Slovenská technická univerzita
Fakulta informatiky a informačných technológií STU v Bratislave
Ilkovičova 2, 842 16 Bratislava 4

Databázové systémy

Zadanie č. 1

Denisa Lipovská

Slovenská technická univerzita
Fakulta informatiky a informačných technológií STU v Bratislave
Ilkovičova 2, 842 16 Bratislava 4

Zadanie

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).

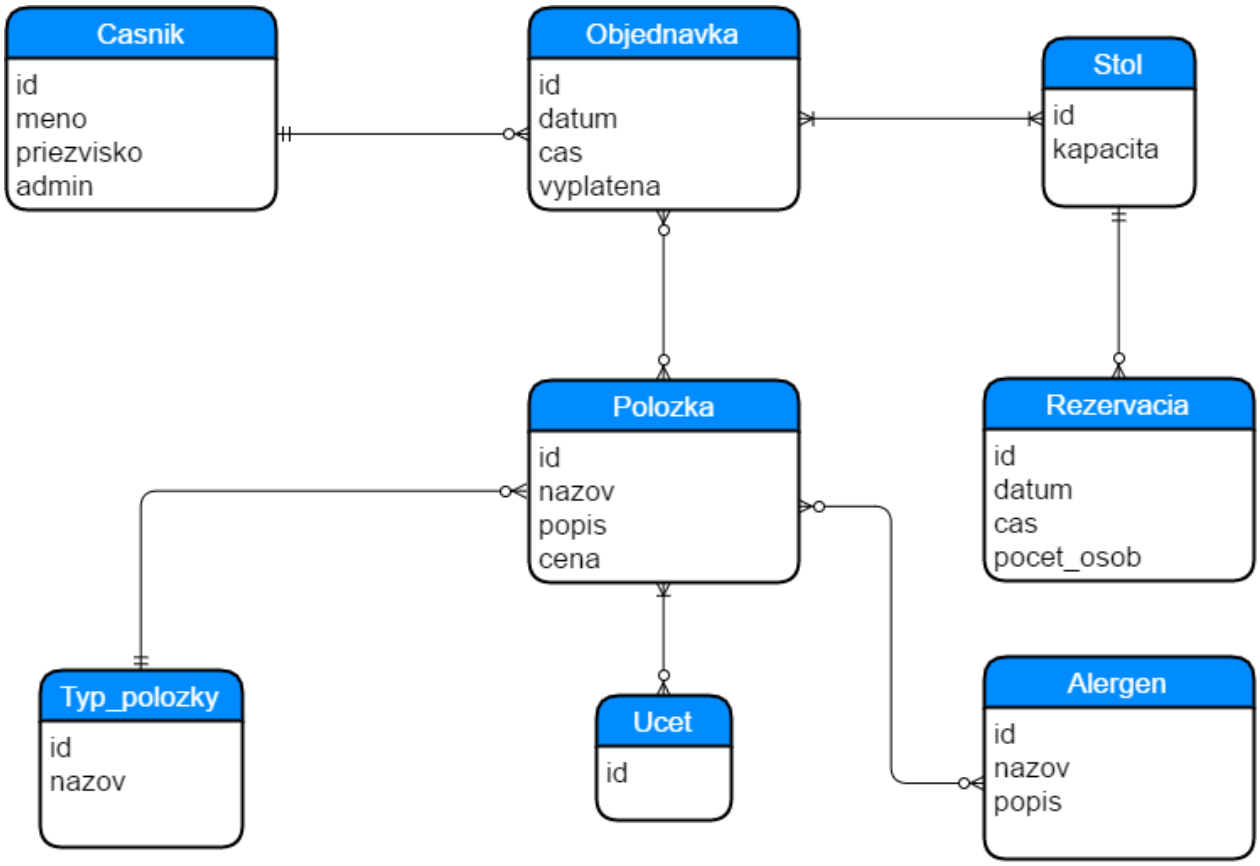
Slovenská technická univerzita
Fakulta informatiky a informačných technológií STU v Bratislave
Ilkovičova 2, 842 16 Bratislava 4

Špecifikácia realizovaných scenárov

1. zobrazenie všetkých jedál v databáze
2. zobrazenie všetkých nápojov v databáze
3. zobrazenie všetkých jedál, v ktorých sa nenachádza zvolený alergén (filtrovanie záznamov spĺňajúcich určité kritériá zadané používateľom)
4. pridanie nového jedla do databázy (vytvorenie nového záznamu)
5. pridanie nového nápoja do databázy (vytvorenie nového záznamu)
6. zmena ceny položky (aktualizácia existujúceho záznamu)
7. zobrazenie všetkých objednávok
8. vytvorenie novej objednávky (vytvorenie nového záznamu)
9. vystavenie účtu (zobrazenie konkrétneho záznamu)
10. zaplatenie účtu (aktualizácia existujúceho záznamu)
11. pridanie položky do objednávky (aktualizácia existujúceho záznamu)
12. storno položky z objednávky (vymazanie záznamu)
13. zobrazenie všetkých alergénov
14. zobrazenie alergénov a počet jedál, v ktorých sa nachádzajú (zobrazenie prehľadu viacerých záznamov (spolu vybranou základnou štatistikou))

Slovenská technická univerzita
Fakulta informatiky a informačných technológií STU v Bratislave
Ilkovičova 2, 842 16 Bratislava 4

Logický dátový model

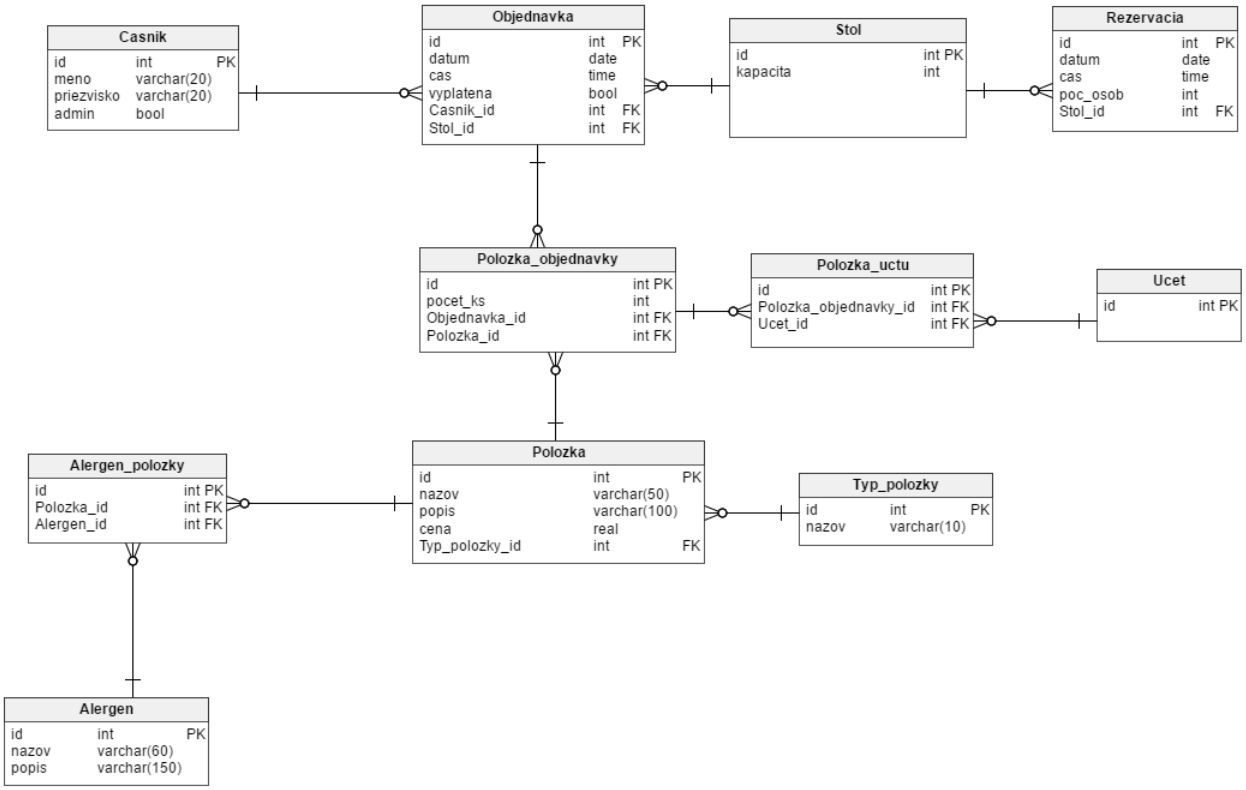


Logický dátový model obsahuje všetky entity, ktoré sa pôvodne mali v databázovej aplikácii využívať. Po neskoršom premyslení všetkých potrebných scenárov na splnenie zadania sa nakoniec využili iba tieto:

- Objednavka
  - entita, ktorá obsahuje zoznam objednávok, ktoré boli kedy vytvorené a údaj o tom, či bola objednávka vyplatená
- Polozka
  - uchováva zoznam položiek, ktoré boli k objednávke pripísané
  - entita, v ktorej sa uchováva zoznam všetkých jedál a nápojov uložených v databáze
- Alergen
  - zoznam všetkých alergénov s ich popisom
- Typ_polozky
  - entita, ktorá pomáha deliť položky na jedlá a nápoje
- Stol
  - zoznam stolov v databáze

Slovenská technická univerzita
Fakulta informatiky a informačných technológií STU v Bratislave
Ilkovičova 2, 842 16 Bratislava 4

Fyzický dátový model



Fyzický dátový model obsahuje všetky entity, ktoré sa pôvodne mali v databázovej aplikácii využívať. Po neskoršom premyslení všetkých potrebných scenárov na splnenie zadania sa nakoniec využili iba tieto:

- Objednavka
  - entita, ktorá obsahuje zoznam objednávok, ktoré boli kedy vytvorené a údaj o tom, či bola objednávka vyplatená
- Polozka_objednavky
  - uchováva zoznam položiek, ktoré boli k objednávke pripísané
- Polozka
  - entita, v ktorej sa uchováva zoznam všetkých jedál a nápojov uložených v databáze
- Alergen_polozky
  - bezodná entita, ktorá priraďuje alergény k položke
- Alergen
  - zoznam všetkých alergénov s ich popisom
- Typ_polozky
  - entita, ktorá pomáha deliť položky na jedlá a nápoje
- Polozka_uctu
  - entita, vďaka ktorej je možné vypočítať účet
- Stol
  - zoznam stolov v databáze

Slovenská technická univerzita
Fakulta informatiky a informačných technológií STU v Bratislave
Ilkovičova 2, 842 16 Bratislava 4

Opis návrhu a implementácie

Projekt bol implementovaný v jazyku Java a ako programovacie prostredie bolo zvolené prostredie Netbeans IDE. Aplikácia je riešená ako konzolová.

Databáza bola vytvorená v PostgreSQL, je to voľne šíriteľný objektovo-relačný databázaový systém. Na manipuláciu s databázou sa používa vývojová platforma pgAdmin.

V každej metóde, ktorá pracuje s databázou sa vytvára nové pripojenie k nej. Spojenie sa vytvára pomocu metódy getConnection triedy DriverManager.

```
try {
     this.conn=DriverManager.getConnection(DB, "postgres", "deni");
     stmt = conn.createStatement();
     ResultSet rs = stmt.executeQuery(query);
     while(rs.next()){
             result.add(processRow(rs));
     } catch (SQLException ex) {
             Logger.getLogger(AllTablesController.class.getName()).log(Level.SEVERE,
             null, ex);
     }finally{
            stmt.close();
            return result;
     }
```

1. zobrazenie všetkých jedál v databáze

   Scenár je realizovaný pomocou triedy PolozkaCtrl a metódy SelectAllJedlo, ktorá vracia spájaný zoznam všetkých jedál v databáze.

   ```
   SELECT * FROM polozka JOIN typ_polozky ON polozka.typ_polozky_id=typ_polozky.id
   WHERE typ_polozky.nazov = 'jedlo'
   ```

2. zobrazenie všetkých nápojov v databáze

   Scenár je realizovaný pomocou triedy PolozkaCtrl a metódy SelectAllPitie, ktorá vracia spájaný zoznam všetkých nápojov v databáze.

   ```
   SELECT * FROM polozka JOIN typ_polozky ON polozka.typ_polozky_id=typ_polozky.id
   WHERE typ_polozky.nazov = 'pitie'
   ```

3. zobrazenie všetkých jedál, v ktorých sa nenachádza zvolený alergén (filtrovanie záznamov spĺňajúcich určité kritériá zadané používateľom)

   Scenár je realizovaný pomocou triedy PolozkaCtrl a metódy vyberJedlaBezAlergenu, ktorá vracia spájaný zoznam jedál bez alergénu zvoleného používateľom.

   ```
   SELECT * FROM polozka WHERE typ_polozky_id=1 AND nazov NOT IN (SELECT p.nazov
   FROM polozka p JOIN alergen_polozky ap ON p.id=ap.polozka_id JOIN alergen a ON
   a.id=ap.alergen_id WHERE a.id=" + alergenId + ");
   ```

4. pridanie nového jedla do databázy (vytvorenie nového záznamu)

Scenár je realizovaný pomocou triedy PolozkaCtrl a metódy pridajPolozku, ktorá dostane cez parameter názov, popis, cenu a typ položky, ktorá sa má pridať do databázy. Následne sa zavolá aj metóda na pridanie alergénu k položke pridajAlergenPolozky triedy AlergenPolozkyCtrl.

```
INSERT INTO polozka (id, nazov, popis, cena, typ_polozky_id) VALUES (DEFAULT, '"
+ nazov + "', '" + popis + "', " + cena + "," + typ + ");
INSERT INTO alergen_polozky (id, polozka_id, alergen_id) VALUES (DEFAULT, " +
idPol + ", " + alergen + ");
```

5. pridanie nového nápoja do databázy (vytvorenie nového záznamu)

   Scenár je realizovaný pomocou triedy PolozkaCtrl a metódy pridajPolozku, ktorá dostane cez parameter názov, popis, cenu a typ položky, ktorá sa má pridať do databázy.

```
INSERT INTO polozka (id, nazov, popis, cena, typ_polozky_id) VALUES (DEFAULT, '"
+ nazov + "', '" + popis + "', " + cena + "," + typ + ");
```

6. zmena ceny položky (aktualizácia existujúceho záznamu)

   Scenár je realizovaný pomocou triedy PolozkaCtrl a metódy zmenCenuPolozky, ktorá dostane cez parameter názov položky a cenu, na ktorú sa má aktuálna cena zmeniť.

```
UPDATE polozka SET cena=" + cena + " WHERE id='" + id + "';
```

7. zobrazenie všetkých objednávok

   Scenár je realizovaný pomocou triedy ObjednavkaCtrl a metódy selectAllObjednavka, ktorá vracia spájaný zoznam všetkých objednávok.

```
SELECT * FROM objednavka
```

8. vytvorenie novej objednávky (vytvorenie nového záznamu)

   Scenár je realizovaný pomocou triedy ObjednavkaCtrl a metódy vytvorObjednavku, ktorá dostane cez parameter číslo stola, na ktorý sa má vytvoriť objednávka.

```
INSERT INTO objednavka(id,datum,cas,vyplatena,casnik_id,stol_id) VALUES
(DEFAULT,current_date,current_time,false,1," + cisloStola + ");
```

9. vystavenie účtu (zobrazenie konkrétneho záznamu)

   Scenár je realizovaný pomocou triedy PolozkaUctuCtrl a metódy VytvorUcet, ktorá dostane cez parameter číslo stola, na ktorý sa má vystaviť účet.

```
SELECT p.nazov, p.cena, po.pocet_ks, p.cena*po.pocet_ks AS \"celk_cena\" FROM
polozka p JOIN polozka_objednavky po ON p.id=po.polozka_id WHERE
po.objednavka_id="+idObj+";
```

10. zaplatenie účtu (aktualizácia existujúceho záznamu)

    Scenár je realizovaný pomocou triedy ObjednavkaCtrl a metódy zaplatObjednavku, ktorá dostane cez parameter číslo stola, na ktorom bola objednávka zaplatená.

```
UPDATE objednavka SET vyplatena=true WHERE id=" + idObj + ";
```

11. pridanie položky do objednávky (aktualizácia existujúceho záznamu)

    Scenár je realizovaný pomocou triedy PolozkaObjednavkyCtrl a metódy pridajPolozkuObjednavky, ktorá dostane cez parameter číslo stola, názov položky a počet kusov,

ktoré sa majú pridať do objednávky. V metóde sa rieši aj či už daná položka v objednávke je zapísaná, ak áno navýši sa iba počet kusov.

```
INSERT INTO polozka_objednavky (id,pocet_ks,objednavka_id,polozka_id) VALUES
(DEFAULT," + pocetKs + "," + idObj + "," + idPol + ");
UPDATE polozka_objednavky SET pocet_ks=" + kusy + " WHERE objednavka_id=" +
idObj + " AND polozka_id=" + idPol + ";
```

12. storno položky z objednávky (aktualizácia existujúceho záznamu)

    Scenár je realizovaný pomocou triedy PolozkaObjednavkyCtrl a metódy zmazPolozkuObjednavky, ktorá dostane cez parameter číslo stola, názov položky a počet kusov, ktoré sa majú v objednávke stornovať. Metóda rieši aj či sa má iba odobrať určitý počet kusov alebo sa daná položka má vymazať celá.

```
UPDATE polozka_objednavky SET pocet_ks=" + aktPocet + " WHERE objednavka_id=" +
idObj + " AND polozka_id=" + idPol + ";
DELETE FROM polozka_objednavky WHERE objednavka_id=" + idObj + " AND
polozka_id=" + idPol + ";
```

13. zobrazenie všetkých alergénov

    Scenár je realizovaný pomocou triedy AlergenCtrl a metódy selectAllAlergen, ktorá vráti spájaný zoznam alergénov.

```
SELECT * FROM alergen
```

14. zobrazenie alergénov a počet jedál, v ktorých sa nachádzajú (zobrazenie prehľadu viacerých záznamov (spolu vybranou základnou štatistikou))

    Scenár je realizovaný pomocou triedy AlergenCtrl a metódy vratStatistikuAlergenov, ktorá vypíše alergény a počet jedál, v ktorých sa alergén nachádza.

```
SELECT a.nazov, COUNT(p.nazov) AS \"pocet_jedal\" FROM alergen a JOIN
alergen_polozky ap ON ap.alergen_id=a.id JOIN polozka p ON ap.polozka_id = p.id
GROUP BY a.id;
```