019.0000	1	30	4	22.5	95.6	15	58	.00	1163.80	1163.80	1163.70	41.00	0	
2017.0000	1	30	3	23.5	97.0	17	15	.00	1530.30	1530.20	1530.10	41.00	0	
2016.0000	1	10	3	24.0	96.9	17	39	.00	1736.00	1736.10	1736.10	41.00	0	
2015.0000	1	10	2	21.0	94.1	18	17	.00	1525.90	1525.80	1525.80	41.00	0	
2014.0100	1	10	2	23.3	96.1	18	35	.00	1918.20	1918.20	1918.10	41.00	0	

A.1.4 Typ 7, LaCoste & Romberg G CPI

AZ090102.006

7 2 1														
6 'Kunratice – Rehenice														
6	1068	1	53.0	16	4	2009	2.0	'Lukavec				1	'VWC4A74983	
2	4													
			.00000000											
9	50	4428	306	-1	4428	406	-265	4428	206	+287	4428	306	+8	
	14	43	4428	232	+1	4428	332	-266	4428	132	+272	4428	232	-8
9	3													
272.0100	1	0	2	18.0	97.0	9	48							
			4428.	296.0	304.0	306.0	26.0	0						
				21.00	2.00	-1.00								

2011.0200	1	0 1	16.0	96.0	10 21				
		4405.	921.0	921.0	920.0	26.0	0		
			-4.00	-5.00	-1.00				
1723.0100	1	0 2	19.0	97.0	10 51				
		4415.	938.0	932.0	930.0	26.0	0		
			-16.00	-3.00	7.00				
2011.0200	1	0 1	18.0	93.0	11 14				
		4405.	915.0	915.0	916.0	26.0	0		
			5.00	8.00	6.00				
272.0100	1	0 3	22.0	97.0	11 40				
		4428.	272.0	277.0	275.0	26.0	0		
			15.00	1.00	13.00				
2011.0200	1	0 2	19.0	96.0	12 7				
		4405.	891.0	892.0	891.0	26.0	0		
			4.00	.00	.00				
1723.0100	1	0 3	24.0	97.0	13 15				
		4415.	887.0	885.0	885.0	26.0	0		
			-2.00	6.00	6.00				
2011.0200	1	10 2	21.0	96.0	14 11				
		4405.	857.0	857.0	861.0	26.0	0		
			11.00	11.00	-6.00				
272.0100	1	10 3	24.0	97.0	14 42				
		4428.	240.0	236.0	232.0	26.0	0		
			-10.00	-3.00	1.00				

*
*
*
*
*

A.1.5 Typ 8, LaCoste & Romberg G a Scintrex

AZ080101.028

8	2	1							
28			'Skutec - Nove Mesto						
25	125	1	67.0	27	8	2008	2.0	'Trakal	1 'VWC 4A7 4983'
13	2								
3346.0200	1	10 2	19.0	97.5	9 18	4403.00	5.0	4.0	46.80 0
3346.7100	1	10 1	22.9	97.8	10 12	4421.00	331.0	332.0	46.80 0
3332.0100	1	10 2	26.7	99.7	11 27	4459.00	307.0	306.0	46.70 0
3327.0100	1	10 3	25.9	99.6	12 29	4469.00	577.0	578.0	46.80 0
3332.0100	1	10 2	27.2	99.6	13 49	4459.00	410.0	409.0	46.80 0
3346.7100	1	20 2	29.0	97.7	14 49	4421.00	517.0	518.0	46.80 0
3346.0200	1	20 3	25.0	97.3	15 24	4403.00	221.0	223.0	46.80 0
3346.7100	1	40 2	28.9	97.6	16 1	4421.00	578.0	578.0	46.80 0
3332.0100	1	10 3	27.7	99.4	17 1	4459.00	572.0	572.0	46.80 0
3327.0100	1	10 2	28.0	99.3	17 52	4469.00	837.0	835.0	46.80 0
3332.0100	1	10 2	26.9	99.3	18 40	4459.00	620.0	622.0	46.70 0
3346.7100	1	10 1	21.3	97.5	19 36	4421.00	683.0	683.0	46.80 0
3346.0200	1	10 1	20.8	97.1	20 17	4403.00	358.0	358.0	46.80 0

*
*
*
*
*

A.1.6 Typ 9, LaCoste & Romberg D

AZ930301.030

9	2	1													
30	'	LITOMERICE	KUNRATICE												
61	180	2	55.0	22	6	1993	2.0	'KOVACIK		'	1				
17	3														
271.0000	1	30	1	28.9		99.7		12	53	1497.00	75.00	75.00	74.50	26.20	0
2009.0000	1	30	1	23.7		100.0		13	30	1656.00	55.00	55.00	54.30	26.10	0
2008.0000	1	30	1	24.7		100.5		14	17	1897.00	14.20	14.50	14.10	26.20	0
2007.0000	1	60	1	26.0		100.5		14	38	1914.00	29.00	29.30	29.00	25.80	0
2008.0000	1	60	1	24.3		100.5		15	0	1897.00	36.40	37.70	38.30	26.20	0
2009.0000	1	60	1	24.1		99.8		15	43	1656.00	45.00	45.90	46.80	26.10	0
271.0000	1	50	1	26.9		99.6		16	26	1497.00	69.20	70.80	72.30	26.20	0
2009.0000	1	50	1	24.9		99.7		17	7	1656.00	72.50	73.60	74.90	26.10	0
2008.0000	1	50	1	23.8		100.4		17	46	1898.00	.40	2.60	4.10	26.20	0
2007.0000	1	50	1	24.5		100.4		18	3	1914.00	92.00	94.00	95.20	25.80	0
2008.0000	1	50	1	24.9		100.4		18	21	1898.00	30.60	33.00	34.30	26.20	0
2009.0000	1	30	1	24.7		99.6		18	55	1657.00	61.20	62.30	63.20	26.10	0
271.0000	1	30	1	24.9		99.5		19	27	1498.00	94.80	95.80	96.90	26.20	0
272.0100	1	60	1	22.6		98.8		20	15	1084.00	21.80	22.00	24.00	25.80	0
271.0000	1	100	1	21.8		99.5		21	0	1499.00	52.80	53.90	54.70	26.20	0
272.0100	1	100	1	20.7		98.8		21	44	1084.00	51.40	52.70	55.70	25.80	0
271.0000	2	100	1	20.6		99.5		22	32	1499.00	65.00	65.90	66.80	26.20	0

A.1.7 Soubory rovnic oprav

R3930102.004

ROV. OPRAV DENNIHO USEKU: 930102.004

DATUM=27.12.1999 CAS=10:05

GRAVIMETR = 22 961

ROVNIC OPRAV = 16

NEZNAMYCH TIZI = 5

1 2014.0100 VOTICE

2 2015.0000 MILICIN

3 2016.0000 PLANA N LUZ

4 2017.0000 RIPEC

5 2019.0000 CHOITYCANY

POCET NEZN. CHODU = 4

6 a'

7 b

8 c

9 d

POCET SKOKU = 1

10

MER. FAKTOR

11 Y1'

12 Y2'

ROVNICE OPRAV:

1.00000	.00000	.00000	.00000	.00000
-1.00000	.00000	.00000	.00000	.00000
.00000	.00000			
1.00000	.00000			
.00000	1.00000	.00000	.00000	.00000

-1.00000	-.33333	-.11111	-.03704	.00000
.00000	423.33800			
1.00000	423.33800			
.00000	.00000	1.00000	.00000	.00000
-1.00000	-.95000	-.90250	-.85737	.00000
.00000	198.47400			
1.00000	198.47400			
.00000	.00000	.00000	1.00000	.00000
-1.00000	-1.83333	-3.36111	-6.16204	1.00000
.00000	418.73200			
1.00000	418.73200			
.00000	.00000	.00000	.00000	1.00000
-1.00000	-2.38333	-5.68028	-13.53800	1.00000
.00000	812.22400			
1.00000	812.22400			
.00000	.00000	.00000	1.00000	.00000
-1.00000	-3.38333	-11.44694	-38.72883	1.00000
.00000	417.63600			
1.00000	417.63600			
.00000	.00000	1.00000	.00000	.00000
-1.00000	-3.81667	-14.56694	-55.59717	1.00000
.00000	196.26900			
1.00000	196.26900			
.00000	1.00000	.00000	.00000	.00000
-1.00000	-4.51667	-20.40028	-92.14125	1.00000
.00000	421.27000			
1.00000	421.27000			
1.00000	.00000	.00000	.00000	.00000
-1.00000	-4.81667	-23.20028	-111.74800	1.00000
.00000	-1.23500			
1.00000	-1.23500			
.00000	1.00000	.00000	.00000	.00000
-1.00000	-5.13333	-26.35111	-135.26904	1.00000
.00000	421.28100			
1.00000	421.28100			
.00000	.00000	1.00000	.00000	.00000
-1.00000	-5.71667	-32.68028	-186.82225	1.00000
.00000	194.96300			
1.00000	194.96300			
.00000	.00000	.00000	.00000	1.00000
-1.00000	-7.41667	-55.00694	-407.96817	1.00000
.00000	809.86600			
1.00000	809.86600			
.00000	.00000	.00000	1.00000	.00000
-1.00000	-8.70000	-75.69000	-658.50300	1.00000
.00000	416.32800			
1.00000	416.32800			

.00000	.00000	1.00000	.00000	.00000
-1.00000	-9.10000	-82.81000	-753.57100	1.00000
.00000	195.06300			
1.00000	195.06300			
.00000	1.00000	.00000	.00000	.00000
-1.00000	-9.73333	-94.73778	-922.11437	1.00000
.00000	421.01600			
1.00000	421.01600			
1.00000	.00000	.00000	.00000	.00000
-1.00000	-10.03333	-100.66778	-1010.03337	1.00000
.00000	-.40400			
1.00000	-.40400			

A.2 Katalogy gravimzu

A.2.1 Katalog měřických skupin

ukázky ze souboru Kat.ms.txt

...

```

MERICKA SKUPINA 85 1 GEODETICKY A^KARTOGRAFICKY PODNIK P R A^H
A^SEZNAM GRAVIMETRU
  TYP CISLO PK NAZEV A^UZIVATEL GRAVIMETRU
  1 1 129 22 GS 12 GU PRAHA
  2 2 174 24 SHARPE GKP PRAHA
  3 22 961 14 WORDEN GKP PRAHA
MERICKA SKUPINA 85 2 GEODETICKY USTAV NP B R A^T I^S^L
A^V^A^SEZNAM GRAVIMETRU
  TYP CISLO PK NAZEV A^UZIVATEL GRAVIMETRU
  1 2 280 21 SHARPE GU BRATISLAVA
  2 22 978 11 WORDEN GU BRATISLAVA
MERICKA SKUPINA 85 3 Z^K^O^U^S^K^Y P R I^S^T R O^J
U^SEZNAM GRAVIMETRU
  TYP CISLO PK NAZEV A^UZIVATEL GRAVIMETRU
  1 2 174 21 SHARPE GU PRAHA
  2 22 961 11 WORDEN GU PRAHA
MERICKA SKUPINA 85 4 DILCI JEDNOTNA GRAVIMETRICKA SIT - CSSR 1985
SEZNAM GRAVIMETRU
  TYP CISLO PK NAZEV A^UZIVATEL GRAVIMETRU
  1 2 174 24 SHARPE GKP PRAHA
  2 22 961 14 WORDEN GKP PRAHA
  3 2 280 21 SHARPE GU BRATISLAVA
  4 22 978 11 WORDEN GU BRATISLAVA
  5 22 1217 2 WORDEN VUGK BLAVA
  6 2 181 11 SHARPE ELGI BUDAPEST
  7 2 256 11 SHARPE ELGI BUDAPEST
  8 22 971 1 WORDEN ELGI BUDAPEST
  9 6 1919 1 LA COSTE ELGI BUDAPEST
MERICKA SKUPINA 85 5 M A^E L G I^B U^D A^P E S^T
SEZNAM GRAVIMETRU
  TYP CISLO PK NAZEV A^UZIVATEL GRAVIMETRU
  1 2 181 11 SHARPE ELGI BUDAPEST
  2 2 256 11 SHARPE ELGI BUDAPEST
  3 22 971 1 WORDEN ELGI BUDAPEST

```

4 6 1919 2 LA COSTE ELGI BUDAPEST

...

...

MERICKA SKUPINA 10 1 ZEMEMERICKY URAD PRAHA + VUGIK ZDIBY

SEZNAM GRAVIMETRU

	TYP	CISLO	PK	NAZEV A [~] UZIVATEL	GRAVIMETRU
1	25	125	1	SCINTREX	ZU PRAHA
2	6	1068	1	LA COSTE G	ZU PRAHA
3	6	137	1	LA COSTE G	VUGIK ZDIBY
4	62	20	1	ZLS	VUGIK ZDIBY

MERICKA SKUPINA 10 2 GEODETICKY A[~]KARTOGRAFICKY USTAV BRATISLAVA

SEZNAM GRAVIMETRU

	TYP	CISLO	PK	NAZEV A [~] UZIVATEL	GRAVIMETRU
1	61	180	2	LA COSTE D	GU BRATISLAVA
2	6	1011	1	LA COSTE G	GKU BLAVA
3	2	280	21	SHARPE	GU BRATISLAVA
4	22	978	11	WORDEN	GU BRATISLAVA

...

A.2.2 Katalog gravimetrů

ukázka ze souboru Kat_grav.txt

```

...
V^E T A^S^O^U^B O^R U^K^T G R P
CISLO GR NAZEV GR MAJITEL GR D-UL.K D-VT.K D-ZC.P D-KN.P UK VZ KA S1-DM S1-HM S2-DM S2-HM S3-DM S3-HM T T-PK VZ-MS
01 0129 01 GS 12 GTU PRAHA 880819 850612 580101 591231 11 1 1 0 5 0 81 40 40 1 40.0 -400
V^E T Y S^O^U^B O^R U^K^T G K^S^T
-7.081354000000E+03 1.769596000000E+02 1.187500000000E-05 4.610940000000E-05 0.000000000000E+00 1.000560000000E+00
9.374800000000E-05 1.870830000000E-04 3.746970000000E-04 3.740840000000E-04 7.489090000000E-04 0.000000000000E+00
V^E T Y S^O^U^B O^R U^K^T G V^N
V^GRAVIMETR CISLO 01 0129 01 NEMA VETY VNEJSICH VLVIVU
...
V^E T Y S^O^U^B O^R U^K^T G V^N
V^GRAVIMETR CISLO 02 0312 01 NEMA VETY VNEJSICH VLVIVU
V^E T A^S^O^U^B O^R U^K^T G R P
CISLO GR NAZEV GR MAJITEL GR D-UL.K D-VT.K D-ZC.P D-KN.P UK VZ KA S1-DM S1-HM S2-DM S2-HM S3-DM S3-HM T T-PK VZ-MS
03 0001 01 GAG 2 IFZ MCKBA 880819 810619 680101 951231 1 4 4 0 360 0 60 0 0 65 1 35.0 -200
V^E T Y S^O^U^B O^R U^K^T G K^S^T
6.000000000000E+01 1.000000000000E+00 1.000000000000E+00 1.000000000000E+00 1.000000000000E+00 5.000000000000E+00
V^E T Y S^O^U^B O^R U^K^T G V^N
V^GRAVIMETR CISLO 03 0001 01 NEMA VETY VNEJSICH VLVIVU
...
V^E T A^S^O^U^B O^R U^K^T G R P
CISLO GR NAZEV GR MAJITEL GR D-UL.K D-VT.K D-ZC.P D-KN.P UK VZ KA S1-DM S1-HM S2-DM S2-HM S3-DM S3-HM T T-PK VZ-MS
25 0125 01 SCINTREX ZU PRAHA 050617 050617 100617 1 2 5 0 7000 0 1100 0 0 1100 0 0 1 67.0 -202
V^E T Y S^O^U^B O^R U^K^T G K^S^T
0.000000000000E+00 6.700000000000E+01 1.000000000000E+03 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00
0.000000000000E+00 1.000000000000E+01 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00
1.000000000000E-20 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00
1.000000000000E-20 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00
1.000000000000E+00 1.000000000000E+00 1.000000000000E+00 1.000000000000E+00 1.000000000000E+00 1.000000000000E+00
V^E T Y S^O^U^B O^R U^K^T G V^N
V^GRAVIMETR CISLO 25 0125 01 NEMA VETY VNEJSICH VLVIVU
...

```

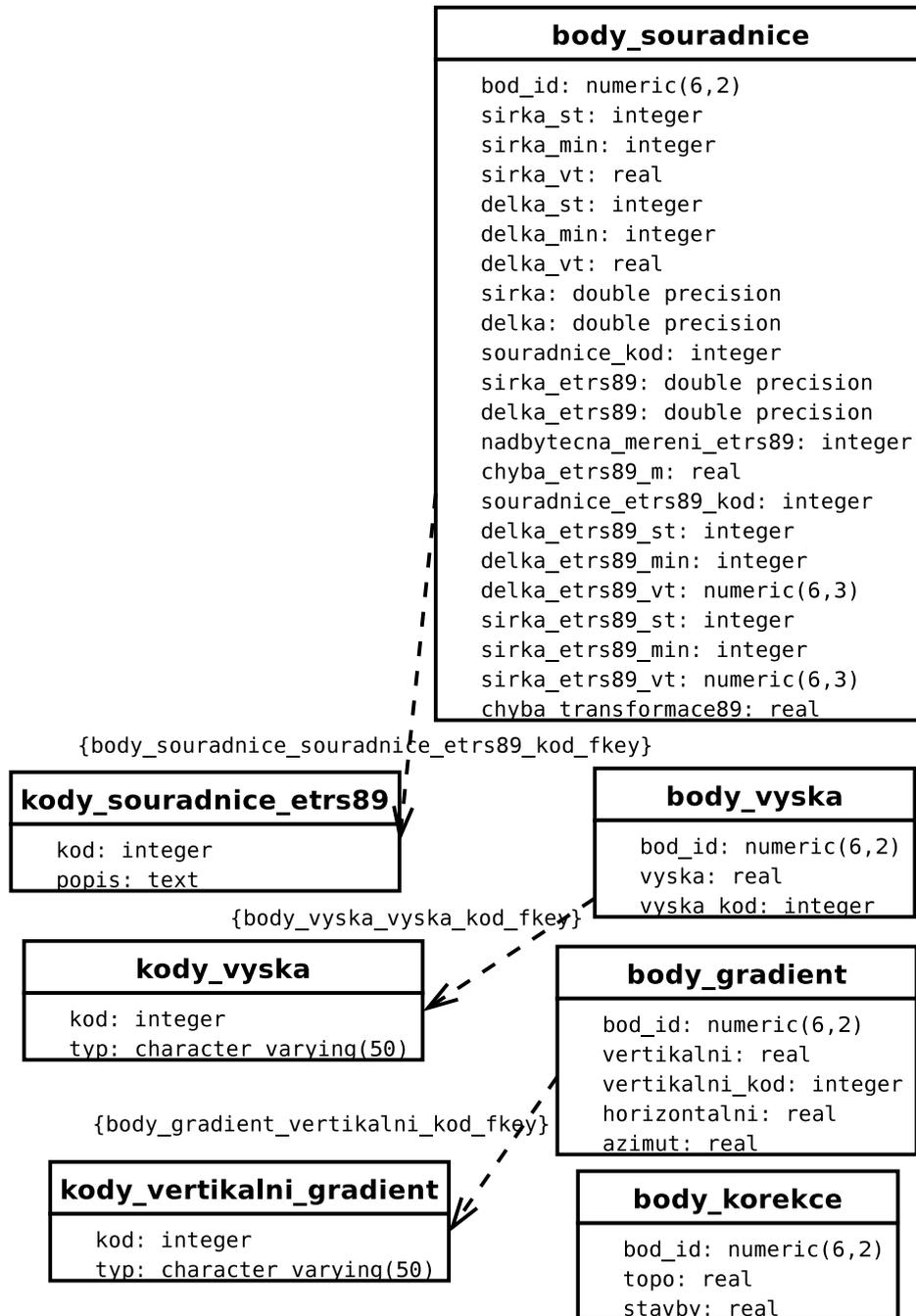
A.2.3 Katalog tíhových bodů

ukázka ze souboru KAT_BODY.txt

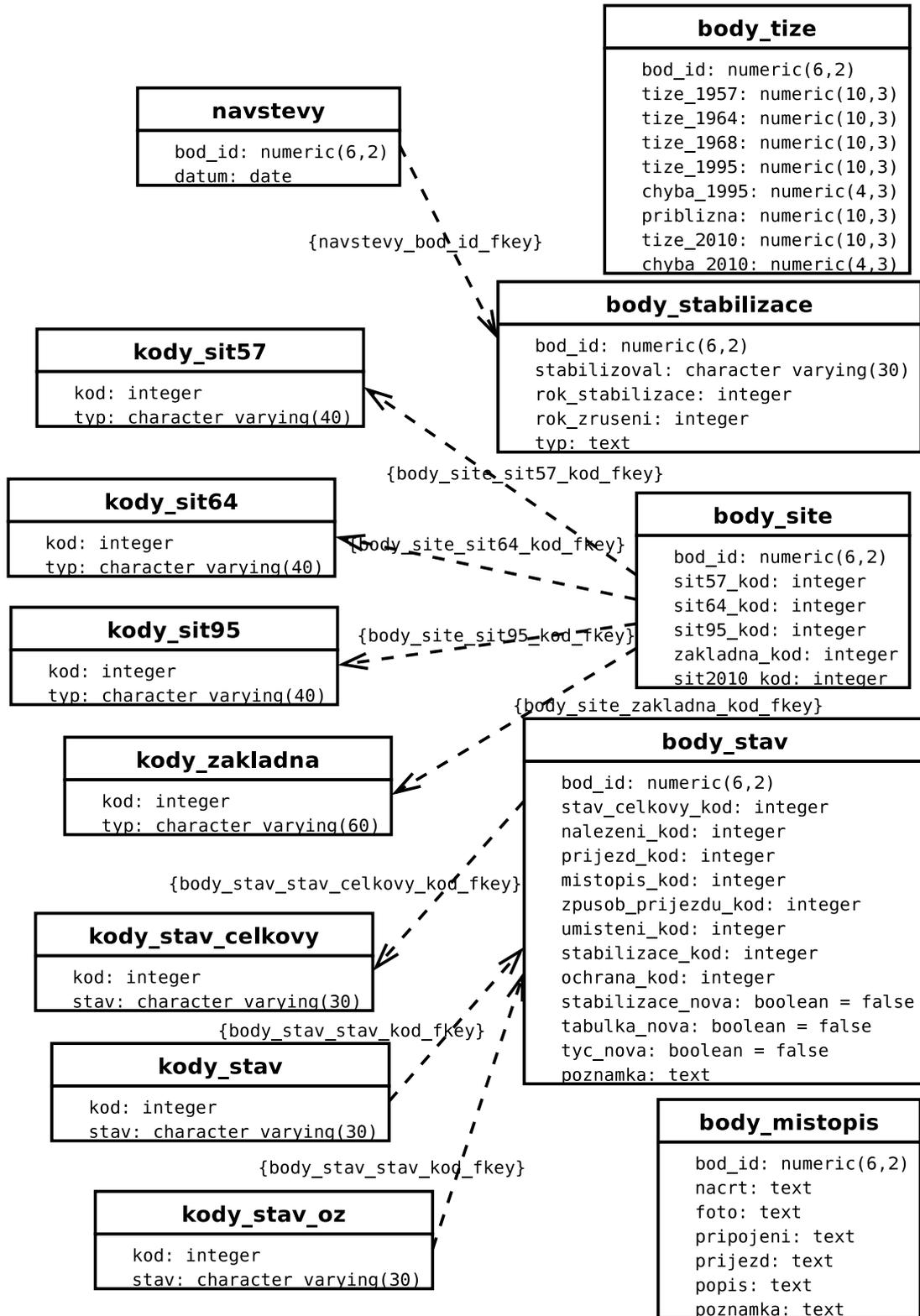
21	TALLINN	68	880603	59	9	60.00	24	37	60.00	.000	0	0	-3084	8	71	9818300.99	0	.00
21.01	TALLINN	74	880603	59	9	60.00	24	37	60.00	.000	0	0	-3084	8	71	9818302.39	0	.00
22	VILNIUS	68	880603	54	37	60.00	25	17	60.00	.000	0	0	-3085	8	71	9814377.70	0	.00
22.01	VILNIUS	74	880603	54	37	60.00	25	17	60.00	.000	0	0	-3085	8	71	9814379.76	0	.00
23	MOCKBA		880603	56	11	60.00	37	19	60.00	.000	0	0	-3085	8	0	9815732.80	0	.00
24	POISDAM S~2		880603	52	22	60.00	13	3	60.00	85.370	0	0	-3085	3	71	9812607.10	0	.00
25	BERLIN		880603	52	21	60.00	13	31	60.00	.000	0	0	-3085	3	71	9812577.93	0	.00
26	WARSAWA	68	880603	52	7	60.00	21	0	.00	.000	0	0	-3085	4	71	9812224.03	0	.00
26.01	WARSAWA	74	880603	52	7	60.00	21	0	.00	.000	0	0	-3085	4	71	9812223.63	0	.00
27	PRAHA RUZYNE		880603	50	6	3.00	14	16	27.00	361.690	11	291	-3047	1	64	9810018.00	68	1.70
27.50	PRAHA RUZ EX		880603	50	5	57.00	14	17	22.00	364.220	0	0	-3130	1	64	9810009.70	68	1.70
28	KRAKOW	68	880603	50	7	60.00	19	54	60.00	.000	0	0	-3085	4	71	9810370.51	0	.00
28.01	KRAKOW BALIC		880603	50	7	60.00	19	54	60.00	.000	0	0	-3085	4	68	9810510.50	0	.00
28.02	KRAKOW BALI		890721	50	7	60.00	19	54	60.00	.000	0	0	-3085	4	0	9810510.00	0	.00
29	BUDAPEST		880603	47	26	58.00	19	13	20.00	136.777	0	0	-3085	5	68	9808284.40	0	.00
29.01	BUDAPEST		880603	47	26	60.00	19	13	60.00	140.000	0	0	-3085	5	0	9808280.00	0	.00
29.13	BUDAPEST EX1		880603	47	26	60.00	19	13	60.00	140.000	0	0	-3085	5	0	9808280.00	0	.00
29.14	BUDAPEST EX2		880603	47	26	60.00	19	13	60.00	140.000	0	0	-3085	5	0	9808280.00	0	.00
29.15	BUDAPEST EX3		880603	47	26	60.00	19	13	60.00	140.000	0	0	-3085	5	0	9808280.00	0	.00
30	BUCURESTI		880603	44	27	60.00	26	5	60.00	.000	0	0	-3086	7	0	9805440.00	0	.00
31	SOFIA		880603	42	42	5.00	23	24	12.00	534.000	0	0	-3085	6	68	9802615.70	0	.00
40	GDANSK		880603	54	22	60.00	18	37	60.00	.000	0	0	-3085	4	71	9814389.20	0	.00
41	BARTH		880603	54	20	60.00	12	40	60.00	.000	0	0	-3085	3	71	9814511.00	0	.00
42	DRESDEN		880603	51	3	60.00	13	49	60.00	.000	0	0	-3085	3	71	9811099.80	0	.00
...																		
2024	D DVORISTE		880603	48	38	59.00	14	26	25.00	670.250	52	283	-3013	1	64	9807706.38	68	1.40
2024.10	DOL DVORISTE		880603	48	38	59.00	14	26	25.00	670.250	52	283	-3013	1	0	9807844.90	0	.00
2024.71	GPS DODO		070528	48	38	59.00	14	26	25.00	670.250	0	0	-3013	1	0	9807800.00	0	.00
2025	D DVOR.STHRA		880603	48	38	42.00	14	27	19.00	627.220	0	0	-3153	1	64	9807909.00	68	1.40
2026	H DVORISTE		880603	48	36	40.00	14	24	26.00	661.930	100	346	-3245	1	64	9807840.10	68	1.40
2026.01	H DVORISTE		880603	48	36	40.00	14	24	26.00	661.940	100	346	-3245	1	64	9807840.10	68	1.40
...																		

B Struktura databáze

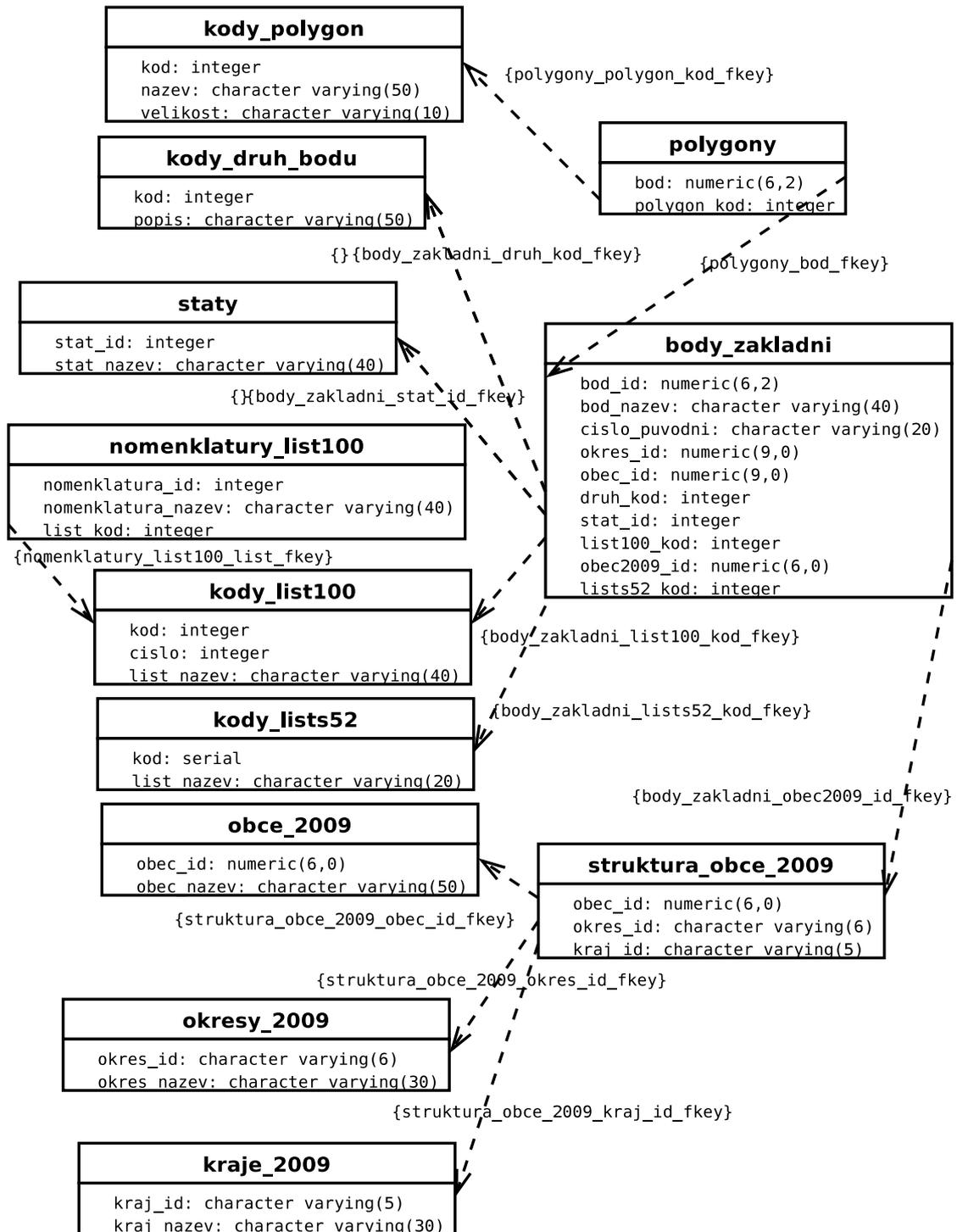
B.1 Struktura uložení tíhových bodů



Obrázek B.1: Struktura uložení bodů 1

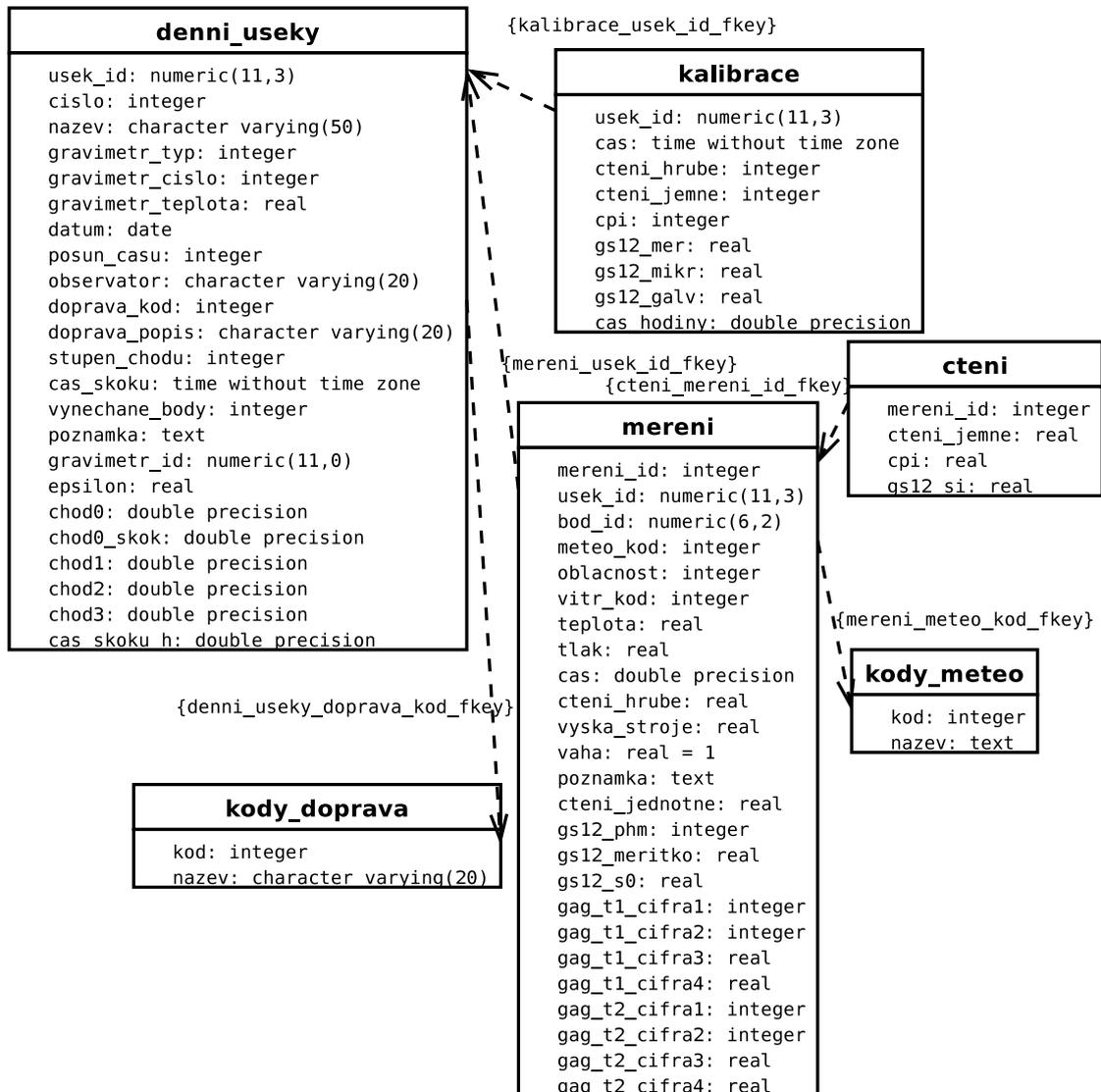


Obrázek B.2: Struktura uložení bodů 2



Obrázek B.3: Struktura uložení bodů 3

B.2 Struktura uložení denních úseků



Obrázek B.4: Struktura uložení denních úseků

B.3 Struktura uložení gravimetrů

gravimetry	gravimetry (pokračování)
gravimetr_id: numeric(11,0) typ: integer cislo: integer tabulka_cislo: integer nazev: character varying(20) uzivatel: character varying(30) datum_ulozeni: date datum_urceni: date zacatek_platnosti: date konec_platnosti: date zpusob_urceni: integer vzor_zapisniku: integer kod_vypoctu: integer stupnice1_dolni: integer stupnice1_horni: integer stupnice2_dolni: integer stupnice2_horni: integer stupnice3_dolni: integer stupnice3_horni: integer termostat: integer teplota: real svisla_vzdalenost: real epsilon: double precision t0: double precision zeta: double precision gag_konstanta1: double precision gag_konstanta2: double precision gag_konstanta3: double precision gag_konstanta4: double precision gag_konstanta5: double precision gag_konstanta6: double precision gs12_runova_chyba: double precision gs12_hmota1: double precision gs12_hmota2: double precision gs12_hmota3: double precision gs12_hmota4: double precision gs12_hmota5: double precision vaha: real = 1	a_jemne: double precision b_jemne: double precision c_jemne: double precision d_jemne: double precision e_jemne: double precision amplituda1_jemne: double precision amplituda2_jemne: double precision amplituda3_jemne: double precision amplituda4_jemne: double precision amplituda5_jemne: double precision amplituda6_jemne: double precision faze1_jemne: double precision faze2_jemne: double precision faze3_jemne: double precision faze4_jemne: double precision faze5_jemne: double precision faze6_jemne: double precision perioda1_jemne: double precision perioda2_jemne: double precision perioda3_jemne: double precision perioda4_jemne: double precision perioda5_jemne: double precision perioda6_jemne: double precision a_hrube: double precision b_hrube: double precision c_hrube: double precision d_hrube: double precision e_hrube: double precision amplituda1_hrube: double precision amplituda2_hrube: double precision amplituda3_hrube: double precision amplituda4_hrube: double precision amplituda5_hrube: double precision amplituda6_hrube: double precision faze1_hrube: double precision faze2_hrube: double precision faze3_hrube: double precision faze4_hrube: double precision faze5_hrube: double precision faze6_hrube: double precision perioda1_hrube: double precision perioda2_hrube: double precision perioda3_hrube: double precision perioda4_hrube: double precision perioda5_hrube: double precision perioda6_hrube: double precision

Obrázek B.5: Struktura uložení gravimetrů 1

gravimetry (pokračování)

```

amp_sin1_jemne: double precision
amp_cos1_jemne: double precision
freq1_jemne: double precision
amp_sin2_jemne: double precision
amp_cos2_jemne: double precision
freq2_jemne: double precision
amp_sin3_jemne: double precision
amp_cos3_jemne: double precision
freq3_jemne: double precision
amp_sin4_jemne: double precision
amp_cos4_jemne: double precision
freq4_jemne: double precision
amp_sin5_jemne: double precision
amp_cos5_jemne: double precision
freq5_jemne: double precision
amp_sin6_jemne: double precision
amp_cos6_jemne: double precision
freq6_jemne: double precision
amp_sin1_hrube: double precision
amp_cos1_hrube: double precision
freq1_hrube: double precision
amp_sin2_hrube: double precision
amp_cos2_hrube: double precision
freq2_hrube: double precision
amp_sin3_hrube: double precision
amp_cos3_hrube: double precision
freq3_hrube: double precision
amp_sin4_hrube: double precision
amp_cos4_hrube: double precision
freq4_hrube: double precision
amp_sin5_hrube: double precision
amp_cos5_hrube: double precision
freq5_hrube: double precision
amp_sin6_hrube: double precision
amp_cos6_hrube: double precision
freq6 hrube: double precision
    
```

Obrázek B.6: Struktura uložení gravimetrů 2

C Naplnění databáze

C.1 Skript pro program RegisterImport

```
#!/bin/bash

# skript pro hromadne nacteni zapisniku do databaze
# prvni argument je cesta k adresary a podadresarum, ve kterych se hledaji soubory
# adjustovanych zapisniku
# druhy argument je cesta k adresary a podadresarum, ve kterych se hledaji soubory
# rovnice oprav
# treti argument je cesta k souboru s katalogem skupin, cesta je relativni od
# pracovniho adresare
# v aktualnim adresari je potreba mit soubory RegisterImport.java, manifest.mf a
# slozku /lib s knihovnou
# postgresql-8.3-603.jdbc3.jar
# je take potreba vytvorit ssh tunel portu 5432 na 3333
# ssh -L 3333:127.0.0.1:5432 honza@gravimetry.nesvadba.eu
if [ $1 -a -d $1 ] && [ -r $3 ]; then
  gp=$(pwd)/$3
  for file in $(find $1 -depth -regextype posix-egrep -type f -iregex '^.*(AZ)
  {1}|(az){1}|([0-9]{6})\.[0-9]{3}$')
  do
    nazev="${file: -10:10}"
    if [ $2 -a -d $2 ]; then
      adresar=$2
    else
      adresar=$(pwd)
      echo "Nebyl zadán název adresare s rovníci oprav, bude prohledávan
      adresar $adresar včetně podadresaru."
    fi
    regexRO="^.*(r|R)([0-3]{1})($nazev: 0:6)\.($nazev: 7:10)$"
    ro=$(find "$adresar" -depth -regextype posix-egrep -type f -iregex "$
    $regexRO" | head -n 1)
    if [ ! $ro ] || [ ! -r $ro ]; then
      echo "Žádný soubor rovníce oprav odpovídající masce $regexRO nebyl
      nalezen"
      echo "Data budou importována do databáze bez informací o stupni
      polynomu, vynechaných bodech a skoku při zpracování."
      java -jar RegisterImport.jar -r $file -gp $gp
    elif [ ! -e RegisterImport.jar -a ! -x RegisterImport.jar ]; then
      echo "V aktuálním adresari $(pwd) chybí program RegisterImport."
    else
      echo "java -jar RegisterImport.jar -r $file -e $ro -gp $gp"
      java -jar RegisterImport.jar -r $file -e $ro -gp $gp
    fi
  done
else
  echo "Neplatný název adresare s adjustovanými záznamy nebo souboru s
  katalogem skupin."
fi
```

D PHP webové rozhraní

D.1 Snímky obrazovky

Databáze Gravimetrie ZÚ

<p>Body</p> <ul style="list-style-type: none"> • Základní údaje • Stav • Údaje • Tíže • Místopis • Vložit nový <p>Denní úseky</p> <ul style="list-style-type: none"> • Zobrazit údaje • Vložit nový 	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> Vyhledávání bodů Číslo: <input type="text"/> Název: <input type="text"/> <input type="button" value="Hledat"/> </div> <div style="border: 1px solid #ccc; padding: 5px;"> Vyhledávání úseků Číslo: <input type="text"/> Název: <input type="text"/> <input type="button" value="Hledat"/> </div>
---	--

Obrázek D.1: Úvodní obrazovka

2019.00

Název bodu:

Původní číslo bodu:

Druh bodu:

Stát:

Mapový list 1:100 000

Mapový list S-52:

Polygony :

- Geodynamický polygon Lišov | malý
- Geodynamická síť České republiky | střední
- U E G N | velký
- Karpatský polygon | velký

Ctrl a levé tlačítko myši pro výběr více možností.

Obec :

Síť 1957:

Síť 1964:

Síť 1995:

Základna:

Obrázek D.2: Základní údaje

2019.00 Chotyčany

Stabilizoval:

Rok stabilizace:

Rok zrušení:

Datum poslední návštěvy:

Popis stabilizace:

Celkový stav:

Vyhledatelnost bodu:

Popis příjezdu:

Místopis (popis, náčrt):

Příjezd:

Umístění bodu:

Kvalita stabilizace:

Ochranné zařízení:

Obrázek D.3: Stav část 1.

Ochranné zařízení:

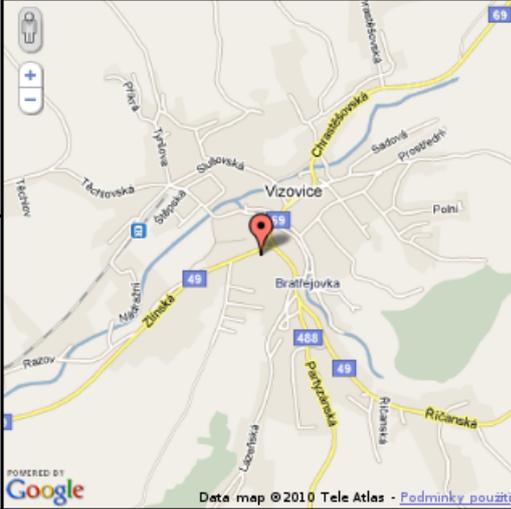
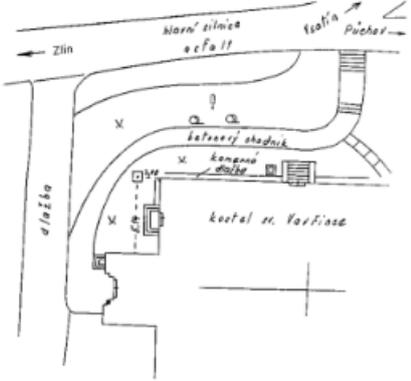
Nová stabilizace:

Nová tabulka:

Nová ochranná tyč:

Poznámka:

Obrázek D.4: Stav část 2.

Místopis tíhového bodu České gravimetrické sítě	
<p>Tíhový bod: Vizovice Kraj: Zlínský kraj List mapy: 19 Zlín Obec: Vizovice φ: 49° 13' 8.170" λ: 17° 51' 4.626" H = 302.69 m g = 9809041.800 μms^{-2} $\sigma = 0.098 \mu\text{ms}^{-2}$</p>	<p>Číslo bodu: 3536.01 Řád bodu: Bod 2. řádu</p>  <p style="font-size: small;">Data map © 2010 Tele Atlas - Podmínky použití</p>
<p>Místopisný popis: u severozápadního rohu kostela sv. Vavřince. Příjezd: Výškové připojení: Lfh Vsetín - Zádveřice, Č 37, kostel sv. Vavřince.</p>	
<p>Stabilizace: typ I, betonová deska, výstražná tabulka</p>	
<p>Bartoš, 1960</p>	
<p>Místopisný náčrt :</p> 	
<p>Poznámka:</p>	

Obrázek D.5: Místopis tíhového bodu

2012.01 Sulice

Souřadnice v systému ETRS-89:

φ : ° ' "

λ : ° ' "

Střední chyba [m]: Nadbytečná měření:

Chyba transformace souřadnic do ETRS-89 [m]:

Popis souřadnic:

Souřadnice v systému S-52:

φ : 49°55'38" λ : 14°34'20"

Výška bodu [m]:

Druh výšky:

Vertikální gradient [E]:

Typ vertikálního gradientu:

Horizontální gradient [E]:

Azimut [°]:

Topografická korekce:

Vliv stavby:

Obrázek D.6: Údaje

Předchozí Následující

Číslo		Název												
19890101.080		HUMPOLEC JIHLAVA												
Grav. 10200174024		Datum			Observátor			Dopravní prostředek						
Typ	Číslo	t [C°]	Datum		T	Kód Název			Popis					
2	174	35	21.11.1989		1	TRAKAL			1 auto			VK ACK 1347		
Počet měření 5		Počet odečtení 3												
1. bod	met	obl	vitr	t	p	čas		hr.čt.	jm.čt	jm.čt	jm.čt	h	váha	
3354.03	1	100	1	2.7	95.2	10	56	772	220.7	220.8	220.8	45	1	
2. bod	met	obl	vitr	t	p	čas		hr.čt.	jm.čt	jm.čt	jm.čt	h	váha	
3358.02	1	100	1	1.5	95.4	11	31	772	232.3	232.3	232.2	45	1	
3. bod	met	obl	vitr	t	p	čas		hr.čt.	jm.čt	jm.čt	jm.čt	h	váha	
3354.03	1	100	1	2.9	95.1	12	24	772	219.9	220	219.9	45	1	
4. bod	met	obl	vitr	t	p	čas		hr.čt.	jm.čt	jm.čt	jm.čt	h	váha	
3358.02	1	100	1	1.9	95.3	12	58	772	231.7	231.8	231.7	45	1	
5. bod	met	obl	vitr	t	p	čas		hr.čt.	jm.čt	jm.čt	jm.čt	h	váha	
3354.03	1	100	1	2.2	95.1	13	33	772	219.5	219.5	219.5	45	1	

Obrázek D.7: Zobrazit údaje

E Výstupy z databáze

E.1 XML výstupy pro Gravimzung

E.1.1 query_point.sql

```

SELECT XMLELEMENT (name "station", XMLATTRIBUTES(BZ.bod_id AS "id", BZ.bod_nazev AS
    "name")),
-- tize
    XMLELEMENT (name "neuron", XMLATTRIBUTES(BZ.bod_id || ':gravity' AS "id"),
    XMLELEMENT (name "inp", XMLATTRIBUTES('epoch' AS "id"),
        XMLELEMENT(name "pow", XMLATTRIBUTES('0' AS "order", 'g' AS "id")),
            COALESCE(BT.tize_1995, BT.priblizna, NULL)),
        XMLELEMENT (name "inp", XMLATTRIBUTES('epoch' AS "id"),
            XMLELEMENT(name "pow", XMLATTRIBUTES('1' AS "order", 'gt' AS "id")), '0.0'
        ),
        XMLELEMENT (name "inp", XMLATTRIBUTES('epoch' AS "id"),
            XMLELEMENT(name "sin", XMLATTRIBUTES('1' AS "freq", 'gst' AS "id")), '0.0'
        ),
        XMLELEMENT (name "inp", XMLATTRIBUTES('epoch' AS "id"),
            XMLELEMENT(name "cos", XMLATTRIBUTES('1' AS "freq", 'gct' AS "id")), '0.0'
        ),
-- souradnice a vyska
    XMLELEMENT (name "inp", XMLATTRIBUTES('epoch' AS "id"),
        XMLELEMENT(name "cwte", XMLATTRIBUTES(BS.sirka_etr89 AS "B", BS.
            delka_etr89 AS "L", BV.vyska AS "H")), '1.0'),
    XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':sensor_hoffset'
        AS "id"),
        XMLELEMENT(name "pow", XMLATTRIBUTES('1' AS "order", 'ghgrad' AS "id")),
            COALESCE(0.0001*BG.vertikalni, -0.3086))))
FROM body_zakladni BZ
JOIN body_tize BT
ON BT.bod_id = BZ.bod_id
JOIN body_souradnice BS
ON BS.bod_id = BZ.bod_id
JOIN body_vyska BV
ON BV.bod_id = BZ.bod_id
JOIN body_gradient BG
ON BG.bod_id = BZ.bod_id
JOIN useky_mereni M
ON M.bod_id = BZ.bod_id
JOIN useky_denni_useky U
ON U.usek_id = M.usek_id
JOIN gravimetry_gravimetry G
ON G.gravimetr_id = U.gravimetr_id
WHERE U.usek_id = 19910402.002
GROUP BY BZ.bod_id, BZ.bod_nazev, BT.tize_1995, BT.priblizna, BS.sirka_etr89, BS.
    delka_etr89, BV.vyska, BG.vertikalni, G.gravimetr_id;

```

Výsledek dotazu query_point.sql

```

<station id="271.00" name="Post&#x159;i&#x17E;&#xED;n">
  <neuron id="271.00:gravity">
    <inp id="epoch"><pow order="0" id="g" />9810382.170</inp>
    <inp id="epoch"><pow order="1" id="gt" />0.0</inp>
    <inp id="epoch"><sin freq="1" id="gst" />0.0</inp>
    <inp id="epoch"><cos freq="1" id="gct" />0.0</inp>

```

```

    <inp id="epoch"><cwte B=" 50.231621" L=" 14.387353" H=" 234.07" />1.0</inp>
    <inp id="10600432002:sensor_hoffset"><pow order="1" id="ghgrad" />-0.31</inp>
  </neuron>
</station>
<station id="2009.00" name="Stra&#x161;kov">
  <neuron id="2009.00:gravity">
    <inp id="epoch"><pow order="0" id="g" />9810536.960</inp>
    <inp id="epoch"><pow order="1" id="gt" />0.0</inp>
    <inp id="epoch"><sin freq="1" id="gst" />0.0</inp>
    <inp id="epoch"><cos freq="1" id="gct" />0.0</inp>
    <inp id="epoch"><cwte B=" 50.364812" L=" 14.252261" H=" 207.24" />1.0</inp>
    <inp id="10600432002:sensor_hoffset"><pow order="1" id="ghgrad" />-0.302</inp>
  </neuron>
</station>
<station id="2007.00" name="Litom&#x11b;&#x159;ice">
  <neuron id="2007.00:gravity">
    <inp id="epoch"><pow order="0" id="g" />9810788.250</inp>
    <inp id="epoch"><pow order="1" id="gt" />0.0</inp>
    <inp id="epoch"><sin freq="1" id="gst" />0.0</inp>
    <inp id="epoch"><cos freq="1" id="gct" />0.0</inp>
    <inp id="epoch"><cwte B=" 50.518791" L=" 14.141066" H=" 146.86" />1.0</inp>
    <inp id="10600432002:sensor_hoffset"><pow order="1" id="ghgrad" />-0.3035</inp>
  </neuron>
</station>
<station id="272.01" name="Kunratice">
  <neuron id="272.01:gravity">
    <inp id="epoch"><pow order="0" id="g" />9809977.100</inp>
    <inp id="epoch"><pow order="1" id="gt" />0.0</inp>
    <inp id="epoch"><sin freq="1" id="gst" />0.0</inp>
    <inp id="epoch"><cos freq="1" id="gct" />0.0</inp>
    <inp id="epoch"><cwte B=" 50.015875" L=" 14.469508" H=" 289.46" />1.0</inp>
    <inp id="10600432002:sensor_hoffset"><pow order="1" id="ghgrad" />-0.301</inp>
  </neuron>
</station>
<station id="2010.00" name="Praha">
  <neuron id="2010.00:gravity">
    <inp id="epoch"><pow order="0" id="g" />9810180.010</inp>
    <inp id="epoch"><pow order="1" id="gt" />0.0</inp>
    <inp id="epoch"><sin freq="1" id="gst" />0.0</inp>
    <inp id="epoch"><cos freq="1" id="gct" />0.0</inp>
    <inp id="epoch"><cwte B=" 50.069207" L=" 14.42519" H=" 203.3" />1.0</inp>
    <inp id="10600432002:sensor_hoffset"><pow order="1" id="ghgrad" />-0.282</inp>
  </neuron>
</station>
(5 rows)

```

E.1.2 query_section.sql

```

SELECT XMLELEMENT (name "du", XMLATTRIBUTES(U.usek_id AS "id", U.nazev AS "name", U
.observator AS "observer", U.doprava_kod AS "transport_code", U.doprava_popis
AS "transport_description"),
XMLELEMENT (name "instrument", XMLATTRIBUTES(G.gravimetr_id AS "id", G.
nazev AS "name", G.uzivatel AS "owner", G.konec_platnosti AS "valid"),
-- staticke retezce
XMLELEMENT (name "sensor", XMLATTRIBUTES(G.gravimetr_id || ':clock' AS "id
")),
XMLELEMENT (name "sensor", XMLATTRIBUTES(G.gravimetr_id || ':sensor_height
' AS "id")),

```

```

XMLELEMENT (name "sensor", XMLATTRIBUTES(G.gravimetr_id || ':CPI' AS "id")
),
XMLELEMENT (name "sensor", XMLATTRIBUTES(G.gravimetr_id || ':z' AS "id")),
-- kalibracni polynom sroubu
XMLELEMENT (name "neuron", XMLATTRIBUTES(G.gravimetr_id || ':
precalibration' AS "id")),
-- absolutni clen
XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':z' AS "id")),
XMLELEMENT(name "pow", XMLATTRIBUTES('0' AS "order", 'pa' AS "id")), G.
a_jemne),
-- linearni clen
XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':z' AS "id")),
XMLELEMENT(name "pow", XMLATTRIBUTES('1' AS "order", 'pl' AS "id")), G.
b_jemne),
-- kvadraticky clen
XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':z' AS "id")),
XMLELEMENT(name "pow", XMLATTRIBUTES('2' AS "order", 'pq' AS "id")), G.
c_jemne),
-- kubicky clen
XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':z' AS "id")),
XMLELEMENT(name "pow", XMLATTRIBUTES('3' AS "order", 'pc' AS "id")), G.
d_jemne),
-- bikubicky clen
XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':z' AS "id")),
XMLELEMENT(name "pow", XMLATTRIBUTES('4' AS "order", 'pbq' AS "id")), G.
e_jemne),
-- periodicka chyba 1
-- cos
XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':z' AS "id")),
XMLELEMENT(name "cos", XMLATTRIBUTES(COALESCE(G.freq1_jemne, 0) AS "freq"
, 'pfc1' AS "id")), COALESCE(G.amp_cos1_jemne, 0)),
-- sin
XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':z' AS "id")),
XMLELEMENT(name "sin", XMLATTRIBUTES(COALESCE(G.freq1_jemne, 0) AS "freq"
, 'pfs1' AS "id")), COALESCE(G.amp_sin1_jemne, 0)),
-- periodicka chyba 2
-- cos
XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':z' AS "id")),
XMLELEMENT(name "cos", XMLATTRIBUTES(COALESCE(G.freq2_jemne, 0) AS "freq"
, 'pfc2' AS "id")), COALESCE(G.amp_cos2_jemne, 0)),
-- sin
XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':z' AS "id")),
XMLELEMENT(name "sin", XMLATTRIBUTES(COALESCE(G.freq2_jemne, 0) AS "freq"
, 'pfs2' AS "id")), COALESCE(G.amp_sin2_jemne, 0)),
-- periodicka chyba 3
-- cos
XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':z' AS "id")),
XMLELEMENT(name "cos", XMLATTRIBUTES(COALESCE(G.freq3_jemne, 0) AS "freq"
, 'pfc3' AS "id")), COALESCE(G.amp_cos3_jemne, 0)),
-- sin
XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':z' AS "id")),
XMLELEMENT(name "sin", XMLATTRIBUTES(COALESCE(G.freq3_jemne, 0) AS "freq"
, 'pfs3' AS "id")), COALESCE(G.amp_sin3_jemne, 0)),
-- periodicka chyba 4
-- cos
XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':z' AS "id")),
XMLELEMENT(name "cos", XMLATTRIBUTES(COALESCE(G.freq4_jemne, 0) AS "freq"
, 'pfc4' AS "id")), COALESCE(G.amp_cos4_jemne, 0)),

```

```

-- sin
XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':z' AS "id"),
  XMLELEMENT(name "sin", XMLATTRIBUTES(COALESCE(G.freq4_jemne, 0) AS "freq",
    , 'pfs4' AS "id")), COALESCE(G.amp_sin4_jemne, 0)),
-- neuron drift
XMLELEMENT (name "neuron", XMLATTRIBUTES(G.gravimetr_id || ':drift' AS "id"),
-- linearni
  XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':time' AS "id"),
    XMLELEMENT(name "pow", XMLATTRIBUTES('1' AS "order", 'd1' AS "id")), U.
      chod1),
-- kvadraticky
  XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':time' AS "id"),
    XMLELEMENT(name "pow", XMLATTRIBUTES('2' AS "order", 'dq' AS "id")), U.
      chod2),
-- kubicky
  XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':time' AS "id"),
    XMLELEMENT(name "pow", XMLATTRIBUTES('3' AS "order", 'dc' AS "id")), U.
      chod3),
-- skok
  XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':time' AS "id"),
    XMLELEMENT(name "hstep", XMLATTRIBUTES(COALESCE(U.cas_skoku_h, 0) AS "
      offset", 'step' AS "id")), '0.1')),
-- neuron height offset
  XMLELEMENT (name "neuron", XMLATTRIBUTES(G.gravimetr_id || ':sensor_hoffset' AS
    "id"),
-- linearni
  XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':sensor_height'
    AS "id"),
    XMLELEMENT(name "pow", XMLATTRIBUTES('1' AS "order", 'hol' AS "id")), G.
      svisla_vzdalenost)),
-- neuron kalibrace CPI
  XMLELEMENT (name "neuron", XMLATTRIBUTES(G.gravimetr_id || ':calCPI' AS "id"),
-- epsilon z gravimetru
  XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':CPI' AS "id"),
    XMLELEMENT(name "linhstep", XMLATTRIBUTES('0.0' AS "offset", 'xpcpi' AS "
      id")), COALESCE(G.epsilon, 0.0)),
-- epsilon z dennich useku
  XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':CPI' AS "id"),
    XMLELEMENT(name "linhstepi", XMLATTRIBUTES('0.0' AS "offset", 'xmcp' AS "
      id")), COALESCE(U.epsilon, 0.0))),
-- neuron sloucení barometrického efektu, chodu a kalibrovaného ctení
  XMLELEMENT (name "neuron", XMLATTRIBUTES(G.gravimetr_id || ':relg' AS "id"),
-- tlak
  XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':pressure' AS "
    id"),
    XMLELEMENT(name "pow", XMLATTRIBUTES('1' AS "order", 'barcoef' AS "id")), '
      0.01'),
-- drift
  XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':drift' AS "id")
    ,
    XMLELEMENT(name "pow", XMLATTRIBUTES('1' AS "order", 'drift' AS "id")), '
      1.0')),
-- neuron epoch
  XMLELEMENT (name "neuron", XMLATTRIBUTES('epoch' AS "id"),
-- casovy ofset
  XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':clock' AS "id")
    ,
    XMLELEMENT(name "pow", XMLATTRIBUTES('0' AS "order")), COALESCE(U.
      posun_casu*0.04166666666666667, 0)),

```

```

-- datum z hlavicky ve dnech
      XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':clock' AS "id")
        ,
      XMLELEMENT(name "pow", XMLATTRIBUTES('0' AS "order")), TO_CHAR(U.datum, 'J
        ')),
-- korekce hodin na dny
      XMLELEMENT (name "inp", XMLATTRIBUTES(G.gravimetr_id || ':clock' AS "id")
        ,
      XMLELEMENT(name "pow", XMLATTRIBUTES('1' AS "order")),
        0.04166666666666667)))
FROM gravimetry.gravimetry G
JOIN useky.denni_useky U
ON U.gravimetr_id = G.gravimetr_id
WHERE U.usek_id = 20070102.021;

```

Výsledek dotazu query_section.sql

```

<du id="20070102.021" name="HOCHKAR LINE" observer="LEDERER" transport_code="1"
  transport_description="VWC 4A7 4983">
  <instrument id="10601068001" name="LA COSTE G" owner="ZU PRAHA" valid="2001-07-12
    ">
    <sensor id="10601068001:clock"/>
    <sensor id="10601068001:sensor_height"/>
    <sensor id="10601068001:CPI"/>
    <sensor id="10601068001:z"/>
    <neuron id="10601068001:precalibration">
      <inp id="10601068001:z"><pow order="0" id="pa"/>5.8795892</inp>
      <inp id="10601068001:z"><pow order="1" id="p1"/>10.1147338</inp>
      <inp id="10601068001:z"><pow order="2" id="pq"/>1.9468127e-05</inp>
      <inp id="10601068001:z"><pow order="3" id="pc"/>-2.2152698e-09</inp>
      <inp id="10601068001:z"><pow order="4" id="pbq"/>8.2192228e-14</inp>
      <inp id="10601068001:z"><cos freq="6.28318530717959" id="pfc1"/>
        6.28318530717959e-40</inp>
      <inp id="10601068001:z"><sin freq="6.28318530717959" id="pfs1"/>1e-20</inp>
      <inp id="10601068001:z"><cos freq="6.28318530717959" id="pfc2"/>0</inp>
      <inp id="10601068001:z"><sin freq="6.28318530717959" id="pfs2"/>0</inp>
      <inp id="10601068001:z"><cos freq="6.28318530717959" id="pfc3"/>0</inp>
      <inp id="10601068001:z"><sin freq="6.28318530717959" id="pfs3"/>0</inp>
      <inp id="10601068001:z"><cos freq="6.28318530717959" id="pfc4"/>0</inp>
      <inp id="10601068001:z"><sin freq="6.28318530717959" id="pfs4"/>0</inp>
    </neuron>
    <neuron id="10601068001:drift">
      <inp id="10601068001:time"><pow order="1" id="dl"/></inp>
      <inp id="10601068001:time"><pow order="2" id="dq"/></inp>
      <inp id="10601068001:time"><pow order="3" id="dc"/></inp>
      <inp id="10601068001:time"><hstep offset="11.95" id="step"/>0.1</inp>
    </neuron>
    <neuron id="10601068001:sensor_hoffset">
      <inp id="10601068001:sensor_height"><pow order="1" id="hol"/>-170</inp>
    </neuron>
    <neuron id="10601068001:calCPI">
      <inp id="10601068001:CPI"><linhstep offset="0.0" id="xpcpi"/>0</inp>
      <inp id="10601068001:CPI"><linhstepi offset="0.0" id="xmcpi"/>0</inp>
    </neuron>
    <neuron id="10601068001:relg">
      <inp id="10601068001:pressure"><pow order="1" id="barcoef"/>0.01</inp>
      <inp id="10601068001:drift"><pow order="1" id="drift"/>1.0</inp>
    </neuron>
  </instrument>

```

```

<neuron id="epoch">
  <inp id="10601068001:clock"><pow order="0"/>0.08333333333333334</inp>
  <inp id="10601068001:clock"><pow order="0"/>2454240</inp>
  <inp id="10601068001:clock"><pow order="1"/>0.04166666666666667</inp>
</neuron>
</du>
(1 row)

```

E.1.3 query_calibration.sql

```

SELECT  XMLELEMENT (name "obs", XMLATTRIBUTES(U.usek_id AS "id"),
-- neuron epoch
  XMLELEMENT (name "neuron", XMLATTRIBUTES('relgrav' AS "id"),
-- relg
    XMLELEMENT (name "inp", XMLATTRIBUTES(U.gravimetr_id || ':relg' AS "id"),
      XMLELEMENT(name "pow", XMLATTRIBUTES('1' AS "order")), '-1.0'),
-- bod a gravity
    XMLELEMENT (name "inp", XMLATTRIBUTES('0.' || to_char(K.cas_hodiny, '
      FM00V99') || ':gravity' AS "id"),
      XMLELEMENT(name "pow", XMLATTRIBUTES('1' AS "order")), '1.0' ),
    XMLELEMENT (name "clock", K.cas_hodiny ),
    XMLELEMENT (name "counter", K.cteni_hrube), XMLAGG(XMLCONCAT(
    XMLELEMENT (name "CPI", K.cpi),
    XMLELEMENT (name "screw", K.cteni_jemne))))
FROM useky.denni_useky U
JOIN useky.kalibrace K
ON K.usek_id = U.usek_id
WHERE U.usek_id = 20090102.002
GROUP BY U.usek_id , U.gravimetr_id , K.cas_hodiny , K.cteni_hrube;

```

Výsledek dotazu query_calibration.sql

```

<obs id="20090102.002"><neuron id="relgrav"><inp id="10601068001:relg"><pow order=
  "1"/>-1.0</inp><inp id="0.1488:gravity"><pow order="1"/>1.0</inp></neuron><
  clock>14.88333333333333</clock><counter>4369</counter><CPI>1</CPI><screw>776</
  screw><CPI>287</CPI><screw>676</screw><CPI>-248</CPI><screw>876</screw><CPI>5<
  /CPI><screw>776</screw></obs>
<obs id="20090102.002"><neuron id="relgrav"><inp id="10601068001:relg"><pow order=
  "1"/>-1.0</inp><inp id="0.1038:gravity"><pow order="1"/>1.0</inp></neuron><
  clock>10.38333333333333</clock><counter>4369</counter><CPI>-6</CPI><screw>761</
  screw><CPI>272</CPI><screw>661</screw><CPI>-293</CPI><screw>861</screw><CPI>
  -24</CPI><screw>761</screw></obs>
(2 rows)

```

E.1.4 query_observation.sql

```

SELECT  XMLELEMENT (name "obs", XMLATTRIBUTES(M.mereni_id AS "id"),
-- neuron epoch
  XMLELEMENT (name "neuron", XMLATTRIBUTES('relgrav' AS "id"),
-- relg
    XMLELEMENT (name "inp", XMLATTRIBUTES(U.gravimetr_id || ':relg' AS "id"),
      XMLELEMENT(name "pow", XMLATTRIBUTES('1' AS "order")), '-1.0' ),
-- bod a gravity
    XMLELEMENT (name "inp", XMLATTRIBUTES(M.bod_id || ':gravity' AS "id"),
      XMLELEMENT(name "pow", XMLATTRIBUTES('1' AS "order")), '1.0' ),
    XMLELEMENT (name "clock", M.cas ),
    XMLELEMENT (name "sensor_height", M.vyska_stroje*10.0),
    XMLELEMENT (name "pressure", M.tlak),

```

```

      XMLELEMENT (name "counter", M.cteni_hrube), XMLAGG(XMLCONCAT(
          XMLELEMENT (name "CPI", C.cpi),
          XMLELEMENT (name "screw", C.cteni_jemne))))
FROM useky.denni_useky U
JOIN useky.mereni M
ON M.usek_id = U.usek_id
JOIN useky.cteni C
ON C.mereni_id = M.mereni_id
WHERE U.usek_id = 20070102.021
GROUP BY M.mereni_id, U.gravimetr_id, M.bod_id, M.cas, M.vyska_stroje, M.tlak, M.
cteni_hrube
ORDER BY M.cas ASC;

```

Ukázka z výsledku dotazu query_observation.sql

```

<obs id="9565">
  <neuron id="relgrav">
    <inp id="10601068001:relg"><pow order="1"/>-1.0</inp>
    <inp id="8091.00:gravity"><pow order="1"/>1.0</inp>
  </neuron>
  <clock>9.13333333333333</clock>
  <sensor_height>272.999992370605</sensor_height>
  <pressure>95.7</pressure>
  <counter>4159</counter><CPI>2</CPI><screw>720</screw><CPI>9</CPI><screw>720</
  screw><CPI>8</CPI><screw>720</screw>
</obs>
<obs id="9566">
  <neuron id="relgrav">
    <inp id="10601068001:relg"><pow order="1"/>-1.0</inp>
    <inp id="8092.00:gravity"><pow order="1"/>1.0</inp>
  </neuron>
  <clock>9.61666666666667</clock>
  <sensor_height>273.999996185303</sensor_height>
  <pressure>93.8</pressure>
  <counter>4119</counter><CPI>24</CPI><screw>270</screw><CPI>-2</CPI><screw>260</
  screw><CPI>0</CPI><screw>260</screw>
</obs>
<obs id="9567">
  <neuron id="relgrav">
    <inp id="10601068001:relg"><pow order="1"/>-1.0</inp>
    <inp id="8093.00:gravity"><pow order="1"/>1.0</inp>
  </neuron>
  <clock>10</clock>
  <sensor_height>272.000007629395</sensor_height>
  <pressure>89.1</pressure>
  <counter>4035</counter><CPI>10</CPI><screw>117</screw><CPI>-12</CPI><screw>110</
  screw><CPI>-10</CPI><screw>110</screw>
</obs>
<obs id="9568">
  ...

```

E.2 Výstupy pro program Surfer

E.2.1 Funkce get_points(point_id)

```

DROP FUNCTION public.get_points(point_id numeric(6,2));
CREATE OR REPLACE FUNCTION public.get_points (point_id numeric(6,2)) RETURNS text
AS $$

```

```

DECLARE
point record;
text_output text := '';
BEGIN
FOR point IN EXECUTE 'SELECT CAST(BZ.bod_id AS text), COALESCE(CAST(BS.
  delka_etr89 AS text), ''not found'') AS delka_etr89, ' || ' COALESCE(
  CAST(BS.sirka_etr89 AS text), ''notfound'') AS sirka_etr89, ' || '
  COALESCE (BSI.sit95_kod, ''99'') AS trida '
  || ' FROM body_zakladni BZ'
  || ' JOIN body_souradnice BS'
  || ' ON BS.bod_id = BZ.bod_id'
  || ' JOIN body_site BSI'
  || ' ON BSI.bod_id = BZ.bod_id'
  || ' WHERE BZ.bod_id = ' || point.id
  || ' GROUP BY BZ.bod_id, delka_etr89, sirka_etr89, trida '
  || ' ORDER BY BZ.bod_id;'
LOOP
  text_output := text_output || point.bod_id || E'\t' ;
  text_output := text_output || point.delka_etr89 || E'\t' || point.
    sirka_etr89;
  text_output := text_output || E'\t' || point.trida;
END LOOP;
RETURN text_output;
END;
$$ LANGUAGE plpgsql;

```

E.2.2 Funkce get_polygons(usek_id)

```

DROP FUNCTION public.get_polygons(section_id numeric(11,3));
CREATE OR REPLACE FUNCTION public.get_polygons (section_id numeric(11,3)) RETURNS
text AS $$
DECLARE
polygon record;
text_output text;
pocet_bodu integer;
BEGIN
  SELECT SUM(V.pocet)
  INTO pocet_bodu
  FROM (
    SELECT (COUNT( DISTINCT CAST(BS.sirka_etr89 AS text) || CAST( BS.
      delka_etr89 AS text))) AS pocet
    FROM useky.merani M
    JOIN useky.denni_useky U
    ON U.usek_id = M.usek_id
    JOIN body_souradnice BS
    ON BS.bod_id = M.bod_id
    WHERE U.usek_id = section_id
    GROUP BY BS.sirka_etr89, BS.delka_etr89) V;
  IF pocet_bodu > 1 THEN
text_output := CAST(pocet_bodu AS text) || E'\n';
FOR polygon IN EXECUTE 'SELECT COALESCE(CAST(BS.delka_etr89 AS text), ''not
  found'') AS delka_etr89, ' || ' COALESCE( CAST(BS.sirka_etr89 AS text
  ), ''notfound'') AS sirka_etr89 '
  || ' FROM body_zakladni BZ'
  || ' JOIN body_souradnice BS'
  || ' ON BS.bod_id = BZ.bod_id'
  || ' JOIN useky.merani M'
  || ' ON M.bod_id = BZ.bod_id'

```

```

|| ' JOIN useky.denni_useky U'
|| ' ON U.usek_id = M.usek_id '
|| ' WHERE U.usek_id = ' || section_id
|| ' GROUP BY BS.delka_etrs89 , BS.sirka_etrs89 '
|| ' ORDER BY MIN(M.cas);'
LOOP
    text_output := text_output || polygon.delka_etrs89 || E'\t' ||
    polygon.sirka_etrs89 || E'\n';
END LOOP;
RETURN text_output;
ELSE
RETURN NULL;
END IF;
END;
$$ LANGUAGE plpgsql;

```

E.3 Tvorba statistik

E.3.1 Skript get_index

```

#!/bin/bash
napoveda="\nSkript pro vytvoreni seznamu identifikatoru dennich useku v databazi
pro dane soubory rovnic oprav nebo adjustovanych zapisniku , ktere jsou obsazeny
v zadanem adresari nebo jeho podadresarich.\n\n"
napoveda=$napoveda"Prepinace - prvni argumenty skriptu je mozne zvolit bud -a nebo
-r: \n\n"
napoveda=$napoveda"\t-a\t adresar s adjustovanymi zapsinky \n"
napoveda=$napoveda"\t-r\t adresar s rovnicemi oprav \n"
napoveda=$napoveda"\nNasleduje zadani cesty k danemu adresari"
napoveda=$napoveda"Priklad pouziti: \n\n"
napoveda=$napoveda"\t./get_index.sh -r Rovopr/Klet/\n\n"
#napoveda=$napoveda"\t-h\t zobrazi tuto napovedu \n"

optstr=a:r:h
while getopts $optstr var
do
    case $var in
        a) maska="^.*((AZ){1}|(az){1}){1}([0-9]{6})\.[0-9]{3}$" ; adresar=$OPTARG ;;
        r) maska="^.*((R){1}|(r){1}){1}([1-3]{1}){1}([0-9]{6})\.[0-9]{3}$" ; adresar=$OPTARG
            ;;
        h) echo -ne $napoveda; exit ;;
    esac
done
shift $(( $OPTIND - 1 ))

if [ $adresar -a -d $adresar ]; then
    seznam=""
    for file in $(find $adresar -depth -regextype posix-egrep -type f -iregex
    $maska)
    do
        nazev="${file: -10:10}"
        rok="${nazev: 0:2}"
        if [ $rok -gt 30 ]; then
            seznam="$seznam 19$nazev,\n"
        elif [ $rok -ge 0 ]; then
            seznam="$seznam 20$nazev,\n"
        fi
    done

```

```

done
echo -ne $seznam | sort | uniq
else
    echo "Neplatny nazev adresare s adjustovanymi zapisniky."
fi

```

E.3.2 Četnost měření na bodech

SQL dotaz:

```

SELECT M.bod_id, COALESCE(BS.delka_etrs89, -1) AS delka, COALESCE(BS.sirka_etrs89,
    -1) AS sirka, '12' AS kod, COUNT(M.bod_id) AS pocet
FROM useky.denni-useky U
JOIN useky.mereni M
ON M.usek_id = U.usek_id
JOIN body_souradnice BS
ON BS.bod_id = M.bod_id
JOIN body_zakladni BZ
ON BZ.bod_id = M.bod_id
JOIN gravimetry.gravimetry G
ON G.gravimetr_id = U.gravimetr_id
WHERE BZ.stat_id = 1 AND G.typ NOT IN (1,11) AND EXTRACT (YEAR FROM(U.datum)) <
    1995 AND U.usek_id IN (
    19720101.001,
    ...
    20100102.059
)
GROUP BY M.bod_id, delka_etrs89, sirka_etrs89
ORDER BY pocet DESC;

```

Výsledek dotazu:

bod_id	delka	sirka	kod	pocet
2019.00	14.519811	49.067586	12	1083
272.01	14.469508	50.015875	12	959
271.00	14.387353	50.231621	12	735
2017.00	14.719577	49.226034	12	594
2020.00	14.470703	48.972465	12	593
2022.00	14.455623	48.86276	12	575
2009.00	14.252261	50.364812	12	564
2014.01	14.643605	49.638589	12	523
2008.00	14.151709	50.51097	12	523
...				

E.3.3 Počet změřených denních úseků s daným počtem bodů

SQL dotaz:

```

SELECT V.pocet, COUNT(V.usek_id) AS cetnost
FROM (
    SELECT U.usek_id AS usek_id, (COUNT( DISTINCT CAST(BS.sirka_etrs89 AS text) ||
        CAST( BS.delka_etrs89 AS text))) AS pocet
    FROM useky.mereni M
    JOIN useky.denni-useky U
    ON U.usek_id = M.usek_id
    JOIN body_souradnice BS
    ON BS.bod_id = M.bod_id

```

```

JOIN gravimetry.gravimetry G
ON G.gravimetr_id = U.gravimetr_id
WHERE G.typ NOT IN (1,11) AND EXTRACT (YEAR FROM(U.datum)) < 1995 AND U.
    usek_id IN (
        19720101.001,
        ...
        20100102.059
    )
GROUP BY U.usek_id , U.datum
) V
GROUP BY V.pocet
ORDER BY V.pocet ;

```

Výsledek dotazu:

pocet	cetnost
1	6
2	1071
3	1376
4	572
5	423
6	131
7	103
8	10
9	9
10	2

(10 rows)

E.4 Grafické znázornění v programu Surfer

