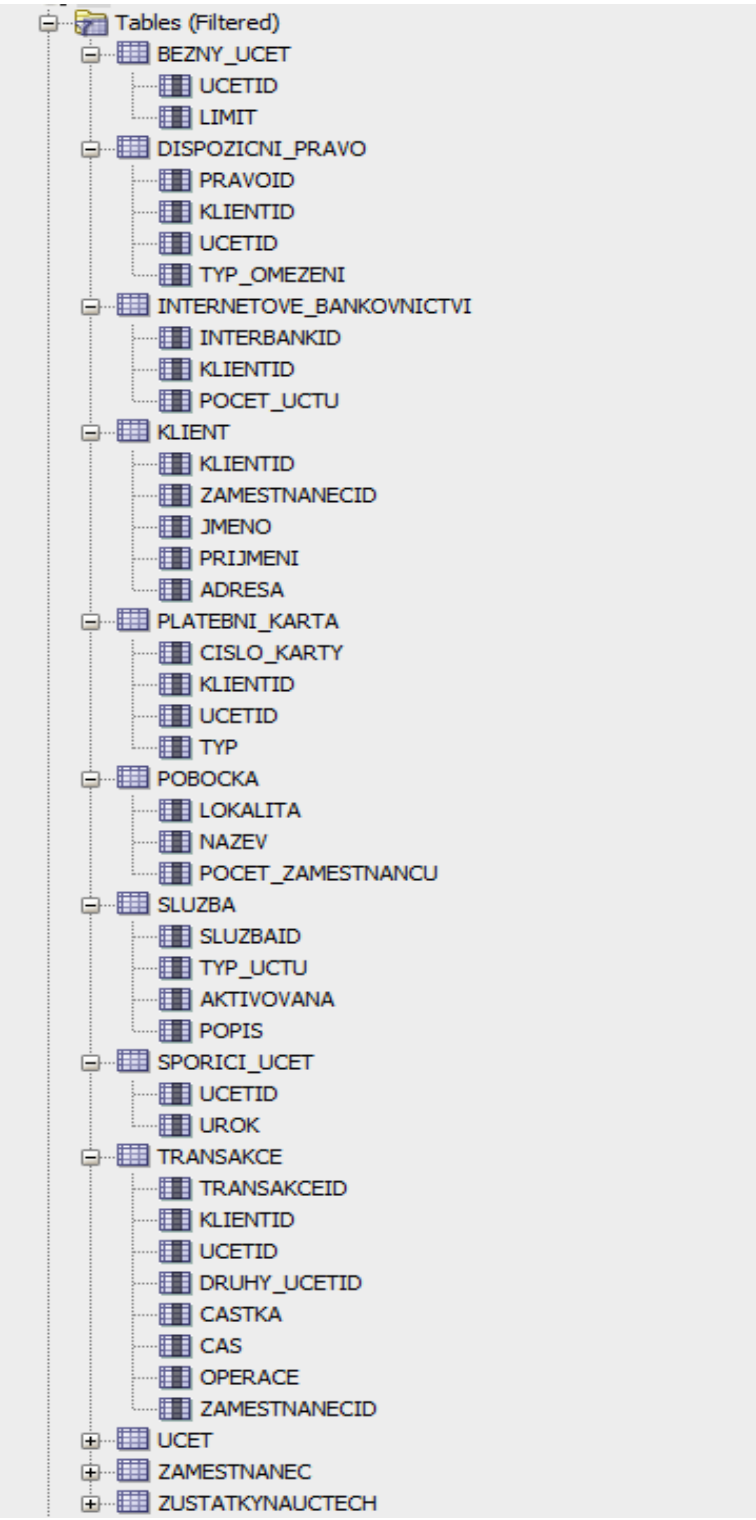
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Databázové systémy
SQL skript

Daniel Olearčin (xolear00) 2.5.2021 Vojtech Kronika(xkroni01)

1 Vytvorenie objektov

Našou prvou úlohou bolo vytvoriť objekty v jazyku SQL podľa predom vytvoreného ER diagramu. Každému objektu sme vytvorili primárne kľúče pomocou autoinkrementu a ďalšie veci ktoré ich špecifikovali. Nakoniec sme objekty poprepajali tak ako boli prepojené na ER Diagrame.



2 Vytvorenie selectov

Našou druhou úlohou bolo vytvoriť rôzne typy selectov.

- Prvý select na vypísanie mien a priezviska ktorý pracujú na pobočke ”Banka v Brně”.
- Druhý select vypíše mien a priezviská všetkých klientov so zostatkom viac ako 10000 na účte.
- Tretí select vypíše transakcie zaznamenané v pobočke ”Banka v Prahe”.
- Štvrtý select vypíše počet vytvorených klientov zamestnancom.
- Piaty select vypíše súčet zostatkov jedneho klienta na všetkých účtoch.
- Šiesty select vypíše transakcie medzi dvoma existujúcimi účtami.
- Posledný select vypíše klientov ktorý vlastnia debitní platební kartu.

3 Tvorba procedúr a triggerov

3.1 Procedúry

Naša prvá procedúra po zavolaní vypíše všetky banky a počet zamestnancov ktorý tam pracujú. Ak banka nemá zamestnancov vypíše sa error. V prvej procedúre sme využili CURSOR ktorý sme využil na cyklus LOOP ktorý sa opakuje toľko krát koľko je bánk uložených v databáze. Nakoniec sa v cykle rozhodujeme na základe IF konštrukcie či sa zavolá error alebo výpis na výstup.

Druhá procedúra pri zavolaní vypíše všetký účty ktoré majú zostatok vyšší ako 50000. Na vytvorenie tejto procedúry sme využili taktiež CURSOR, cyklus LOOP a aj IF konštrukciu ale navyše sme použili aj premennú s datovým typom ”ucet.zustatek

3.2 Triggery

Náš prvý trigger sa zavolá vždy pri vložení do tabuľky transakce. Slúži na aktualizáciu zostatkov na úctoch klientov po vykonaní transakcii.

Druhý trigger slúži na vytvorenie ID pri tabuľke zamestnanec. Zavolá sa vždy pred vložením do tabuľky. Slúži ako autoinkrement ID.

4 Tvorba Explain plan s indexom a materialized view

4.1 Explain plan

Súčasťou tretej úlohy bolo aj vytvoriť explain plain pomocou indexu a materialized view. Explain plan sme vytvorili pomocou selectu pri ktorom sme využili Agregačnú funkciu COUNT a taktiež sme zakomponovali do selectu GROUP BY. Navyše sme skombinovali Explain plan s indexom. Následným výpisom sme zistili že na realizáciu potrebujeme 30 operácií.

```
PLAN_TABLE_OUTPUT
-------------------------------------------------------------------------------------------
Plan hash value: 2833454906

-------------------------------------------------------------------------------------------
| Id  | Operation                     | Name        | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |             |     4 |   252 |     6  (34)| 00:00:01 |
|   1 |  HASH GROUP BY                |             |     4 |   252 |     6  (34)| 00:00:01 |
|   2 |   NESTED LOOPS                |             |     4 |   252 |     5  (20)| 00:00:01 |
|   3 |    NESTED LOOPS               |             |     4 |   252 |     5  (20)| 00:00:01 |
|   4 |     VIEW                      | VW_GBF_7    |     4 |   104 |     4  (25)| 00:00:01 |
|   5 |      HASH GROUP BY            |             |     4 |    52 |     4  (25)| 00:00:01 |

PLAN_TABLE_OUTPUT
-------------------------------------------------------------------------------------------
|   6 |       TABLE ACCESS FULL       | KLIENT      |     4 |    52 |     3   (0)| 00:00:01 |
|*  7 |     INDEX UNIQUE SCAN         | SYS_C008855 |     1 |       |     0   (0)| 00:00:01 |
|   8 |    TABLE ACCESS BY INDEX ROWID| ZAMESTNANEC |     1 |    37 |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   7 - access("ITEM_1"="ZAMESTNANEC"."ZAMESTNANECID")
```

4.2 Materialized view

Na vytvorenie materialized view sme najprv potrebovali dat opravnenia druhému účtu. Následne sme vytvorili materialized view, ktorý slúži na vytvorenie pohľadu na tabuľku súčtov zostatkov na účtoch klienta. Tento výpis je akutálny a nemení sa až pokým sa nezavolá COMMIT. Pri zavolaní COMMIT sa nahrajú do tabuľky aktuálne údaje.