Slovenská technická univerzita

Fakulta informatiky a informačných technológii

Ilkovičova 3, 812 19 Bratislava



**DBS**

**Zadanie č. 1**

**Matej Adamov**

Cvičiaci: Ing. Eduard Kuric

Študijný odbor: Informatika

Ročník: 2. Bc

Akademický rok: 2015/2016

**1. Znenie zadania**

Vo vami zvolenom prostredí vytvorte databázovú aplikáciu, **ktorá komplexne rieši minimálne 6 scenárov** vo vami zvolenej doméne. Presný rozsah a konkretizáciu scenárov si dohodnete s Vašim cvičiacim na cvičení. Aplikáciu vytvoríte v dvoch iteráciach. V prvej iterácii, postavenej nad relačnou databázou, musí aplikácia realizovať tieto všeobecné scenáre:

* Vytvorenie nového záznamu,
* Aktualizácia existujúceho záznamu,
* Vymazanie záznamu,
* Zobrazenie prehľadu viacerých záznamov (spolu vybranou základnou štatistikou),
* Zobrazenie konkrétneho záznamu,
* Filtrovanie záznamov spĺňajúcich určité kritériá zadané používateľom.

Aplikácia môže mať konzolové alebo grafické rozhranie. Je dôležité aby scenáre boli realizované realisticky - teda aby aplikácia (a teda aj jej používateľské rozhranie) naozaj poskytovala časť funkcionality tak, ako by ju očakával zákazník v danej doméne.

Scenáre, ktoré menia dáta musia byť realizované **s použitím transakcií** a aspoň jeden z nich musí zahŕňať **prácu s viacerými tabuľkami** (typicky vytvorenie záznamu a naviazanie cudzieho kľúča).

**2. Špecifikácia scenárov**

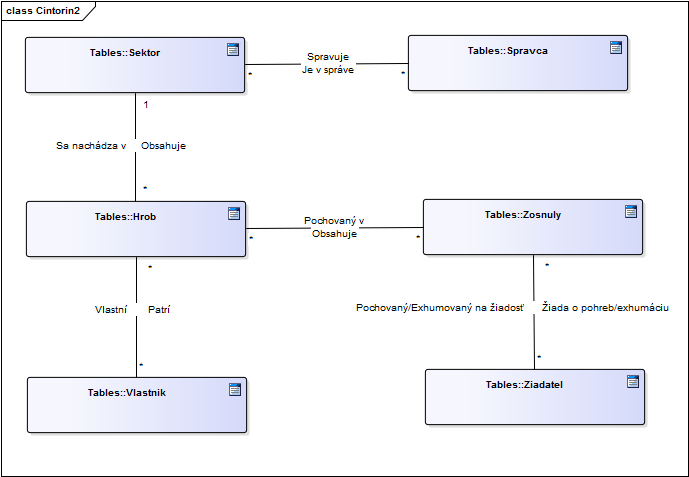
Ako doménu, pre ktorú databázovú aplikáciu vytváram, som sa rozhodol zvoliť správu cintorína.

Moja aplikácia rieši nasledovné scenáre:

* Komplexnejšie pridanie záznamu na základe iných záznamov a rôznych kritérií.
* Naplánovane pohrebu
* Registrácia exhumácie
* Zobrazenie viacerých komplexných štatistík
* Vyhľadanie osoby podľa typu a mena, alebo priezviska
* Komplexný prehľad schémy sektorov a hrobových miest
* Systematické zobrazenie detailov väčšiny prvkov z prehľadu
* Vytváranie a úprava prvkov

**3. Riešenie**

**Logický model**:

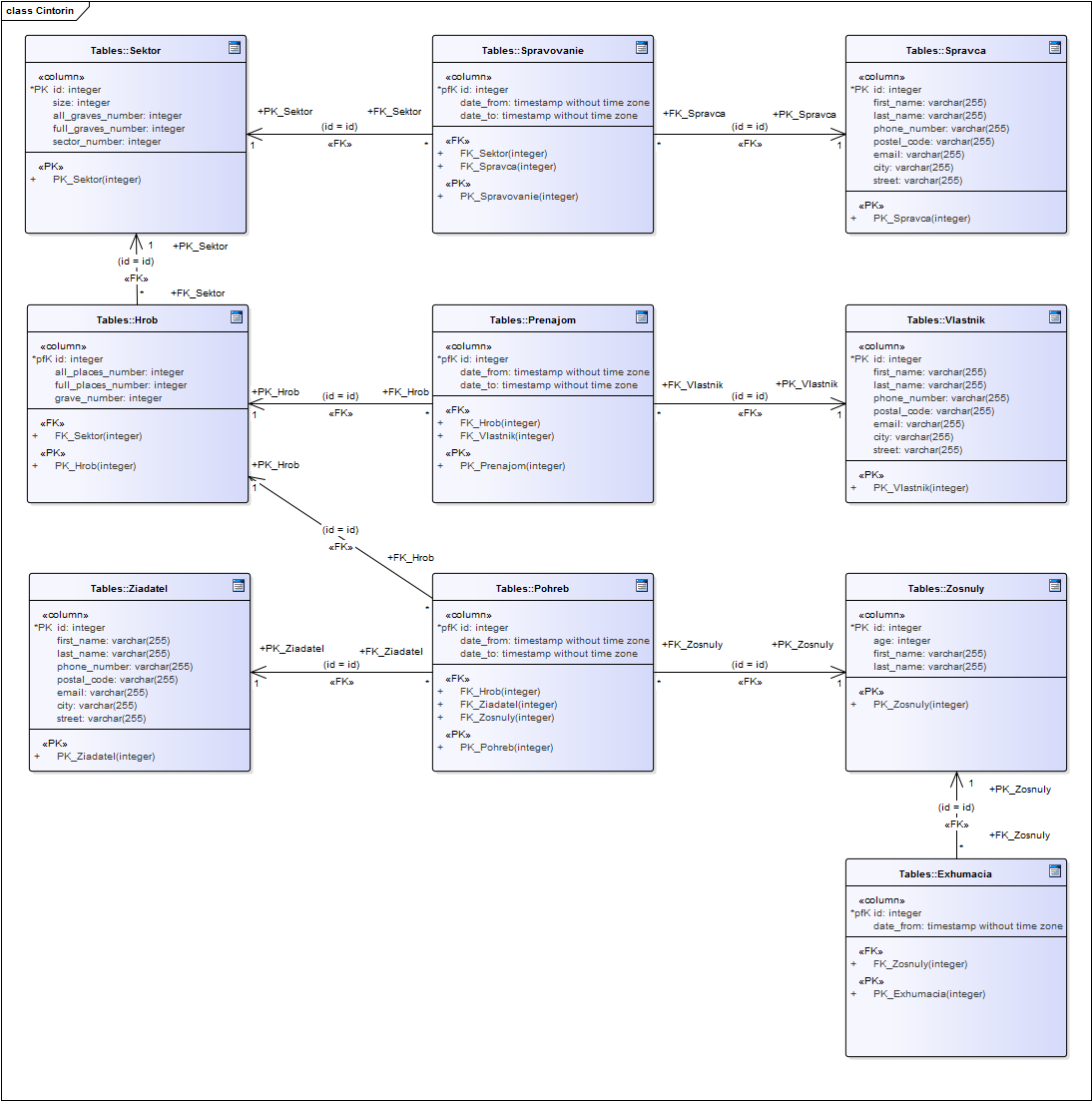


Model zachytáva podstatné vzťahy medzi hlavnými prvkami figurujúcimi v doméne správy cintorína.

Medzi tie patria:

* Správca spravuje sektor / Sektor spravovaný správcom, pričom správca môže spravovať viac sektorov, a sektor môže mať viac správcov v čase.
* Sektor môže obsahovať množstvo hrobov / Hrob patrí do jedného sektoru.
* Hrob je v prenájme viacerých vlastníkov v čase / Vlastník môže vlastniť niekoľko hrobov
* Zosnulý je pochovaný v hroboch podľa času
* Žiadateľ vybavuje pohreb alebo exhumáciu zosnulých / Zosnulý je pochovaný alebo exhumovaný na žiadosť žiadateľa

**Fyzický model:**



Na základe logického modelu som vytvoril fyzický model, ktorý zobrazuje implementačné detaily. Bolo nutné rozložiť "many-to-many" vzťahy ďalšími entitami. Taktiež som pridal samostatnú tabuľku pre exhumáciu.

Medzi takto rozložené vzťahy patria:

* Spravovanie (entity sektor a správca)
* Prenájom (entity hrob a vlastník)
* Pohreb(entity hrob a zosnulý)

Na rozdiel od logického modelu som pridal a popísal atribúty s presnými dátovými typmi a foreign/private kľúče do vzťahov.

**4. Implementácia**

Riešenie som sa rozhodol implementovať v objektovo orientovanom programovacom jazyku Java, a ako databázu som použil PostgreSQL.

V projekte do databázy pristupujem pomocou metód knižnice Hibernate.org, ktorá mi umožňuje pomerne jednoduché pripojenie na databázu.

Pri spustení aplikácie sa vytvorí "SessionFactory", ktorá slúži na vytváranie spojení s databázou. Následne pri dopytoch do databázy vytváram spojenia, ktoré sú otvorené čo najkratšiu dobu. Po vykonaní potrebných operácií ich okamžite zatváram.

V prípadoch kde v databáze mením dáta a je to potrebné, používam transakcie, ktoré dočasne bránia iných užívateľom v prístupe k objektom, aby nedošlo ku kolíznym situáciám.

Dopyty do databázy som vytváral pomocou funkcie createSQLQuery(String), kde som ako argument ručne písal dopyt v jazyku SQL.

Implementácia scenárov:

1. Pridávanie a zmena údajov v databáze je vyriešená pomocou dialógových okien, kde používateľ vyplní alebo zmení potrebné dáta. Tie sú skontrolované a následne pridané do databázy.
2. Registrácia pohrebu je samostatná kategória, nakoľko je nutné overiť niekoľko údajov. Je v prvom rade potrebné skontrolovať stav hrobového miesta, či je tam voľné miesto, či má vlastníka, a nakoniec je nutné overiť, či žiadateľ vlastní hrob. Ak nie tak používateľ musí potvrdiť, že žiadateľ má potrebné povolenie.
3. Žiadosť o exhumáciu sa vykonáva podobne ako registrácia pohrebu. V troch krokoch sa zadajú informácie o žiadateľovi, následne o zosnulom, ktorý sa vyhľadá v databáze či vôbec existuje. Nakoniec sú vypísané informácie o vlastníkovi hrobového miesta, kde je potrebné zadať čas exhumácie a potvrdiť, že vyššie uvedený vlastník súhlasí s tým súhlasí.
4. Pri vyhľadávaní osôb musí používateľ vybrať "typ" osoby a meno alebo priezvisko. Následne sa vypíšu všetky nájdene osoby vyhovujúce parametrom. Ak nie je zadané meno ani priezvisko zobrazia sa všetky osoby vybraného typu.
5. Zobrazenie prehľadu spočíva v zobrazení sektorov pričom všetky hrobové miesta sú znázornené ako potomkovia ich sektoru. Tu je možné pomocou tlačidiel realizovať zmeny, mazanie alebo pridávanie prvkov. Pri označení určitého sektoru, alebo hrobu sú zobrazené detailné informácie o tomto mieste, spolu so zoznamom osôb prislúchajúcim tomuto miestu. Pri sektoroch sú to správcovia, a pri hrobových miestach zase vlastníci a zosnulí. Po označení konkrétnej osoby sa zobrazia jej osobné informácie.
6. Štatistiky obsahujú tri panely. Každý z nich pozostáva z grafu, ktorý zobrazuje jednu štatistiku na základe zložitejšieho dopytu do databázy.

**5. Záver/Zhodnotenie**

V projekte som sa pokúsil splniť zadanie a snažil som sa vytvoriť aplikáciu, ktorá by sa teoreticky dala využiť aj v reálnom svete. Zachytil som praktické a nutné scenáre, ktoré sa často vyskytujú vo vybranej doméne.