

VYHOTOVIL Matej Kamenický	KONTROLOVAL doc. Ing. Renata ĎURAČIOVÁ, PhD.	KATEDRA GLOBÁLNEJ GEODÉZIE A GEOINFORMATIKY STAVEBNÁ FAKULTA STU RADLINSKÉHO 11 810 05 BRATISLAVA	
Názov: <b>NÁVRH A REALIZÁCIA PROJEKTU DATABÁZY V SYSTÉME POSTGRESQL/POSTGIS</b>		FORMÁT	A4
		POČET STRÁN	14
		POČET PRÍLOH	2
		ZADANÉ	21.11.2023
		ODOVZDANÉ	15.12.2023
DATABÁZOVÉ SYSTÉMY V GIS		HODNOTENIE	KONTROLOVAL

*Zadanie:*

## Návrh a realizácia projektu databázy v systéme PostgreSQL/PostGIS

Navrhnete a potom v prostredí PostgreSQL s rozšírením PostGIS vytvorte vlastný projekt databázy.

Projekt bude obsahovať:

- E-R diagram (alebo diagram tried v jazyku UML) pre navrhnutú databázovú štruktúru,
- technickú správu s opisom projektu (požiadavky používateľa, cieľ), použitých entít, atribútov a ich dátových typov, ako aj všetkých vytvorených objektov v databáze (tabuľky, domény, pohľady, používatelia a pod.)
- vzorové dopyty (ich znenie aj dopyty v jazyku SQL) ako prílohu k technickej správe,
- súbor so zálohou celej databázy (súbor **priezvisko.backup**).

a) Databáza bude minimálne v 3NF a bude obsahovať:

- **minimálne šesť tabuliek** (minimálne tie tabuľky, ktoré neobsahujú priestorové dáta, budú navzájom previazané kľúčmi),
- **z toho minimálne dve tabuľky, ktoré obsahujú aj priestorové dáta (geografické)** a z nich aspoň jedna s dátovým typom polygón,
- aspoň jedna priestorová tabuľka bude obsahovať **dáta importované z externých zdrojov (ZBGIS, OSM a pod.)**,
- **v každej tabuľke bude minimálne päť záznamov**, pričom v týchto vzorových záznamoch budú vyplnené všetky hodnoty atribútov (okrem atribútov, v ktorých sú povolené prázdne hodnoty),
- **minimálne jednu doménu hodnôt**, ktorá bude aspoň raz použitá v databáze,
- **minimálne jednu sekvenciu** (aj použitú v tabuľke),
- **minimálne dva pohľady**, ktoré vzniknú výberom dát vždy minimálne z dvoch vytvorených tabuliek,
- **minimálne dva typy používateľov** (napr. administrátor a bežný používateľ databázy).

b) Dopyty do databázy (**minimálne 8 dopytov**), prostredníctvom ktorých budú realizované výbery vždy aspoň z dvoch tabuliek, budú obsahovať agregačnú funkciu (aspoň v 1 dopyte), spojenie tabuliek cez podmienku (aspoň v jednom dopyte), spojenie tabuliek pomocou kľúčového slova JOIN (aspoň v jednom dopyte), vnorený dopyt (aspoň jeden), priestorové funkcie (aspoň v 3 dopytoch a aspoň jedna funkcia bude obsahovať metrické vzťahy (dĺžka, plocha, vzdialenosť a pod.)).

Technickú správu napíšete **formou technickej správy pre používateľa**.

Termín odovzdania zadania v AIS: **15.12.2023** (za každý týždeň oneskorenia bude hodnotenie znížené o 1 bod, max. počet bodov je 6).

# Dokumentácia databázy find\_your\_book

## Prehľad

Táto databáza je navrhnutá na vyhľadávanie pre študentov základných a stredných škôl, ktorý potrebujú zistiť kde sa nachádza kniha ktorú hľadajú. Môže ísť o povinné čítanie alebo kľudne aj o obľúbenú knižku na čítanie vo voľnom čase.

## Tabuľky

### 1. district

V tejto tabuľke sa nachádzajú dáta o okresoch na slovensku. Súčasťou tabuľky je aj geometrická informácia o jednotlivých okresoch. Dáta pochádzajú zo stránky geoportal (<https://www.geoportal.sk/sk/zbgis/na-stiahnutie/>).

- **Atribúty:**

- **id** (Primárny Kľúč): Unikátny identifikátor okresu.
- **name**: Názov okresu.
- **geom**: Geometrické údaje pre okres.

### 2. student

V tabuľke študentov sú údaje o jednotlivých žiakoch a študentoch. Tabuľka má geometrický bodový údaj o bydlisku. Dáta v tabuľke sú fiktívne.

- **Atribúty:**

- **id** (Primárny Kľúč): Unikátny identifikátor študenta.
- **class\_id** (Cudzí Kľúč): Odkazuje na **id** v tabuľke class.
- **name**: Meno študenta.
- **surname**: Priezvisko študenta.
- **home\_geom**: Geometrické údaje o mieste bydliska študenta.

### 3. class

V tabuľke sú základné údaje o triedach. Dáta v tabuľke sú fiktívne.

- **Atribúty:**

- **id** (Primárny Kľúč): Unikátny identifikátor triedy.
- **school\_id** (Cudzí Kľúč): Odkazuje na **id** v tabuľke school.
- **name**: Názov triedy.

#### 4. school

V tabuľke sú informácie o školách spolu s bodovou geometrickou reprezentáciou polohy školy. Jedná sa o OSM dáta stiahnuté z nasledujúcej stránky (<https://download.geofabrik.de/>).

- **Atribúty:**

- **id** (Primárny Kľúč): Unikátny identifikátor školy.
- **name**: Názov školy.
- **geom**: Geometrické údaje pre polohu školy.

#### 5. library

V tabuľke sú informácie o knižniciach spolu s bodovou geometrickou reprezentáciou ich polohy. Jedná sa o OSM dáta stiahnuté z nasledujúcej stránky (<https://download.geofabrik.de/>).

- **Atribúty:**

- **id** (Primárny Kľúč): Unikátny identifikátor knižnice.
- **name**: Názov knižnice.
- **geom**: Geometrické údaje pre polohu knižnice.

#### 6. book\_store

V tabuľke sú informácie o knihkupectvách spolu s bodovou geometrickou reprezentáciou ich polohy. Jedná sa o OSM dáta stiahnuté z nasledujúcej stránky (<https://download.geofabrik.de/>).

- **Atribúty:**

- **id** (Primárny Kľúč): Unikátny identifikátor knihkupectva.
- **name**: Názov knihkupectva.
- **geom**: Geometrické údaje pre polohu knihkupectva.

#### 7. book

V tabuľke sú názvy kníh doplnené o jazyk a ďalšie údaje. Jazyk kníh je obmedzený na angličtinu, slovenčinu ale taktiež je možné zadať aj neznámy jazyk. Dáta v tabuľke sú fiktívne.

- **Atribúty:**

- **id** (Primárny Kľúč): Unikátny identifikátor knihy.
- **author\_id** (Cudzí Kľúč): Odkazuje na **id** v tabuľke **author**.
- **title**: Názov knihy.
- **language**: Jazyk knihy.

## 8. author

V tabuľke sú základné informácie o autoroch. Dáta v tabuľke sú fiktívne.

- **Atribúty:**

- **id** (Primárny Kľúč): Unikátny identifikátor autora.
- **name**: Meno autora.
- **surname**: Priezvisko autora.

## Väzobné tabuľky

Väzobné tabuľky bolo potrebné vytvoriť medzi entitami kde sa nachádza vzťah many to many.

### 1. library\_book

- **Atribúty:**

- **id** (Primárny Kľúč): Unikátny identifikátor pripojenia.
- **library\_id** (Cudzí Kľúč): Odkazuje na **id** v tabuľke library.
- **book\_id** (Cudzí Kľúč): Odkazuje na **id** v tabuľke book.

### 2. book\_store\_book

- **Atribúty:**

- **id** (Primárny Kľúč): Unikátny identifikátor
- **book\_store\_id** (Cudzí Kľúč): Odkazuje na **id** v tabuľke book\_store.
- **book\_id** (Cudzí Kľúč): Odkazuje na **id** v tabuľke book.

## Pohľady

### Pohľad 1

V prvom pohľade `students_in_class` je možné si pozrieť mena a priezviská študentov spoločne s informáciou do ktorej triedy chodia.

```
1 create view students_in_class as (  
2 select student.name, student.surname, class.name as class  
3 from student  
4 inner join class on student.class_id = class.id);
```

## Pohľad 2

V druhom pohľade `library_in_ba` je možné si pozrieť mená všetkých knižníc v Bratislave. Mesto Bratislava sa delí na 5 okresov. Knižnice v tomto pohľade sa môžu nachádzať v ktoromkoľvek z piatich okresov mesta Bratislava. K názvu knižnice sa vypíše aj v ktorom okrese sa knižnica nachádza.

```
1 create view library_in_BA as(  
2 select library.name, district.nm3 as district  
3 from library, district  
4 where ST_intersects(library.geom,district.geom) and district.nm3 like  
5 'Bratislava%');
```

## Používatelia

Administrátorské práva v databáze má používateľ postgres. Okrem neho majú do databázy prístup dvaja ďalší používatelia a to Alice a Bob. Používateľ Alice má právo vo všetkých tabuľkách používať príkazy SELECT a UPDATE.

```
1 create user alice with password 'alice';  
2 grant select,update on all tables in schema public to alice;
```

Používateľ Bob má právo vo všetkých tabuľkách používať príkaz SELECT

```
1 create user bob with password 'bob';  
2 grant select on all tables in schema public to bob;
```

## Doména

V databáze sa nachádza aj jedna doména ktorá obmedzuje pridávanie do atribútu `language` v tabuľke `book`. Do atribútu je možné vložiť iba jednu z troch možností a to: `unknown`, `en`, `sk`.

```
1 create domain language_dm  
2 as varchar(10)  
3 default 'unknown'  
4 check (value in('unknown','sk','en'));
```

## Sekvencia

Sekvencie sa v databáze nachádzajú v každej tabuľke okrem tabuľky `district`. Sú vytvorené vždy pri použití dátového typu `serial` pri atribúte `id`.

## Prílohy

Príloha 1 - Vzorové dopyty  
Príloha 2 - ER diagram

```

1  --Find your dream book
2  create extension postgis;
3  set search_path to public;
4
5  --Table school
6
7  create table school(
8  id serial,
9  name varchar(255),
10 geom geometry(POINT,4326),
11 primary key (id)
12 );
13
14 insert into school(name,geom)
15 select name, ST_centroid(ST_transform(way,4326))
16 from planet_osm_polygon
17 where amenity = 'school';
18
19 insert into school(name,geom)
20 select name, ST_transform(way,4326)
21 from planet_osm_point
22 where amenity = 'school';
23
24 --Table class
25 create table class(
26 id serial,
27 name varchar(10) NOT NULL,
28 school_id int,
29 primary key (id),
30 foreign key (school_id) references school(id)
31 );
32
33
34 --Table student
35 create table student(
36 id serial,
37 class_id int,
38 name varchar(50),
39 surname varchar(50),
40 home_geom geometry(POINT,4326),
41 primary key (id),
42 foreign key (class_id) references class(id)
43 );
44
45
46 --Table library
47 create table library(
48 id serial,
49 name varchar(255),
50 geom geometry(POINT,4326),

```



```

51 primary key (id)
52 );
53
54 insert into library(name,geom)
55 select name, ST_centroid(ST_transform(way,4326))
56 from planet_osm_polygon
57 where amenity = 'library';
58
59 insert into library(name,geom)
60 select name, ST_transform(way,4326)
61 from planet_osm_point
62 where amenity = 'library';
63
64
65 --Table book_store
66 create table book_store(
67 id serial,
68 name varchar(255),
69 geom geometry(POINT,4326),
70 primary key (id)
71 );
72
73 insert into book_store(name,geom)
74 select name, ST_centroid(ST_transform(way,4326))
75 from planet_osm_polygon
76 where shop = 'books';
77
78 insert into book_store(name,geom)
79 select name, ST_transform(way,4326)
80 from planet_osm_point
81 where shop = 'books';
82
83
84
85 --Table author
86 create table author(
87 id serial,
88 name varchar(50),
89 surname varchar(50),
90 primary key (id)
91 );
92
93 --Table book
94 create domain language_dm
95 as varchar(10)
96 default 'unknown'
97 check (value in('unknown','sk','en'));
98
99 alter table book
100 add language language_dm;

```

```

101
102 drop table book;
103 create table book(
104 id serial,
105 title varchar(255),
106 author_id int,
107 language language_dm,
108 primary key (id),
109 foreign key (author_id) references author(id)
110 );
111
112 --Table district
113 create table district as(
114 select id,nm3, geom
115 from district2
116 );
117
118 alter table district
119 add primary key (id);
120
121 --Conection tables
122 create table library_book(
123 id serial,
124 library_id int,
125 book_id int,
126 primary key (id),
127 foreign key (library_id) references library(id),
128 foreign key (book_id) references book(id)
129 );
130
131 create table book_store_book(
132 id serial,
133 book_store_id int,
134 book_id int,
135 primary key (id),
136 foreign key (book_store_id) references book_store(id),
137 foreign key (book_id) references book(id)
138 );
139
140 --Add values into tables
141 --Table class
142 insert into class(school_id,name)
143 values
144 (1,'5.A'),
145 (1,'9.A'),
146 (2,'1.A'),
147 (2,'4.A'),
148 (3,'7.A');
149
150 --Table student

```

```

151 insert into student(class_id,name,surname,home_geom)
152 values
153 (1,'Miroslava','Novak',ST_GeomFromText('POINT(17.09976 48.14910)')),
154 (1,'Martin','Hrivnak',ST_GeomFromText('POINT(17.15045 48.15725)')),
155 (2,'Eva','Kovacova',ST_GeomFromText('POINT(17.03582 48.18880)')),
156 (2,'Juraj','Mikula',ST_GeomFromText('POINT(20.21712 49.13912 )')),
157 (3,'Zuzana','Kovac',ST_GeomFromText('POINT(21.23501 48.99800 )')),
158 (3,'Lukas','Varga',ST_GeomFromText('POINT(18.78043 49.43734)')),
159 (4,'Katarina','Kolarova',ST_GeomFromText('POINT(18.86123 48.86298)')),
160 (4,'Michal','Dvorak',ST_GeomFromText('POINT(22.15608 48.99165 )')),
161 (5,'Nina','Barta',ST_GeomFromText('POINT(19.14050 48.73755 )')),
162 (5,'Matej','Horvath',ST_GeomFromText('POINT(19.61720 49.08056 )'));
163
164 --Table author
165 insert into author(name,surname)
166 values
167 ('Sarah','Turner'),
168 ('James','Montgomery'),
169 ('Isabella','Cruz'),
170 ('Robert','Greene'),
171 ('Eleanor','Harper');
172
173
174 --Table book
175 insert into book(author_id,title,language)
176 values
177 (1,'Whispers in the Wind','en'),
178 (1,'Ephemeral Echoes','sk'),
179 (2,'Echoes of Eternity','en'),
180 (2,'The Last Alchemist','en'),
181 (3,'Sirens of the Sea','sk'),
182 (3,'Labyrinth of Lies','unknown'),
183 (4,'The Quantum Paradox','en'),
184 (4,'Art of Illusion','sk'),
185 (5,'Lost Legends','unknown'),
186 (5,'Shadows of Serenity','en');
187
188 --Conection table
189 insert into book_store_book(book_store_id,book_id)
190 values
191 (42,7),
192 (19,3),
193 (73,9),
194 (64,5),
195 (95,8),
196 (37,2),
197 (50,10),
198 (12,1),
199 (88,6),
200 (5,4),

```

```

201 (29,7),
202 (68,2),
203 (21,9),
204 (35,1),
205 (96,5),
206 (7,8),
207 (48,3),
208 (10,6),
209 (81,9),
210 (53,4);
211
212 insert into library_book(library_id,book_id)
213 values
214 (56,3),
215 (18,6),
216 (72,8),
217 (39,2),
218 (91,9),
219 (27,7),
220 (64,1),
221 (13,4),
222 (85,5),
223 (47,10),
224 (31,3),
225 (77,6),
226 (22,8),
227 (58,1),
228 (43,5),
229 (95,7),
230 (15,4),
231 (69,9),
232 (54,2),
233 (36,10);
234
235
236
237 --Views
238 create view students_in_class as (
239 select student.name, student.surname, class.name as class
240 from student
241 inner join class on student.class_id = class.id);
242
243 create view library_in_BA as(
244 select library.name, district.nm3 as district
245 from library, district
246 where ST_intersects(library.geom,district.geom) and district.nm3 like
      'Bratislava%'
247 );
248
249 --Users

```

```

250 create user alice with password 'alice';
251 grant select,update on all tables in schema public to alice;
252
253 create user bob with password 'bob';
254 grant select on all tables in schema public to bob;
255
256 --Queries in DB
257
258 --Q1 number of librarys in every district
259 select district.nm3 as name, count(library.id) as number_of_librarys
260 from district,library
261 where ST_intersects(library.geom,district.geom)
262 group by district.nm3
263 order by number_of_librarys desc;
264
265 --Q2 number of students in Bratislava city
266 select count(student.id) as number_of_students_in_Kosice
267 from student, district
268 where ST_intersects(student.home_geom,district.geom) and district.nm3
      like 'Bratislava%';
269
270 --Q3 authors with book titles
271 select author.name, author.surname, book.title
272 from author
273 inner join book on author.id = book.author_id;
274
275 --Q4 closest library from home of student with id 1
276 select student.name, student.surname,
      ST_distance(ST_transform(student.home_geom,5514),
277 ST_transform(library.geom,5514)) as distance_to_library, library.name
      as library
278 from library, student
279 where student.id = 1
280 order by distance_to_library
281 limit 1;
282
283 --Q5 number of schools in Poprad district
284 select count(school.id) as number_of_schools_in_Poprad
285 from school, district
286 where ST_intersects(school.geom,district.geom) and district.nm3 =
      'Poprad';
287
288 --Q6 in how many bookstores have auther some book
289 select author.id as author_id,author.name as author_name,
290 count(distinct book_store_book.book_store_id) as bookstores_count
291 from author
292 join book on author.id = book.author_id
293 join book_store_book on book.id = book_store_book.book_id
294 group by author.id, author.name
295 order by bookstores_count desc;

```

```

296
297 --Q7 nearest school from home for every student
298 select student.id as student_id, student.name as student_name,
      student.surname as student_surname,
299 class.name as class_name, school.name as school_name
300 from student
301 join class on student.class_id = class.id
302 join school on class.school_id = school.id
303 join district on ST_Within(student.home_geom, district.geom);
304
305 --Q8 in which book stores have books from Sarah Turner
306 select book.title, book_store.name
307 from book_store
308 join book_store_book ON book_store_book.book_store_id = book_store.id
309 join book on book.id = book_store_book.book_id
310 where book.author_id = (select id from author where name = 'Sarah' and
      surname = 'Turner');

```

district		
PK	id	int
	name	varchar 100
	geom	geometry

student		
PK	id	serial
FK	class_id	int
	name	varchar 50
	surname	varchar 50
	home_geom	geometry

class		
PK	id	serial
FK	school_id	int
	name	varchar 10

school		
PK	id	serial
	name	varchar 255
	geom	geometry

library		
PK	id	serial
	name	varchar 255
	geom	geometry

library_book		
PK	id	serial
FK	library_id	int
FK	book_id	int

book		
PK	id	serial
FK	author_id	int
	title	varchar 255
	language	varchar 10

author		
PK	id	serial
	name	varchar 50
	surname	varchar 50

book_store		
PK	id	serial
	name	varchar 255
	geom	geometry

book_store_book		
PK	id	serial
FK	book_store_id	int
FK	book_id	int

is in

has

contains

is available in

writes

sell

is available in