

Projekt IDS

4 a 5.část

SQL skript pro vytvoření pokročilých
objektů schématu databáze

Zdeněk Lapeš (xlapes02)
Andrej Bínovský (xbinov00)

2.5.2022

Zadání (Banka):

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 nebo disponentům vystaveny platební karty a zřízeno internetové bankovníctví. V rámci jednoho internetového bankovníctví klient spravuje všechny své účty a všechny účty, ke kterým má dispoziční právo.

Banka nabízí více typů účtu, přičemž každý typ účtu má dostupné různé služby, které zákazník může nebo nemusí mít aktivovány. Klienti banky zadávají požadavky prostřednictvím zaměstnanců na pobočce, kartou v platebních terminálech, nebo přes internetové bankovníctví. Základní operace nad účty jsou vklad, výběr z účtu, převod mezi účty, nebo platba kartou. 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 byla provedena a, v případě zadání operace na pobočce, kterým zaměstnancem byla operace provedena. Banka může mít více poboček, u zaměstnanců je možné zjistit na které pobočce pracují a jaké účty jím byly na dané pobočce zřízeny.

Popis:

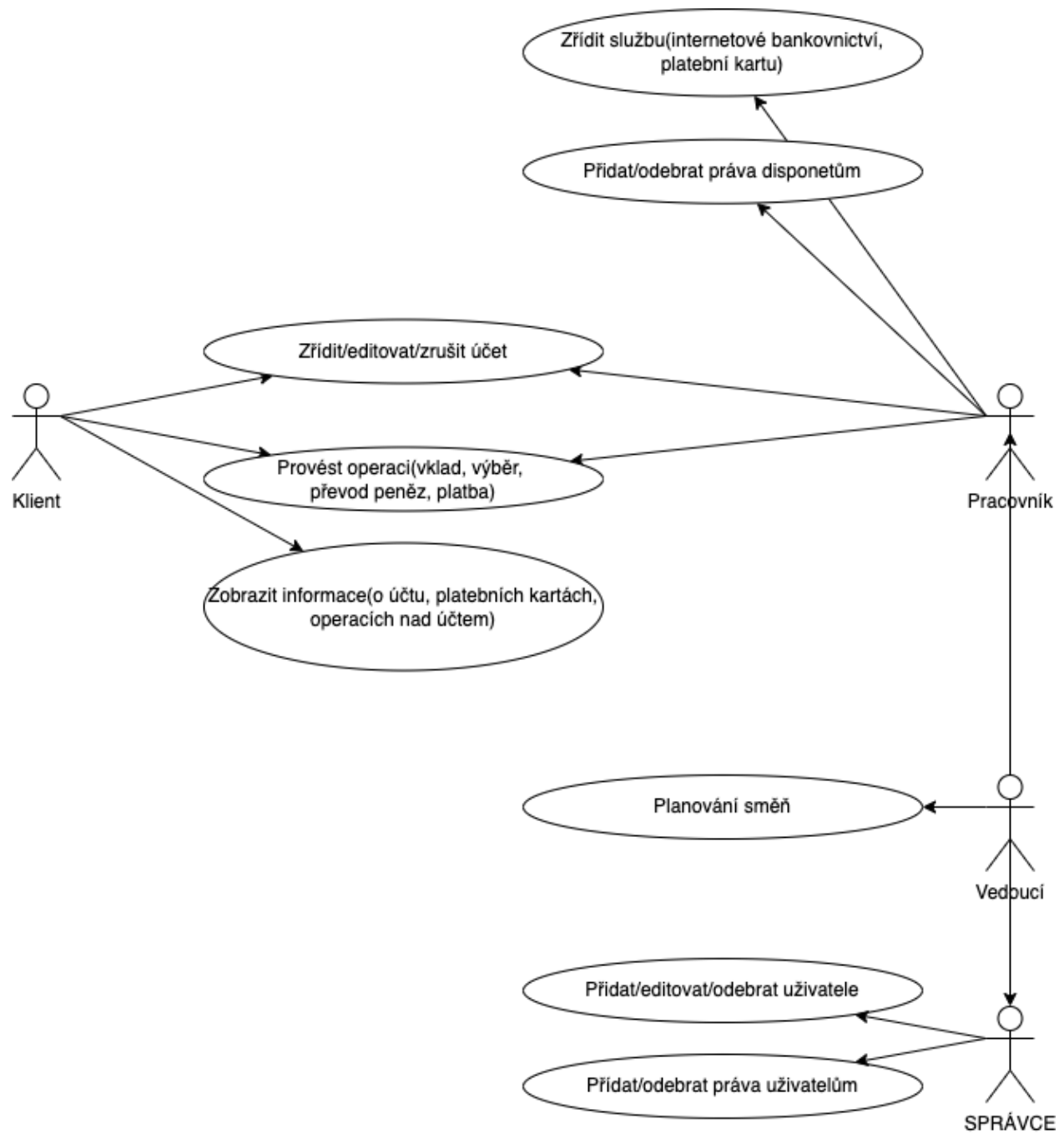
Entity Relationship Diagram

Návrh Datového modelu (ERD) jsme vytvářeli vzhledem ke zkušenostem s bankami a jejich informačními systémy. V ERD jsme považovali za vhodné použít 2x vztahový atribut. Ve spojení s právem disponenta, neboť každý disponent může provádět různé úkony nad bankovním účtem. A při operacích nad bankovními účty, které mohou být též různého typu jako transakce (banka, banka) anebo vytvoření bankovního účtu (zaměstnanec, banka).

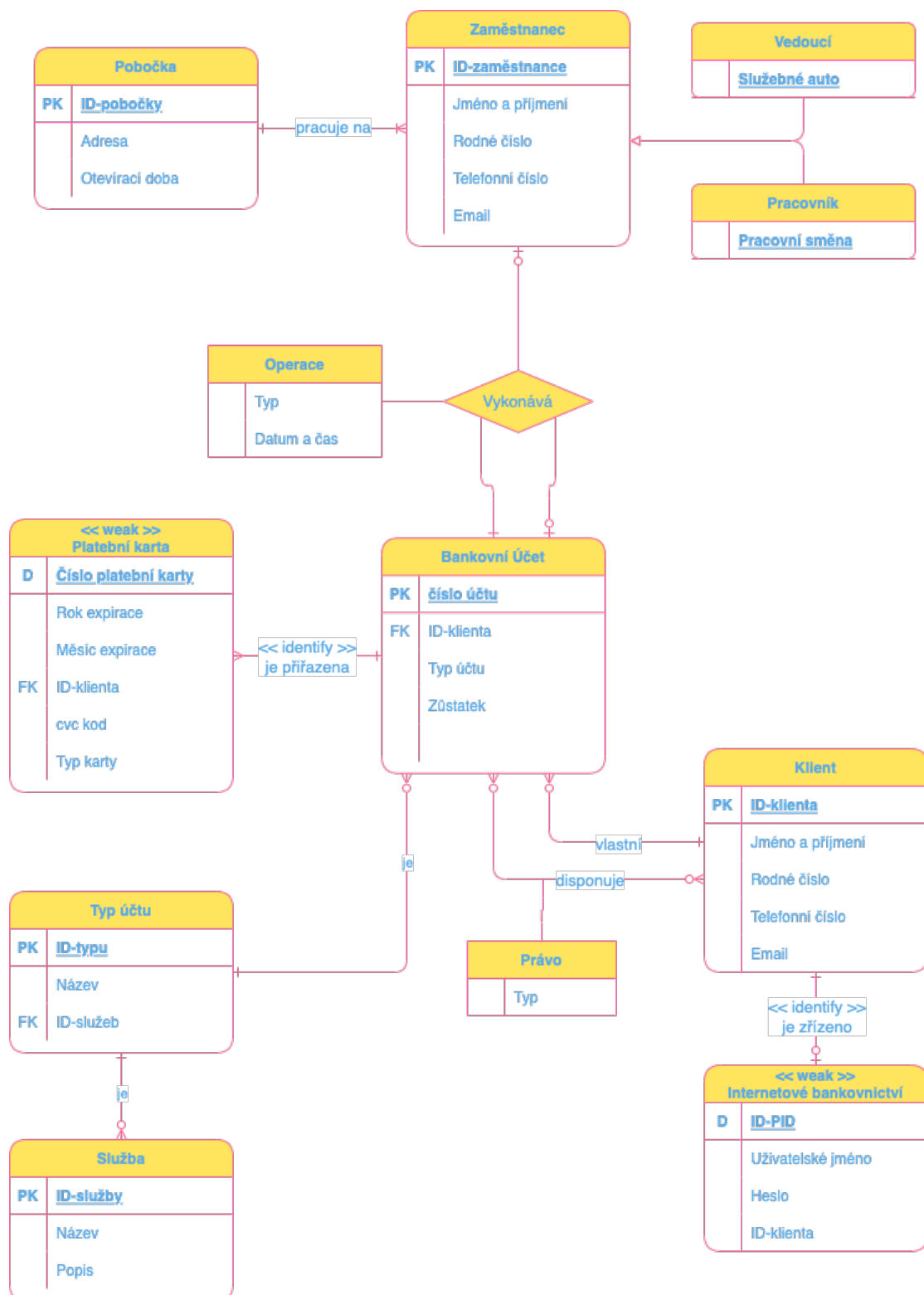
Use Case Diagram

Při návrhu Use Case Diagramu jsme rozlišovali 4 skupiny uživatelů bankovního systému, kterými jsou klient, pracovník, vedoucí a správce.

Za požadavek na pracovníka jsme považovali případ užití „Provést operaci“, kterou může provádět klient sám anebo požádáním pracovníka na pobočce. Pracovník se stará o účty klientům.



Use Case Diagram Bankovního Systému



Entity Relationship Diagram pro Bankovní systém

Implementace:

DROP

Ze začátku je potřeba vymazat tabulky z důvodu možných kolizí s již existujícími tabulkami.

CREATE

Po smazání tabulek se vytvoří nové, totožné s těmi, které byly vytvořeny ve 2.části projektu.

TRIGGER

Oba trigger se vytvoří před vložením záznamů do tabulky.

1.Trigger zkonvertuje jméno a příjmení klienta na malá písmena.

2.Trigger zkontroluje zda nad daným účtem provádí operaci uživatel, který je majitel a nebo je disponentem daného účtu.

INSERT

Po úspěšném vytvoření triggerů se tabulky naplní ukázkovými daty, pro demonstraci dalších úkonů.

PROCEDURE

1.Procedure ukáže kolik má banka klientů, s kolika účty její klienti disponují a procentuální rozložení v 3 různých typech účtů.

2.Procedura přijímá jako argument typ účtu a ukáže které účty operují s daným typem účtu a kolik takových klientů je celkem. Pro iteraci se používá cursor, který provede fetch pouze pro jeden column (Bank_account.client_fk%type) tabulky.

EXPLAIN PLAN

Ukazuje jak SQL prochází daty a zpracovává příkaz. My jsme pro ukázkou použili SELECT, který ukazuje všechny klienty kteří mají více jak 2 účty a již provedli WITHDRAWAL request připřikayu je proveden join přes 3 tabulky. Následně se provede indexace pro column client_fk v Bank_account tabulce.

Druhý explain plan by měl vykazovat zrychlení při provádění příkazu, díky indexaci.

PRIVILEGES

Uživatel xlapes02 přidá práva uživateli xbinov00, aby mohl editovat všechny tabulky a pracovat s oběma 2 procedures a používat user_accounts view, přičemž tyto všechny objekty existují pouze v databázi uživatele xlapes02.

MATERIALIZED VIEW

Materializovaná pohled vypíše informace o uživateli(rodné číslo, jméno, příjmení a kolik mají účtů). Materializovaný pohled je rychlejší, protože je uložený, ovšem nevýhoda je, že při příkazu UPDATE se neaktualizuje automaticky.

Zdroje:

<https://www.techonthenet.com/oracle/indexes.php>

https://docs.oracle.com/cd/A57673_01/DOC/server/doc/SCN73/ch15.htm

https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_6009.htm