

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií

DATABÁZOVÉ SYSTÉMY
2018/2019

Projekt IDS - Upíří krvní banka

Matej Záhorský (xzahor04)

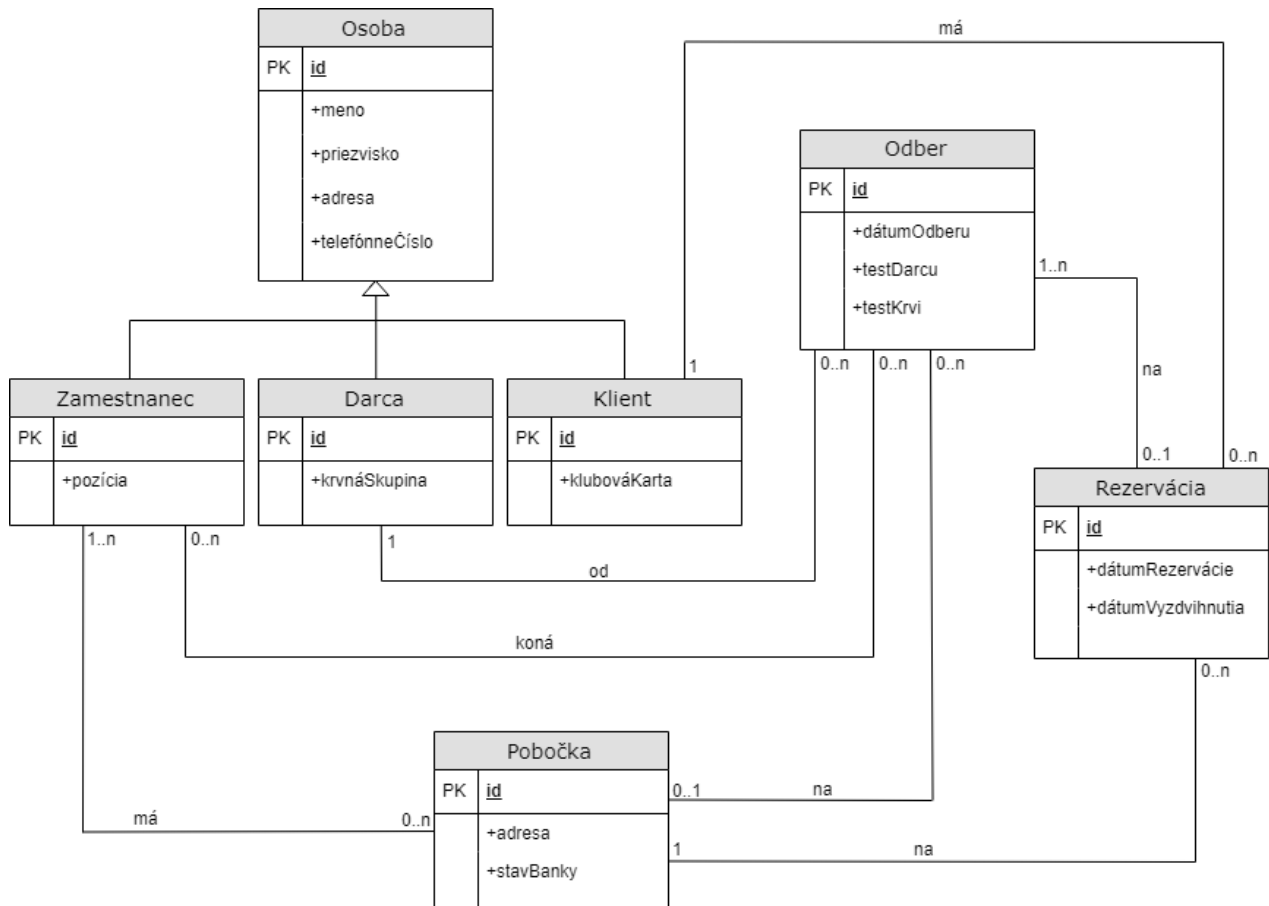
Adam Abrahám (xabrah04)

Brno, 29. dubna 2019

1. Zadanie

Navrhňte informační systém pro upíří krevní banku. Krevní banka si zve své dárce k dalšímu odběru podle stavu krevní banky a data posledního odběru daného dárce. Každý takový dárce obdrží pozvánku na konkrétní pobočku krevní banky. V IS jsou uloženy informace, zda dárce na pozvánku zareagoval - dostavil se k odběru či nikoliv. Před vlastním odběrem dárce podstoupí test, zda mu může být odebrána krev, pokud je test v pořádku je dárce odebrána krev (jedna dávka), která dále musí znovu podstoupit test, z něhož se vyhodnotí její vhodnost ke konzumaci. Vzniklou dávku si u krevní banky mohou rezervovat různí klienti, o nichž je v databázi uložen záznam. Rezervace se provádí pro zadané množství dávek u pobočky, na které je materiál uskladněn, na určité datum.

2. ERD Diagram a generalizácia



V našej databáze sa vyskytujú 3 druhy osôb, ktoré zdieľajú väčšinu informácií, a teda entita **Osoba** je generalizáciou entít **Zamestnanec**, **Darca** a **Klient**.

3. Triggery

3.1 ID_Odber_TRG

Tento trigger reaguje na požiadavku vloženia dát do tabuľky `Odber` predtým, ako k tomu dôjde. Jeho úlohou je vygenerovať jednoznačné identifikačné číslo (primárny kľúč) pre každý záznam v tejto tabuľke. Toto číslo sa vyberá zo sekvencie `ID_Odber_SEQ` aj v prípade, že sa užívateľ pokúsi kľúč explicitne zadať, čím zároveň dochádza k zabezpečeniu proti duplicitným primárnym kľúčom.

3.2 Rezervacia_Davky_TRG

Do tabuľky `Odber` sa pridávajú záznamy všetkých plánovaných odberov, i tých nevykonaných. Z tohto dôvodu je nevyhnutné vykonať kontrolu, či rezervovaná dávka skutočne existuje. Tento trigger testuje, či bol test krvi úspešný a ak áno, dávka je pripravená k užitiu. V opačnom prípade trigger vyvolá výnimku `VALUE_ERROR` a záznam v tabuľke `Odber` sa neaktualizuje.

3.3 Zvys_StavBanky_TRG a Zniz_StavBanky_TRG

Stav banky reprezentuje množstvo krvi dostupnej na danej pobočke. Túto hodnotu je možné zadať manuálne, napríklad v prípade prevozu krvi z jednej pobočky do druhej. V momente, keď sa úspešne vykoná odber na určitej pobočke, jej stav sa aktualizuje (hodnota sa zvýši) alebo naopak po vyzdvihnutí dávky (dávok) sa hodnota zníži. Toto nám pomôže predísť duplicitným záznamom s rezerváciami, ktoré už boli zadane.

4. Procedúry

4.1 `Uspesnost_Darcov_PRC`

Táto procedúra slúži na výpočet štatistiky, ktorá udáva úspešnosť jednotlivých darcov na ich odberoch. Táto informácia môže byť užitočná v tom prípade, keď by vedenie chcelo prestať posielat' neaktívnym darcom pozvánky na odbery, alebo naopak odmeniť tých aktívnych.

4.2 `Mnozstvo_Davok_PRC`

Podobne ako predchádzajúca procedúra, aj táto slúži na výpis štatistických informácií. Tá pre zadaného klienta vypíše, koľko percent zo všetkých dostupných dávok daný klient odobral. Na základe tejto informácie môže vedenie zamedziť danému klientovi prístup k ďalším dávkam krvi, alebo naopak ponúknuť dávky tým, ktorí si ešte žiadnu nevzali.

5. Explain plan

Explain plan slúži na výpis operácii, ktoré sú nutné na vykonanie zadanej požiadavky. Nasledujúca tabuľka popisuje plán vygenerovaný bez použitia indexu:

PLAN_TABLE_OUTPUT

Plan hash value: 2259862485

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		2	18	4 (25)	00:00:01
1	HASH GROUP BY		2	18	4 (25)	00:00:01
* 2	TABLE ACCESS FULL	REZERVACIA	2	18	3 (0)	00:00:01

Predicate Information (identified by operation id):

2 - filter("REZERVACIA"."DATUMVYZDVIHNUTIA" IS NOT NULL)

Note

- dynamic statistics used: dynamic sampling (level=2)

18 rows selected

Po použití indexu IDX, ktorý filtroval dáta v tabuľke Rezervacia podľa stĺpca DatumVyzdvihnutia sme dostali nasledovný výstup:

PLAN_TABLE_OUTPUT

Plan hash value: 3832420172

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		2	18	1 (0)	00:00:01
1	SORT GROUP BY NOSORT		2	18	1 (0)	00:00:01
* 2	INDEX FULL SCAN	IDX	2	18	1 (0)	00:00:01

Predicate Information (identified by operation id):

2 - filter("REZERVACIA"."DATUMVYZDVIHNUTIA" IS NOT NULL)

Note

- dynamic statistics used: dynamic sampling (level=2)

18 rows selected

V stĺpci `Cost` môžeme jasne vidieť zlepšenie oproti plánu bez použitia indexu. Táto hodnota uvádza výstup optimalizátora, ktorý na základe určitých informácií vyhodnotil cenu danej operácie. Z obrázka č. 1 môžeme vidieť, že sa v tabuľke `Rezervacia` pristupuje ku všetkým riadkom a stĺpcom (operácia `TABLE ACCESS FULL`), pričom po použití indexu sa pristupuje len k indexovaným riadkom (operácia `INDEX FULL SCAN`). Z tohto faktu vyplýva, že sme dospeli k optimalizácii za použitia indexu `IDX`, ktorý sa použije na vyhľadanie riadkov potrebných pre našu požiadavku.

6. Materializovaný pohľad a oprávnenia

Materializovaný pohľad slúži na zachytenie a uchovanie výsledku požiadavky na tabuľku bez nutnosti opätovného prechádzania. V našom pohľade sme nastavili rôzne vlastnosti, ako napr. `NOLOGGING`, vďaka ktorému sa nezaznamenávajú operácie s daným pohľadom. `CACHE` ukladá čítané dáta do vyrovnávacej pamäte, vďaka čomu toto čítanie optimalizuje. `REFRESH FAST ON COMMIT` aktualizuje pohľad na základe novo pridaných alebo aktualizovaných dát v tabuľke na základe vytvoreného logu. Tento pohľad sa naplní hneď po jeho vytvorení pomocou nastavenia `BUILD IMMEDIATE`. Pohľad sa v našom prípade vytvára pre tabuľku druhého užívateľa, pre ktorého sa zároveň nastavujú práva na operácie s tabuľkami pomocou `GRANT`. Na základe ukážky v kóde je možné vidieť, že pohľad je zastaralý i napriek vloženiu dát do tabuľky, a teda je nutné ho aktualizovať pomocou požiadavky `COMMIT`.

7. Záver

SQL skript bol vytvorený a testovaný na Oracle serveri gort programom SQL Developer. Všetky časti boli dôkladne otestované, a preto sa očakáva plná funkcionálna.