

Banka

Závěrečná správa

Projekt na Databázy (2)

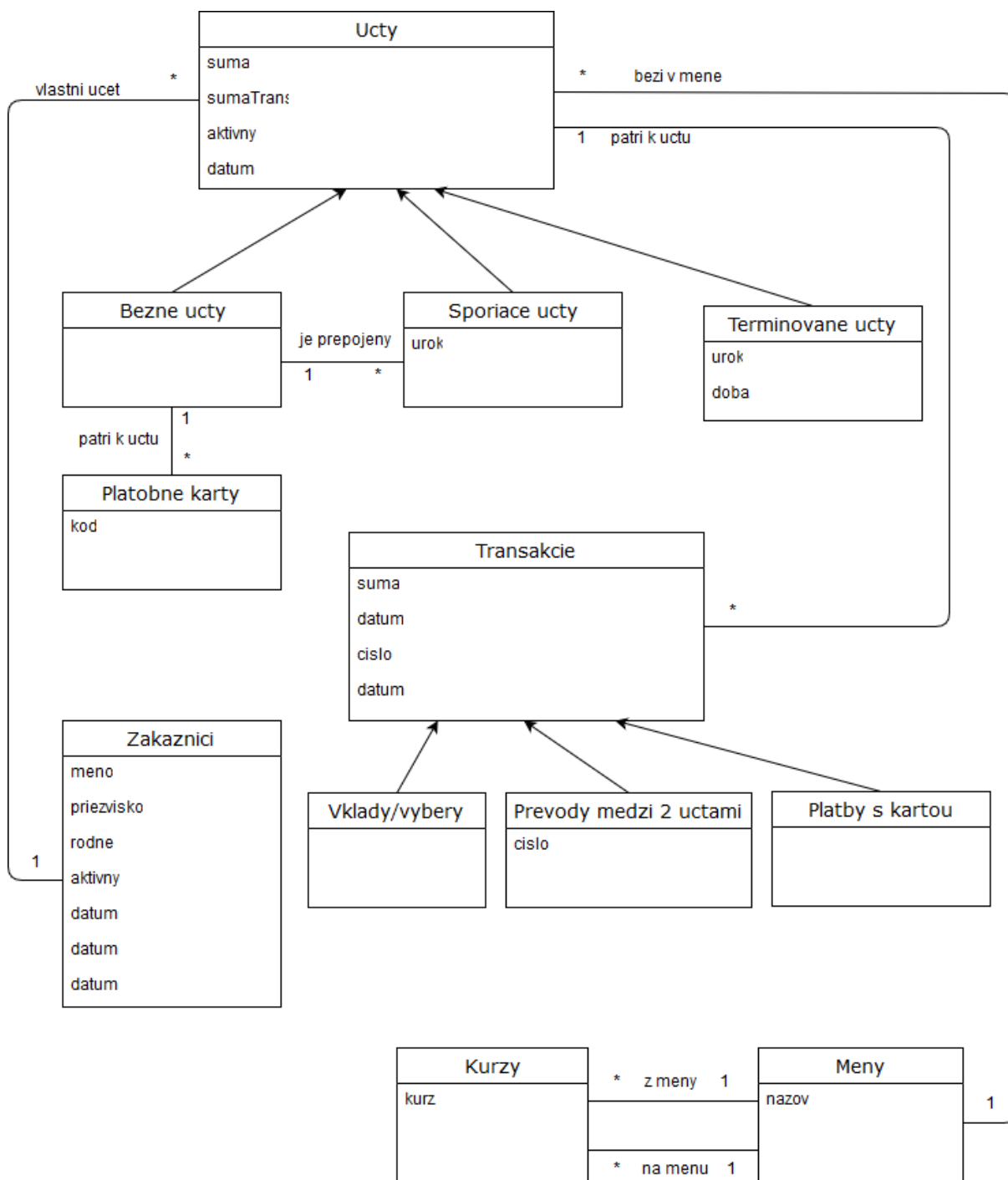
Konrád Müller

08.05.2019

1 O dokumente:

Tento dokument predstavuje dátový model odovzdávaný v rámci predmetu Databázy (2). Opisuje dátový model časti bankového systému. Zaoberá sa hlavne s uchovávaním účtov a s transakciami.

2 Dátový model



1. Obrázok - entitni-relačný model dát

Zákazníci sú uchovávané v tabuľke „Zákazníci“. O každom zákazníkovi sa uloží meno, priezvisko, rodné číslo a či je aktívny zákazník.

Každý zákazník môže mať viac účtov (aj toho istého druhu), každý účet môže byť aktívny, alebo nie, takisto ako aj zákazníci. V tabuľke účtov sa nachádza suma, dátum vytvorenia účtu, či je daný účet aktívny.

Účty môžu byť 3 typov:

- bežný účet
- sporiaci účet – zapamätá sa ešte úrok
- termínovyný účet – zapamätá sa ešte úrok a doba viazanosti

V tabuľke „Meny“ sú meny, v ktorých môžu byť vedené účty.

V tabuľke „Kurzy“ sú kurzy medzi dvoma menami.

K bežným účtom môžu byť vydané aj platobné karty, ktoré sú uložené v tabuľke „Platobné karty“. O každej karte je kód karty.

Vklady, výbery, prevody sú evidované v tabuľke „Transakcie“. Je tu uložené dátum transakcie, suma, poplatok. Transakcie sú rozdelené do typov:

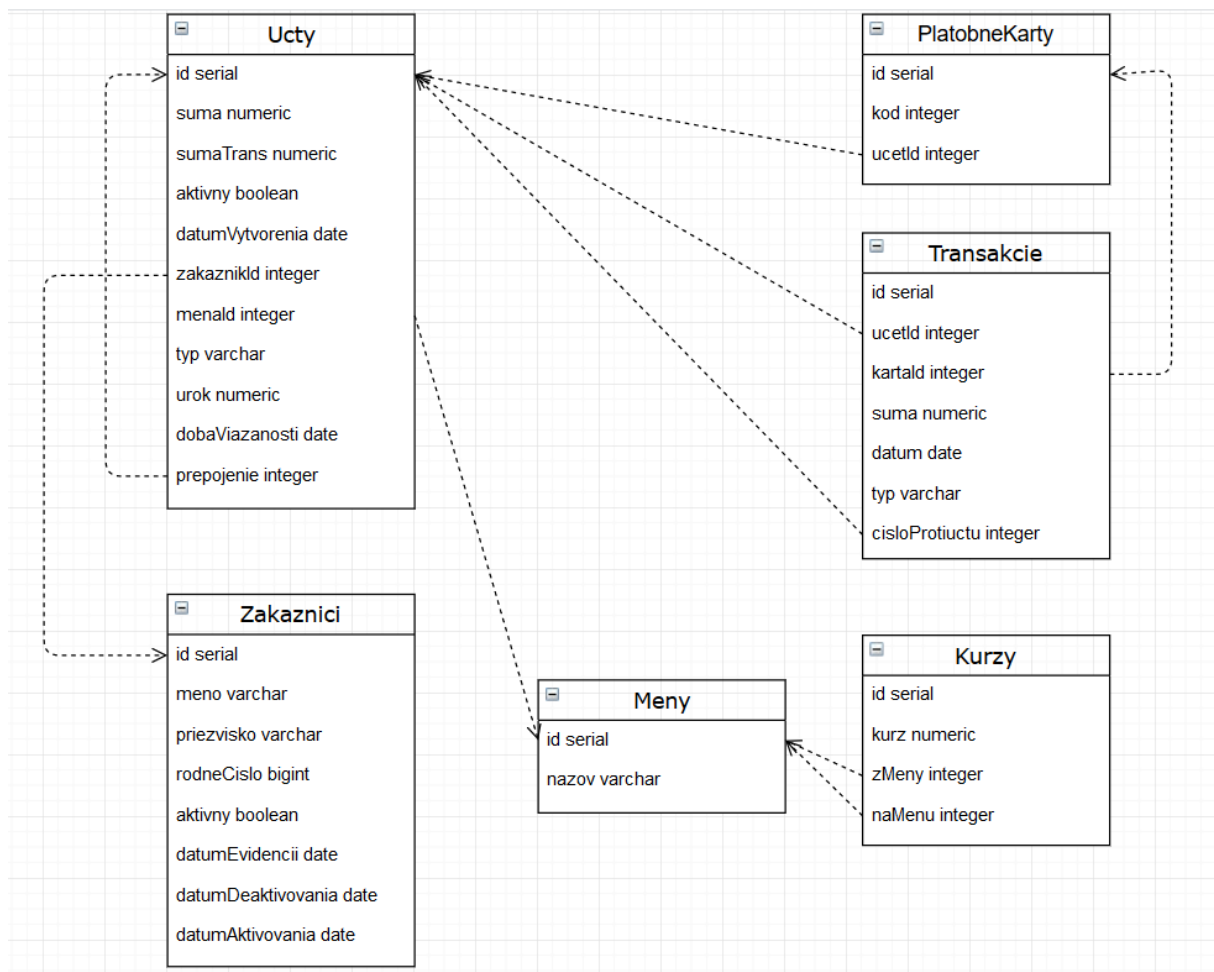
- vklad/výber (v hotovosti)
- (transakcia) medzi dvomi účtami
- (transakcie) s kartou

3 Zmeny voči zadaniu

Nie sú žiadne zmeny.

4 Relačná databáza

Zobrazuje výslednú relačnú databázu, ktorá vznikla transformovaním entitno relačného modelu. Podmnožiny *Bezne ucty*, *Sporiace ucty*, *Terminovane ucty*, *Vklady/Vybery*, *Prevody medzi 2 uctami*, *Platby s kartou* boli transformované stratégiou *všetky podmnožiny do jednej tabuľky*. (2. obrázok, ďalšia strana)



2. Obrázok - relačný model dát

5 Organizácia kódu

Aplikácia je naprogramovaná v jazyku Java. Prístup do databázy je riešený cez JDBC. Zdrojový kód je rozdelený do viacerých balíkov.

- sk.uniba.fmph.simko.db2.application – Trieda Main slúži na spúšťanie celej aplikácie, spojenie s databázovým systémom sa udržiava v triede DBContext.
- sk.uniba.fmph.simko.db2.application.rdg – Triedy v tomto balíku reprezentujú každú tabuľku v databáze a aj ich finder-y, ktoré slúžia na vyhľadávanie entít. Podmnožiny *Bezne*, *Terminovane* a *Sporiace* ucty sú realizované ako jedna trieda (*Ucty*), v ktorej sa zapamätá typ účtu a podobne aj pri transakciách.
- sk.uniba.fmph.simko.db2.application.ts – Nachádza sa tu trieda *ZlozitejsieDomenoveOperacie*, ktorá obsahuje funkcie pre zložitejšie doménové operácie s databázou a ďalšia trieda pre chybové hlášky.
- sk.uniba.fmph.simko.db2.application.ui – Obsahuje triedu *MainMenu* pre používateľské rozhranie a viacerých *Printer*-ov, ktoré vypisujú dáta o danej entite.

6 Optimalizácia SQL

Pri veľkom množstve dát som potreboval rýchlejšie nájsť všetkých zákazníkov, ktorí sa vrátili v danom roku po n rokov, kedy boli deaktivovaní. Potreboval som to, pri štatistike o znovuzískaných zákazníkov. Pridal som index na tabuľku zákazníkov.

```
SELECT count(*) as pocet
FROM zakaznici WHERE aktivny IS TRUE AND datumAktivovania IS NOT NULL
                AND datumDeaktivovania IS NOT NULL
                AND EXTRACT(YEAR FROM datumAktivovania) = ?
                AND (EXTRACT(YEAR FROM datumAktivovania) - EXTRACT(YEAR
FROM datumDeaktivovania)) = ?
```

```
CREATE INDEX zIndex ON zakaznici(aktivny, datumAktivovania, EXTRACT(YEAR
FROM datumAktivovania));
```

Rýchlosti vykonania pri rôznych variantoch a množstve dát sú uvedené v tabuľke. Časy vykonania som meral pomocou príkazu **EXPLAIN ANALYSE**.

Počet riadkov v tabuľke zakaznici	Čas vykonania pôvodného kódu (bez indexu)	Čas vykonania opraveného kódu
10000	2.580	0.547
25000	8.371	1.040
50000	11.985	2.186
100000	22.869	4.086

7 Riešený problém

Pôvodne som nerátal s faktom, že číslo protiúčtu môže byť aj z našej a aj z inej banky.

To by spôsobovalo problém, že keď si vytvoríme jednu novú transakciu medzi dvoma účtami ktoré sú oba evidované v našej banke, tak sa vytvorí jedna transakcia, pri ktorej strhne suma z účtu a aj prirába druhému účtu, ale pri transakciách účte, ktorý prijal sumu, by sme nevideli žiadnu transakciu ktorá by nám vysvetlila odkiaľ pribudli tie peniaze.

Preto pri vytvorení transakcii kontrolujem, či protiúčet nie je evidovaný v našej banke a ak áno, tak sa vytvorí ešte jedna transakcia, ktorá informuje zákazníka o príjme.