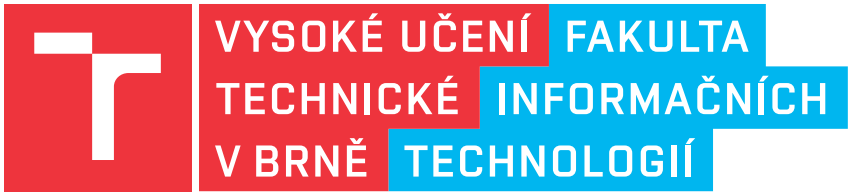


IDS - Projektová dokumentácia

29. apríla 2022

Knapovský Jan (xknapo05)
Hanus Igor (xhanus19)

Obsah

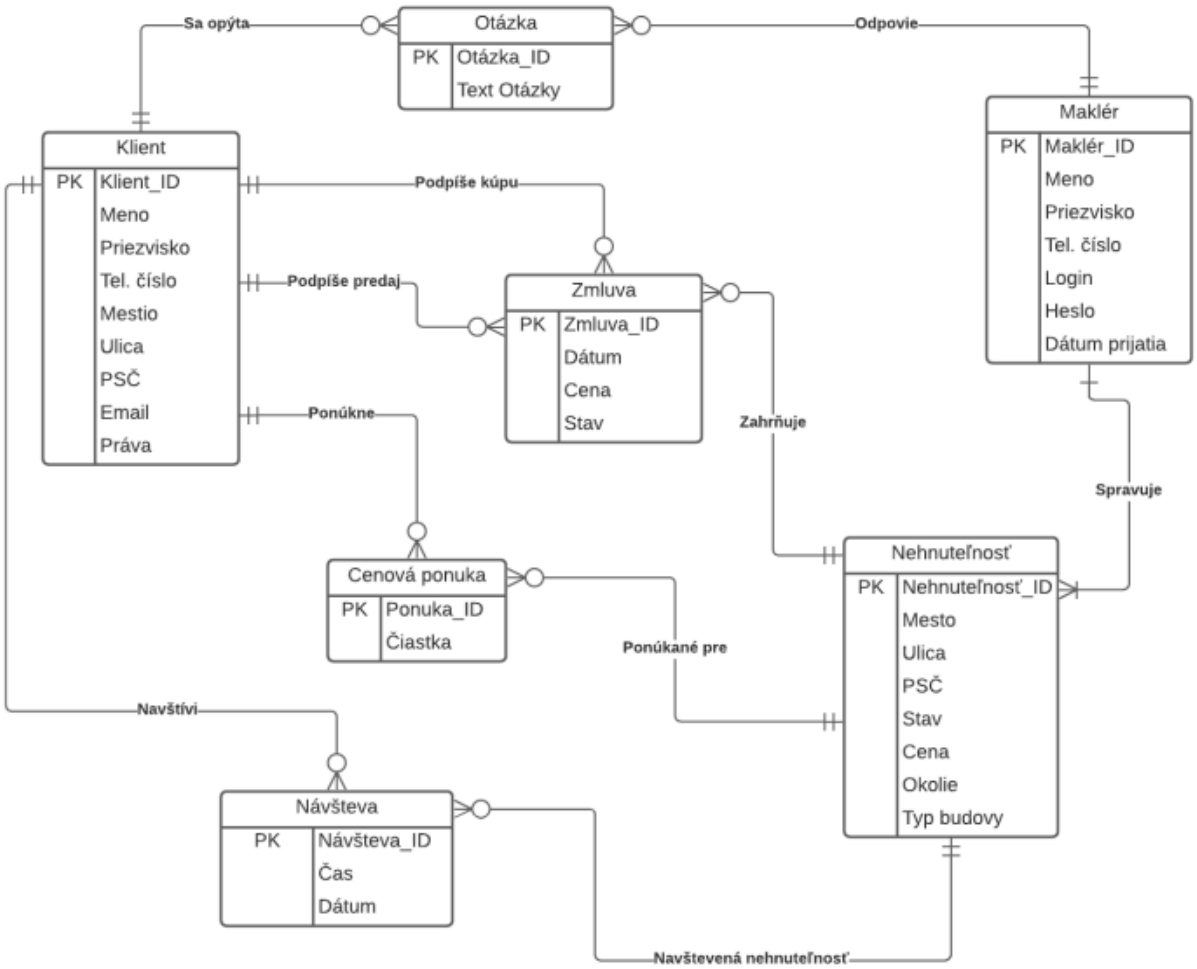
1 Úvod

Cieľom projektu je implementovať skript v jazyku SQL na základe vytvoreného ER diagramu. V našom prípade sa jedná o implementáciu informačného modelu pre realitnú kanceláriu. V pôvodnom diagrame sme museli vyriešiť niektoré nezrovnalosti pre správnu logickú štruktúru v skripte.

2 Prvá časť

V rámci prvého odovzdanie sme mali za úlohu vytvoriť ER diargram a diagram prípadov použitia.

2.1 ER Diagram



3 Druhá časť

V druhej časti sme následne implementovali jednotlivé tabuľky na základe nami vytvoreného ER diagramu. Pre vytváranie jedinečných identifikátorov pre primárne kľúče sme použili automatické generovanie na základe vygenerovanej sekvencie. Pre zobrazenie času sme použili typ `timestamp`. Následne jednotlivé tabuľky boli vytvárané cez funkciu `create table` v správnom poradí na základe jednotlivých závislostí objektov podľa spomenutého ER diagramu. Taktiež boli vytvorené funkcie pre potrebné zmazanie tabuľiek cez `drop table`.

```
create table visits
(
    visit_id         number generated as identity
        constraint VISITS_PK
            primary key,
    visit_date       timestamp default current_timestamp not null,
    visiting_user_id number
        constraint VISITS_USERS_USER_ID_FK
            references USERS,
    estate_id        number
        constraint VISITS_ESTATES_ESTATE_ID_FK
            references ESTATES
);
```

Po vytvorení jednotlivých tabuliek sme ich následne naplnili obraznými dátami pre kontrolu zobrazenia jednotlivých dát, ktoré sa v nich nachádzajú. Toto sme docielili funkciou `INSERT INTO [parametre] VALUES [hodnoty]`.

```
INSERT INTO ESTATES (ADDRESS, STATE, PRICE, BUILDING_TYPE, MANAGING_BROKER_ID)
VALUES ('Okurkova 22, Brno 61300', 'Old', 123.10, 'House', 3);
```

4 Tretia časť

Úlohou tretej časti bolo vytvorenie selectov pre jednotlivé dáta z databázy kde bolo potrebné použiť príkazy `GROUP BY`, `EXISTS`, `IN` a taktiež využitie príkazu `JOIN` pre prepojenie viacerých tabuliek do jednej.

Prvý select sa zameriava na použitie príkazu `JOIN` a `GROUP BY`. Úlohou prvého selectu je výpis jednotlivých maklérov a ich zoradenie podľa nehnuteľností, ktoré spravujú.

```
SELECT brokers.user_id, brokers.name AS name, brokers.surname AS surname, COUNT(*) AS estate_count
FROM brokers INNER JOIN estates ON brokers.user_id = estates.managing_broker_id
GROUP BY brokers.user_id, name, surname
ORDER BY estate_count DESC;
```

Druhý select sa zameriava na použitie príkazu `IN`. Daný select vráti všetkých maklérov, ktorý nespracovávajú žiadne nehnuteľnosti s pridelenou zmluvou.

```
SELECT * FROM brokers WHERE user_id NOT IN (
    SELECT brokers.user_id
    FROM brokers
             JOIN estates ON estates.managing_broker_id = brokers.user_id
             JOIN contracts ON contracts.estate_id = estates.estate_id
    GROUP BY brokers.user_id, brokers.surname
);
```

Tretí select sa zameriava na použitie príkazu `EXISTS`. Select vyberie maklérov, ktorí predávaju nehnuteľnosti za viac ako 200czk.

```
SELECT * FROM brokers WHERE EXISTS(
    SELECT * FROM estates WHERE estates.price > 200 AND estates.managing_broker_id = brokers.user_id
                              );
```

5 Štvrtá časť

5.1 Triggery

V skripte sú vytvorené dva triggery. Jedným z nich je `auto_contract_date`. Tento trigger sa aplikuje pre každý riadok v tabuľke `contracts`. Úloha tohto triggeru je automatické priradenie dátumu podľa času kedy bol daný záznam pridelený do databázy. K prideleniu aktuálneho času sa používa `SYSDATE`.

Ďalší trigger s názvom `auto_managing_broker`, ktorý v prípade pridania nového inzerátu na nehnuteľnosť vyhladá z pomedzi všetkých maklérov takého makléra, ktorý v čase pridania spravuje najmenej nehnuteľností. V prípade, že sa v tabuľke nenachádza žiadny maklér tak sa vyhodí `NO_DATA_FOUND` výnimka.

5.2 Procedúry

Procedúra `estate_prices` používa ako vstup premenné `max_price`, `low_price`, `broker_count`, `estate_count` a `broker_estate_ratio`. Tieto premenné sú následne naplnené potrebnými hodnotami, ktoré sa následne vypíšu na výstup. Taktiež sa vypíše priemerné množstvo nehnuteľností na jedného makléra. V prípade, že počeť nehnuteľností bude nulový(čiže počet nehnuteľností bude rovný nule) vyhodí sa výnimka `ZERO_DIVIDE`. Procedúra slúži na vypísanie základných štatistík pre nehnuteľnosti.

Procedúra `registered_users_questions` na výstup vypíše otázky vytvorené registrovanými uživateľmi. V procedúre sa využíva kurzor `cursor_users`, do ktorého sa načítajú jednotlivý uživatelia. Následne pomocou príkazu `LOOP` sa cez jednotlivé záznamy iteruje a v prípade, že sa jedna o registrovaného uživateľa tak sa ich otázka vypíše na výstup. Procedúra ma obmedzenie iba na uživateľa s jednou otázkov, inak sa spustí výnimka `TOO_MANY_ROWS`.

5.3 Prístupové práva

V tejto časti sa pridelia práva jednému členovi z tímu k všetkým tabuľkám a taktiež práva k spúšťaniu procedúry `estate_prices`.

5.4 Materializovaný pohľad

V rámci skriptu je vytvorený jeden materializovaný pohľad pre všetky nehnuteľnosti, ktoré boli zrekonštruované.