Vytvořil: Michal Řezník
Login: xrezni28

Dokumentace k projektu v předmětu IDS 2020/2021

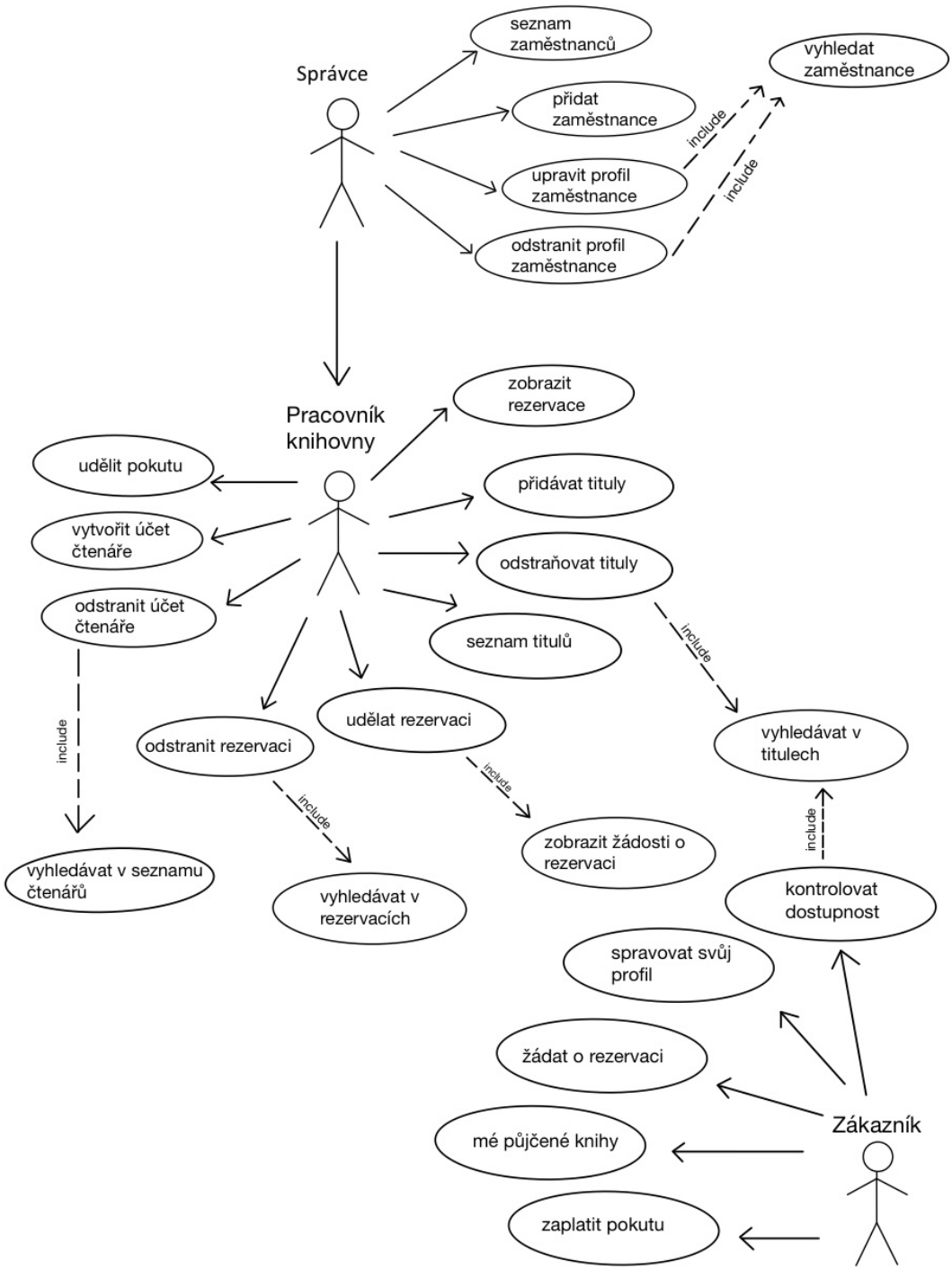
Zadání č.21 :Knihovna1

Vytvořil: Michal Řezník
Login: xrezni28

Zadání:

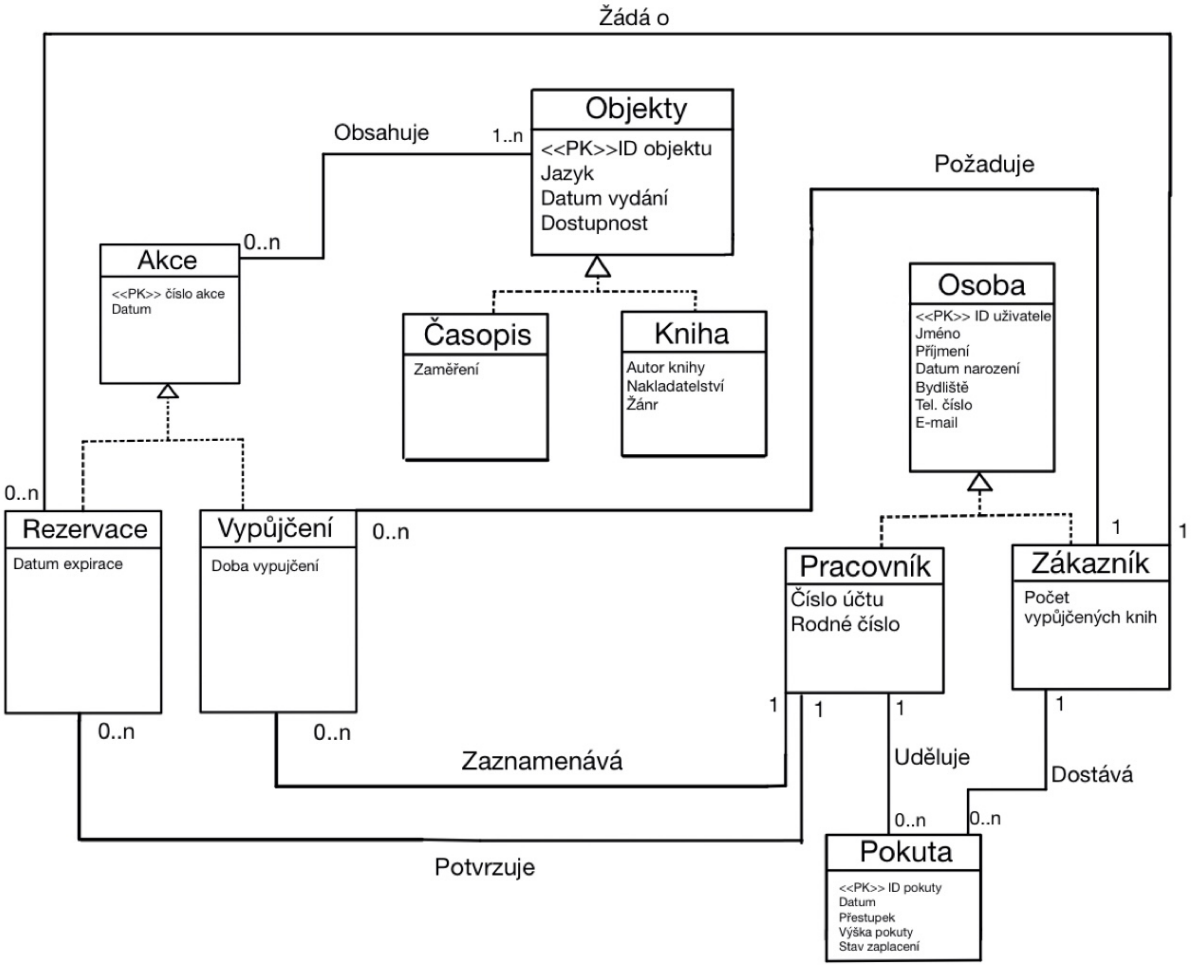
Vytvořte jednoduchý IS knihovny, který by poskytoval informace o titulech knih a časopisů, o registrovaných čtenářích, vypůjčených exemplářích apod. a umožňoval také provádět rezervace žádaných titulů. Z hlediska přístupu k datům uvažujte dvě skupiny uživatelů: pracovníky knihovny a čtenáře.

Use case diagram:



Vytvořil: Michal Řezník
Login: xrezni28

Entity relationship diagram:



Týmová práce

Našim úkolem bylo spolupracovat ve dvojici na řešení, což se v našem případě zrovna nepovedlo. Bohužel můj kolega nekomunikoval a nějak se neúčastnil na vytváření samotného skriptu a troufám si tvrdit, že tento finální produkt neobsahuje ani jeden řádek, který bych nepsal já. Proto jsem byl nucen po třech odevzdáních, které jsem řešil téměř bez jeho vědomí, se domluvit, že poslední část si odevzdáme každý sám. Děkuji za pochopení.

Implementace

Skript před samotným vytvářením tabulek obsahuje několik příkazů DROP, které slouží k zahození databázových objektů z minulého spuštění. Díky těmto příkazům se zamezuje konfliktům.

Po vytvoření tabulek reprezentující entity se nastaví klíče, triggery, procedury a tabulky se naplní testovacími daty, na kterých se poté testují příkazy SELECT, prohledávající danou databázi.

Vytvořil: Michal Řezník
Login: xrezni28

TRIGGERY

Mé řešení obsahuje 2 triggery, které by měly pomáhat automatizovat práci s danou databází.

První z nich je trigger „osobaID," který má za úkol vytvořit ID číslo uživatele v případě, že během jeho vytváření není zadáno. Využívá se zde sekvence „IDs,"která se inkrementuje pomocí NEXTVAL.

Druhý trigger slouží k dekrementaci množství daného objektu v případě, že je zákazník pokutován za jeho ztrátu.

PROCEDURY

Projekt obsahuje 2 procedury. Obě jsou netriviální, obsahují CURSOR a datový typ odkazující na řádek %ROWTYPE.

První z nich slouží ke spočítání, jaký podíl uživatelů naší databáze používá doménu, kterou zadáváme v argumentu.

Druhá slouží pro vypočítání výdajů na výplaty na daný měsíc.

MATERIALIZOVANÝ POHLED

Za úkol jsme dostali také vytvořit materializovaný pohled. I přes absenci kolegy jsem se úkol snažil splnit. Po vytvoření logů jsme použili CACHE pro optimalizaci čtení a BUILD IMMIDIATE, který nám pohled naplní okamžitě po spuštění. Dále jsem využil REFRESH ON COMIT, který zajišťuje to, že se nebude dotaz spouštět pokaždé celý.

POPIS EXPLAIN PLAN

Plán dotazu bez indexu

```
-------------------------------------------------------------------------------
| Id  | Operation             | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |           |     2 |    36 |     7  (15)| 00:00:01 |
|   1 |  HASH GROUP BY        |           |     2 |    36 |     7  (15)| 00:00:01 |
|   2 |   MERGE JOIN CARTESIAN|           |     2 |    36 |     6   (0)| 00:00:01 |
|   3 |    TABLE ACCESS FULL  | Vypujceni |     1 |       |     3   (0)| 00:00:01 |
|   4 |    BUFFER SORT        |           |     2 |    36 |     4  (25)| 00:00:01 |
```

Plán dotazu s použitým indexem

```
-------------------------------------------------------------------------------
| Id  | Operation             | Name    | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |         |     2 |    36 |     5  (20)| 00:00:01 |
|   1 |  HASH GROUP BY        |         |     2 |    36 |     5  (20)| 00:00:01 |
|   2 |   MERGE JOIN CARTESIAN|         |     2 |    36 |     4   (0)| 00:00:01 |
|   3 |    INDEX FULL SCAN    | MYINDEX |     1 |       |     1   (0)| 00:00:01 |
|   4 |    BUFFER SORT        |         |     2 |    36 |     4  (25)| 00:00:01 |
|   5 |     TABLE ACCESS FULL | Osoba   |     2 |    36 |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------
```

Funkce byla a vliv indexu jsem demonstroval na dotazu SELECT, který spojuje 2 tabulky a využívá agregační funkci jak je považováno v zadání. Díky indexu je možné rapidně snížit časovou složitost(například Binary tree)

Vytvořil: Michal Řezník
Login: xrezni28

PŘÍSTUPOVÁ PRÁVA

Samotná přístupová práva jsem vytvářel taktéž bez přítomnosti mého kolegy. Myslím ale, že z teoretického hlediska by mělo být vše v pořádku. Používal jsem příkaz GRANT. Práva jsem potřeboval k implementaci materializovaného pohledu.