



Dokumentace k projektu pro předmět IDS

**Banka**

projekt č.: 26

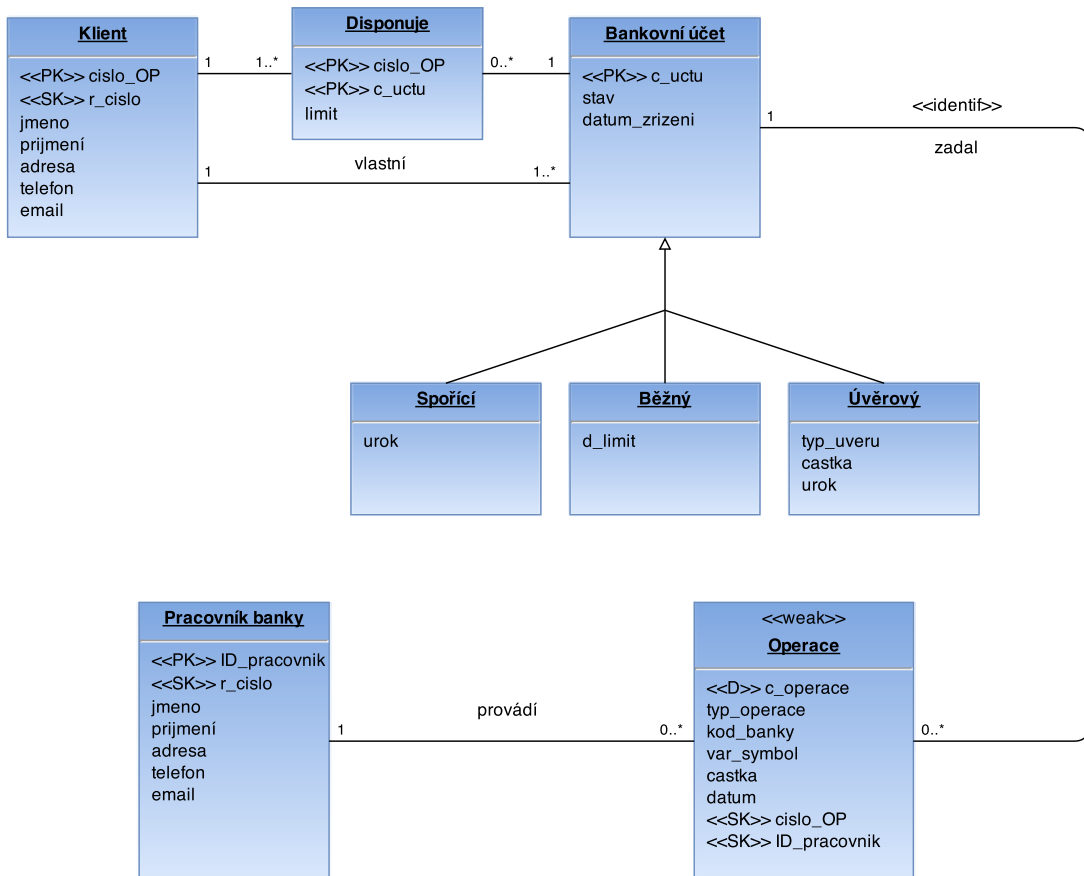
**1. května 2017**

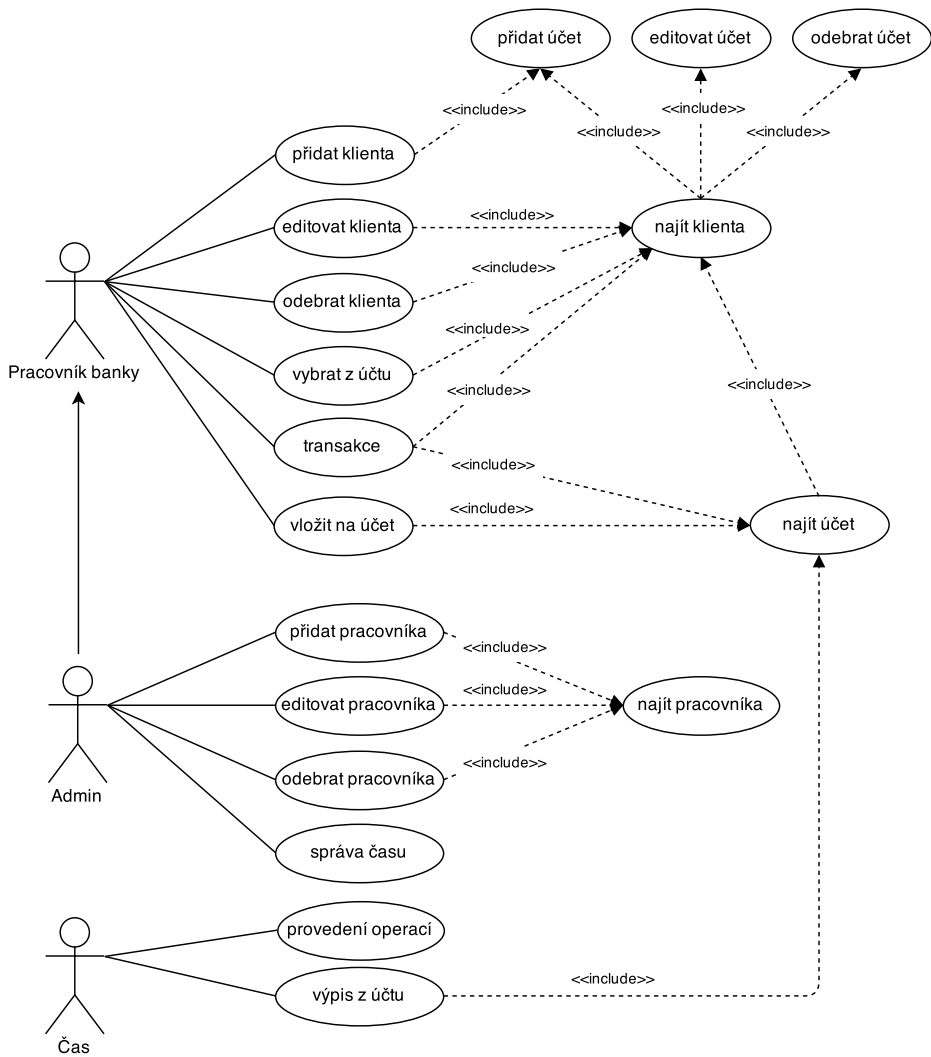
Autoři: Štěpán Vích, xvichs00@stud.fit.vutbr.cz  
Michal Klhůfek, xklhuf00@stud.fit.vutbr.cz  
Fakulta Informačních Technologí  
Vysoké Učení Technické v Brně

**Název projektu: Banka**

**Zadání:**

Navrhněte modul informačního systému banky pro správu účtů. Modul musí evidovat klienty, jejich účty a operace s nimi. Předpokládejte, že každý účet má jednoho vlastníka, ale s účtem může disponovat více osob, které určí vlastník. Operace zahrnují vklad na účet, výběr z účtu a převod na jiný účet (téže či jiné banky). Systém musí ukládat informaci o všech operacích s účtem (kdo zadal, kdy, jaká operace a částka, kdo provedl). Se systémem vždy přímo komunikuje pouze pracovník banky. Systém musí také mimo jiné poskytovat výpis z účtu, který se posílá vlastníkově, tj. výpis všech operací s účtem za dané období.





## GENERALIZACE/SPECIALIZACE

- Ve skriptu jsme využili bankovní účet pro účely generalizace/specializace. Ten pak dále rozdělujeme na tři dílčí typy: úvěrový účet, spořicí účet a běžný účet. V databázi je to řešeno za pomoci tří tabulek, kde každá tabulka neprezentuje daný typ účtu. Vzájemné propojení tabulek je řešeno pomocí sekundárního klíče (číslo účtu) do tabulky bankovní účet, která je nadřazená.

## TRIGGER

- První trigger je vytvořený pro účely kontroly rodného čísla u tabulek PRACOVNÍK BANKY a KLIENT. Kontrola se provádí při každém vložení do tabulky a následně při updatu dat. Samotná kontrola se vztahuje na rodné číslo, které vybere ze sloupce r\_cislo. Z něho se následně vypočítá den, měsíc a rok. Pokud rodné číslo nebude dělitelné jedenácti, vypíše se chybové hlášení. Je zde bráno v úvahu i ošetření výjimek, které nastanou při neplatných datech.
- Druhý trigger slouží pro generování hodnot primárního klíče u čísla účtu v bankovním účtu. Generování čísel využívá sekvenci čísel které se zvyšují o jedničku.

## PROCEDURE, CURSOR

- Nacházejí se zde dvě procedury, kde jedna z nich je na odstranění pracovníka a druhá na přidání pracovníka.
  - V proceduře na odstranění pracovníka si vytváříme kurzor, který nám zpracovává dotaz vybírající pracovníky s daným id, které je předáno jako parametr procedury. V těle procedury si otevíráme kurzor a cyklíme v něm pomocí LOOP, kde si pokaždé příkazem FETCH uložíme výsledek dotazu do proměnné. Výsledek obsahuje id pracovníka, který použijeme k odstranění pracovníka. Cyklus končí při nenalezení výsledku.
  - V proceduře na přidání pracovníka, používáme kurzor, který vybírá id pracovníka a pokud id pracovníka v databázi neexistuje, vloží nového pracovníka do tabulky s danými hodnotami převzatými ze vstupních parametrů. Jestliže byly nalezeny nějaké výsledky, klient již existuje a není povoleno jej přidat.

## EXPLAIN PLAN

- Pro ukázkou jsou ve skriptu vytvořeny dva explain plány. První pro výstup bez indexu, druhý pro výstup s dvěma indexy.
  - První plán pro výstup bez indexu obsahuje dotaz SELECT, který nepoužije žádný index což nám dává vysokou cenu za použití, ale nízké nároky na výkon CPU.

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		7	490	7 (15)	00:00:01
1	HASH GROUP BY		7	490	7 (15)	00:00:01
* 2	HASH JOIN		7	490	6 (0)	00:00:01
3	TABLE ACCESS FULL	KLIENT	5	185	3 (0)	00:00:01
4	TABLE ACCESS FULL	DISPONUJE	7	231	3 (0)	00:00:01

- V porovnání s druhou variantou ve které jsme použili výstup s dvěma indexy, je výsledná cena na výpočet nižší, ale nároky na CPU jsou vyšší.

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		7	490	3 (34)	00:00:01
1	HASH GROUP BY		7	490	3 (34)	00:00:01
* 2	HASH JOIN		7	490	2 (0)	00:00:01
3	INDEX FULL SCAN	INDEX1	5	185	1 (0)	00:00:01
4	INDEX FULL SCAN	INDEX2	7	231	1 (0)	00:00:01

## MATERIALIZED VIEW

- Materializovaný pohled sestaven s příkazem SELECT, kde jsme umožnili přístup do nové tabulky, která obsahuje klienta i jeho účet. Díky tomu se nemusí pokládat složité dotazy na obě tabulky, ale stačí jeden, proto abychom zjistili, jaký účet vlastní který klient. Záznamy se vždy aktualizují po provedení insert do tabulky klient.

## GRANT

- Samotná přístupová práva řešena zvlášť pro každou tabulku. Jednotlivá práva se nastaví pomocí GRANT ALL, kde je dále potřebné uvést název tabulky a uživatele, který bude mít k tabulkám přístup. ALL značí, že mu jsou dána veškerá práva, co lze dát.