Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4

Databázový systém antikvariátu

Richard Belan

Cvičiaci: Ing. Lukáš Turský

FIIT INFO 2013/14

1. Zadanie

V zadaní som spracoval informačný systém databázy pre Antikvariát. Systém rieši základnú prácu s databázami, ako:

* Pridávanie nových záznamov,
* Upravovanie záznamov,
* Mazanie záznamov,
* Spájanie záznamov,
* Naviazanie cez cudzie kľúče,
* Zobrazenie viacerých záznamov,
* Zobrazenie štatistiky pre záznamy,
* Vyhľadanie konkrétneho záznamu.

Databázu som vytvoril v objektovo-relačnom databázovom systéme PostgreSQL 9.3. Užívateľské rozhranie a funkcionalitu som programoval v jazyku JAVA 7 v prostredí Eclipse KEPLER, kde som sa pomocou Java Database Connectivity(JDBC) pripojil na databázu a pracoval s ňou. Pri vkladaní, upravovaní a mazaní vecí z databázy som použil transakcie.

* 1. Možné scenáre

Do môjho systému som implementoval rôzne scenáre, aby práca s databázou bola interaktívna ľahko použiteľná.

Zoznam a krátke vysvetlenie implementovaných scenárov, zotriedených podľa kategórie typu funkčnosti:

* Pridávanie nových záznamov,
  + Pridanie nového titulu,
  + Pridanie nového typu titulu,
  + Pridanie nového autora,

Používateľ zadá všetky potrebné informácie do TextFieldov a stlačí tlačidlo. Stlačením sa vykoná query do databázy v podobne transakcie. Ak sa query neskončí úspešne, všetky zmeny v danej transakcii sa vrátia späť a vypíše sa chyba.

Pri pridaní nového titulu treba z ComboBoxu vybrať typ titulu a sklad kde je uložený, keďže som neimplementoval vkladanie typov a skladov. Typy produktov a sklady väčšinou nemusia byť pridávané a stačí ak sú napevno dané, keďže Antikvariát nemá široký sortiment typov produktov ani veľa skladov.

* Naviazanie cez cudzie kľúče,
  + Naviazanie autora na titul,
  + Naviazanie žánru na titul,

Pri naviazaní sa zadá id titulu a id záznamu z naväzovanej tabuľky(žáner alebo autor). Naväzovanie sa tiež vykonáva v transakcii. Po úspešnom naviazaní má titul nastavený buď žáner, autora alebo obe veci zároveň.

* Upravovanie záznamov,
  + Upravenie titulu,

Scenár rieši upravenie záznamu titulu. Používateľ napíše informácie znova korektne a záznam sa zmení. Používateľovi je umožnené pomocou ComboBoxov zmeniť aj záznamy pridružené cez cudzie kľúče ako sú typ a sklad.

* Zmazanie záznamu,
  + Zmazanie titulu podľa id,
  + Zmazanie titulov podľa roku(väčších ako zadaný rok),

Používateľ buď zadá id podľa ktorého sa záznam v tabuľke titul so zadaným id zmaže. Alebo ak používateľ zadá nejaký rok, vymažú sa všetky záznamy s rokom väčším ako zadaný rok. Ak je už záznam naviazaný cez cudzí kľúč na nejakého autora tak sa vypíše chyba a nedovolí sa zmazať. Riešené tiež pomocou transakcií.

* Zobrazenie záznamov
  + Zobrazenie autorov,
  + Zobrazenie titulov,
  + Zobrazenie žánrov,
  + Zobrazenie titulov naviazaných na všetky súvisiace tabuľky

Tieto scenáre zobrazia všetky záznamy z tabuliek autor, titul alebo žáner. Vypíšu všetky záznamy zo zvolenej tabuľky aj s informáciami k nim prislúchajúcimi. Štvrtý scenár naviaže pomocou JOINu všetky tabuľky súvisiace s tabuľkou titul. Tiež vypíše všetky záznamy pre titul aj s informáciami z naviazaných záznamov

* Zobrazenie štatistiky
  + Zobrazanie štatistiky počtu titulov v databáze podľa autorov

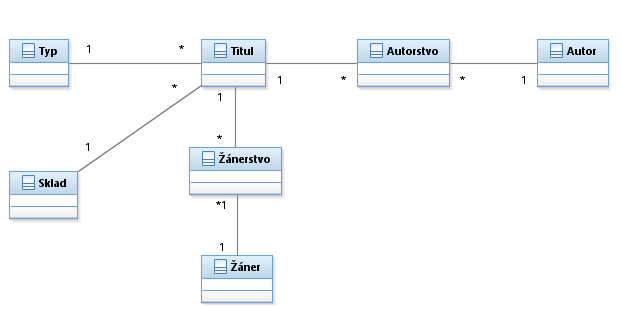
Riešené pomocou GROUP BY a agregačnej funkcie count(). Najprv si naJOINujem tabuľky autorov a titulov cez cudzie kľúče na seba a potom pomocou GROUP BY ich rozdelím podľa mena autorov. Následne využijem funkciu count(), ktorá mi pre každého autora vypíše počet jeho titulov v databáze.

* Vyhľadanie titulu v databáze
  + Vyhľadanie titulu podľa názvu titulu a mena autora

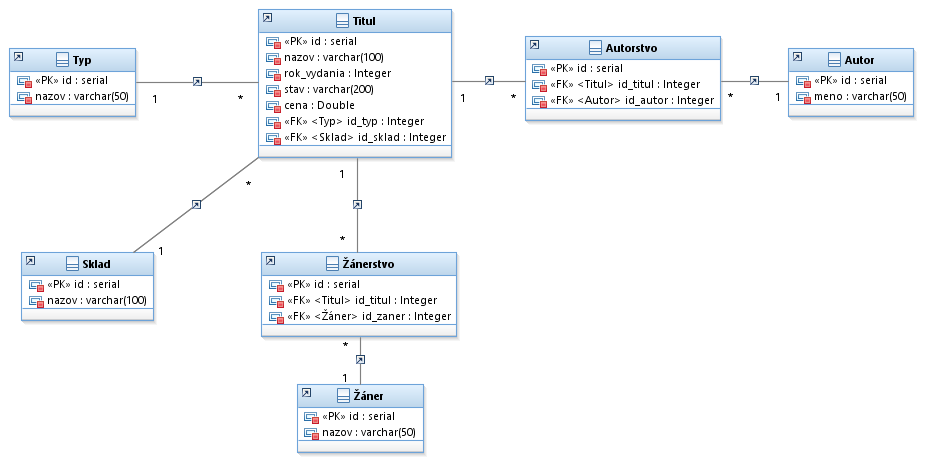
Daný scenár po zadaní názvu titulu a mena autora vypíše všetky zhodné záznamy aj s informáciami.

1. Model

2.1 Logický model



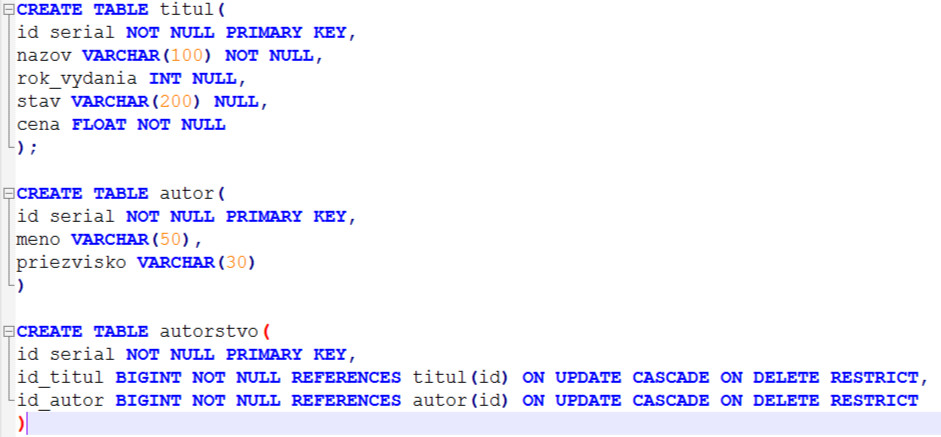
2.2 Fyzický model



Základom Antikvariátu je titul. Každý titul musí byť na nejakom sklade a musí mať nejaký typ(Kniha, platňa, kazeta alebo CD). Ak nie je na sklade tak skladom sa stáva hodnota nie je na sklade, ale musí byť zadané či sa nachádza titul v Antikvariáte a ak hej tak kde. Ak sa nenachádza, tak musí byť sprístupnená informácia o nedostupnosti titulu. Titul môže mať viac autorov a takisto aj autor mohol napísať viacero titulov, preto sa využíva medzientita autorstvo. Autor nemusí mať naviazaný žiadny titul. Takisto ani titul nemusí mať autora. Sú tituly, ktorých autor je neznámi, ale pri nich sa odporúča použiť autor ako Anonym. Taký istý vzťah ako medzi autorom a titulom je aj medzi titulom a žánrom. Každá entita má svoj primárny kľúč a medzi sebou sú naviazané pomocou cudzích kľúčov.

1. Implementácia
   1. PostgreSQL

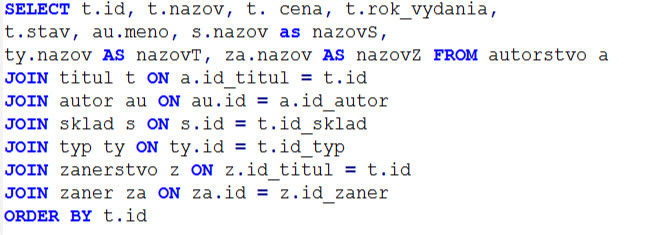
**V PostgreSQL som vytvoril databázu aj tabuľky pomocou SQL príkazou. Využíval som hlavne príkazy CREATE TABLE:**



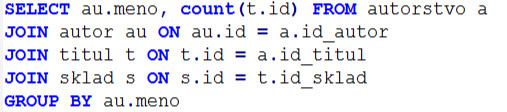
**alebo ALTER TABLE:**

http://i.snag.gy/kwG37.jpg

**Pri zobrazovaní a vypisovaní záznamov som cez Javu Eclipse JDBC posielal rôzne veľké query s veľa JOINami v jazyku SQL do databázy a čítal z nej výsledky:**

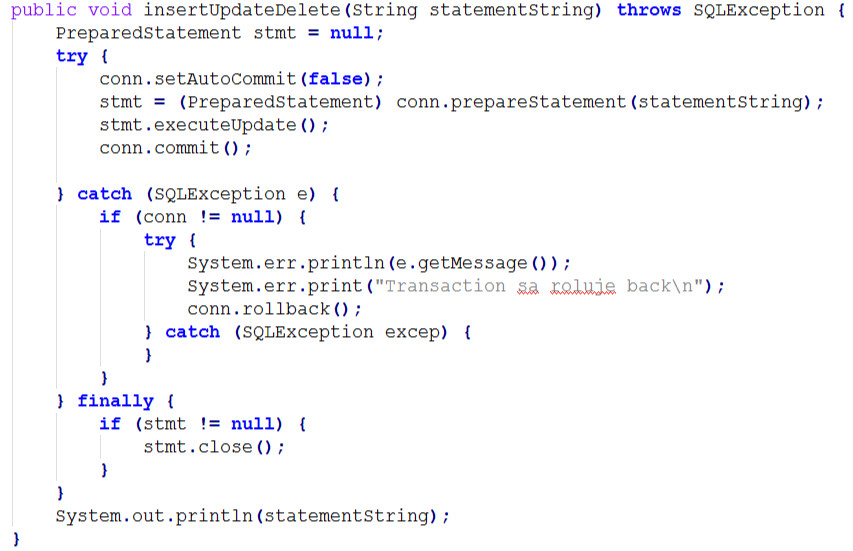


**V projekte som využil aj GROUP BY, JOIN a aj agregačnú funkciu count() v jednom SELECTe:**

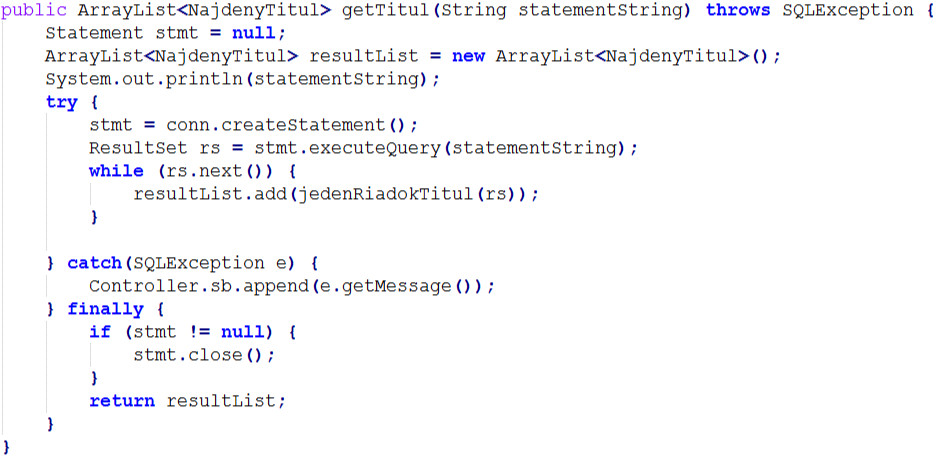


* 1. Java 7, Eclipse KEPLER

Všetky vloženia pridania aj mazania som riešil pomocou jednej funkcie insertUpdateDelete(String statementString). Do funkcie som poslal už vyskladané SQL query, a vo funkcii som ho skúsil vykonať ak sa nepodarilo, tak všetko vrátilo do pôvodného stavu a vykonal sa rollback databázy. Tiež sa vypísala chyba. Celé vykonávanie query, je obalené try a catch zachyteniami. Ak nastane chyba v try tak catch spustí rollback transakcie, a vypíše sa chyba. Funkcia je univerzálna. Keďže pri vkladaní, menení a mazaní sa nič nečíta z databázy, tak celý query String si môžeme predpripraviť v rôznych funkciách a sem len poslať a vykonať bezpečne v databáze.



Pri scenároch kde sa aj čítalo z databázy voľačo to bolo zložitejšie. Riešil som to pomocou dvoch funkcií, jedna ktorá mala na vstupe statementString a po vykonaní databázou čítala výsledok do ResultSetu:



Každý ResultSet poslala do pomocnej funkcie spracujRiadok(ResultSet rs). Funkcia spracujRiadok prijala ResultSet, vytvorila inštanciu objektu ktorý bol čítaný z databázy a naplnila ho informáciami z ResultSetu. Potom ho vrátila pôvodnej funkcii, kde sa pridal do ArrayListu a tá ho nakonci vrátila volajúcje funkcii:

