

Slovenská technická univerzita

Fakulta informatiky a informačných technológií
Ilkovičova 3, 812 19 Bratislava

Databázové systémy

Projekt 1. iterácia

Jakub Pullmann

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 Vaším 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.

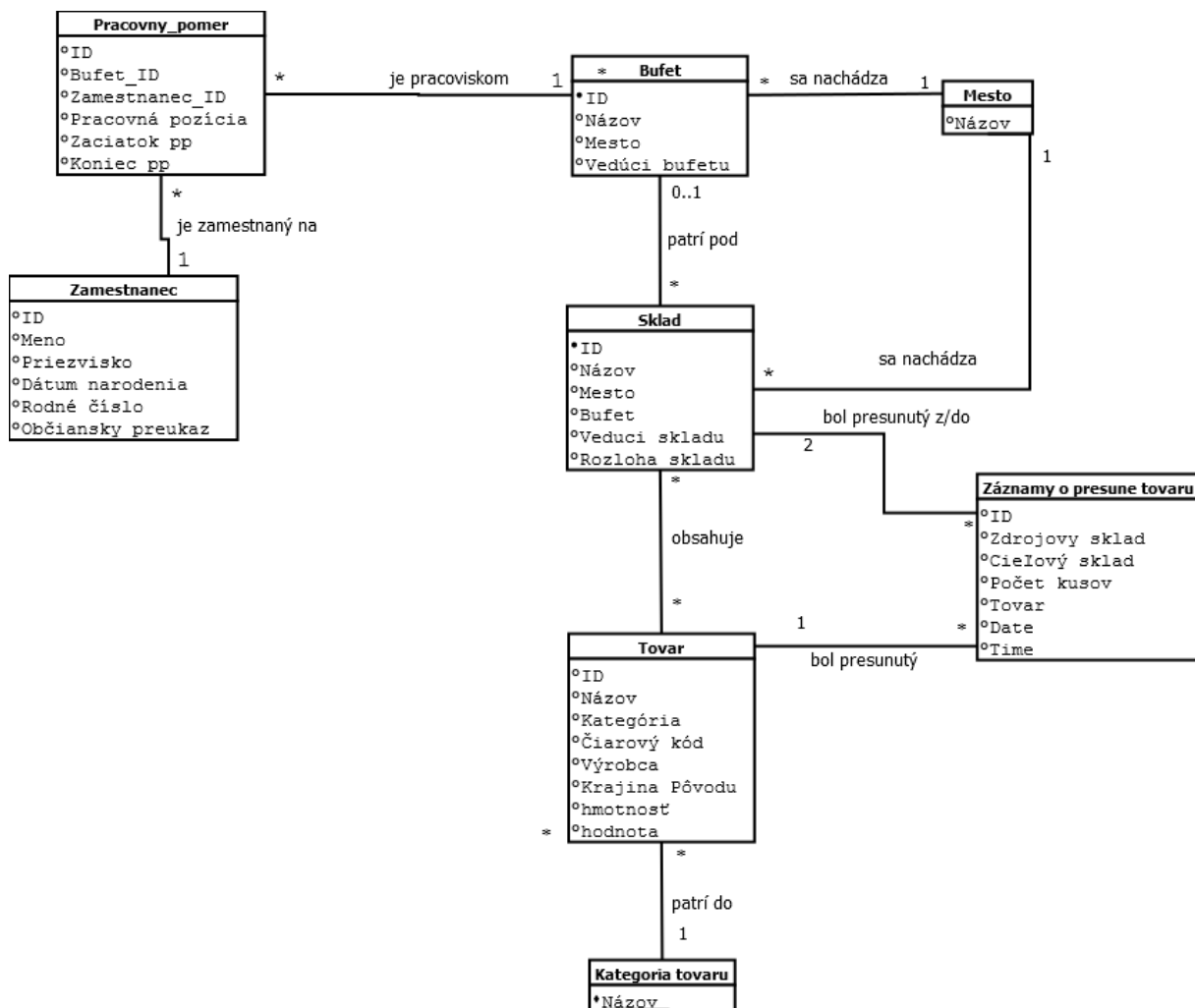
Scenáre, ktoré menia dáta musia byť realizované **s použitím transakcií** a aspoň jeden z nich musí zahŕňať **prácu s viacerými tabuľkami** (typicky vytvorenie záznamu a naviazanie cudzieho kľúča).

Dátový model

Logický dátový model

Logický dátový model pre sieť bufetov sebe zahŕňa entity:

- **Bufet:** obsahujúci základné informácie o bufete.
- **Pracovný pomer:** obsahuje informácie o pracovnom pomere zamestnanca
- **Zamestnanec:** obsahuje osobné údaje o zamestnancovi, ktoré má jeho zamestnávateľ
- **Mesto:** slúži ako zoznam miest
- **Sklad:** obsahujúci základné informácie o bufete
- **Tovar:** obsahuje informácie o tovare, ktorý má bufet v ponuke
- **Kategória Tovarů:** slúži ako zoznam kategórií
- **Záznamy o presune tovarů:** Je záznam o transporte tovarů z jedného skladu do druhého (Logistika)



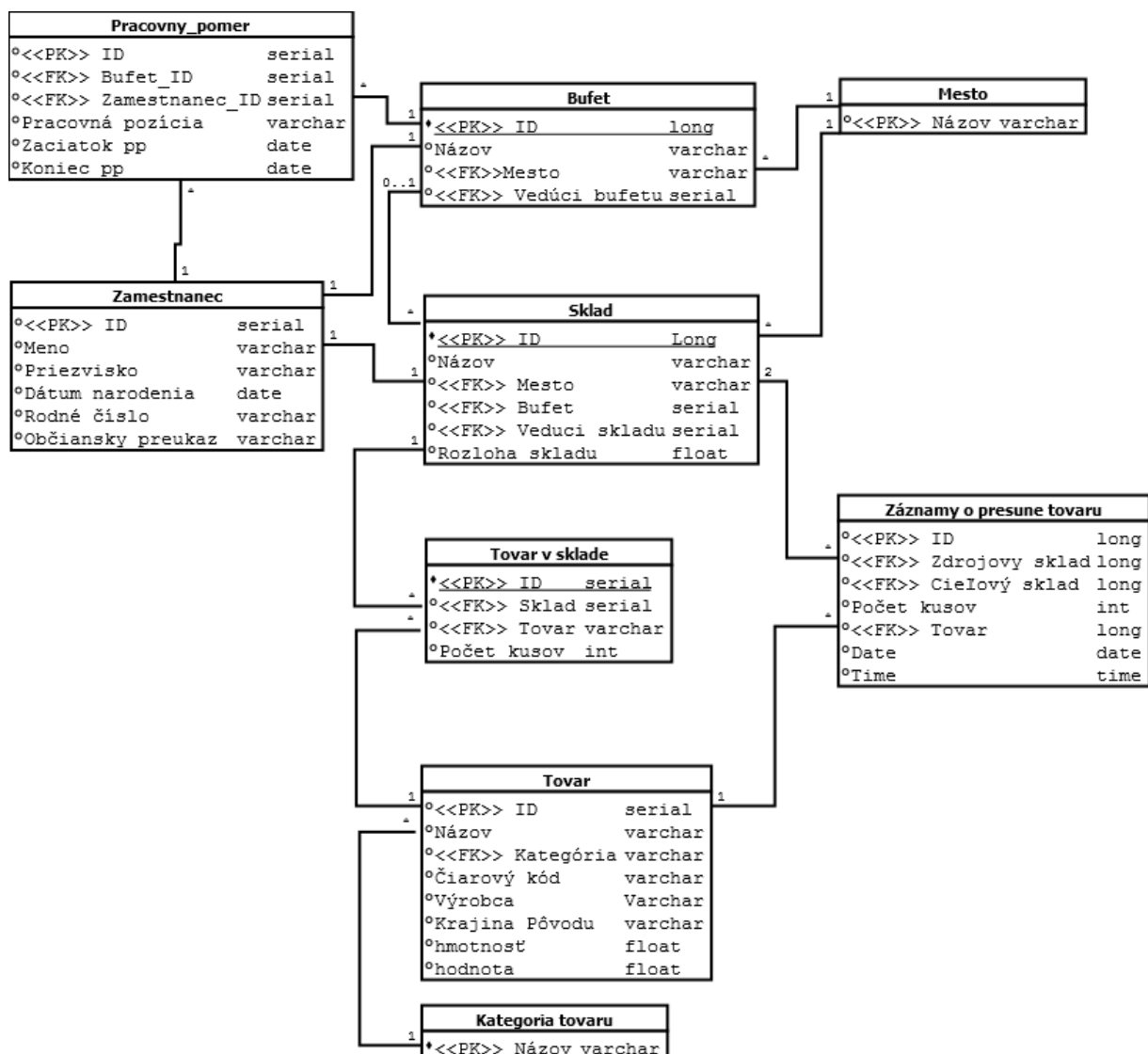
Skrytý rezeň, Skrytý rezeň, Skrytý rezeň, Skrytý rezeň, Skrytý rezeň, Skrytý rezeň, Skrytý rezeň, Skrytý

Fyzický dátový model

Pri vytváraní logického dátového modelu som musel:

- Pridať typy atribútov
- Pridať primárne a cudzie kľúče
- „Rozbiť“ N:M vzťahy

V logickom dátovom modeli sa nachádza N:M vzťah medzi entitou **Sklad** a **Tovar**. Vzťah som rozbil vytvorením Entity Tovar v Sklade, ktorá obsahuje informácie o tom koľko produkt sa nachádza v ktorom sklade (aj jeho počet kusov). Medzi entitou **Sklad** a **Tovar**, tak vznikli 2 nové 1:N vzťahy.



Programové prostredie:

GUI Správa tovaru

- Slúži na spravovanie tovaru v ponuke (pridávanie a editáciu). Po kliknutí za zobrazia detailne informácie o vybranom tovare

Name	Category	Value (€)
Fanta 2L	alkoholicke napoje	0.8
Fanta 2L ver2	alkoholicke napoje	0.82
Kofola 2L	nealkoholicke napoje	0.89
Sprite 2L	nealkoholicke napoje	0.73
Makový štrudla	pecivo	0.72
Osie hniezdo	pecivo	0.6
Oškvarkový pagáč	pecivo	0.76
Oškvarkový pagáč 3	pecivo	0.35
Tvarohová štrudla	pecivo	0.65
Tvarohový koláč	pecivo	0.74
kávenky	sladkosti	0.5
Mila	sladkosti	0.49

GUI Správa skladov

- Slúži na riadenie transportu tovaru medzi skladmi. Zobrazuje stav a základné štatistiky tovaru na skladoch.

ID	Warehouse name	Goods	Count	Value/unit
1	Sklad_1_BA	kávenky	12	0.50
24	Sklad_2_BA	kávenky	10	0.50
5	Sklad_3_BA	kávenky	8	0.50
6	Sklad_1_BA	Makový štrudla	20	0.72
8	Sklad_3_BA	Makový štrudla	15	0.72
7	Sklad_2_BA	Makový štrudla	10	0.72
17	Sklad_3_BA	Osie hniezdo	19	0.60
16	Sklad_2_BA	Osie hniezdo	16	0.60
15	Sklad_1_BA	Osie hniezdo	10	0.60
12	Sklad_1_BA	Oškvarkový pagáč	12	0.76
13	Sklad_2_BA	Oškvarkový pagáč	10	0.76
20	Sklad_3_BA	Oškvarkový pagáč	10	0.76
18	Sklad_2_BA	Tvarohový koláč	38	0.74
9	Sklad_1_BA	Tvarohový koláč	30	0.74
22	Sklad_3_BA	Tvarohový koláč	20	0.74

ID	Warehouse name	Total value	Total weight
1	Sklad_1_BA	57.720	5.400
2	Sklad_2_BA	57.520	5.250
3	Sklad_3_BA	48.600	4.360

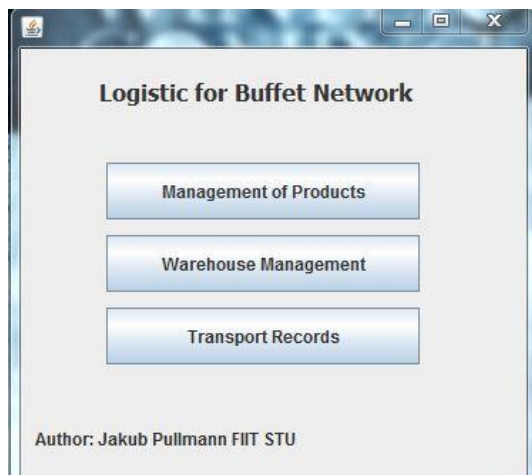
GUI Záznamy o presune tovaru

- Zobrazuje uskutočnené presuny tovaru medzi skladmi

ID	Goods	Category	Source	Destination	Count	Value	Date	Time
9	Makový štrudla	pecivo	Sklad_2_BA	Sklad_1_BA	10	7.20	2016-04-04	15:02:54
8	Tvarohový koláč	pecivo	Sklad_2_BA	Sklad_3_BA	15	11.10	2016-04-04	13:18:15
7	Tvarohový koláč	pecivo	Sklad_2_BA	Sklad_1_BA	15	11.10	2016-04-04	10:11:29

GUI Menu

- Slúži na prepojenie a výber pre ostatné GUI



Three Rings for the Elven-kings under the sky,
Seven for the Dwarf-lords in their halls of stone,
Nine for Mortal Men doomed to die,
One for the Dark Lord on his dark throne
In the Land of Mordor where the Shadows lie.
One Ring to rule them all, One Ring to find them,
One Ring to bring them all and in the darkness bind them
In the Land of Mordor where the Shadows lie."

—J.R.R. Tolkien's epigraph to The Lord of The Rings

Špecifikáciu scenárov

Vytvorenie nového záznamu

- Vytvorenie nového produktu do tabuľky **Tovar**. (GUI **Správa tovaru**)
- V prípade presunu tovaru , ak v cieľovom sklade nemajú ani jeden kus tovaru. (GUI **Správa skladov**)
- V prípade presunu tovaru do tabuľky **Záznamy o presune tovaru**, pri každom presune. (GUI **Záznamy o presune tovaru**)

Aktualizácia existujúceho záznamu

- Aktualizáciu záznamu v tabuľke **tovar**. (GUI **Správa tovaru**)
- Aktualizáciu záznamu (počtu kusov) v tabuľke **Tovar v sklade** pri presune tovaru. (GUI **Záznamy o presune tovaru**)

Vymazanie záznamu

- Vymazanie záznamu – z tabuľky produkt v sklade, v prípade že pri presune nezostane na sklade ani jeden kus tovaru. (GUI **Správa skladov**)

Zobrazenie prehľadu viacerých záznamov

- Zobrazenie záznamov z tabuľky **tovar**. (GUI **Správa tovaru**)
- Zobrazenie záznamov z tabuľky **produkt v sklade**. (GUI **Správa skladov**)
- Zobrazenie základnej štatistiky (celková hodnota a hmotnosť) z tabuľky **produkt v sklade**. (GUI **Správa skladov**)
- Zobrazenie záznamov z tabuľky **Záznamy o presune tovaru**. (GUI **Záznamy o presune tovaru**)

Zobrazenie konkrétneho záznamu

- Zobrazenie konkrétneho záznamu z tabuľky **tovar**. (GUI **Správa tovaru**)

Filtrovanie záznamov spĺňajúcich určité kritériá zadané používateľom

- Filtrovanie záznamov z tabuľky **Záznamy o presune tovaru**, podľa kategórie, ceny (od - do) a zdrojového/cieľového skladu. (GUI **Záznamy o presune tovaru**)

Realizácia scenárov

Použitie transakcie v realizovaných scenároch

Pri presun tovaru je potrebných niekoľko dopytov do databázy, ktoré musia všetky prebehnúť úspešne. V scenároch využívam transakciu na to aby sa databáza, v prípade chyby vrátila do pôvodného stavu.

Transakcia v prípade presunu tovaru obsahuje tieto dopyty:

- Odobratie posielaného počtu kusov tovaru zo zdrojového skladu.
- Pridanie posielaného počtu kusov tovaru do cieľového skladu.
- Vytvorenie záznamu o presune tovaru.
- Vytvorenie záznamu, ak v cieľovom sklade nemajú ani jeden kus tovaru.
- Vymazanie záznamu ak pri presune nezostane na sklade ani jeden kus tovaru

Práca s viacerými tabuľkami

S viacerými tabuľkami pracuje v prípade zobrazovania prehľadu viacerých záznamov tovaru v sklade, záznamoch o presune tovaru. Práca s viacerými tabuľkami taktiež využítá vo vyššie spomínanej transakcií.

Návrhové rozhodnutia

Program je naprogramovaný v jazyku **JAVA**. Na programovanie som využil vývojové prostredie Eclipse. Ako databázový systém využívam **PSQL**. Databáza beží na serveri pgAdmin III. Na prepojenie databázy s programom využívam JDBC.

V programe využívam architektonický vzor Model-view-controller.

Opis implementácie jednotlivých scenárov

Metódy na prácu s databázou sú spravené univerzálne, tak aby sa pre každý scenár nemusela vytvárať nová metóda na prácu s databázou.

Príklady implementácie metód pracujúcich s databázov:

Metóda *executeTransaction*: vykoná zoznam dopytov na databázu vložených cez argument funkcie.

```
public void executeTransaction(ArrayList<String> listOfQueries, GuiBase myGui)
throws SQLException, ClassNotFoundException {
```

```
    Connection conn = null;
    Statement stmt = null;
    try {
        Class.forName(JDBC_DRIVER);
        conn = DriverManager.getConnection(DB_URL, USER, PASS);
        conn.setAutoCommit(false);
        //System.out.println("Opened database successfully");

        stmt = conn.createStatement();
        String sql;
        for (int i = 0; i < listOfQueries.size(); i++) {
            sql = listOfQueries.get(i);
            stmt.executeUpdate(sql);
        }
        conn.commit();
    } catch (SQLException e) {
        if (conn != null) {
            try {
                System.err.println(e.getMessage());
                myGui.displayErrorMessage(e.getMessage() +
                    "Transaction is being rolled back");
                conn.rollback();
            } catch (SQLException excep) {

            }
        }
    } finally {
        if (stmt != null) {
            stmt.close();
        }
    }
}
```



```
    }  
}
```

Metóda *loadDataForTable*: Vrátí dáta z databázy pre naplnenie tabulky. Argumentmi sú dopyt a požadované stĺpce.

```
public ArrayList<Object> loadDataForTable(String query, ArrayList<String>  
listOfColumnns) {  
    Connection c = null;  
    Statement stmt = null;  
    ArrayList<Object> tableData = new ArrayList<Object>();  
    try {  
        Class.forName(JDBC_DRIVER);  
        c = DriverManager.getConnection(DB_URL, USER, PASS);  
        c.setAutoCommit(false);  
  
        stmt = c.createStatement();  
        ResultSet rs = stmt.executeQuery(query);  
  
        while (rs.next()) {  
            for (int i = 0; i < listOfColumnns.size(); i++) {  
  
tableData.add(rs.getObject(listOfColumnns.get(i)));  
            }  
        }  
  
        rs.close();  
        stmt.close();  
        c.close();  
    } catch (Exception e) {  
        System.err.println(e.getClass().getName() + ": " +  
e.getMessage());  
        System.exit(0);  
    }  
    //          System.out.println("Operation done successfully");  
  
    return tableData;  
}
```

Príklady implementovaných dopytov na databázu:

```
-- tabulka in GUI zaznamy_presunov_tovaru
SELECT zpt.id,
       t.nazov,
       k.nazov AS kategoria,
       sz.nazov AS zdrojovy_sklad,
       sc.nazov AS cielovy_sklad,
       zpt.pocet_kusov,
       (t.hodnota*zpt.pocet_kusov) AS hodnota_tovaru,
       zpt.time,
       zpt.date
FROM zaznamy_presunov_tovaru AS zpt
JOIN sklady sz ON sz.id = zpt.zdrojovy_sklad
JOIN sklady sc ON sc.id = zpt.cielovy_sklad
JOIN tovar t ON t.id = zpt.tovar
JOIN kategoria_tovaru k ON k.nazov = t.kategoria;

-- tabulka zoznam tovaru in GUI sprava tovaru
SELECT nazov,kategoria,hodnota FROM tovar;

-- tabulka statistic in GUI sprava skladov
SELECT tvs.sklad,s.nazov,sum(t.hodnota*tvs.pocet_kusov) AS celkova_hodnota_tovaru,
sum(t.hmotnost*tvs.pocet_kusov/1000) AS celkova_hmotnost_tovaru_kg
FROM tovar_v_skladoch tvs
JOIN sklady s ON s.id = tvs.sklad
JOIN tovar t ON t.id = tvs.tovar
GROUP BY tvs.sklad,s.nazov
ORDER BY celkova_hodnota_tovaru desc;

-- tabulka tovar v sklade in GUI sprava skladov
SELECT tvs.id,s.nazov,t.nazov AS tovar,tvs.pocet_kusov,t.hodnota AS cena_za_kus
FROM tovar_v_skladoch tvs
JOIN sklady s ON s.id = tvs.sklad
JOIN tovar t ON t.id = tvs.tovar
ORDER BY tvs.sklad;

-- odcita tovar (9ks) zo zdrojoveho skladu
UPDATE tovar_v_skladoch
SET pocet_kusov = (pocet_kusov - 9)
WHERE sklad = 1 AND tovar = 1

-- pricita tovar (5ks) do cieloveho skladu
UPDATE tovar_v_skladoch
SET pocet_kusov = (pocet_kusov + 5)
WHERE sklad = 2
AND tovar = 2

-- Create do tabulky zaznamy_presunov_tovaru
INSERT INTO zaznamy_presunov_tovaru (tovar,zdrojovy_sklad,cielovy_sklad,pocet_kusov,date,time) VALUES
(2,1,2,2,'1.1.2016','15:20')

-- Create do tabulky tovar_v_skladoch
INSERT INTO tovar_v_skladoch (sklad,tovar,pocet_kusov) VALUES
(1,1,5),

-- delete do tabulky tovar_v_skladoch
DELETE FROM tovar_v_skladoch
WHERE id = 1

-- pridanie noveho tovaru
INSERT INTO tovar (nazov,kategoria,ciarovy_kod,vyrobcia,krajina_povodu,hmotnost,hodnota) VALUES
('kávenky','sladkosti','555444333222','Sedita','Slovensko',50,0.50),
```