

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií

DATABÁZOVÉ SYSTÉMY
2016/2017

Dokumentace popisující finální schéma databáze
Zadání č. 16 – Banka

Martin Nizner (xnizne00)
Ján Gula (xgulaj00)

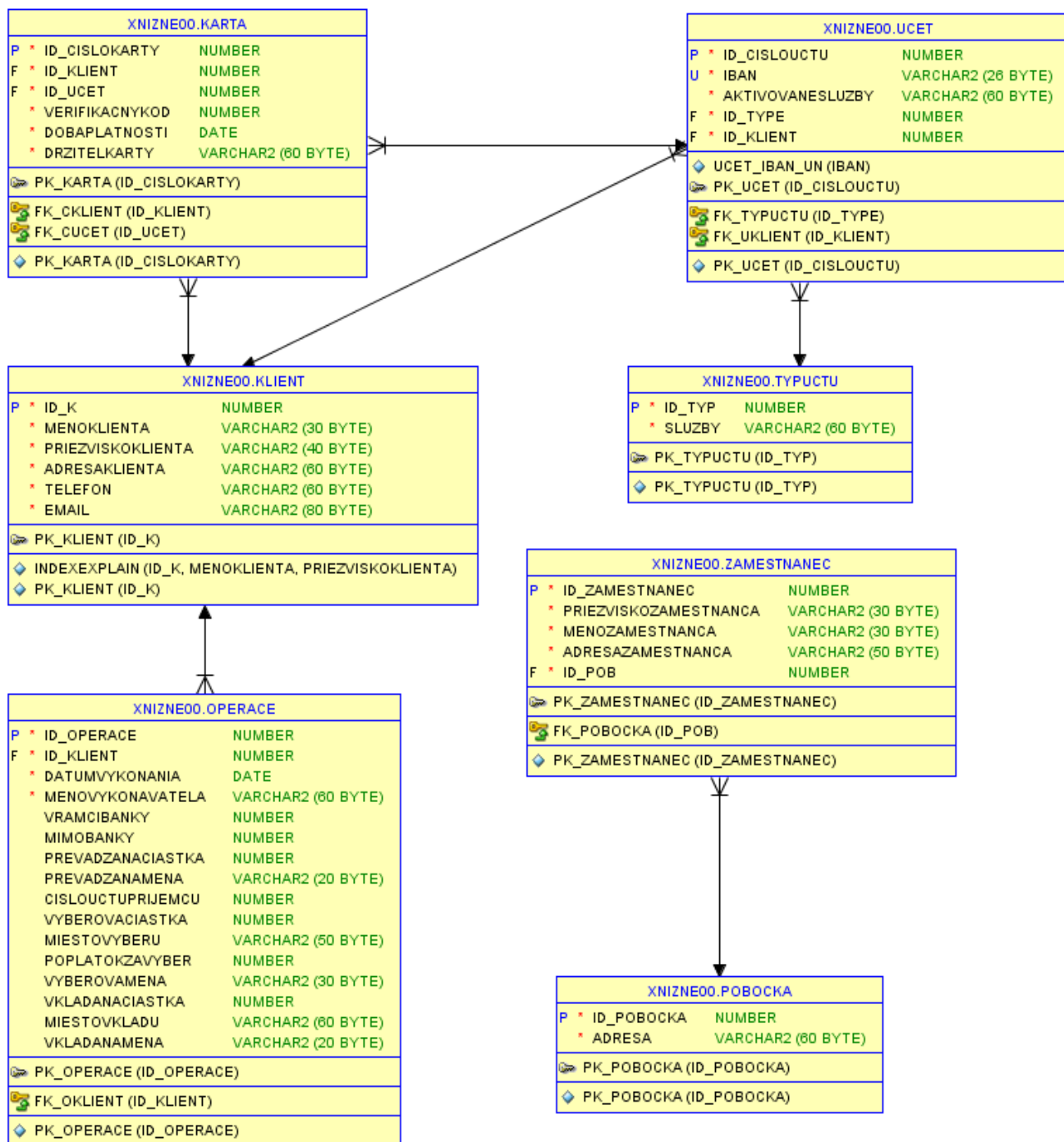
Brno, 1. května 2017

1. Zadaní

Navrhněte modul jednoduchého informačního systému banky, který bude umožňovat správu bankovních účtů. V systému jsou evidovány základní údaje o zaměstnancích banky i jejich klientech, kteří mají minimálně jeden účet u banky nebo jsou disponenty s nějakým účtem. Systém uchovává informace o různých operacích nad účty. Předpokládejte, že každý účet má pouze jednoho vlastníka, ale může mít více disponentů, kteří mají omezená práva u daného účtu. K účtu mohou být klientům vystaveny platební karty.

Banka nabízí více typů účtu, přičemž každý typ účtu má dostupné různé služby, které zákazník může nebo nemusí mít aktivovány. Základní operace nad účty jsou vklad, výběr z účtu, nebo převod mezi účty. Pokud se jedná o převod v rámci dané banky, transakce je dohledatelná z obou účtů. U každé transakce je možné zpětně dohledat, který klient o operaci zažádal, kdy a kým byla operace provedena, atd. Přičemž přímý přístup k informačnímu systému banky mají pouze pracovníci banky. Klienti pouze bance zadávají požadavky, které zaměstnanci zpracují a provedou požadovanou transakci. Banka může mít více poboček, u zaměstnanců je možné zjistit na které pobočce pracují a jaké účty jím byly na dané pobočce zřízeny.

2. Schéma databázy



2.1 Generalizácia/špecializácia

Tento vzťah je v našej implementácii využitý v prípade tabuľky OPERACE kde operáciou môžu byť napríklad vklad, výber alebo prevod avšak nemôžu byť súčasne.

3. Implementácia

Náš skript reprezentuje zjednodušený informačný systém pre správu bankových účtov. Informačný systém eviduje klientov, ktorý majú založený účet alebo účtom disponujú, ale aj zamestnancov a pobočky v ktorých títo zamestnanci pracujú. Ďalej naša implementácia zaznamenáva aj všetky operácie, ktoré boli s účtom vykonané ako aj kto ich vykonal. V neposlednom rade informačný systém udržiava informácie o type účtu a aj to, či je k danému účtu priradená platobná karta. Skript najprv vytvorí základné objekty relačnej databázy, tabuľky. Potom tieto objekty naplní vzorovými dátami. Na konci používame príkazy na zobrazenie dát z tabuliek ako ukážku správnej funkčnosti našej databázy. V skripte sú taktiež prítomne dva triggery a dve procedúry, ktoré budú popísané nižšie.

3.1 Triggery

Skript obsahuje dva databázove triggery. Prvý trigger slúži na kontrolu IBANu. Trigger sa aktivuje vždy pri VLOŽENÍ(INSERT) alebo AKTUALIZÁCIÍ(UPDATE) položky IBAN v tabuľke účet. Trigger kontroluje, či je dĺžka IBANu rovná 24 znakom, či je kód štátu zložený len z písmen od A po Z, a taktiež, či je zvyšok IBANu zložený iba z číslíc. Na konci je ešte špecifická kontrola, ktorá zisťuje deliteľnosť 11 súčtu násobenia jednotlivých číslíc IBANu špecifickým vzorom daným ČNB. Na tieto činnosti využívame príkazy na orezávanie reťazcov (SUBSTR a TRIM), ako aj prevod reťazca na číslo (TO_NUMBER), a taktiež prevod všetkých znakov vo vzore na prázdny reťazec (TRANSLATE), čo slúži na overenie prítomnosti iba čísel alebo iba písmen. Druhý implementovaný trigger slúži na autoinkrementáciu primárneho kľúča v tabuľke klient. Tento trigger je implementovaný pomocou sekvencie, ktorej úloha je uchovávanie posledného priradeného čísla.

3.2 Procedúry

Prítomné sú taktiež dve netriviálne procedúry, v ktorých využívame kurzor, aby sme boli schopní zachytiť viacriadkové výsledky. Taktiež je využité zachytávanie výnimiek (príkaz EXCEPTION) pre určenie reakcie na prípadné chybové stavy ako je napríklad delenie nulou, či iné. Použité procedúry sú typUctuPercentualne a pocZamVPob. Prvá procedúra očakáva na vstupe id typu účtu, ktorého percentuálne zastúpenie nás zaujíma a výstupom je zaokrúhlený počet percent účtov tohto typu, z celkového počtu účtov. Táto procedúra vyberie všetky údaje z tabuľky účet, ktoré skúma riadok po riadku. Ak je v niektorom riadku zhoda v id typ účtu, inkrementuje sa premenná, ktorá počíta účty daného typu a pri každej iterácii sa inkrementuje premenná pre všetky účty. Výstupom

tejto procedúry je vlastne zaokrúhlený podiel týchto dvoch hodnot, ktorý je následne vynásobený číslom 100 pre percentuálne zobrazenie. Druhá procedúra funguje na veľmi podobnom princípe ako prvá. Jej vstupom je id pobočky v ktorej pracujú zamestnanci a výstupom je určit presný počet zamestnancov, ktorí v tejto pobočke pracujú.

3.3 Explain plan s použitím indexu

Explain plan zobrazuje jednotlivé kroky, ktoré boli potrebné pre vykonanie zadanej operácie vrátane štatistických informácií k týmto krokom (napr. čas vykonania kroku, percentuálne vyťaženie procesoru apod.). Vytvorenie indexu slúži k urýchleniu vykonávania zadanej operácie. V nižšie uvedených tabuľkách je zobrazený priebeh zadanej operácie pred a po použití indexu.

PLAN_TABLE_OUTPUT

Plan hash value: 3930712395

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		36	1404	7 (15)	00:00:01
1	HASH GROUP BY		36	1404	7 (15)	00:00:01
2	MERGE JOIN CARTESIAN		36	1404	6 (0)	00:00:01
3	TABLE ACCESS FULL	KLIENT	6	234	3 (0)	00:00:01
4	BUFFER SORT		6		4 (25)	00:00:01
5	INDEX FAST FULL SCAN	PK_UCET	6		1 (0)	00:00:01

PLAN_TABLE_OUTPUT

PLAN_TABLE_OUTPUT

Plan hash value: 3700525591

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		36	1404	5 (20)	00:00:01
1	HASH GROUP BY		36	1404	5 (20)	00:00:01
2	MERGE JOIN CARTESIAN		36	1404	4 (0)	00:00:01

```
| 3 | INDEX FULL SCAN | INDEXEXPLAIN | 6 | 234 | 1 (0)|
00:00:01 |
| 4 | BUFFER SORT | | 6 | | 4 (25)| 00:00:01 |
| 5 | INDEX FAST FULL SCAN| PK_UCET | 6 | | 1 (0)|
00:00:01 |
PLAN_TABLE_OUTPUT
-----
```

3.4 Prístupové práva

Prístupové práva sú definované pre zamestnanca, ktorý má obmedzený prístup k niektorým tabuľkám : zamestnanec, pobočka. Tieto tabuľky slúžia pre riešenie konfliktov a preto nie je potrebné zamestnancovi udeľovať prístup.

3.5 Materializovaný pohľad

Materializovaný pohľad patrí druhému členovi tímu a využíva tabuľky prvého člena tímu. Pre funkčnosť tohto kroku je potrebné, aby prístupové práva boli už nastavené. Najprv boli vytvorené logy, ktoré sú potrebné pre optimalizáciu materializovaného pohľadu. Následne bol vytvorený materializovaný pohľad.