

UNIVERZITA KARLOVA

Přírodovědecká fakulta

Katedra aplikované geoinformatiky a kartografie

Studijní program: Geografie (bakalářské studium)

Studijní obor: Fyzická geografie a geoinformatika



Hugo-Miroslav MAJER

NÁVRH DATABÁZY LETOKRUHOVÝCH CHRONOLÓGIÍ

DESIGN OF A DENDROCHRONOLOGICAL DATABASE

Bakalářská práce

Vedoucí bakalářské práce: Mgr. Lukáš Brůha, Ph.D.

Praha 2021

Vysoká škola: Univerzita Karlova v Praze

Katedra: Aplikované geoinformatiky a kartografie

Fakulta: Přírodovědecká

Školní rok: 2020/2021

Zadání bakalářské práce

pro Huga-Miroslava Majera

obor Fyzická geografie a geoinformatika

Název tématu: Návrh databáze letokruhových chronologií

Zásady pro vypracování

Cílem práce je návrh prostorové databáze letokruhových chronologií. Navržená databáze bude sloužit jako centrální uložení datových podkladů sloužících predikci reakce hlavních lesních dřevin na klimatickou změnu. Databáze bude navržena tak, aby splňovala datové a funkční požadavky, které jsou dány jejím účelem. Při návrhu struktury databáze bude minimalizována redundance uložení dat (normalizace). Implementace návrhu zohlední i četnost očekávaných dotazů pro optimalizaci rychlosti dotazů (tvorba indexů). Řešení bude implementováno nad databázovou platformou PostgreSQL/PostGIS.

Rozsah grafických prací: dle potřeby

Rozsah průvodní zprávy: cca 50 stran

Seznam odborné literatury:

Brewer, P. W., Guiterman C. H. A new digital field data collection system for dendrochronology. *Dendrochronologia*. 2016;38:131–5

Brewer, P. W. TELLERVO, A guide for users and developers. 2016. Accessed online: <http://tellervo.org/support/tellervo-manual.pdf>

Hsu, L. S., Regina O. PostGIS in Action, Third Edition Spring 2021 (*estimated*) ISBN 9781617296697

Vedoucí bakalářské práce: Lukáš Brůha, Ph.D.

Konzultant bakalářské práce:

Datum zadání bakalářské práce: 01. 12. 2020

Termín odevzdání bakalářské práce: jaro 2021

Platnost tohoto zadání je po dobu jednoho akademického roku.

.....
Vedoucí bakalářské práce

.....
Vedoucí katedry

V Praze dne: 1. 12. 2020

Prohlášení

Prohlašuji, že jsem závěrečnou práci zpracoval samostatně a že jsem uvedl všechny použité informační zdroje a literaturu. Tato práce ani její podstatná část nebyla předložena k získání jiného nebo stejného akademického titulu.

V Praze dne 3. 5. 2021

.....

podpis

Pod'akovanie

V prvom rade chcem veľmi pod'akovať vedúcemu práce Mgr. Lukášovi Brůhovi, Ph.D. za venovaný čas, trpezlivosť a každé povzbudivé slovo v priebehu tvorby práce. Ďakujem patrí aj Mgr. Jiřímu Mašekovi za poskytnutie a pomoc s dendrochronologickými dátami a ochotu odpovedať na moje otázky. Za priblíženie koncepcie a účelu tvorenej databázy ďakujem doc. Mgr. Václavovi Tremlovi, Ph. D. a za nasadenie databázy na fakultný server ďakujem jeho správcovi Ing. Janovi Kurandovi. V neposlednom rade chcem pod'akovať rodine za podporu nielen pri tvorbe tejto práce, ale v priebehu celého štúdia.

Abstrakt

Bakalárska práca sa zaoberá návrhom priestorovej databázy letokruhových chronológií. Teoretická časť práce sa venuje databázovým systémom, princípom ukladania priestorových dát v relačných databázach, existujúcim databázam letokruhových chronológií, ale aj základným princípom dendrochronológie. V ďalšej časti práce sú popísané poskytnuté dendrochronologické dáta, ktoré sú používané v metodologickej časti práce. Metodická časť popisuje dátové požiadavky pre tvorenú databázu, postup jej návrhu a implementáciu v rámci PostgreSQL/PostGIS. Následne sa venuje tvorbe funkcií tvorených v procedurálnom rozšírení SQL, ktoré slúžia na zefektívnenie vkladu dát a neskôr je popisovaná tvorba SQL príkazu slúžiaceho ako dátový filter. Hlavným výsledkom práce je vytvorená priestorová databáza spĺňajúca dátové aj funkčné požiadavky, ktorá je naplnená poskytnutými dátami a pripojiteľná ku GIS. Záverom práce je zhrnutie dosiahnutých výsledkov a stručný popis prínosu práce v prebiehajúcom výskumnom projekte.

Kľúčové slová: priestorová databáza, návrh databázy, letokruhovú chronológiu, PostgreSQL/PostGIS, Procedural Language/PostgreSQL

Abstract

The bachelor thesis focuses on the design of a spatial database of tree-ring chronologies. The theoretical part of the thesis describes database systems and the storage of spatial data in relational databases. It also reviews existing databases of tree-ring chronologies and the basics of dendrochronology. The next section describes provided dendrochronological data, which are used in methodical section of the thesis. The methodical section describes the data requirements for the created database, the process of its design and implementation within PostgreSQL/PostGIS. Subsequently, it deals with the creation of functions developed in procedural extension of SQL. The stored functions are used to improve import of data. Afterwards, the design of a data filter in the form of SQL statement is mentioned. The main result of the thesis is a spatial database that meets both data and functional requirements, contains provided data and is connectable to GIS. The thesis concludes with a summary of the achieved results and contributions to the ongoing scientific project.

Keywords: spatial database, database design, tree-ring chronologies, PostgreSQL/PostGIS, Procedural Language/PostgreSQL

Obsah

Zoznam obrázkov a tabuliek.....	9
Zoznam skratiek.....	10
1. Úvod a ciele práce	11
2. Databázový systém	12
2.1. Komponenty databázového systému.....	12
2.2. Delenie databázových systémov	14
2.3. História SRBD	14
3. Priestorové dáta a relačné databázy	16
3.1. Priestorové objekty ukladané v budovaným numerickým dátovým typom.....	16
3.2. Priestorové objekty ukladané formou binárnej reprezentácie	16
3.3. Priestorové objekty ukladané formou geometrického dátového typu.....	17
3.4. Implementácia priestorových objektov v PostGIS.....	18
3.4.1. PostgreSQL/PostGIS	18
3.4.2. Dátový typ <i>Geometry</i>	18
3.4.3. Dátový typ <i>PostGIS Geography</i>	21
4. Letokruhové chronológie.....	22
4.1. Princípy letokruhov	22
4.2. Tvorba letokruhových chronológií.....	23
4.3. Využitie letokruhových chronológií	24
5. Databázy letokruhových chronológií.....	26
5.1. International Tree-Ring Data Bank (ITRDB)	26
5.2. TRiDaBase	27
5.3. Tellervo	28
6. Dáta a použitý softvér.....	30
6.1. Použitý softvér	30
6.1.1. R.....	31

6.1.2. ArcGIS Pro	31
7. Tvorba databázy letokruhových chronológií	32
7.1. Dátové požiadavky	32
7.2. Analýza dát.....	32
7.3. Konceptuálny model	33
7.3.1. Normalizácia	34
7.4. Logický model	35
7.5. Vklad dát do databázy	37
7.5.1. Vklad záznamov lokalít a stromov	37
7.5.2. Vklad záznamov vzoriek	39
7.5.3. Vklad chronológií širok letokruhov	39
7.6. Dátový filter	43
8. Výsledky	44
9. Diskusia	46
10. Záver	48
11. Použité zdroje	50
12. Zoznam príloh.....	54

Zoznam obrázkov a tabuliek

Obrázok 1: Schéma komponentov databázového systému.....	14
Obrázok 2: Geometrický model OGC.	19
Obrázok 3: SQL schéma pre implementáciu dátového typu Geometry.	20
Obrázok 4: Tvorba chronológie pomocou referenčnej chronológie.	23
Obrázok 5: Entitno-relačný diagram navrhovanej databázy.....	35
Obrázok 6: Relačná schéma navrhovanej databázy.....	37
Obrázok 7: Podoba CSV súboru letokruhových chronológií s požadovanou úpravou názvu prvého stĺpca.	40
Tabuľka 1: Prehľad atribútov, ktoré možno vybrať z databázy a dostupné podmienky aplikovateľné na ich záznamy.	45

Zoznam skratiek

ADT – abstraktný dátový typ

CSV – Comma-separated values

DBH – Priemer kmeňu stromu vo výške prs

DML – Data Manipulation Language

ERD – Entitno-relačný diagram

GIS – Geoinformačný systém

ITRDB – International Tree-Ring Data Bank

KFGG – Katedra fyzickej geografie a geoekológie PrF UK

NCEI – National Centers for Environmental Information

NOAA – National Oceanic and Atmospheric Administration

ODK – Open Data Kit

OGC – Open Geospatial Consortium

OOSRBD – Objektovo-orientovaný systém riadenia bázy dát

ORSRBD – Objektovo-relačný systém riadenia bázy dát

PL/pgSQL – Procedural Language/PostgreSQL

RSRBD – Relačný systém riadenia bázy dát

SFA – Simple Feature Access

SQL – Structured Query Language

SRBD – Systém riadenia bázy dát

SRS – Súradnicový referenčný systém

TRiDaS – Tree-Ring Data Standard

WKT – Well-Known Text

XML – eXtensible Markup Language

1. Úvod a ciele práce

Joe Celko, americký databázový odborník na adresu databáz vyslovil výrok: (Pokorný and Valenta 2020)

„Kde je múdrosť? Stratená v znalostiach.

Kde sú znalosti? Stratené v informáciách.

Kde sú informácie? Stratené v dátach.

Kde sú dáta? Stratené v databázach.“

Databázy nie sú vo svete ničím novým. Vznikli pred viac ako polstoročím za účelom zefektívniť ukladanie a spracovanie digitálnych dát a odstrániť nedostatky vtedy aktuálneho súborového prístupu. Postupom času sa stali fenomenálnou technológiou, s ktorou sa každý z nás stretáva dennodenne, nakoľko spektrum ich uplatnenia je veľmi široké a je obdobné rozsahu všetkých možných typov digitálnych dát. V tejto práci sú predmetom záujmu dendrochronologické dáta, medzi ktoré patria aj letokruhovú chronológiu, a ich aplikácia v rámci dendroklimatológie. Ambíciou súdobých výskumných projektov, ako napr. TAČR (2021), je vďaka priestorovej a časovej zložke týchto dát vysvetliť vplyv nielen súčasných klimatických podmienok na dreviny na území ČR, ale aj predikovať ich reakciu v budúcich desaťročiach na zmenu klímy, ktorá je predikovaná klimatickými scenármi. Pri pohľade na vyššie uvedený výrok sa dá povedať, že databáza nie je cieľom, ale len prostriedok ako zo samotných dát získať efektívne čo najviac informácií a znalostí. Rovnako sa dá povedať, že vlastnosti a kvalita výstupov nezáleží teda len na samotných dátach, ale aj na kvalite ich uloženia a spracovania na strane databázy.

Hlavným cieľom práce je vytvorenie priestorovej databázy letokruhovú chronológií, ktorá spĺňa dátové požiadavky a ktoré sú špecifikované v kapitole 7.1. Medzi dlhšie ciele patria funkčné požiadavky, ako je definovanie metód vkladu dát do databázy, možnosť výberu podmnožiny dát z databázy na základe požiadaviek užívateľa aplikovateľných na všetky atribúty z dátových požiadaviek a možnosť pripojiť sa k databáze v geoinformačnom systéme. Ďalšími cieľmi práce je rešerš literatúry so zameraním sa na databázové systémy vo všeobecnosti a následne na interakciu priestorových dát s relačnými databázami. Druhým cieľom rešerše je stručné oboznámenie sa so základnými princípmi dendrochronológie a prieskum už existujúcich dendrochronologických databáz.

2. Databázový systém

Databázový systém je počítačový systém, ktorého základnými službami je uchovávanie dát, len na požiadavku užívateľa vkladanie nových dát, modifikácia alebo mazanie uložených dát a ich získavanie zo systému (Date 2004). Jeho ďalšími službami je manažment ochrany systému, dát, zálohovania, obnovy a dátovej integrity, riadenie prístupu viacerých užívateľov, prezentácia a transformácia dát a iné (Coronel and Morris 2015).

2.1. Komponenty databázového systému

Databázový systém sa skladá zo štyroch hlavných komponentov, ktorými sú zabezpečované všetky jeho funkcie (pozri Obrázok 1) (Bal Gupta and Mittal 2017).

Dáta

Všetky dáta v systéme sú ukladané spoločne do jednej *databázy*, ktorá sa dá definovať ako organizovaný súbor navzájom logicky súvisiacich dát, ktoré sú jednak v nej integrované, ale aj zdieľané (Bal Gupta and Mittal 2017). Databáza je nazývaná aj samo vysvetľujúcou kolekciou dát, pretože okrem samotných dát sú uchovávané aj metadáta, ktoré popisujú databázovú schému (Elmasri and Navathe 2010).

Databázy sú organizované pomocou *polí*, čo je najmenší prvok definovaný určitou hodnotou, ktorá užívateľovi ponúka určitú informáciu, ďalej pomocou *záznamov*, čo je súbor logicky súvisiacich polí a vďaka *súborom*, ktoré predstavujú kolekciu súvisiacich záznamov (Bal Gupta and Mittal 2017).

Databáza je tvorená štyrmi komponentami: 1) určitými, samostatnými informáciami nazývanými *dátovými prvkami* alebo elementami; 2) *vzťahmi* medzi dátovými prvkami; 3) *integritnými obmedzeniami*, ktoré predstavujú požiadavky pre dáta ukladané v databázy, pričom je pomocou nich definovaný stav databázy; 4) *schémou*, ktorá popisuje usporiadanie dát a vzťahov pochopiteľne pre užívateľa (Bal Gupta and Mittal 2017, Pokorný and Valenta 2020).

Softvér

Najdôležitejším, ale ani zďaleka nie jediným softvérom databázového systému je *Systém riadenia báze dát* (SRBD) (angl. *Database Management System*), ktorý je považovaný za srdce databázového systému. Okrem SRBD sa na chode databázového systému podieľa operačný systém a užívateľské programy (Coronel and Morris 2015).

SRBD používateľovi slúži ako jediný poskytovateľ prístupu k databáze a to prostredníctvom integrovaného pohľadu na dáta v nej, čím chráni užívateľa pred vnútornou zložitou databázy (Coronel and Morris 2015). SRBD je takisto zodpovedný za riadenie súbežného prístupu užívateľov a za zálohu a obnovu databázy (Bal Gupta and Mittal 2017). Pokorný and Valenta (2020) uvádza, že SRBD je poskytovateľom všetkých spomínaných služieb databázového systému.

Musí zahrňovať dva databázové jazyky: (Silberschatz et al. 2011)

- ❖ **Jazyk pre definíciu dát** (angl. *Data Definition Language* – DDL) pomocou ktorého sa definuje schéma databázy a vlastnosti dát, ako je integritné obmedzenie atribútov a referenčná integrita spojená s primárnymi a cudzími kľúčmi. Takisto sa využíva na udeľovanie určitých právomocí užívateľom pre manipuláciu s dátami (autorizácia).
- ❖ **Jazyk pre manipuláciu dát** (angl. *Data Manipulation Language* – DML) slúži na vklad, úpravu, mazanie dát v databáze a vďaka súčasť DML nazývanej *dotazovací jazyk* (angl. *query language*) slúži aj na získavanie dát z databázy na základe určitých podmienok.

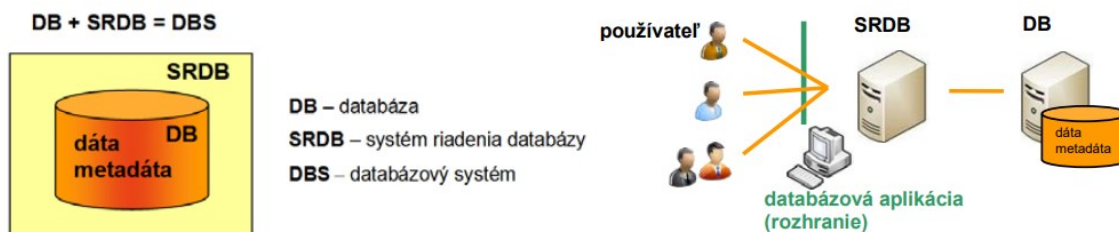
Hardvér

Skladá sa z dvoch komponentov: (Bal Gupta and Mittal 2017)

- ❖ **Procesor a hlavná pamäť**, ktoré sa starajú o funkčnosť softvérovej zložky systému.
- ❖ **Externé zariadenia**, ako sú pevné alebo magnetické disky, na ktoré sú ukladané dáta v systéme.

Užívatelia

Tvorí ho všetci užívatelia, ktorý nejakým spôsobom interagujú s databázovým systémom od administrátora databázy po koncového užívateľa (Bal Gupta and Mittal 2017).



Obrázok 1: Schéma komponentov databázového systému.

Zdroj: Ďuračiová and Cibulka 2015

2.2. Delenie databázových systémov

Databázové systémy sa môžu deliť *podľa počtu užívateľov*, ktorým je umožnený prístup v jednej chvíli na systémy pre jedného používateľa resp. pre viacerých používateľov. Ďalším delením je *podľa ich využitia*. Produkčné alebo transakčné systémy sú využívané na dohľad nad výrobou produktov, druhé menované v bankovníctve. Existujú databázové systémy podporujúce rôzne formy modelovania využívané v rôznych odvetviach na tvorbu rozhodnutí. Databázový systém sa bežne využíva ako centralizované úložisko dát (Bal Gupta and Mittal 2017). Tretím delením je *podľa umiestnenie databázy*. Ak sú všetky dáta uložené na jednom mieste (počítači alebo serveri), jedná sa o centralizovaný systém. Naopak, ak sú dáta ukladané na rôznych fyzických miestach, tak hovoríme o distribuovanom databázovom systéme (Coronel and Morris 2015).

2.3. História SRBD

Prvé SRBD, napr. IMS pre Apollo zo 60. rokov, vychádzali z hierarchického dátového modelu, ktorý organizuje jednotky do stromu od koreňa po potomkov so vzťahmi medzi jednotkami 1:N. Sieťové SRBD boli vyvinuté na odstránenie nedostatkov hierarchického modelu, konkrétne priniesli možnosť M:N vzťahov medzi jednotkami. V 70. rokoch E. F. Codd predstavil relačný model. Dáta sú ukladané v tabuľkách nazývaných reláciami, ktoré pozostávajú z riadkov (n-tíc, definovaných ako súbor informácií jedného záznamu) a zo stĺpcov (atribútov) (Bal Gupta and Mittal 2017). V tomto období takisto začal vznikať neprocedurálny a v relačných SRBD (RSRBD) všadeprítomný jazyk – *Structured Query Language* (SQL) využívaný pre realizáciu všetkých služieb RSRBD. Na dominantnosti RSRBD vo svete nič nezmenil príchod objektovo-orientovaných SRBD (OOSRBD) koncom 80. rokov, ktorých využitie zostalo na nízkej úrovni a tak sa začalo špekulovať o ich vzťahu k RSRBD. Koncom 90. rokov vznikli objektovo-relačné SRBD (ORSRBD) kombinujúce vlastnosti relačných systémov (SQL, relačná implementácia dát) s dobrými vlastnosťami

OOSRBD (modelovanie zložitých dát apod.). V 90. rokoch s príchodom *eXtensible Markup Language* (XML) sa vyvinula potreba ukladať a spracovávať tieto dáta v databázach, čím vznikol ďalší databázový model. Medzi novšie smery vo vývoji SRBD patria NoSQL databázy, ktoré ale majú ďaleko od vyspelej databázovej technológie. Dáta v nich môžu byť rozložené aj horizontálne aj vertikálne, vyznačujú sa intuitívnym dátovým modelom, teda modelom bez akýchkoľvek formálnych základov, čím sa stráca rozdiel medzi konceptuálnym a databázovým pohľadom na dáta a ponúkajú veľmi obmedzené dotazovanie paradoxne s ohľadom na ich názov pomocou obmedzeného SQL (Pokorný and Valenta 2020).

3. Priestorové dáta a relačné databázy

Vývoj ukladania priestorových dát sa odvodzuje pomocou vývoja geoinformačných systémov (GIS). Ich prvá generácia je založená na súborovom prístupe spracovania dát – ukladanie priestorových dát a ich atribútov do súborov. Zo súborového prístupu sa vyvinul databázový prístup, ktorý bol sčasti aplikovaný v druhej generácii GIS a to tak, že priestorové dáta sa stále nachádzali v súboroch, ale nepriestorové dáta boli uchované v relačnej databáze. V GIS tretej generácie sú obidve zložky dát uchovávané v relačnej databáze, pričom existuje viacero prístupov uloženia týchto komplexných dát (Pokorný 2001).

Vo všeobecnosti platí, že podobné geografické objekty (v GIS nazývané vrstvami) tvoria tabuľku (reláciu), ktorej riadky (záznamy) predstavujú jednotlivé geografické objekty. Geografický objekt má geometrický (priestorový) atribút (Rigaux et al. 2002).

3.1. Priestorové objekty ukladané v budovaným numerickým dátovým typom

Geometrický atribút priestorového objektu je zaznamenaný pomocou cudzieho kľúča do normalizovanej tabuľky geometrií. Je tvorený jednou alebo viacerými primitívnymi časťami, ktoré sú zoradené pomocou čísiel. Každá primitívna časť môže byť zložená z viacerých častí, takže môže tvoriť jeden alebo viacero riadkov v tejto tabuľke. Ich riadky a ich správna postupnosť je opäť identifikovaná pomocou čísla a rovnakým spôsobom je definovaný aj typ ich geometrie. Geometria (súradnice) týchto primitívnych geometrických objektov je uchovávaná formou preddefinovaných SQL numerických dátových typov (OGC 2010).

Rigaux et al. (2002) uvádza nedostatky tohto prístupu ako je porušenie nezávislosti dát, pretože užívateľ musí poznať štruktúru uloženia dát a tiež zlý výkon, pretože akákoľvek priestorová informácia vyžaduje množstvo záznamov a aj celková nevhodnosť pre užívateľa.

3.2. Priestorové objekty ukladané formou binárnej reprezentácie

Je to prístup podobný predošlému prístupu s tým rozdielom, že v tabuľke geometrií sú geometrické objekty ukladané binárnou reprezentáciou *Well-known Binary Representation for Geometry*. Spoločne s nimi sú v tabuľke geometrií uchovávané aj súradnice minimálneho

obdĺžnika ohraničujúceho daný geometrický objekt a to opäť klasickým numerickým dátovým typom (OGC 2010).

Obidva dosiaľ spomenuté prístupy pristupujú k priestorovým dátam ako ku ktorýmkoľvek iným dátam a nevyužívajú sa geometrické vlastnosti priestoru. Vnútna štruktúra priestorových objektov je pre RSRBD neznáma a nedá sa teda zahrnúť do dotazov mierených na jej obsah. Pre akýkoľvek manažment alebo spracovanie priestorových dát musí byť využitý špeciálny softvér – GIS (Pokorný and Valenta 2020).

3.3. Priestorové objekty ukladané formou geometrického dátového typu

Optimálnym prístupom je rozšírenie SQL o geometrický dátový typ využitím *užívateľom definovaných typov a užívateľom definovaných funkcií* univerzálnejšie nazývané *abstraktné dátové typy* (ADT) (Pokorný 2001).

Mechanizmus ADT umožňuje vytvárať nové dátové typy a pracovať s nimi ako s bežnými vbudovanými dátovými typmi. To je dosiahnuté vďaka princípu *zapuzdrenia* využívaného v objektovo orientovanom programovaní. Zapuzdrením sa nová dátová štruktúra spoločne s funkciami (metódami), ktoré definujú nielen správanie inštancií daného typu ale napríklad umožňujú aj ich porovnávanie, „zabalí“ do modulu, ktorým sa rozšíri SRBD. Takýto modul sa nazýva *extender* (Pokorný and Valenta 2020).

Využívať resp. tvoriť ADT v relačných systémoch je dôsledok štandardu SQL:1999, ktorý rozširuje databázový jazyk relačných databáz SQL o objektovú orientáciu. Práve jeho implementáciou je dosiahnutá koexistencia relácií a objektov a z RSRBD sa stáva ORSRBD. ORSRBD boli vyvinuté práve za účelom odstrániť limity relačných systémov v súvislosti s narábaním so zložitými dátami a rozšíriť využitie relačnej technológie do odvetví vyžadujúcich integráciu ako klasických, tak aj komplexných dát, medzi ktoré patria napríklad multimedialne dáta ale aj priestorové dáta (Pokorný and Valenta 2020).

Po vložení geometrických dátových typov mechanizmom ADT môžu byť ich hodnoty umiestnené do stĺpcov tabuliek ako akýkoľvek iný, klasický dátový typ. Systém im už nielenže „rozumie“, ale s využitím priložených metód sa takisto bežnými prostriedkami SQL dá dotazovať na geometrické vlastnosti priestorových objektov, čím sa odstraňujú nedostatky predošlých prístupov (Pokorný 2001, Pokorný and Valenta 2020).

Príkladom priestorového extenderu pre proprietárne databázové systémy môže byť Oracle Spatial and Graph pre Oracle, priestorovo rozšírený je aj DB2 a Informix, oba

vyvíjané spoločnosťou IBM. Z voľne dostupných systémov môžeme spomenúť MySQL a PostgreSQL, ktorý disponuje priestorovým modulom PostGIS.

3.4. Implementácia priestorových objektov v PostGIS

Táto kapitola bližšie popisuje PostgreSQL a jeho priestorový extender PostGIS a následne popisuje implementáciu vektorových dát doň pomocou dátového typu *Geometry*, pričom stručne popísaný je aj a dátový typ *PostGIS Geography*.

3.4.1. PostgreSQL/PostGIS

PostgreSQL je voľne dostupný objektovo-relačný databázový systém využívajúci jazyk SQL. Jeho začiatky siahajú do 80. rokov minulého storočia, kedy bol ešte prezentovaný pod názvom *Postgres*. Je považovaný za najlepší a najdokonalejší voľne dostupný systém najmä vďaka osvedčenej architektúre, spoľahlivosti, bezpečnosti, integrite dát, veľkému množstvu dostupných funkcií, neustálym aktualizáciám zdokonaľujúcim systém a vďaka jeho možnostiam rozširiteľnosti. Rozširiteľnosť systému je možná napríklad spomínaným ADT, vloženými funkciami a procedúrami, ktoré môžu byť vytvorené v procedurálnych jazykoch ako je *Procedural Language/PostgreSQL* (PL/pgSQL), Python, Java a iné. Dostupné sú aj rôzne extrendre, najpopulárnejším je priestorové rozšírenie PostGIS (Postgresql 2021a).

PostGIS je voľne dostupné priestorové rozšírenie pre PostgreSQL. Pomocou nových dátových typov umožňuje jednak uchovávať priestorové objekty v databáze, ale aj podrobovať ich analýze a spracovaniu v prostredí SQL (Postgis 2021). Podobne ako ostatné priestorové databázy teda kombinuje vlastnosti GIS s výhodami databázových systémov, napríklad transakčné spracovanie, paralelný prístup alebo indexovanie (Strobl 2008). Okrem vektorových dát plne podporuje aj spracovanie a analýzu rastrových dát, 3D modelov povrchu a dostupný je aj modul pre topológiu (Postgis 2021).

3.4.2. Dátový typ *Geometry*

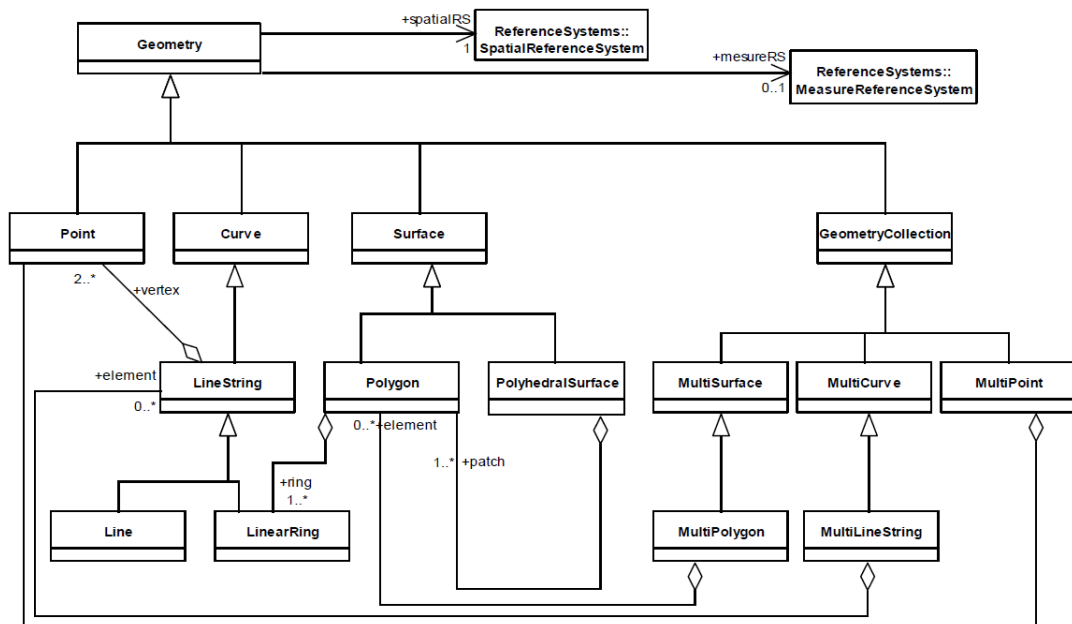
PostGIS sa riadi a je plne vyhovujúci štandardu konzorcia *Open Geospatial Consortium* (OGC) s názvom *Simple Feature Access* (SFA) (Postgis 2021). Táto norma sa delí na dve časti. Prvá časť definuje model geometrie OGC spolu s metódami, z ktorého vychádza priestorový dátový typ *Geometry*. Druhá časť definuje jeho SQL implementáciu.

Prvá časť SFA - geometrický model OGC

Priestorový dátový typ *Geometry* používaný v PostGIS vychádza z modelu geometrie OGC (pozri Obrázok 2). Ten predstavuje hierarchiu geometrických typov, ktorej koreňom je trieda *Geometry*. Podľa tohto modelu stále platí, že podobné priestorové objekty sú ukladané v jednej tabuľke. Priestorový objekt má aspoň jeden geometrický atribút, ktorý odpovedá niektorej podtriede koreňa *Geometry* (Yeung et al. 2007). Každá inštancia geometrického typu alebo lepšie povedané každý geometrický atribút, má pridelený súradnicový referenčný systém (SRS) (OGC 2011).

Každý geometrický typ v opisovanom modeli obsahuje súbor metód, ktoré sú využívané na test ich príslušných geometrických vlastností, na definíciu priestorových vzťahov medzi nimi a metódy na podporu priestorových analýz (OGC 2011).

Táto časť SFA definuje aj dve štandardné reprezentácie priestorových dát *Well-Known Text* (WKT) a *Well-Known Binary* (WKB), ktoré sú využívané v SQL funkciách na tvorbu nových inštancií geometrických typov (OGC 2010).



Obrázok 2: Geometrický model OGC.

Zdroj: OGC 2011

Druhá časť SFA - SQL implementácia

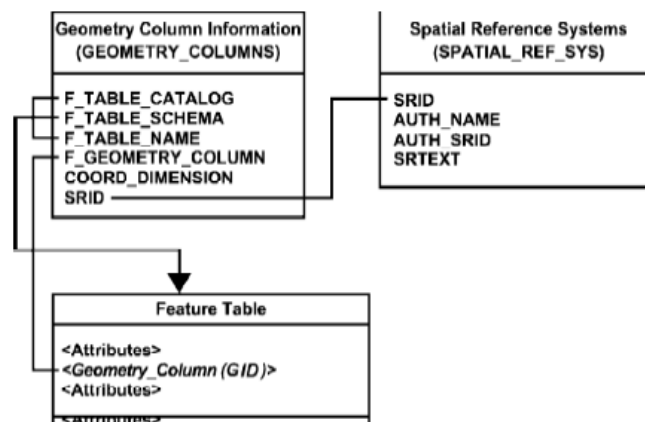
PostGIS implementuje všetky geometrické typy špecifikované v modeli geometrie OGC spoločne so všetkými ich metódami (Postgis 2021), rozšírením SQL o dátový typ *Geometry*. Podľa OGC (2010) sa na rozšírenie SQL týmto dátovým typom používa mechanizmus ADT,

pričom jednotlivé geometrické typy sú definované metódou *čiernej skrinky*, čo znamená, že všetok prístup k informáciám o nich je cez SQL funkcie.

Táto časť SFA navyše stanovuje aj potrebnú SQL schému na ukladanie a prácu s priestorovými objektmi spomínaných geometrických typov. Jedná sa o dve metadátové tabuľky (pozri Obrázok 3).

Tabuľka (pohľad) geometrií zaznamenáva každý jeden geometrický atribút (stĺpec) v databáze, pričom uchovávaný je jeho názov, identita tabuľky z ktorej pochádza, jeho priestorová dimenzia a identifikátor využívaného SRS (OGC 2010). V prípade PostGIS je tabuľka geometrií doplnená o typ geometrie nachádzajúcej sa v danom geometrickom atribúte (Postgis 2021).

V tabuľke geometrií je identifikátor SRS cudzím kľúčom odkazujúcim do druhej metadátovej tabuľky SRS, čím súradniciam v danom geometrickom atribúte dáva význam. Okrem identifikátoru SRS tabuľka obsahuje jeho EPSG a WKT reprezentáciu daného SRS (OGC 2010). V PostGIS táto tabuľka obsahuje cez 3 000 SRS a detaily potrebné na transformáciu medzi nimi (Postgis 2021).



Obrázok 3: SQL schéma pre implementáciu dátového typu Geometry.

Zdroj: OGC 2010

Pre úplnosť dodajme, že geometrické typy dostupné dnes v PostGIS sú nadmnožinou typov definovaných v SFA. PostGIS totižto podporuje aj viacero kruhovo interpolačných kriviek definovaných normou *SQL Multimedia Applications Spatial* od *International Organization for Standardization* (Postgis 2021).

3.4.3. Dátový typ *PostGIS Geography*

Alternatívou pre vektorové dáta je dátový typ *PostGIS Geography*, ktorý zohľadňuje sférický tvar Zeme na rozdiel od predošlého typu, ktorého základom je rovina. Akékoľvek výpočty na sfére sú komplikovanejšie ako v karteziánskej sústave, takže pre tento dátový typ je zatiaľ dostupných aj menej funkcií. *PostGIS Geography* podporuje akýkoľvek SRS založený na zemepisných súradniciach (Postgis 2021).

4. Letokruhové chronológie

Dendrochronológia je veda, ktorá prostredníctvom štruktúry dreva skúma nielen prirodzené zmeny životného prostredia prírodnými procesmi, ale aj zmeny prostredia vyvolané pôsobením človeka v čase, pričom ich dokáže aj datovať (Speer 2010). Premennivosť prostredia je zaznamenávaná v letokruhoch stromov, ktoré sú považované za kľúčový zdroj informácií pre rekonštrukciu podmienok v pre-inštrumentálnom období (Esper and Gärtner 2001).

4.1. Princípy letokruhov

Letokruhy vytvára kambium – úzka vrstva deliteľného tkaniva medzi kôrou a drevom, v ktorom prebieha bunkové delenie. Každý letokruh sa skladá z jarného dreva, ktoré vzniká na jar a skoro v lete, a z letného dreva, ktoré sa formuje neskôr v lete a je typické svojim tmavším sfarbením (Speer 2010).

Tvorba letokruhov je ovplyvnená ekologickou valenciou druhu a limitujúcim faktorom prostredia, ktorý determinuje rýchlosť rastu stromu (Panyushkina 2011). Stromy vo vysokých nadmorských výškach sú limitované teplotou, v suchých oblastiach zrážkami, pričom je možné, že hlavný limitujúci faktor sa behom života stromu zmení, prípadne strom môže mať aj viac limitujúcich faktorov naraz (Speer 2010).

Väčšina druhov v temperátnom a subarktickom pásme, kde sa strieda vegetačné obdobie s dormanciou, produkuje každý rok jeden letokruh (Speer 2010). Drevinám v oblastiach s minimálnymi výkyvmi klímy behom roka, napr. tropické dažďové lesy, letokruhy častokrát úplne chýbajú (Kyncl 2017). Strom môže za rok vyprodukovať aj viac ako jeden letokruh a to v prípade výskytu dočasného stresu, ktorý po čase stromu znovu umožní pokračovať v raste alebo v prípade dlhšie pôsobiaceho stresu sa nemusí vytvoriť žiaden. Takýto letokruh sa nazýva falošným, resp. chýbajúcim (Nash 2008).

Prostredie nemá vplyv len na počet letokruhov, ale aj na ich šírku, čo je najskúmanejší parameter štruktúry dreva, pretože ponúka informácie o množstve ekologických faktoroch, čím je ale kladený veľký dôraz na kvalitný výber lokality (Esper and Gärtner 2001). Nash (2008) v tejto súvislosti uvádza dôležitosť výskytu práve jedného limitujúceho faktoru rastu stromov na uvažovanej lokalite, ktorý sa zároveň musí vyznačovať každoročnou premenlivosťou, aby sa odzrkadlil v šírke letokruhov. Rovnakú dôležitosť pripisuje jeho plošnému rozsahu a dostatočnej sile, aby boli stromy ovplyvňované

v podobnej miere. Torbenson (2015) uvádza, že najvyšší potenciál pre výskum a presné datovanie majú stromy žijúce na hranici svojej ekologickej valencie.

4.2. Tvorba letokruhových chronológií

Zo závislosti produktivity stromov na prostredí, ktorá spôsobuje jednotnú podobu letokruhov medzi jednotlivými stromami na určitej lokalite (Esper and Gärtner 2001), vychádza základný princíp dendrochronológie – krížové datovanie (Speer 2010).

Vďaka krížovému datovaniu, ktoré predstavuje vzájomné porovnávanie širok letokruhov medzi jednotlivými stromami na určitej lokalite, sa dajú účinne identifikovať a odstrániť letokruhové anomálie, ako sú falošné a chýbajúce letokruhy a tým pádom je možné presne stanoviť rok vzniku každého letokruhu (Speer 2010). Ďalšou funkciou krížového datovania je datovanie neznámej letokruhovej série priložením k tzv. referenčnej chronológii, ktorá môže pochádzať zo živého stromu a je teda datovaná rokom odberu vzorky. Pri ich porovnaní musí dôjsť k tzv. spoľahlivému prekrytiu, ktoré sa počíta štatistickou matematikou a takýmto spôsobom sa chronológia predlžuje smerom do minulosti (pozri Obrázok 4) (Kyncl 2017).



Obrázok 4: Tvorba chronológie pomocou referenčnej chronológie.

Zdroj: Kyncl 2017

Po odatovaní letokruhovej série sa pristúpi k meraniu parametrov letokruhov. Najčastejšie sa zaznamenáva šírka každého letokruhu, šírka jarného a letného dreva s presnosťou aspoň na stotiny milimetra. Ďalšími veličinami môže byť napríklad priemerná hustota letokruhov (Panyushkina 2011). Na meranie sa používajú skenery s vysokým rozlíšením spojené so softvérom, prípadne pohyblivý mechanizmus s mikroskopom. Využívajú sa aj softvéry, ktoré štatistickými testami overujú vykonané datovanie krížovým datovaním (Torbenson 2015).

Tieto série nespracovaných meraní nie sú vhodné na interpretáciu a vyžadujú šandardizáciu, pretože sú ovplyvnené rastovým trendom spojeným s vekom stromu, rôznou rýchlosťou rastu stromov, prípadne sú poznačené v dôsledku disturbancií (Carrer 2011). Speer (2010) uvádza, že táto procedúra sa vykonáva zložitými matematickými vzorcami, najčastejšie zápornou exponenciálnou krivkou alebo inými empirickými modelmi, napríklad kubickým *spline*, pričom upozorňuje, že metóda detrendovania sa musí voliť opatrne, s ohľadom na výskumnú otázku tak, aby potlačila nežiadúci šum a zvýšila požadovaný signál.

Jedným z možných postupov po šandardizácii hodnôt je vydelenie hodnôt predpovedanou hodnotou rastu, čím sa vytvorí index rastu stromu. Tieto indexy pre každý rok sa medzi všetkými zozbieranými sériami navzájom spriemerujú a vznikne chronológia letokruhových indexov nesúca informáciu už na úrovni lokality (*Tree-Ring Standardized Growth Index*) (Panyushkina 2011). Nejedná sa o jediný možný postup.

4.3. Využitie letokruhových chronológií

Táto kapitola sa zameriava na využitie letokruhových chronológií v rámci dendroklimatológie. Okrem nej sú chronológie využívané aj v dendroarcheológii, geomorfológií, dendroekológii a v iných odvetviach.

Vďaka letokruhovým chronológiám, dokážeme rekonštruovať teplotu, zrážky či obdobia sucha v minulosti vzdialenej kľudne aj tisíc rokov a takisto dopad tejto klimatickej variability na stromy, ktorých letokruhy plnia funkciu indikátorov (Jones and Mann 2004).

Büntgen et al. (2006) využili chronológiu pokrývajúcu obdobie 735 – 2004 na rekonštrukciu letnej teploty v Alpách, pričom zistili, že vyššie teploty sa vyskytovali v 10. a 13. storočí, následne do 18. storočia teplota klesala a 20. storočie sa už podmienkami podobalo na dobu pred ochladzovaním. Až šesť z desať najteplejších dekád bolo

zaznamenaných v minulom storočí. Najvyšší rozdiel za skúmané obdobie medzi najteplejšou a najchladnejšou dekadou bol približne 3,1 °C.

Cook et al. (2010) využili chronológiu pozostávajúcu z hodnôt Palmerovho indexu intenzity sucha (PDSI) odvodenú z viac ako 300 monzúnom zasahovaných lesných lokalít v Ázii na vytvorenie atlasu monzúnového sucha pre túto oblasť. Tento atlas ponúka za posledné tisícročie absolútne datovanú a každoročne zrekonštruovanú časopriestorovú variabilitu ázijských monzúnov, popisuje výskyt a závažnosť pred tým nepoznaných monzúnových období sucha a odhaľuje ich súvislosť nielen s El-Niño. Takéto informácie sú nutné napríklad pre validáciu klimatických modelov a predpovede variability monzúnu do budúcnosti.

Rôzne štúdie sa prostredníctvom letokruhov zaoberajú aj dopadom globálneho otepľovania na stromy, napríklad Gedalof and Berg (2010) alebo van der Sleen et al. (2015) skúmali vplyv zvyšujúcej sa koncentrácie CO₂ v atmosfére na produktivitu lesov. Muzika et al. (2004) sledovali dlhodobu znečistenú lokalitu škodlivinami NO₂ a SO₂ a ich vplyv na rast stromov.

5. Databázy letokruhových chronológií

Táto kapitola sa zaoberá prieskumom existujúcich dendrochronologických databáz, ich stavbou, funkčnosťou, obsahom a vlastnosťami.

5.1. International Tree-Ring Data Bank (ITRDB)

ITRDB je dostupná na web stránke *National Climatic Data Center* (www.ncdc.noaa.gov/paleo-search/). Databáza bola vytvorená v roku 1974 za účelom uchovávanía vysokokvalitných dendrochronologických dát a stala sa primárnou databázou letokruhových dát (Zhao et al. 2018). Jedná sa o verejnú databázu, ktorá obsahuje cez 10 000 datasetov odvodených z prírodných zdrojov, necelých 3000 z letokruhov stromov.

Databáza poskytuje prístup k dátam od troch poskytovateľov dát, pričom si používateľ môže vybrať jedného alebo viacerých poskytovateľov. Prvým poskytovateľom je portál *Neotoma*, ktorý prevádzkuje databázu rôznych typov paleoekologických dát. Druhým poskytovateľom je samotný prevádzkovateľ ITRDB – *National Oceanic and Atmospheric Administration* (NOAA) a *National Centers for Environmental Information* (NCEI), pri ktorého voľbe má užívateľ možnosť výberu typu dát ktoré požaduje. Letokruhové dáta nie sú teda jediným dostupným typom dát, v ponuke sú napríklad dáta už z klimatických rekonštrukcií, peľové alebo paleolimnologické dáta a mnoho iných. Tretím poskytovateľom dát je portál *Pangaea*. Pri voľbe tohto poskytovateľa si ale užívateľ nemôže vybrať typ dát o ktoré má záujem, čo spôsobuje veľmi zdĺhavé čakanie pri hľadaní dát, ktoré sa častokrát ukončí bez výsledku. Lepším postupom je navštíviť priamo portál *Pangaea*, kde sa dajú kritéria vyhľadávania špecifikovať. Dostupné sú dáta naprieč všetkými zemskými sférami a dendrochronologické dáta nie sú výnimkou.

Pri hľadaní v ITRDB sa dá využiť aj rozšírené vyhľadávanie. Pri dátach z *Neotoma* a *Pangaea* sa ale výsledky dajú filtrovať len podľa mena osoby, ktorá dáta zozbierala a podľa lokality, ktorá sa definuje zemepisnou šírkou a dĺžkou. Rozšírené vyhľadávanie je omnoho efektívnejšie pri dátach z NOAA a NCEI. Definovanie lokality je obohatené o zoznam s výberom kontinentu, časti kontinentu, prípadne štátu, pričom výber na základe súradníc je zachovaný a doplnený o hodnotu nadmorskej výšky.

Ďalším kritériom je výber premenných alebo kľúčových slov. V prípade výberu určitých premenných sa užívateľovi sprístupní výber ďalších doplňujúcich kritérií, napr. materiálu a sezonality. Nechýba možnosť voľby druhu stromu, ktorý je predmetom záujmu.

Je nutné podotknúť, že pri každom kritériu je možné vybrať viac ako jednu užívateľom požadovanú hodnotu. V takomto prípade sa objaví aj možnosť, či vyhladané dáta majú obsahovať iba jednu z požadovaných premenných alebo všetky.

Poslednou možnosťou je špecifikácia začiatku a konca časovej rady, kde sú zaujímavé možnosti, či vyhladávané datasety môžu len sčasti prekryvať uvedenú časovú radu alebo majú byť celé prekryté danou chronológiou.

Po vyhladaní a výbere vhodného datasetu je možné si prečítať informácie o datasete a všetky dáta, ktoré obsahuje, sa dajú bezplatne stiahnuť. V závislosti na lokalite sú dostupné chronológie z nespracovaných meraní širok letokruhov, hustoty letokruhov, izotopových meraní ale aj už štandardizované chronológie indexu rastu, prípadne pre niektoré lokality sú dostupné aj chronológie rekonštruovaných klimatických premenných. Pri súboroch sú dostupné aj informácie o jednotke merania, metóde a iné dodatočné informácie.

Zhao et al. (2018) uvádzajú, že napriek veľkému potenciálu, ITRDB bola využitá len v málo štúdiách skúmajúcich komplexné procesy. Ako jedny z dôvodov určujú formátovanie dát vo formáte *Tucson*, ktorý sa dnes používa vďaka svojim špecifikám už iba v dendrochronológií. Tento formát bol zvolený kvôli pôvodnej funkcii ITRDB – uchovávanie len dendrochronologických dát slúžiacich na rekonštrukciu klimatických pomerov (Grissino-Mayer and Fritts 1997). Časom zvyšujúcim sa využívaním týchto dát v rôznych vedných odboroch, ktoré majú vyššie nároky na dáta a metadáta, sa *Tucson* formát stal limitujúcim faktorom pre využitie databázy (Zhao et al. 2018). Zang (2015) dopĺňa, že ďalším nedostatkom je absencia okamžitej možnosti preskúmať dáta a za takýmto účelom je nutné dáta zakaždým stiahnuť.

5.2. TRiDaBase

TRiDaBase je voľne dostupná k stiahnutiu na web stránke TRiDaS (www.tridas.org/software.php). Fakt, že databáza je vyvinutá v Microsoft Access, spôsobuje „bežne akceptovanú funkčnosť a ľahkú dostupnosť pre používateľov Windows. Vytváranie pokročilých databázových funkcií pomocou MS Access je náročné z dôvodu špecifickosti softvéru.“ (Jansma et al. 2012, s. 210)

Je to normalizovaná relačná databáza založená na dátovom štandarde *Tree Ring Data Standard* (TRiDaS) (Jansma et al. 2012), ktorý vznikol kvôli tomu, že dendrochronológovia ukladajú svoje dáta a metadáta v navzájom nekompatibilných formátoch a prevod dát medzi

nimi spôsobuje straty a nepresnosti. TRiDaS, ktorý je založený na XML, potlačil mnoho nedostatkov starších formátov dendrochronologických dát a stal sa medzinárodne uznávaným štandardom pre dendrochronologické dáta a metadáta (Jansma et al. 2010).

TRiDaBase nebudeme podrobnejšie popisovať, pretože v poradí ďalšia popisovaná databáza je založená na rovnakom koncepte a zároveň predstavuje kvalitnejšie a modernejšie spracovanie.

5.3. Tellervo

Tellervo je voľne dostupná cross-platform aplikácia pre Windows, Mac a Linux. Slúži na meranie a uchovávanie dendrochronologických dát (Brewer 2014). Je zložená zo samotnej aplikácie a z databázového serveru, ktorý slúži ako zdieľaná centrála, kam sa ukladajú dáta a metadáta do bezpečnej objektovo-relačnej databázy. Prístup k nej poskytuje samotná aplikácia prostredníctvom prihlasovacích údajov a URL adresy serveru, ktoré sa vytvoria pri inštalácii serveru (Brewer 2016), takže sa na rozdiel od ITRDB, jedná o neverejnú databázu.

Databáza bola tvorená v PostgreSQL s rozšírením PostGIS, čím je priestorovo rozšírená, čo využíva samotná aplikácia v budovaným mapovaním lokalít, pričom pri výbere lokality sa zobrazia jej charakteristiky, čerpané z databázy (Brewer 2014).

Tellervo taktiež využíva dátový štandard TRiDaS (Brewer 2016), ktorému odpovedá pohľad užívateľa na databázu, ktorá obsahuje tri základné tabuľky (pohľady). Prvá reprezentuje lokalitu. Jej atribútmi sú jedinečný kód lokality, názov lokality, súradnice, typ, popis a komentár a iné. Ďalšou tabuľkou je predmet záujmu, napríklad strom. Záznamy stromov sú definované kódom lokality, ku ktorej patrí a vlastným unikátnym kódom, taxónom, ale aj mnohými nepovinnými atribútmi, napr. výškou, súradnicami, typom pôdy a pod. Tretou tabuľkou je tabuľka vzoriek odobraných v tomto prípade zo stromu. Opäť je nutné pri každom zázname definovať kód lokality a kód stromu, ku ktorým daná vzorka prináleží a takisto priradiť každej vzorke vlastný unikátny kód a typ. Opäť nechýbajú doplňujúce atribúty.

Následne je treba vyplniť tabuľky odpovedajúce zvyšným dátovým entitám TRiDaS s názvom *Radius* a *Series*. *Radius* je odvodený z práve jednej vzorky a predstavuje líniu od drene po kôru, ktorá môže byť meraná mnohokrát. Dátová entita *Series* je séria nespracovaných meraní po rádiuse pre každý letokruh. Namerané hodnoty sa ukladajú do zvlášť určenej entity združujúcej hodnoty. Tu je zaznamenávaná aj jednotka nameraných

hodnôt. Existuje aj entita odvodených sérií (*Derived Series*), ktorej hodnoty sú odvodené z jednej alebo viacerých sérií nespracovaných meraní a môžu predstavovať, napr. index (Jansma et al. 2010, Brewer 2016).

Ručný import metadát a dát nie je jedinou možnosťou, do databázy ich je možné importovať aj zo súborov s rôznymi formátmi. Pri spojení Tellervo s meracím prístrojom je dokonca možné namerané hodnoty do databázy automaticky odosielať z prístroja, ktorý danú vzorku práve meria.

Tellervo podporuje *Open Data Kit* (ODK), čo je súbor nástrojov určených na zber dát mobilnými zariadeniami. Využívané sú dva nástroje ODK: 1) *ODK Build*, ktorý je dostupný priamo v aplikácii Tellervo, vytvára formuláre, do ktorých sa zaznamenávajú informácie o lokalite a stromoch (resp. vzorkách); 2) *ODK Collect*, ktorý je dostupný pre Android smartfóny a tablety. Pri jeho spustení používateľ zadá URL adresu Tellervo Serveru a prihlasovacie údaje. Následne sa stiahnu vytvorené formuláre z *ODK Build*, ktoré užívateľ môže v teréne začať vyplňať hodnotami. Keďže v teréne je málokedy dostupné internetové pripojenie, úplne absentuje kontrola vkladaných dát, ako napríklad kontrola duplicity unikátnych kódov stromov. Po pripojení na internet je možné zozbierané dáta nahráť na Tellervo Server (Brewer and Guiterman 2016).

Databáza oplýva rôznymi funkciami skonštruovanými prevažne v PLJava, ktoré sa využívajú na komplexné procesy, kalkulácie a analýzy v databáze (Brewer 2016).

6. Dáta a použitý softvér

Dendrochronologické dáta používané v tejto práci boli poskytnuté Katedrou fyzickej geografie a geoekológie PrF UK (KFGG). Konkrétne sa jednalo o dataset Kokořínsko, ktorý obsahuje dáta z 20 lokalít zozbierané v severnej časti CHKO Kokořínsko v okolí vrchu Kostelec na rôznych častiach reliéfu – údolie, plošina, južný alebo severný svah. Na všetkých lokalitách boli skúmané borovice lesné. Na veľkej väčšine lokalít kolísal počet vzorkovaných stromov okolo čísla 26, na niektorých lokalitách to bolo výrazne viac. Atribúty lokalít tvorili názov a kód lokality, jej zemepisné súradnice a nadmorská výška.

Pre každý strom sú v datasete dostupné dendrometrické údaje, ktoré tvoria základné dendrometrické parametre, ako je výška stromu, koruny, priemer kmeňu v prsnej výške a pod.

Lokalitám nechýbajú ani nespracované a ani už štandardizované (detrendované) letokruhovú chronológiu pozostávajúce z hodnôt širok letokruhov danej vzorky (patriacej určitému stromu na lokalite) v danom roku. Metódou detrendovania bola *spline* krivka s dĺžkou okna 50 rokov.

Dve lokality sa odlišovali od zbytku tým, že z každého stromu nebola odobratá len jedna vzorka, ale boli vykonané dva vývrty a to z východu a z juhu. Navyše na týchto lokalitách bola nadmorská výška, sklon a orientácia zaznamenávaná na úrovni stromov, a na jednej z nich boli dokonca zaznamenávané aj súradnice stromov.

Dataset Kokořínska teda obsahuje 20 lokalít, pre každú nespracovanú, ale aj štandardizovanú chronológiu širok letokruhov s výnimkou dvoch lokalít, ktoré disponujú len nespracovanou chronológiou, 644 stromov s dendrometrickými údajmi a 764 vzoriek.

6.1. Použitý softvér

Použitým softvérom v tejto práci je už v kapitole 3.4.1. bližšie spomínaný PostgreSQL 13 s priestorovým rozšírením PostGIS 3.1, v ktorom bola tvorená priestorová databáza letokruhovú chronológií. Okrem neho bol využitý softvér R 4.0.4 (R Core Team 2020) s rozšírením dplR (Bunn 2008), ktorý bol využitý na manipuláciu so súbormi letokruhovú chronológií. ArcGIS Pro 2.7.0 bol použitý na vizualizáciu priestorových dát uložených vo výslednej databáze.

6.1.1. R

R (R Core Team 2020) je voľne dostupná integrovaná kolekcia softvérov a zároveň aj programovací jazyk využívaný na manipuláciu a kalkuláciu s dátami prevažne v oblasti štatistiky, ale aj na ich grafickú prezentáciu pomocou množstva vbudovaných grafických techník. Taktiež ponúka možnosť rozširiteľnosti pomocou *balíkov* (R-project 2021).

Balík dplR (Bunn 2008) je určený pre dendrochronológov a obsahuje množstvo funkcií využívaných na analýzu a manipuláciu s letokruhovými dátami, ako napríklad vykonávanie krížového datovania, tvorba chronológií, detrendovanie hodnôt alebo zápis dát do štandardných dendrochronologických súborových formátov (Bunn and Korpela 2021).

6.1.2. ArcGIS Pro

ArcGIS Pro ponúka možnosť pripojenia k databáze za účelom vizualizácie, analýzy, dotazovania, filtrovania a manažovania dát uložených v pripojenej databáze. Pri pripojovaní sa k databáze ju nutné vyplniť vlastnosti pripojenia, ako je typ databázovej platformy (napr. Oracle, PostgreSQL, DB2 a iné), názov databázy, prihlasovacie údaje a názov alebo IP adresu serveru, na ktorom je PostgreSQL nainštalovaný. Po úspešnom pripojení sa k databáze je možné z dát z databázovej tabuľky spraviť vrstvu a tú vizualizovať alebo podrobovať rôznym analýzami. V databáze je možné aj vytvárať nové tabuľky vrstiev, atribútové tabuľky a definovať databázové pohľady pre tabuľky v databáze, upravovať schému tabuliek, samozrejme všetko len s patričnými právami prihláseného užívateľa. Možné je aj prekopírovanie dát z jednej databázy do druhej, prípadne do geodatabázy alebo zverejnenie dát pomocou ArcGIS Server (ESRI 2021a, ESRI 2021b).

7. Tvorba databázy letokruhových chronológií

Táto kapitola popisuje vznik priestorovej databázy letokruhových chronológií od samotných dátových požiadaviek, cez ich analýzu, návrh databázy, jej implementáciu, tvorbu funkcií slúžiacich na vklad dendrochronologických dát až po konštrukciu dátového filtru.

7.1. Dátové požiadavky

Dátové požiadavky pre výslednú databázu letokruhových chronológií boli vznesené zo strany KFGG formou zoznamu atribútov, ktoré by sa mali dať v databáze uchovávať. Zoznam tvorili:

- ❖ Kód lokality (*Site code*)
- ❖ Kód stromu (*Tree ID*)
- ❖ Zemepisné súradnice (*Latitude, Longitude*)
- ❖ Názov lokality (*Site name*)
- ❖ Nadmorská výška (*Elevation*), Sklon (*Slope*), Orientácia (*Aspect*)
- ❖ Hĺbka koreňov (*Rooting depth*)
- ❖ Pôdny typ (*Soil*)
- ❖ Skratka druhu (*Species*)
- ❖ Výška stromu (*Height*), výška koruny (*Crown height*), polomer koruny (*Crown projection*), priemer kmeňu vo výške prs (*DBH*)
- ❖ Kompetičný index (*Social status*)
- ❖ Šírka beľového dreva (*Sapwood*)
- ❖ Chronológia nespracovaných meraní širok letokruhov (*Raw ring-width measurement*)
- ❖ Chronológia detrendovaných meraní širok letokruhov (*Standardized ring-width measurement*)

V zátvorke za názvami atribútov je uvedený ich anglický ekvivalent pre lepšiu orientáciu v rámci práce, nakoľko samotné atribúty v databáze nesú ich anglické názvy.

7.2. Analýza dát

Pre akýkoľvek návrh databázy je nutné spoznať dáta, pre ktoré je databáza navrhovaná. Je zrejmé, že z pohľadu na vyššie uvedený zoznam atribútov sa bez dostatočných poznatkov

nedá odvodit' podoba dát alebo pre návrh databázy úplne kľúčové vzťahy medzi jednotlivými atribútmi.

V tomto smere aspoň v určitej miere pomohla rešerš existujúcich dendrochronologických databáz, ale aj pochopenie základných princípov dendrochronológie. V uvedených zdrojoch sa ale nepodarilo objasniť všetky nejasnosti a zistilo sa, že podoba dát a vzťahy atribútov uvedených v zozname vyššie, môžu byť odlišné, či už v závislosti na výskumnej otázke alebo v prípade niektorých atribútov môžu byť odlišné dôsledkom rozhodnutí zberateľa dát. Záver prieskumu dostupných zdrojov nebol teda v mnohých smeroch dostatočne jednoznačný.

Dôsledkom toho boli dátové požiadavky a celá zamýšľaná koncepcia databázy prekonzultovaná s členmi KFGG. Okrem odpovedí na všetky otázky ohľadom dát bolo výsledkom aj poskytnutie celého datasetu z Kokořínska, ktorý je bližšie popisovaný v kapitole 6. Mať pred sebou konkrétne dáta z KFGG, pre ktoré je koniec koncov databáza navrhovaná, bolo najväčším krokom vpred.

7.3. Konceptuálny model

Výsledkom analýzy dát je konceptuálny model databázy, ktorého úlohou je znázorniť objekty a vzťahy v časti reálneho sveta, ktorá je predmetom záujmu, v tomto prípade v dendrochronológii. Ako typ konceptuálneho modelu bol zvolený *entitno-relačný diagram* (ERD) so Chenovou notáciou.

Tvorba ERD začala určením objektov reálneho sveta – *entít*, ktoré boli odvodené zo spozorovanej hierarchie jestvujúcej v dendrochronologických dátach:

Druh stromu → Lokalita → Strom → Vzorka → Meranie

Druh stromu bol pri tvorbe entít vynechaný, nakoľko nie je nutné ho považovať za entitu.

Následne boli vytvoreným entitám priradené prislúchajúce *atribúty*, či už z uvedeného zoznamu alebo iné, prebraté zo samotných dát. Lokalite bol priradený kód, ktorý je jej unikátnym atribútom, názov lokality, zemepisné súradnice, nadmorská výška, sklon, orientácia, pôdny typ, a aj hĺbka koreňov, ktorá je zaznamenávaná na úrovni lokalít. Atribútom tejto entity je aj druh stromu, ktorý bol preskočený v predošlom kroku tvorby ERD.

Entita stromov oplýva kódom stromu, ktorý ale nebol prehlásený unikátnym atribútom, pretože je bežné číslovať stromy celými číslami (v rozmedzí podľa počtu stromov na lokalite), čo spôsobuje, že kód stromu nemusí byť unikátny naprieč lokalitami. Ďalšími atribútmi entity stromov sú nielen dendrometrické údaje – výška stromu, výška koruny, polomer koruny, DBH, kompetičný index, šírka beľového dreva, ale opäť aj zemepisné súradnice, nadmorská výška, sklon a orientácia.

Entita vzoriek obsahuje len jeden atribút a to kód vzorky, ktorý nie je unikátny z rovnakých dôvodov ako kód stromu. Veľmi ľahko sa môže stať, že kód vzorky nie je unikátny ani medzi stromami na jednej lokalite.

Meranie v tomto prípade predstavuje časovú radu širok letokruhov, pričom vznikli pre neho dve entity, jedna pre nespracované chronológie a druhá pre štandardizované (detrendované) chronológie. Ich atribúty sú totožné a to rok a hodnota šírky letokruhu pre daný rok.

Ďalej bolo nutné definovať *vzťahy* (väzby) medzi entitami. Entity boli prepojené tak, ako uvádza vyššie uvedená hierarchia.

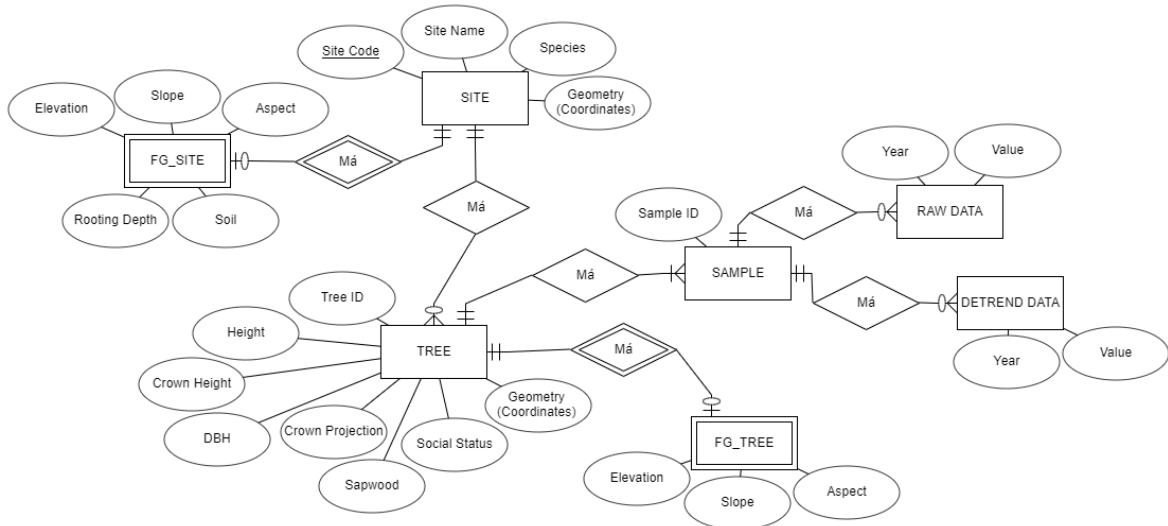
So vzťahmi sa spája aj *kardinalita* a *parcialita*. Entita lokalít a stromov je vo vzťahu 1:N, pretože na lokalite je vzorkovaných viacero stromov, ale strom patrí práve jednej lokalite. Povinnú účasť má len lokalita, pretože strom nemôže byť bez lokality, ale lokalita bez stromu (teoreticky) áno. Strom a vzorka sú opäť vo vzťahu 1:N, ako bolo už načrtnuté - zo stromu môže byť vytvorených viacero vzoriek ale vždy aspoň jedna a vzorka prináleží práve jednému stromu. Medzi vzorkou a obidvomi entitami meraní je vzťah 1:N, ale len s povinnou účasťou vzorky, pretože pri meraní môže byť vzorka aj vynechaná, napríklad z dôvodu zlej zhody s referenčnou chronológiou.

7.3.1. Normalizácia

Už pri tvorbe ERD sa predvídal jeho prevod do relačnej schémy a s tým súvisiaca normalizácia relačných tabuliek, za účelom zbaviť sa nežiadúcej redundancie dát a zabezpečiť integritu dát. Pre atribúty nadmorskej výšky, sklonu, orientácie, hĺbky koreňov a pôdneho typu pri entite lokalít bola vytvorená zvlášť entita pomenovaná *fg_site*, aby sa eliminoval výskyt tranzitívnej závislosti s atribútom zemepisných súradníc. Z rovnakého dôvodu entitu stromov opustili atribúty nadmorskej výšky, sklonu a orientácie stromov za vzniku entity s názvom *fg_tree*. Novovzniknuté entity sú entity slabého typu so vzťahom

k príslušnej silnej entite 1:1 a s povinnou účasťou silnej entity a nepovinnou účasťou slabej entity.

Takouto úpravou bola dosiahnutá tretia normálna forma, avšak výsledný ERD (pozri Obrázok 5) je normalizovaný minimálne do Boyce-Coddovej normálnej formy.



Obrázok 5: Entitno-relačný diagram navrhovanej databázy.

Zdroj: vlastné spracovanie

7.4. Logický model

Aby vznikla výsledná relačná databáza bolo nutné konceptuálny model previesť do relačnej schémy, to znamená definovať schémy relácií (tabuliek), stanoviť primárne a cudzie kľúče resp. integritné obmedzenia, atribútom (stĺpcom) priradiť dátové typy, resp. definovať ich domény a pod.

Pre každú entitu z navrhnutého ERD modelu bola vytvorená rovnomenná tabuľka, ktorej stĺpce tvorili atribúty spojené s danou entitou v ERD. V tabuľke lokalít je primárnym kľúčom jej kód tvorený znakovým reťazcom. Atribút skratky druhu je obmedzený štyrmi znakmi, čo je maximálna dĺžka každej existujúcej skratky druhu a nemôže byť bez hodnoty. Prázdnej hodnoty nemôže byť ani názov lokality. Zemepisné súradnice lokalít sú zaznamenávané v atribúte *geom*, ktorý je geometrického dátového typu *Geometry* s obmedzením na geometrický typ *bod* a na SRS *WGS84 (lon/lat)*.

Keďže entita *fg_site* je slabého typu a žiaden jej atribút nedokáže unikátne identifikovať každý jej riadok (záznam), tak primárny kľúč jej tabuľky je kód lokality prevzatý z tabuľky jej silnej entity, do ktorej zároveň ako cudzí kľúč odkazuje. Atribúty

sklonu a orientácie sú obmedzené hodnotami, ktoré môžu reálne dosahovať, to znamená sklon od 0° do 90° a orientácia od 0° do 360°, ale možná je aj hodnota orientácie -1, ktorú môžu dosahovať ploché plochy bez zostupného svahu.

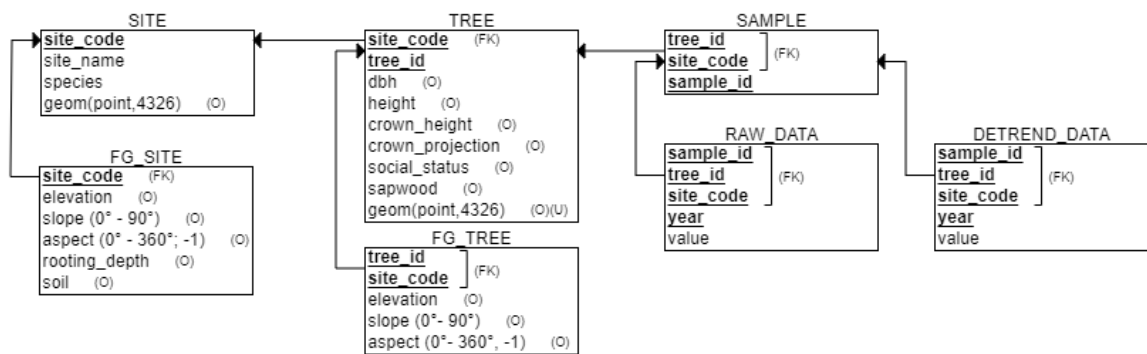
Tabuľka stromov má zložený primárny kľúč z kódu stromu a kódu lokality. Druhý menovaný je cudzím kľúčom odkazujúcim do tabuľky lokalít. Kód stromu môže dosahovať maximálnu dĺžku šesť znakov, čo je štandardom v dátach. Dendrometrické údaje sú numerického dátového typu. Zemepisné súradnice sú zaznamenávané presne tak isto ako v tabuľke lokalít – v atribúte *geom*, ale s jediným rozdielom a to, že súradnice stromu (hodnota atribútu *geom*) musia byť unikátne, pretože dva stromy nemôžu mať rovnakú polohu. Dodajme, že pri súradniciach lokalít by takéto obmedzenie nebolo správne, pretože ak sa na lokalite vzorkuje viac druhov, pre každý jeden druh vzniká nový záznam pre lokalitu (nová lokalita), ale ich zaznamenané súradnice môžu byť potenciálne rovnaké, aj keď sa takéto prípad takmer nevyskytuje.

Zložený primárny kľúč tabuľky stromov je takisto primárnym kľúčom tabuľky *fg_tree*, pričom je v nej (celý) zároveň aj cudzím kľúčom do tabuľky stromov. Atribúty sklonu, orientácie a súradníc sú riešené presne tak isto ako v tabuľke *fg_site*.

Atribút kód lokality a kód stromu vstupujú aj do tabuľky vzoriek, kde sú jednak cudzím kľúčom do tabuľky stromov ale, spoločne s maximálne šesťmiestnym kódom vzorky tvoria primárny kľúč tejto tabuľky.

Primárnym kľúčom tabuliek surových a detrendovaných chronológií sú atribúty kód lokality, kód stromu, kód vzorky spoločne so stĺpcom rok. Jedine tak je možné unikátne rozlíšiť záznamy v týchto tabuľkách, pretože určitá vzorka stromu má šírky letokruhov zaznamenané pre množstvo rokov, ale nikdy nemá dve hodnoty pre jeden rok (časová rada). Okrem atribútu rok sú zvyšné atribúty primárneho kľúču cudzím kľúčom odkazujúcim do tabuľky vzoriek.

Obrázok 6 predstavuje výslednú relačnú schému databázy. Tučným a zároveň podčiarknutým písmom sú znázornené primárne kľúče tabuliek. Cudzí kľúč tabuľky je označený iniciálovou skratkou FK (*Foreign Key*) v zátvorke za názvom atribútu, pričom šípka znázorňuje do ktorej tabuľky, resp. na ktorý atribút v nej, cudzí kľúč referuje. Písmeno O (*Optional*) v zátvorke za atribútom znamená, že daný atribút môže zostať prázdnu hodnotou, písmeno U (*Unique*) hovorí, že hodnota atribútu musí byť unikátna.



Obrázok 6: Relačná schéma navrhovanej databázy.
Zdroj: vlastné spracovanie

Po relačnej schéme databázy boli vytvorené aj referenčné akcie, ktoré sa vykonajú ak dôjde k porušeniu integrity cudzích kľúčov v tabuľkách obsahujúcich cudzí kľúč. Podmienkou ich spustenia je aktualizácia alebo zmazanie záznamu v referovanej tabuľke, následne prebehne aktualizácia, resp. zmazanie dotknutého záznamu v referujúcej tabuľke. Pre priestorové atribúty *geom* bol vytvorený index typu *Generalized Search Tree* (GiST), ktorý je najvhodnejší pre priestorové dátové typy.

7.5. Vklad dát do databázy

Po implementovaní relačnej schémy a teda vytvorení databázy v PostgreSQL/PostGIS je namieste sa zaoberať vkladom dát do nej. Pri pohľade na schému databázy (pozri Obrázok 6) je zrejmé, že vďaka vzťahom musí vklad akýchkoľvek dát začínať od vrcholu hierarchie, teda zľava smerom doprava.

7.5.1. Vklad záznamov lokalít a stromov

Pre vklad záznamov týkajúcich sa lokalít a stromov boli vytvorené dva pohľady. Prvý pohľad zjednocuje tabuľky lokalít – *site*, *fg_site* a je určený pre vklad záznamov lokalít. Druhý pohľad zjednocuje dvojicu tabuliek stromov a je určený pre vklad záznamov stromov. Pohľady boli pomenované *sites* a *trees*. Keďže sa jedná o pohľady tvorené dvomi tabuľkami, PostgreSQL do takýchto pohľadov neumožňuje vklad, úpravu a mazanie dát (záznamov).

Aby sa dokázali vytvorené pohľady využiť na vklad dát, bolo nutné vytvoriť *trigger*, ktorý po vklade dát do pohľadu spustí PL/pgSQL funkciu, ktorá automaticky zabezpečuje, že vložené hodnoty do atribútov týchto pohľadov budú zároveň roztriedené do príslušných databázových tabuliek, do ktorých patria. V tabuľkách, ktoré pohľady zjednocujú, sa nachádza aj priestorový atribút *geom* dátového typu *Geometry*, do ktorého sú zaznamenávané zemepisné súradnice. Aby ale súradnice mohli byť vložené do tohto

atribútu, tak je nutné ich mať zapísané v jednom z formátov, ktorý je systémom podporovaný pre reprezentáciu súradníc a ich SRS (napr. WKT). Užívateľ ale vo svojich súboroch zemepisné súradnice lokalít, resp. stromov, reprezentuje klasicky číselne, čo pre vklad do *geom* nestačí. Z tohto dôvodu vytvorené pohľady obsahujú aj dva atribúty navyše a to konkrétne atribúty *latitude* a *longitude* tvorené PostGIS funkciou *ST_Y*, resp. *ST_X* a vychádzajúce z atribútu *geom*. Opisovaná PL/pgSQL funkcia po vložení číselne zapísaných súradníc do atribútov *latitude* a *longitude* v pohľadoch ich hodnoty automaticky prevedie do vyhovujúcej reprezentácie, pomocou ktorej súradnice automaticky vloží do priestorového atribútu *geom* nielen v pohľade, ale samozrejme aj v databázovej tabuľke *site* resp. *tree*.

Vytvorený pohľad *sites* s PL/pgSQL funkciou s *triggerom* umožňujúcou vklad záznamov doň a s prevodom číselne zapísaných súradníc do vhodnej reprezentácie je v Prílohe 1. V Prílohe 2 je uvedená funkcia pre pohľad *trees*.

Druhá PL/pgSQL funkcia s *triggerom* umožňuje, aby sa v pohľade dali mazať záznamy. Po vyvolaní takejto požiadavky na pohľad sa následne požadovaný záznam z pohľadu zmaže a vďaka funkciám sa daný záznam identifikuje aj v databázových tabuľkách, ktoré daný pohľad spája a dotknutý záznam sa zmaže aj v nich (pozri Príloha 3). Úprava záznamov v pohľadoch nie je umožnená, nakoľko sa to neodporúča. Riešením je zmazanie záznamu a jeho opätovný vklad už s požadovanými úpravami.

V praxi vklad záznamov lokalít a stromov pomocou vytvorených pohľadov je oproti vkladu záznamov do každej tabuľky zvlášť tou lepšou alternatívou a je odporúčané ju primárne využívať. Jednak z dôvodu, že užívateľovi nemusí byť známa štruktúra databázy a schéma tabuliek, ale výhoda je aj pri vklade dát skopírovaním zo súboru. Užívateľ nemusí mať kvôli vlastnosti SQL funkcie *COPY FROM* viacero súborov s rovnakou štruktúrou ako sú databázové tabuľky a obsah súborov kopírovať do príslušných tabuliek. Stačí mu jeden súbor, ktorý bude obsahovať dáta pre všetky atribúty patriace lokalitám, resp. stromom, ktorého obsah nakopíruje do pohľadu *sites*, resp. *trees* a systém si už vďaka vyvinutým funkciám s distribúciou záznamov do správnej databázovej tabuľky poradí sám. Navyše pohľady si poradia aj s prevodom číselne zapísaných súradníc do správnej reprezentácie a ich vložení do atribútu *geom*, čo by užívateľ, ak by si zvolil vklad dát do databázových tabuliek zvlášť, musel robiť ručne.

7.5.2. Vklad záznamov vzoriek

Po vložení záznamov lokalít a stromov je nutné vyplniť tabuľku vzoriek. V nej treba vytvoriť záznamy, ktoré pridelujú určitému stromu na určitej lokalite kód alebo kódy vzoriek, ktoré sa nachádzajú (a identifikujú záznamy) v súboroch nespracovaných a detrendovaných chronológií (súbory s príponou RWL), ktoré budú následne vkladané do databázy.

7.5.3. Vklad chronológií širok letokruhov

Vklad chronológií širok letokruhov do databázy je proces zložitejší a je potreba ho popísať postupne, krok po kroku.

Prevod chronológií do formátu CSV

Chronológie hodnôt širok letokruhov jednotlivých vzoriek sú ukladané v súboroch vo formáte *Tucson* s príponou RWL, z ktorých PostgreSQL nedokáže skopírovať dáta do databázy. Z tohto dôvodu bolo nutné tieto súbory previesť do formátu *Comma-separated values* (CSV).

Prevod súborov RWL do formátu CSV bol vykonaný vytvoreným skriptom v programe R s využitím rozšírenia *dplR*. Tento dendrochronologický balík obsahuje funkciu, ktorá umožňuje načítať RWL súbor do R. Hodnoty širok letokruhov sú v súbore RWL zobrazované celými číslami, ale po načítaní súboru v R a zobrazení jeho hodnôt sú hodnoty desatinné. Spôsobuje to jednotka merania, ktorou sú stotiny milimetra, prípadne tisíciny milimetra, ktorú RWL súbor pri jeho otvorení a zobrazení hodnôt v ňom berie do úvahy, no program R nie, čo spôsobuje, že CSV súbor vyexportovaný z R bez úpravy by obsahoval desatinné hodnoty, čo nie je žiadúce. Skript teda okrem načítania súboru RWL do R obsahuje aj príkazy na posun desatinnej čiarky, pričom užívateľ si vyberie, ktorý z nich využije a to na základe jednotky, v ktorej sú hodnoty širok letokruhov v danom súbore zaznamenané. Takisto bol vytvorený príkaz, ktorý umožní v prípade potreby zaokrúhliť tieto hodnoty na ľubovoľný počet desatinných miest. Následne skript obsahuje príkaz s funkciou na export načítanej (a upravenej) chronológie do CSV súboru.

Opisovaný R skript bol doplnený príkazmi pre vykonanie procedúry štandardizácie (detrendovania) hodnôt širok letokruhov rôznymi metódami podľa výberu užívateľa, ktoré boli prevzaté zo skriptov využívaných dendrochronológmi KFGG.

Užívateľ pri vyvolaní funkcie *load_csv_file* musí zadať tri parametre funkcie. Prvým je názov fázovej tabuľky, ktorý nesmie obsahovať žiadne veľké písmená a nesmie začínať číslom. Druhým je cesta ku (už patrične upravenému) CSV súboru, resp. umiestnenie CSV súboru v súborom systéme. Posledným parametrom je počet vzoriek v danom súbore chronológií. Vyvolanie funkcie *load_csv_file* je možné demonštrovať na príklade lokality 1SS z dát Kokořínska, ktorá obsahuje 26 vzoriek, jednak pre CSV súbor s nespracovanou chronológiou a aj pre súbor s detrendovanou chronológiu:

```
SELECT load_csv_file ('ko_1ss', 'D:/kokorinsko/1SS.csv', 26);  
SELECT load_csv_file ('ko_1ss_d', 'D:/kokorinsko/1SS_det.csv', 26);
```

Funkcia v prvom kroku vytvorí fázovú tabuľku v databázovej schéme *import* a vytvorí v nej taký počet stĺpcov, koľko vzoriek má daný súbor, teda podľa čísla, ktoré užívateľ definoval do tretieho parametru funkcie. Pre správnosť treba dodať, že funkcia automaticky vytvorí v tabuľke ešte jeden stĺpec navyše, určený pre hodnoty rokov chronológie. Následne do vytvorenej tabuľky pomocou COPY FROM skopíruje dáta zo súboru CSV spoločne s názvami stĺpcov, ktoré v ňom predstavujú kódy vzoriek a ktoré vo fázovej tabuľke budú uložené v prvom riadku. Stĺpce tabuľky sa automaticky premenujú na kódy vzoriek z prvého riadku, ktorý sa následne zmaže. Práve z tohto dôvodu je nutné upraviť názov prvého stĺpcu v CSV súbore, pretože systém nedokáže vytvoriť stĺpec bez názvu. Názov stĺpcu podľa kódu vzorky je dôležitý, aby sa dalo v tabuľke rozlíšiť, ktorý stĺpec nesie hodnoty širok letokruhov ktorej vzorky, nakoľko v samotnom CSV súbore sa o nejakom konzistentnom poradí stĺpcov podľa kódu vzorky nedá hovoriť. Následne sa tabuľka premenuje podľa názvu uvedeného v prvom parametre funkcie. Jej posledným krokom je zmena dátových typov všetkých stĺpcov fázovej tabuľky z *text* na *numeric*. Ak by sa takáto zmena nevykonala, neskôr po prekopírovaní hodnôt širok letokruhov do databázovej tabuľky, by sa tieto hodnoty v nej tvárili ako text, čo je problém pri akomkoľvek dotaze na ich hodnoty.

Presun chronológií z fázovej tabuľky do tabuliek chronológií

Prevod dát z fázovej tabuľky do príslušnej databázovej tabuľky zabezpečujú ďalšie dve vytvorené PL/pgSQL funkcie. Prvú funkciu je nutné vyvolať, ak sa jedná o nespracované chronológie vo fázovej tabuľke, čo znamená, že takéto dáta z fázovej tabuľky je nutné prekopírovať do tabuľky určenej pre nespracované chronológie. Táto funkcia sa nazýva *insert_raw_data* (pozri Príloha 5). Druhú funkciu je nutné využiť v prípade detrendovaných

chronológií vo fázovej tabuľke a ich prenosu do tabuľky detrendovaných chronológií. Funkcia sa nazýva *insert_detrend_data* (pozri Príloha 6).

Pri vyvolávaní oboch funkcií užívateľ musí stanoviť ich štyri parametre. Prvý parameter predstavuje kód lokality, z ktorej pochádzajú dané vzorky v chronológií, resp. z ktorej lokality daná chronológia pochádza. Druhým parametrom je kód stromu a tretím je kód vzorky. Posledným štvrtým parametrom je názov fázovej tabuľky, z ktorej sa majú dáta kopírovať.

Už z týchto parametrov je jasné, že aby sa vložila celá chronológia do patričnej tabuľky, vyvolanie funkcie musí užívateľ vykonať opakovane, konkrétne toľkokrát, koľko je vzoriek na lokalite alebo lepšie povedané, koľko vzoriek obsahuje daná chronológia. Je nutné, aby užívateľ vypĺňal tieto prvé tri spomenuté parametre, pretože bez ani jedného z nich by sa nedokázal unikátne identifikovať záznam v tabuľke vzoriek, pretože primárnym kľúčom tabuľky sú atribúty kód lokality, kód stromu a kód vzorky, pričom tieto atribúty sú takisto primárnym kľúčom tabuliek chronológií a zároveň cudzími kľúčmi do tabuľky vzoriek. V praxi užívateľ pri využívaní funkcie len mení kód stromu a kód vzorky a spúšťa príkaz.

Vyvolanie funkcie *insert_raw_data* na prekopírovanie dát z fázovej tabuľky do tabuľky nespracovaných chronológií opäť na príklade lokality 1SS z dát Kokořínska:

```
SELECT insert_raw_data ('1SS', '1SS1', '001SS1', 'ko_1ss');
SELECT insert_raw_data ('1SS', '1SS2', '001SS2', 'ko_1ss');
...
SELECT insert_raw_data ('1SS', '1SS26', '01SS26', 'ko_1ss');
```

Použitie funkcie *insert_detrend_data* na prekopírovanie dát z fázovej tabuľky do tabuľky detrendovaných chronológií na príklade lokality 1SS z dát Kokořínska:

```
SELECT insert_detrend_data ('1SS', '1SS1', '001SS1', 'ko_1ss_d');
SELECT insert_detrend_data ('1SS', '1SS2', '001SS2', 'ko_1ss_d');
...
SELECT insert_detrend_data ('1SS', '1SS26', '01SS26', 'ko_1ss_d');
```

Opisované funkcie sú skonštruované tak, že obsahujú príkaz INSERT, ktorý vkladá užívateľom definovaný kód lokality, stromu a vzorky do tabuľky chronológie. Z definovanej fázovej tabuľky sa vezmú len tie hodnoty, ktoré sa nachádzajú v stĺpci, ktorého názov odpovedá definovanému kódu vzorky, ale len tie, ktoré sú nenulové. Spoločne s takýmito hodnotami širok letokruhov sa vezmú z fázovej tabuľky aj hodnoty rokov zo

stĺpcu “y” patriace jednotlivým hodnotám širok letokruhov. Názov stĺpcu rokov vo fázovej tabuľke do týchto funkcií vstupuje „natvrdo“, preto je veľmi dôležité, aby CSV súbor bol upravený správne, ako bolo popisované v predošlej kapitole.

V prípade, že kód stromu je totožný s kódom vzorky, tak kód vzorky nie je nutné pri vyvolávaní týchto funkcií znova špecifikovať, čím sa užívateľovi ušetrí práca. Príklad vyvolania funkcie v takomto prípade môže byť:

```
SELECT insert_raw_data ('ABC', '01', '', 'faz_tab');
```

7.6. Dátový filter

Za účelom vrátenia ľubovoľnej podmnožiny dát z databázy bol vytvorený univerzálny SQL dotaz, ktorý umožňuje užívateľovi stanoviť určité požiadavky na atribúty nachádzajúce sa v databáze, pričom mu umožňuje aj definovať množinu atribútov, ktorú chce užívateľ mať ako výstup tohto príkazu (pozri Príloha 7). Pre upresnenie dodajme, že nie je umožnené filtrovať na základe hodnôt atribútov patriacich tabuľkám chronológií, teda atribútom rok a hodnota šírky letokruhov, nakoľko je to zbytočné. V prípade chronológií sa dá ale filtrovať na základe dĺžky chronológie patriacej určitej vzorke vyjadrenej v počte rokov, čo bola explicitná požiadavka KFGG.

Príkaz bol skonštruovaný tak, že najprv v tabuľke chronológií sa vyfiltrujú záznamy spĺňajúce požadovanú dĺžku chronológie, pričom sa vrátia len záznamy atribútov kód lokality, stromu a vzorky. Je irelevantné v tomto kroku užívateľom špecifikovať na ktorú tabuľku chronológií, či nespracovaných alebo štandardizovaných, sa aplikuje výber podľa dĺžky chronológie, pretože dĺžka týchto dvoch typov chronológií je totožná. Následne sa z už užívateľom špecifikovanej tabuľky (typu) chronológií vyberú kompletne záznamy všetkých atribútov, ktoré tabuľky chronológií obsahujú, ale len tie záznamy, ktorých kód lokality, stromu a vzorky odpovedajú tým, ktoré spĺňajú požadovanú dĺžku chronológie. K takto vzniknutej tabuľke sa pripoja databázové tabuľky lokalít na základe kódu lokality a databázové tabuľky stromov na základe kódu stromu a kódu lokality. Nad touto tabuľkou zjednocujúcou všetky atribúty v databáze, sa vykonáva filter na základe vyhovujúcich hodnôt atribútov, podľa ktorých užívateľ chce filtrovať. Následne sa z takejto tabuľky vyberú len tie atribúty, ktoré užívateľ chce, aby tvorili jeho výstup.

8. Výsledky

Hlavným výsledkom práce je priestorová databáza letokruhových chronológií, ktorá vďaka svojmu normalizovanému návrhu, opierajúcim sa o analýzu dendrochronologických dát, spĺňa dátové požiadavky KFGG a zároveň sa vyvarováva zbytočnej redundancii dát. Databáza bola úspešne implementovaná pomocou PostgreSQL/PostGIS. Pomocou vyvinutých PL/pgSQL funkcií boli definované metódy vkladu dendrochronologických dát. Vďaka funkciám s *triggermi* a vytvoreným pohľadom bola zlepšená tvorba záznamov lokalít a stromov vrátane ich súradníc. Vklad letokruhových chronológií do databázy je zabezpečený funkciou *load_csv_file*, ktorá vytvára fázovú tabuľku, do ktorej sú chronológie pomocou SQL funkcie COPY FROM prekopírované z CSV súboru vzniknutého príkazmi v prostredí R s dendrochronologickým balíkom. Zvyšné dve funkcie *insert_raw_data* a *insert_detrend_data* zabezpečujú presun chronológií z fázovej tabuľky do správnej databázovej tabuľky.

Pomocou popísaných metód vkladu dát bola databáza úspešne naplnená všetkými poskytnutými dendrochronologickými dátami z KFGG, konkrétne celým datasetom Kokořínska. Databáza teda obsahuje dáta ohľadom 20 skúmaných lokalít, 644 vzorkovaných stromov, 764 vzoriek a pre každú lokalitu nespracovanú a aj detrendovanú chronológiu s výnimkou dvoch lokalít, ktoré obsahujú len nespracovanú chronológiu. Toto činí 90 487 záznamov v tabuľke nespracovaných chronológií a 58 119 záznamov v tabuľke detrendovaných chronológií.

Po naplnení databázy dátami bol pomocou utility *pg_dump* vytvorený obraz databázy a databáza bola za pomoci administrátora fakultného serveru prenesená na linuxový fakultný server DATAMAP, ktorý tvorí základ budovaného fakultného geoportálu. Vďaka jej preneseniu na server je možné databázu so správnymi prihlasovacími údajmi a heslom spojiť s ArcGIS Pro (prípadne s iným GIS softvérom), a vizualizovať priestorové dáta uložené v databáze ale aj vytvoriť si z nich vlastnú vrstvu a pracovať s ňou ako s inou bežnou vrstvou. Takýmto spôsobom vznikla mapa, už v databáze vložených, skúmaných lokalít zo severu CHKO Kokořínsko, ktorú možno vidieť v Prílohe 8.

Pomocou vytvoreného univerzálneho SQL dotazu, ktorý slúži ako dátový filter je možné vrátiť z databázy takú podmnožinu dát, akú užívateľ požaduje. Filtrovať sa dá na základe väčšiny atribútov v databáze a podľa jeho dátového typu je dostupná podmienka. V prípade atribútov so znakovými záznamami sa môže klásť požiadavka formou zoznamu

vyhovujúcich hodnôt, v prípade číselných hodnôt atribútov vyhovujúce záznamy sa definujú hodnotou od/do, prípadne sa dá nastaviť, že vyhovujúce budú aj prázdne hodnoty (pozri Tabuľka 1). Tento dotaz sa môže uplatniť v pripravovanej webovej aplikácii, ktorá umožní užívateľovi vykonávať analýzy, prípadne iné procesy na základe ním vybraných dát z databázy, ale aj na export ľubovoľných dát z databázy do napríklad textového súboru podľa vôle užívateľa. Nakoľko v Prílohe 7 je uvedený jeho univerzálny zápis, názorný príklad jeho funkčného použitia je v Prílohe 9.

Dáta, ktoré možno vybrať z DB (SELECT)	Dostupné podmienky výberu (WHERE)
Kód lokality	Zo zadaného zoznamu (IN)
Názov lokality	
Skratka druhu	
Zemepisná šírka lokality	Z rozsahu hodnôt od ... do ..., alebo neznáma (NULL) hodnota
Zemepisná dĺžka lokality	
Nadmorská výška lokality	
Sklon lokality	
Orientácia lokality	
Hĺbka koreňov	Zo zadaného zoznamu (IN), alebo neznáma (NULL) hodnota
Pôdny typ	
Kód stromu	Zo zadaného zoznamu (IN)
DBH	Z rozsahu hodnôt od ... do ..., alebo neznáma (NULL) hodnota
Výška stromu	
Výška koruny	
Polomer koruny	
Kompetičný index	
Šírka beľového dreva	
Zemepisná šírka stromu	
Zemepisná dĺžka stromu	
Nadmorská výška stromu	
Sklon stromu	
Orientácia stromu	Zo zadaného zoznamu (IN)
Kód vzorky	
Nespracovaná chronológia letokruhových širok (rok + hodnota)	
Detrendovaná chronológia letokruhových širok (rok + hodnota)	Dĺžka chronológie aspoň ... rokov

Tabuľka 1: Prehľad atribútov, ktoré možno vybrať z databázy a dostupné podmienky aplikovateľné na ich záznamy.

Zdroj: vlastné spracovanie

9. Diskusia

Konceptuálny model a tým pádom aj celý návrh databázy sa prevažne opiera o poskytnuté dáta z Kokořínska. So spomínanou určitou variabilitou podoby dendrochronologických dát by sa mohlo potenciálne jednať o problém, ale v tomto smere je nutné zdôrazniť, že dataset Kokořínsko je dostatočne reprezentatívna vzorka dát, nakoľko obsahuje všetky možné varianty dát, ktorými KFGG disponuje a pre ktorú je databáza tvorená. Pripomeňme napríklad rôznu úroveň, či už lokalít alebo stromov, na ktorej sú zaznamenávané súradnice. Výsledná databáza s takýmito vlastnosti počíta.

Pri návrhu databázy boli využité len dve vrstvy modelovania databázy a to konceptuálny model a logický model. Mnoho zdrojov uvádza aj tretí, fyzický model, ktorý je ale častokrát pri tvorbe relačných databáz vynechávaný. Napríklad Coronel and Morris (2015) uvádza, že pri tvorbe relačnej databázy sa jej tvorca nemusí zaoberať fyzickým modelom, pretože relačný model je silno zameraný na logický model (na rozdiel od svojich predchodcov), ale implementácia relačného modelu si môže vyžadovať určitý zásah do fyzického modelu za cieľom zlepšenia výkonu. Medzi zlepšenie výkonu patrí indexovanie. Vytvorenie indexov pre primárne kľúče tabuliek by malo zmysel, pretože vo vytvorenom dátovom filtri je na ich základe vykonávaný JOIN, ale nebolo to nutné, pretože v použitom PostgreSQL sú indexy vytvorené automaticky pre každý atribút s unikátnym obmedzením alebo pre každý atribút, ktorý je prehlásený za primárny kľúč tabuľky (Postgresql 2021b). Indexovanie priestorových atribútov *geom* ale bolo vykonané, ale v tejto práci bolo zaradené do tvorby logického modelu.

Pre implementáciu navrhovanej databázy bol použitý objektovo-relačný SRBD PostgreSQL s priestorovým rozšírením PostGIS z dôvodu, že sa jedná o voľne dostupný softvér, ktorý je zároveň považovaný za jeden z najlepších vôbec a je využívaný aj svetovými organizáciami, či už sa bavíme o samotnom SRBD alebo jeho priestorovom rozšírení. Wikipedia (2021) uvádza ako používateľov PostgreSQL napríklad Reddit, Instagram či OpenStreetMap. Kvalite a funkčnosti PostGIS pridáva aj riadenie sa OGC normami. Rolu hral aj fakt, že fakultný server využíva práve tento softvér, čo umožnilo vytvorenú databázu naň previesť a databázu spojiť s GIS softvérom. Navyše spojenie GIS softvéru s databázou takisto nie je umožnené pre všetky databázové systémy. PostgreSQL/PostGIS je ale bežne podporovaný GIS softvérm, nielen ArcGIS, ale aj QGIS, SAGA GIS, GRASS GIS a mnohými inými (Strobl 2008).

Funkčnosť definovaných metód vkladu dát do databázy bola overená vykonaním vkladu dát z datasetu Kokořínska do databázy. Funkcie majú aj určité nedostatky v oblasti automatizácie, hlavne čo sa týka vkladu letokruhových chronológií. Pripomeňme nutnosť upravovať názov prvého stĺpcu v ich CSV súbore alebo nutné opakované vyvolávanie vytvorenej PL/pgSQL funkcie (funkcií) na presun chronológií z fázovej tabuľky do príslušnej databázovej tabuľky. Veľmi dôležité je spomenúť, že vyvinuté PL/pgSQL funkcie pre vklad chronológií do databázy využívajú SQL príkaz COPY FROM, ktorý je použiteľný len pre užívateľa s právami *superuser*.

Vytvorený dátový filter do určitej miery odpovedá rozšírenému vyhľadávaniu v ITRDB popisovaného v kapitole 5.1. Spoločným znakom je možnosť nastaviť vyhovujúce hodnoty atribútov od/do a nastavenie, či musia byť splňané všetky požiadavky zároveň alebo len určité množstvo z nich. Takisto je zachovaná voľba druhu stromu a dĺžky chronológie, kde je ale prítomný mierny rozdiel, pretože v ITRDB užívateľ vyhovujúcu chronológiu nenastavuje počtom rokov (dĺžkou) ale jej rozsahom (od/do). Vytvorená databáza ako celok sa do značnej miery podobá databáze Tellervo popisovanej v kapitole 5.3 a to vďaka svojej štruktúre a zároveň aj implementácií pomocou PostgreSQL/PostGIS.

10. Záver

Hlavným cieľom tejto bakalárskej práce bola tvorba priestorovej databázy letokruhových chronológií. Jej tvorbe avšak predchádzala v úvode práce literárna rešerš so zameraním sa na základy databázových systémov, ako je popis ich komponentov, typov a ich stručná história. Nasledovali možnosti uchovávanía priestorových dát v relačných databázach s bližším popisom prístupu opierajúcim sa o OGC normy, ktorý využíva v práci použitý PostGIS. Druhá časť rešerše sa venuje problematike dendrochronológie, konkrétne základným princípom letokruhov, tvorbe letokruhových chronológií a takisto ich využitiu v rámci dendroklimatológie. V neposlednom rade boli popisované a charakterizované už existujúce dendrochronologické databázy.

Metodická časť práce postupne popisuje implementáciu tvorenej priestorovej databázy. Začína oboznámením sa s dátovými požiadavkami definovanými KFGG a poskytnutým datasetom z Kokořínska, pokračuje analýzou a popisom dát za účelom vytvoriť návrh databáze formou ERD. Nasledoval prevod návrhu databáze na relačnú schému a jej implementáciu v objektovo-relačnom PostgreSQL s priestorovým rozšírením PostGIS. Ďalším krokom bolo definovanie metód vkladu dát prostredníctvom vytvorených PL/pgSQL funkcií, pomocou ktorých boli do vytvorenej databázy vložené poskytnuté dáta z Kokořínska, čo poslúžilo aj ako overenie funkčnosti vyvinutých funkcií. Poslednou časťou metodickéj časti je skonštruovanie SQL príkazu predstavujúceho dátový filter, pomocou ktorého užívateľ môže vytvoriť výstup z databázy, ktorý bude obsahovať dáta podľa jeho požiadaviek. Výsledná databáza bola prenesená na fakultný server DATAMAP, čo umožňuje v GIS softvéri pripojiť sa k databáze.

Prínos práce môže byť v rámci projektu Technologickéj agentúry ČR prebiehajúceho v rokoch 2021 – 2024 s názvom „*Databáze letokruhových chronologií jako nástroj pro evidenci a predikci reakce hlavních lesních dřevin na klimatickou změnu*“ (č. SS03010134), pre ktorý môže výsledok práce poslúžiť ako solídny základ. Cieľom projektu je vytvoriť unikátnu databázu letokruhových chronológií spoločne s metadátami popisujúcimi stanovištia, ktorá bude využitá na tvorbu analytických vrstiev, mapových výstupov a modelov, popisujúcich vplyv klimatických zmien na hlavné lesné dreviny ČR v súčasnosti aj budúcnosti (TAČR 2021).

Výsledná databáza by sa do budúcnosti dala vylepšiť pokročilou automatizáciou. Za spomenutie stojí napríklad vložená procedúra, ktorá bude vykonávať detrendovanie

chronológií priamo v databáze. Znamenalo by to, že užívateľ nemusí tento proces robiť manuálne v R, ale priamo v databáze by vyvolal spustenie procedúry, ktorá by hodnoty širok letokruhov z nespracovaných chronológií uložených v databáze štandardizovala a vložila do príslušnej databázovej tabuľky. Kvalite databázy by pridalo aj zlepšenie stávajúcich funkcií opäť v oblasti automatizácie.

11. Použité zdroje

- BAL GUPTA, S.; MITTAL, A. 2017. *Introduction to Database Management System*. 2nd ed. New Delhi : University Science Press. ISBN 978-93-81159-31-6.
- BREWER, P. W. 2014. Data Management in Dendroarchaeology using Tellervo. *Radiocarbon*, 56, 4, s. 79-83.
- BREWER, P. W. 2016. *Tellervo: A guide for users and developers* [online]. Ver. 1.2. Dostupné z: <http://www.tellervo.org/support/tellervo-manual.pdf> [cit. 8. 4. 2021].
- BREWER, P. W.; GUITERMAN Ch. H. 2016. A new digital field data collection system for dendrochronology. *Dendrochronologia*, 38, s. 131-135.
- BUNN, A.; KORPELA, M. 2021. *An Introduction to dplR* [online]. Posledná úprava 29. 1. 2021. Dostupné z: <http://finzi.psych.upenn.edu/R/library/dplR/doc/intro-dplR.pdf> [cit. 8. 4. 2021].
- BÜNTGEN, U.; FRANK, D. C.; NIEVERGELT, D.; ESPER, J. 2006. Summer Temperature Variations in the European Alps, A.D. 755–2004. *Journal of Climate*, 19, 21, s. 5606-5623.
- CARRER, M. 2011. Individualistic and Time-Varying Tree-Ring Growth to Climate Sensitivity. *PLoS ONE* 6(7): e22813, 8 s.
- COOK, E. R. et al. 2010. Asian Monsoon Failure and Megadrought During the Last Millennium. *Science*, 328, 5977, s. 486-489.
- CORONEL, C.; MORRIS, S. 2015. *Database Systems: Design, Implementation, and Management*. 11th ed. Boston (MA) : Course Technology/Cengage Learning. ISBN 978-1-285-19614-5.
- DATE, C. J. 2004. *An Introduction to Database Systems*. 8th ed. London : Addison-Wesley. ISBN 0-321-18956-6.
- ĎURAČIOVÁ, R.; CIBULKA D. 2015. *Databázové systémy v GIS: Návod a cvičenia* [online]. Bratislava : Nakladateľstvo STU. ISBN 978-80-227-4502-4. Dostupné z: https://www.svf.stuba.sk/buxus/docs/dokumenty/skripta/DATABAZOVE_SYSTEMY_V_GIS_Duraciova_Cibulka-komplet_na_www.pdf [cit. 14. 4. 2021].

- ELMASRI, R.; NAVATHE, S. 2010. *Fundamentals of Database Systems*. 6th ed. Hoboken (NJ) : Pearson. ISBN 978-0-136-08620-8.
- ESPER, J.; GÄRTNER, H. 2001. Interpretation of Tree-Ring Chronologies. *Erdkunde*, 55, 3, s. 277-288.
- ESRI 2021a. *Database connections in ArcGIS Pro* [online]. Ver. 2.7. Dostupné z: <https://pro.arcgis.com/en/pro-app/latest/help/data/databases/database-connections-in-arcgis-pro.htm> [cit. 14. 4. 2021].
- ESRI 2021b. *Databases and ArcGIS* [online]. Ver. 2.7. Dostupné z: <https://pro.arcgis.com/en/pro-app/latest/help/data/databases/databases-and-arcgis.htm> [cit. 14. 4. 2021].
- GEDALOF, Z.; BERG, A. A. 2010. Tree ring evidence for limited direct CO₂ fertilization of forests over the 20th century. *Global Biogeochemical Cycles*, 24, GB3027.
- GRISSINO-MAYER, H. D.; FRITTS, H. C. 1997. The International Tree-Ring Data Bank: An enhanced global database serving the global scientific community. *Holocene*, 7, 2, s. 235-238.
- JANSMA, E. et al. 2012. TRiDaBASE: A stand-alone database for storage, analysis and exchange of dendrochronological metadata. *Dendrochronologia*, 30, 3, s. 209-211.
- JANSMA, E.; BREWER, P. W.; ZANDHUIS, I. 2010. TRiDaS 1.1: The tree-ring data standard. *Dendrochronologia*, 28, 2, s. 99-130.
- JONES, P. D.; MANN, M. E. 2004. Climate over past millennia. *Reviews of Geophysics*, 42, 2, s. RG2002 1-42.
- KYNCL, J. 2017. *Letokruhy jako kalendář i záznamník: Zajímavosti z dendrochronologie*. Praha : Grada Publishing. ISBN 978-80-271-9548-0.
- MUZIKA, R. M. et al. 2004. The influence of O₃, NO₂ and SO₂ on growth of *Picea abies* and *Fagus sylvatica* in the Carpathian Mountains. *Environmental Pollution*, 130, 1, s. 65-71.
- NASH, S. E. 2008. Dendrochronology. In: PEARSALL, D. M. (eds.): *Encyclopedia of Archaeology*. San Diego (CA) : Elsevier/Academic. ISBN 9780125480307.

- OGC 2010. *OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 2: SQL option* [online]. Ver. 1.2.1. Dostupné z: <https://www.ogc.org/standards/sfs> [cit. 8. 4. 2021].
- OGC 2011. *OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture* [online]. Ver. 1.2.1. Dostupné z: <https://www.ogc.org/standards/sfa> [cit. 8. 4. 2021].
- PANYUSHKINA, I. P. 2011. Dendrochronology. In: McGROW-HILL (eds.): *McGraw Hill Encyclopedia of Science & Technology*. 11th ed. New York (NY) : McGraw-Hill. ISBN 9780071792738.
- POKORNÝ, J. 2001. *Prostorové objekty a SQL* [online]. Dostupné z: http://gisak.vsb.cz/gis_ostrava/GIS_Ova_2001/Sbornik/Referaty/Pokornyr.htm [cit. 8. 4. 2021].
- POKORNÝ, J.; VALENTA, M. 2020. *Databázové systémy*. 2. vyd. Praha : Česká technika - nakladatelství ČVUT. ISBN 978-80-01-06696-6.
- POSTGIS 2021. *PostGIS Manual* [online]. Ver. 3.1.2dev. Dostupné z: <https://postgis.net/docs/manual-3.1/> [cit. 8. 4. 2021].
- POSTGRESQL 2021a. *About PostgreSQL*[online]. Dostupné z: <https://www.postgresql.org/about/> [cit. 14. 4. 2021].
- POSTGRESQL 2021b. *PostgreSQL Documentation* [online]. Ver. 13.2. Dostupné z: <https://www.postgresql.org/docs/13/index.html> [cit. 14. 4. 2021].
- RIGAUX, P.; SCHOLL, M.; VOISARD, A. 2002. *Spatial Databases With Application to GIS*. San Francisco : Morgan Kaufmann. ISBN 1-55860-588-6.
- R-PROJECT 2021. *What is R?* [online]. Dostupné z: <https://www.r-project.org/about.html> [cit. 14. 4. 2021].
- SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. 2011. *Database System Concepts*. 6th ed. New York (NY) : McGraw-Hill. ISBN 978-0-07-352332-3.
- SPEER, J. H. 2010. *Fundamentals of Tree-Ring Research*. Tucson (AZ) : The University of Arizona Press. ISBN 9780816526857.

STROBL, Ch. 2008. PostGIS. In: SHEKHAR, S., XIONG, H. (eds.): *Encyclopedia of GIS*. Boston (MA) : Springer. s. 891-898. ISBN 978-0-387-35973-1.

TAČR 2021. *Databáze letokruhových chronologií jako nástroj pro evidenci a predikci reakce hlavních lesních dřevin na klimatickou změnu (SS03010134): zadávací dokumentace*. Praha : Technologická agentura ČR. Program Prostředí pro život. Řešitel: Václav Tremel. 2021-2024.

TORBENSON, M. C. A. 2015. Section 4.2.8: Dendrochronology. In: COOK, S. J., CLARKE, L. E., NIELD, J. M. (eds.): *Geomorphological Techniques (Online Edition)*. London : British Society for Geomorphology. ISSN: 2047-0371.

VAN DER SLEEN, P. et al. 2015. No growth stimulation of tropical trees by 150 years of CO₂ fertilization but water-use efficiency increased. *Nature Geoscience*, 8, 1, s. 24-28.

WIKIPEDIA 2021. *PostgreSQL* [online]. Dostupné z: <https://en.wikipedia.org/wiki/PostgreSQL> [cit. 14. 4. 2021].

YEUNG, A. K. W.; BRENT HALL, G. 2007. *Spatial Database Systems: Design, Implementation and Project Management*. Dordrecht : Springer. ISBN 978-1-4020-5393-1.

ZANG, Ch. 2015. Dendrobox - An interactive exploration tool for the International Tree Ring Data Bank. *Dendrochronologia*, 33, s. 31-33.

ZHAO, S. et al. 2018. The International Tree-Ring Data Bank (ITRDB) revisited: Data availability and global ecological representativity. *Journal of Biogeography*, 46, 2, s. 355-368.

Softvér:

ArcGIS Pro 2.7.0

BUNN, A. G. 2008. A dendrochronology program library in R (dplR). *Dendrochronologia*, 26, 115–124 s.

PostgreSQL 13 / PostGIS 3.1

R CORE TEAM 2020. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

12. Zoznam príloh

Príloha 1: PL/pgSQL funkcia s *triggerom*, ktorá umožňuje vklad záznamov do pohľadu *sites* a prevádza číselné zemepisné súradnice do priestorového atribútu *geom*

Príloha 2: PL/pgSQL funkcia s *triggerom*, ktorá umožňuje vklad záznamov do pohľadu *trees* a prevádza číselné zemepisné súradnice do priestorového atribútu *geom*

Príloha 3: PL/pgSQL funkcia s *triggerom* umožňujúca mazanie záznamov z pohľadov *sites* a *trees*

Príloha 4: PL/pgSQL funkcia *load_csv_file*, ktorá vytvára fázovú tabuľku, do ktorej kopíruje chronológiu zo súboru CSV

Príloha 5: PL/pgSQL funkcia *insert_raw_data*, ktorá kopíruje chronologie z fázovej tabuľky do tabuľky *raw_data*

Príloha 6: PL/pgSQL funkcia *insert_detrend_data*, ktorá kopíruje chronologie z fázovej tabuľky do tabuľky *detrend_data*

Príloha 7: Univerzálny zápis SQL dotazu slúžiaceho ako dátový filter

Príloha 8: Mapa skúmaných lokalít v datasete Kokořínsko zo severnej časti CHKO Kokořínsko

Príloha 9: Príklad použitia SQL dátového filtru

Príloha 1: PL/pgSQL funkcia s *triggerom*, ktorá umožňuje vklad záznamov do pohľadu *sites* a prevádza číselné zemepisné súradnice do priestorového atribútu *geom*

```
--tvorba pohľadu
CREATE VIEW sites AS
SELECT s.site_code,site_name,species,
ST_Y(geom) AS latitude,ST_X(geom) AS longitude,geom,
elevation,slope,aspect,rooting_depth,soil
FROM site AS s LEFT JOIN fg_site AS f ON s.site_code=f.site_code;

--funkcia umožňujúca vklad záznamov do pohľadu + prevod súradníc
CREATE OR REPLACE FUNCTION sites_view_insert() RETURNS trigger
AS $$

BEGIN

--ak sú vkladané súradnice, z ich číselného zápisu v atribútoch latitude
a longitude sa prevedú vhodnou reprezentáciou do atribútu geom:

IF NEW.latitude IS NOT NULL AND NEW.longitude IS NOT NULL THEN

INSERT INTO site (site_code,site_name,species,geom)
VALUES (NEW.site_code,NEW.site_name,NEW.species,
'SRID=4326;POINT('||NEW.longitude::text||' '||NEW.latitude::text||')');

--ak elevation, slope, aspect sú špecifikované, treba ich vkladať do
fg_site:

IF NEW.elevation IS NOT NULL OR NEW.slope IS NOT NULL
OR NEW.aspect IS NOTNULL OR NEW.rooting_depth IS NOT NULL
OR NEW.soil IS NOT NULL THEN

INSERT INTO fg_site (site_code, elevation, slope, aspect, rooting_depth,
soil)
VALUES (NEW.site_code,NEW.elevation,NEW.slope,NEW.aspect,
NEW.rooting_depth,NEW.soil);

END IF;
END IF;

--ak nie sú vkladané súradnice, žiadny prevod nie je nutný a hodnoty
vkladané do pohľadu sa môžu rovno vkladať do príslušných databázových
tabuliek:

IF NEW.latitude IS NULL AND NEW.longitude IS NULL THEN

INSERT INTO site (site_code,site_name,species)
VALUES (NEW.site_code,NEW.site_name,NEW.species);
```

```
--ak elevation, slope, aspect sú špecifikované, treba ich vkladať do fg_site:
```

```
IF NEW.elevation IS NOT NULL OR NEW.slope IS NOT NULL  
OR NEW.aspect IS NOTNULL OR NEW.rooting_depth IS NOT NULL  
OR NEW.soil IS NOT NULL THEN
```

```
INSERT INTO fg_site (site_code, elevation, slope, aspect, rooting_depth, s  
oil)  
VALUES (NEW.site_code,NEW.elevation,NEW.slope,NEW.aspect,  
NEW.rooting_depth,NEW.soil);
```

```
END IF;  
END IF;
```

```
RETURN NEW;  
EXCEPTION WHEN unique_violation THEN  
RAISE EXCEPTION 'porusujes unique obmedzenie, chyba s klucmi';
```

```
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER sites_view_insert_trg  
INSTEAD OF INSERT ON sites  
FOR EACH ROW EXECUTE PROCEDURE sites_view_insert();
```


Príloha 2: PL/pgSQL funkcia s *triggerom*, ktorá umožňuje vklad záznamov do pohľadu *trees* a prevádza číselné zemepisné súradnice do priestorového atribútu *geom*

```
--tvorba pohľadu
CREATE VIEW trees AS
SELECT t.site_code,t.tree_id,dbh,height,crown_height,crown_projection,
social_status,sapwood,ST_Y(geom) AS latitude,ST_X(geom) AS longitude,
geom,elevation,slope,aspect
FROM tree AS t LEFT JOIN fg_tree AS f ON t.site_code=f.site_code
AND t.tree_id=f.tree_id;

--funkcia umožňujúca vklad záznamov do pohľadu + prevod súradníc
CREATE OR REPLACE FUNCTION trees_view_insert() RETURNS trigger
AS $$

BEGIN

--ak sú vkladané súradnice, z ich číselného zápisu v atribútoch latitude
a longitude sa prevedú vhodnou reprezentáciou do atribútu geom:

IF NEW.latitude IS NOT NULL AND NEW.longitude IS NOT NULL THEN

INSERT INTO tree (site_code,tree_id,dbh,height,crown_height,
crown_projection,social_status,sapwood,geom)
VALUES(NEW.site_code,NEW.tree_id,NEW.dbh, NEW.height, NEW.crown_height,
NEW.crown_projection,NEW.social_status, NEW.sapwood,
'SRID=4326;POINT('||NEW.longitude::text||' '||NEW.latitude::text||')');

--ak elevation, slope, aspect sú špecifikované, treba ich vkladať do
fg_site:

IF NEW.elevation IS NOT NULL OR NEW.slope IS NOT NULL OR NEW.aspect IS NOT
NULL THEN

INSERT INTO fg_tree (site_code, tree_id, elevation, slope, aspect)
VALUES (NEW.site_code,NEW.tree_id, NEW.elevation,NEW.slope,NEW.aspect);

END IF;
END IF;

--ak nie sú vkladané súradnice, žiadny prevod nie je nutný a hodnoty
vkladané do pohľadu sa môžu rovno vkladať do príslušných databázových
tabuliek:

IF NEW.latitude IS NULL AND NEW.longitude IS NULL THEN
```

```

INSERT INTO tree (site_code,tree_id,dbh,height,crown_height,crown_projecti
on,social_status,sapwood)
VALUES (NEW.site_code,NEW.tree_id,NEW.dbh, NEW.height, NEW.crown_height,NE
W.crown_projection,NEW.social_status, NEW.sapwood);

--ak elevation, slope, aspect sú špecifikované, treba ich vkladať do
fg_site:

IF NEW.elevation IS NOT NULL OR NEW.slope IS NOT NULL OR NEW.aspect IS NOT
NULL THEN

INSERT INTO fg_tree (site_code, tree_id, elevation, slope, aspect)
VALUES (NEW.site_code,NEW.tree_id, NEW.elevation,NEW.slope,NEW.aspect);

END IF;
END IF;

RETURN NEW;
EXCEPTION WHEN unique_violation THEN
RAISE EXCEPTION 'porusujes unique obmedzenie, chyba s klucmi';

END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trees_view_insert_trg
INSTEAD OF INSERT ON trees
FOR EACH ROW EXECUTE PROCEDURE trees_view_insert();

```

Príloha 3: PL/pgSQL funkcia s *triggerom* umožňujúca mazanie záznamov z pohľadov *sites* a *trees*

```
--pre pohľad sites:
CREATE OR REPLACE FUNCTION sites_view_delete() RETURNS trigger
AS $$

BEGIN

DELETE FROM site WHERE site_code=OLD.site_code;
DELETE FROM fg_site WHERE site_code=OLD.site_code;

RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER sites_view_delete_trg
INSTEAD OF DELETE ON sites
FOR EACH ROW EXECUTE PROCEDURE sites_view_delete();

--pre pohľad trees:
CREATE OR REPLACE FUNCTION trees_view_delete() RETURNS trigger
AS $$

BEGIN

DELETE FROM tree WHERE site_code=old.site_code AND tree_id= OLD.tree_id;

DELETE FROM fg_tree WHERE site_code=old.site_code
AND tree_id= OLD.tree_id;

RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trees_view_delete_trg
INSTEAD OF DELETE ON trees
FOR EACH ROW EXECUTE PROCEDURE trees_view_delete();
```

Príloha 4: PL/pgSQL funkcia *load_csv_file*, ktorá vytvára fázovú tabuľku, do ktorej kopíruje chronológiu zo súboru CSV

```
--vytvorenie schémy pre fázové tabuľky
CREATE SCHEMA import;

--funkcia load_csv_file
CREATE OR REPLACE FUNCTION load_csv_file
(
    target_table TEXT,    --názov fázovej tabuľky
    csv_path TEXT,       --cesta k súboru CSV
    col_count INTEGER    --počet vzoriek definujúci počet stĺpcov
)
RETURNS void AS $$

DECLARE

iter INTEGER;    --celé číslo pre iteráciu stĺpcov
col TEXT;        --zaznamenanie názvov stĺpcov pre každú iteráciu
col_first TEXT; --zaznamenanie názvu prvého stĺpcu
t RECORD;       --zaznamenanie výsledku dotazu

BEGIN
--tvorba fázovej tabuľky
CREATE TABLE import.temp_table ();
FOR iter IN 1..(col_count+1)    --vytvorenie stĺpcov podľa počtu vzoriek
LOOP
    EXECUTE FORMAT
        ('ALTER TABLE import.temp_table ADD COLUMN col_%s text;', iter);
END LOOP;

--skopírovanie dát z CSV súboru chronológie, bez CSV HEADER
EXECUTE FORMAT
('COPY import.temp_table from %L with delimiter ',' quote '''' csv ',
csv_path);

--zmena názvov stĺpcov podľa prvého riadku, ktorý nesie ich názvy
iter := 1;
col_first := (SELECT col_1 FROM import.temp_table LIMIT 1);

FOR col IN EXECUTE FORMAT
('select unnest(string_to_array(trim(temp_table::text, '()'), ','))
FROM import.temp_table WHERE col_1 = %L', col_first)
LOOP
    EXECUTE FORMAT ('ALTER TABLE import.temp_table
        RENAME COLUMN col_%s TO "%s"', iter, col);
    iter := iter + 1;
END LOOP;
```

```

--zmazanie prvého riadku, ktorý nesie názvy stĺpcov
EXECUTE FORMAT
('DELETE FROM import.temp_table WHERE %s = %L', col_first, col_first);

--zmena názvu fázovej tabuľky
IF LENGTH(target_table) > 0 THEN
    EXECUTE FORMAT
        ('ALTER TABLE import.temp_table RENAME TO %I', target_table);
END IF;

--zmena dátového typu stĺpcov na numeric
FOR t IN SELECT column_name,table_schema || '.' || table_name AS full_name
FROM information_schema.columns
WHERE table_name=$1 AND data_type='text'
    LOOP
        EXECUTE 'ALTER TABLE ' || t.full_name || ' ALTER COLUMN
            "' || t.column_name || '" TYPE numeric
            USING ("' || t.column_name || '"::numeric)';
    END LOOP;

END;

$$ LANGUAGE plpgsql;

```

Príloha 5: PL/pgSQL funkcia *insert_raw_data*, ktorá kopíruje chronológie z fázovej tabuľky do tabuľky *raw_data*

```
CREATE OR REPLACE FUNCTION insert_raw_data
(
  kod_lokality VARCHAR,      --zadať kód lokality
  kod_stromu VARCHAR,       --zadať kód stromu
  sample_id VARCHAR,        --zadať kód vzorky
  nazov_temp_tab VARCHAR    --názov fázovej tabuľky z kt. sa kopíruje
)
)

RETURNS void AS $$

DECLARE

BEGIN
SET SCHEMA 'import';
IF LENGTH(sample_id) > 0 THEN
  EXECUTE FORMAT
  ('INSERT INTO public.raw_data (site_code, tree_id, sample_id, year,
  value) SELECT (SELECT site_code FROM public.sample
  WHERE site_code='%s' LIMIT 1),(SELECT tree_id FROM public.sample
  WHERE tree_id='%s' AND site_code='%s' LIMIT 1),
  (SELECT sample_id FROM public.sample WHERE sample_id='%s'
  AND site_code='%s' AND tree_id='%s'),y,"%s" FROM %s
  WHERE (%I<>0::numeric)', $1,$2,$1,$3,$1,$2,$3,$4,$3);

  ELSE
  EXECUTE FORMAT
  ('INSERT INTO public.raw_data (site_code, tree_id, sample_id, year,
  value) SELECT (SELECT site_code FROM public.sample
  WHERE site_code='%s' LIMIT 1),(SELECT tree_id FROM public.sample
  WHERE tree_id='%s' AND site_code='%s' LIMIT 1),
  (SELECT sample_id FROM public.sample WHERE site_code='%s'
  AND tree_id='%s'),y,"%s" FROM %s WHERE (%I<>0::numeric)',
  $1,$2,$1,$1,$2,$2,$4,$2);
END IF;
SET SCHEMA 'public';

END;

$$ LANGUAGE plpgsql;
```

Príloha 6: PL/pgSQL funkcia *insert_detrend_data*, ktorá kopíruje chronológiu z fázovej tabuľky do tabuľky *detrend_data*

```
CREATE OR REPLACE FUNCTION insert_detrend_data
(
    kod_lokality VARCHAR,    --zadať kód lokality
    kod_stromu VARCHAR,     --zadať kód stromu
    sample_id VARCHAR,      --zadať kód vzorky
    nazov_temp_tab VARCHAR  --názov fázovej tabuľky z kt. sa kopíruje
)
)

RETURNS void AS $$

DECLARE

BEGIN
SET SCHEMA 'import';
IF LENGTH(sample_id) > 0 THEN
    EXECUTE FORMAT
    ('INSERT INTO public.detrend_data (site_code,tree_id,sample_id,year,
    value) SELECT (SELECT site_code FROM public.sample
    WHERE site_code='%s' LIMIT 1),(SELECT tree_id FROM public.sample
    WHERE tree_id='%s' AND site_code='%s' LIMIT 1),
    (SELECT sample_id FROM public.sample WHERE sample_id='%s'
    AND site_code='%s' AND tree_id='%s'),y,"%s" FROM %s
    WHERE (%I<>0::numeric)', $1,$2,$1,$3,$1,$2,$3,$4,$3);

    ELSE
    EXECUTE FORMAT
    ('INSERT INTO public.detrend_data (site_code,tree_id,sample_id,year,
    value) SELECT (SELECT site_code FROM public.sample
    WHERE site_code='%s' LIMIT 1),(SELECT tree_id FROM public.sample
    WHERE tree_id='%s' AND site_code='%s' LIMIT 1),
    (SELECT sample_id FROM public.sample WHERE site_code='%s'
    AND tree_id='%s'),y,"%s" FROM %s WHERE (%I<>0::numeric)',
    $1,$2,$1,$1,$2,$2,$4,$2);
    END IF;
SET SCHEMA 'public';

END;

$$ LANGUAGE plpgsql;
```

Príloha 7: Univerzálny zápis SQL dotazu slúžiaceho ako dátový filter

```
SELECT DISTINCT      --<tu špecifikovať atribúty, kt. budú výstupom
site_code,site_name,species,lat_site,lon_site,
elev_site,slope_site,asp_site,rooting_depth,soil,
tree_id,dbh,height,crown_height,crown_projection,social_status,sapwood,
elev_tree,slope_tree,asp_tree,
sample_id,year,value      --(boli vymenované všetky atribúty)
FROM                  --tu nič nemeniť:
(
SELECT v.site_code,v.site_name,v.species,v.lat_site,v.lon_site,
v.elev_site,v.slope_site,v.asp_site,v.rooting_depth,v.soil,v.tree_id,
v.dbh,v.height,v.crown_height,v.crown_projection,v.social_status,
v.sapwood,v.lat_tree,v.lon_tree,fgt.elevation AS elev_tree,
fgt.slope AS slope_tree,fgt.aspect AS asp_tree,v.sample_id,v.year,v.value
FROM fg_tree AS fgt RIGHT JOIN
(
SELECT x.site_code,x.site_name,x.species,x.lat_site,x.lon_site,x.elev_site
,x.slope_site,x.asp_site,x.rooting_depth,x.soil,x.tree_id,
dbh,height,crown_height,crown_projection,social_status,sapwood,ST_Y(t.geom
) AS lat_tree,ST_X(t.geom) AS lon_tree,x.sample_id,x.year,x.value
FROM tree AS t INNER JOIN
(
SELECT y.site_code,y.site_name,y.species,y.lat_site,y.lon_site,
elevation AS elev_site,slope AS slope_site,aspect AS asp_site,
rooting_depth,soil,y.tree_id,y.sample_id,y.year,y.value
FROM fg_site AS fgs RIGHT JOIN
(
SELECT s.site_code,site_name,species,ST_Y(s.geom) AS lat_site,
ST_X(s.geom) AS lon_site,tree_id,sample_id,year,value FROM site AS s
INNER JOIN
(
SELECT * FROM raw_data      --<nastaviť či raw_data/detrend_data
AS d WHERE (site_code,tree_id,sample_id) IN
(SELECT DISTINCT site_code,tree_id,sample_id FROM raw_data
GROUP BY site_code,sample_id,tree_id
HAVING COUNT(*)>150)      --<tu špecifikovať požadovanú dĺžku chronológie
) AS z
ON z.site_code=s.site_code
) AS y
ON y.site_code=fgs.site_code
) AS x
ON x.site_code=t.site_code AND x.tree_id=t.tree_id
) AS v
ON fgt.site_code=v.site_code AND fgt.tree_id=v.tree_id
) AS vse
```

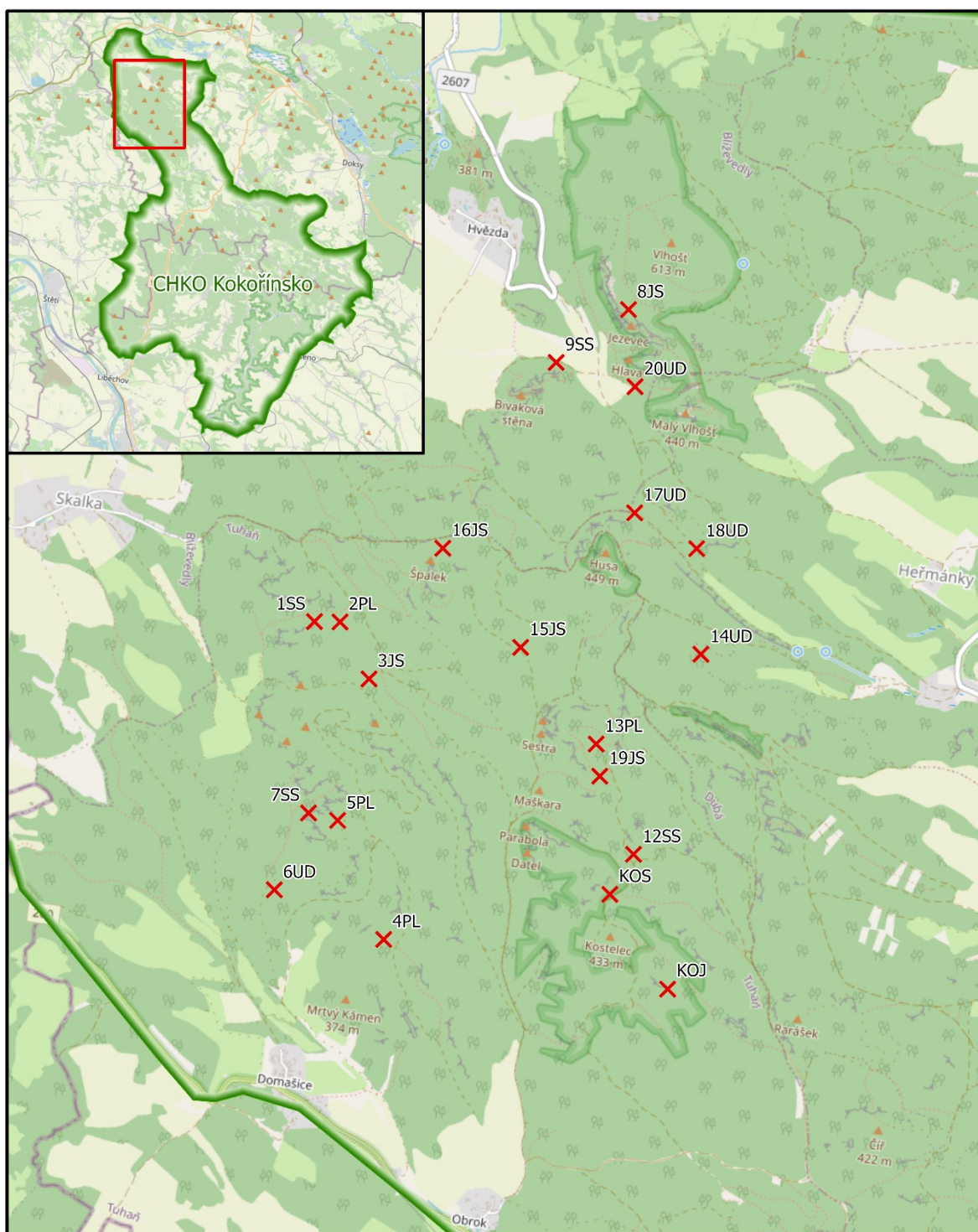


```

WHERE                                     --<tu špecifikovať požiadavky na atribúty:
--kód lokality v zozname
site_code IN ('KOJ','KOS')
--názov lokality v zozname
AND // OR site_name IN ('Kostelec Jih','Kostelec Sever')
--skratka druhu stromu v zozname
AND // OR species IN ('PISY', 'dub')
--latituda lokality od/do alebo neznáma hodnota
AND // OR ((lat_site>48 AND lat_site<52) OR lat_site IS NULL)
--longtitude lokality od/do alebo neznáma hodnota
AND // OR ((lon_site>10 AND lon_site<18) OR lon_site IS NULL)
--elevation lokality od/do alebo neznáma hodnota
AND // OR ((elev_site>390 AND elev_site<400) OR elev_site IS NULL)
--slope lokality od/do alebo neznáma hodnota
AND // OR ((slope_site>10 AND slope_site<50) OR slope_site IS NULL)
--aspect lokality od/do alebo neznáma hodnota
AND // OR ((asp_site>10 AND asp_site<50) OR asp_site IS NULL)
--rooting_depth od/do alebo neznáma hodnota
AND // OR ((rooting_depth>5 AND rooting_depth<10) OR rooting_depth
IS NULL)
--pôdny typ v zozname
AND // OR (soil IN ('leptosol','cernozem') OR soil IS NULL)
--kód stromu v zozname
AND // OR tree_id IN ('1','2')
--dendrometrické údaje od/do alebo neznáma hodnota
AND // OR ((dbh>30 AND dbh<35) OR dbh IS NULL)
AND // OR ((height>15 AND height<30) OR height IS NULL)
AND // OR ((crown_height>10 AND crown_height<25) OR crown_height IS NULL)
AND // OR ((crown_projection>2 AND crown_projection<3) OR crown_projection
IS NULL)
AND // OR ((social_status>2 AND social_status<5) OR social_status IS NULL)
AND // OR ((sapwood>2 AND sapwood<5) OR sapwood IS NULL)
--latituda stromu od/do alebo neznáma hodnota
AND // OR ((lat_tree>48 AND lat_tree<52) OR lat_tree IS NULL)
--longtitude stromu od/do alebo neznáma hodnota
AND // OR ((lon_tree>10 AND lon_tree<18) OR lon_tree IS NULL)
--elevation stromu od/do alebo neznáma hodnota
AND // OR ((elev_tree>350 AND elev_tree<480) OR elev_tree IS NULL)
--slope stromu od/do alebo neznáma hodnota
AND // OR ((slope_tree>10 AND slope_tree<50) OR slope_tree IS NULL)
--aspect stromu od/do alebo neznáma hodnota
AND // OR ((asp_tree>10 AND asp_tree<50) OR asp_tree IS NULL)
--kód vzorky v zozname od/do alebo neznáma hodnota
AND // OR sample_id IN ('01','02')



```

Príloha 8: Mapa skúmaných lokalít v datasete Kokořínsko zo severnej časti CHKO Kokořínsko



0 1 km



1SS
 skúmaná lokalita s kódom
 chránená krajinná oblasť

Zdroj podkladu a dát:

© OpenStreetMap (and) contributors.

ARCDATA Praha (2017): ArcČR500 v. 3.3 Dostupné z:

<https://www.arcdata.cz/produkty/geograficka-data/arccr-500>. [cit. 6.4.2021].

Hugo MAJER
 Bratislava 2021

Príloha 9: Príklad použitia SQL dátového filtru

```
SELECT DISTINCT --vo výstupe chcem tieto atribúty:
site_code,site_name,species,lat_site,lon_site,elev_site,slope_site,
asp_site,rooting_depth,soil,tree_id,dbh,height,crown_height,
crown_projection,social_status,sapwood,elev_tree,slope_tree,asp_tree,
sample_id,year,value
FROM
(
SELECT v.site_code,v.site_name,v.species,v.lat_site,v.lon_site,v.elev_site
,v.slope_site,v.asp_site,v.rooting_depth,v.soil,v.tree_id,
v.dbh,v.height,v.crown_height,v.crown_projection,v.social_status,v.sapwood
,v.lat_tree,v.lon_tree,fgt.elevation AS elev_tree,
fgt.slope AS slope_tree,fgt.aspect AS asp_tree,v.sample_id,v.year,v.value
FROM fg_tree AS fgt RIGHT JOIN
(
SELECT x.site_code,x.site_name,x.species,x.lat_site,x.lon_site,x.elev_site
,x.slope_site,x.asp_site,x.rooting_depth,x.soil,x.tree_id,
dbh,height,crown_height,crown_projection,social_status,sapwood,ST_Y(t.geom
) AS lat_tree,ST_X(t.geom) AS lon_tree,x.sample_id,x.year,x.value FROM tre
e AS t INNER JOIN
(
SELECT y.site_code,y.site_name,y.species,y.lat_site,y.lon_site,elevation A
S elev_site,slope AS slope_site,aspect AS asp_site,rooting_depth,soil,y.tr
ee_id,y.sample_id,y.year,y.value
FROM fg_site AS fgs RIGHT JOIN
(
SELECT s.site_code,site_name,species,ST_Y(s.geom) AS lat_site,ST_X(s.geom)
AS lon_site,tree_id,sample_id,year,value FROM site AS s INNER JOIN
(
SELECT * FROM detrend_data --<vo výstupe chcem detrendované hodnoty
AS d WHERE (site_code,tree_id,sample_id) IN
(SELECT DISTINCT site_code,tree_id,sample_id FROM raw_data GROUP BY site_c
ode,sample_id,tree_id
HAVING COUNT(*)>150) --<chcem aspoň 150 ročnú chronológiu
) AS z
ON z.site_code=s.site_code
) AS y
ON y.site_code=fgs.site_code
) AS x
ON x.site_code=t.site_code AND x.tree_id=t.tree_id
) AS v
ON fgt.site_code=v.site_code AND fgt.tree_id=v.tree_id
) AS vse
```

```
WHERE          --chcem:
--DBH od 30 do 35 alebo neznáme
((dbh>30 AND dbh<35) OR dbh IS NULL)
--výšku stromu od 15 m do 30 m alebo neznáme
AND ((height>15 AND height<30) OR height IS NULL)
--polomer koruny od 2 m do 3 m alebo neznáme
AND ((crown_projection>2 AND crown_projection<3) OR crown_projection
IS NULL);
```