Universitatea Tehnică "Gheorghe Asachi" din Iași
Facultatea de Automatică și Calculatoare
Baze de Date - Temă

Rezervări hotele

Beldiman Vladislav
Grupa 1305A
Coordonator: Cătălin Mironeanu

January 10, 2021

Cuprins

1 Descrierea proiectului

Aplicația dată are scopul de a gestiona baza de date a rezervărilor unui lanț hotelier printr-o interfață grafică ușor de utilizat.

Aceasta asigură o interfață separată pentru clienți și administratorii lanțului.

Cea destinată clienților le oferă posibilitatea de a căuta oferte în perioada dorită, putând fi selectat și orașul, numărul de camere al apartamentului căutat, numărul de adulți și copii, precum și alte opțiuni. Iar, după selectarea unei oferte acesta va introduce datele personale pentru a confirma rezervarea, care implică adăugarea unei rezervări în baza de date, cât și a unui client, dacă nu există.

Cea pentru administrare permite celelalte opțiuni, mai avansate, de actualizare și ștergere în toate entitățile, și adăugarea unui oraș, hotel sau apartament.

În plus, aplicația șterge clientul din baza de date atunci când nu mai are nici o rezervare.

2 Tehnologii folosite

- Front-end

  Qt

- Back-end

  Python și Oracle Database

3 Structura și inter-relațioarea tabelelor

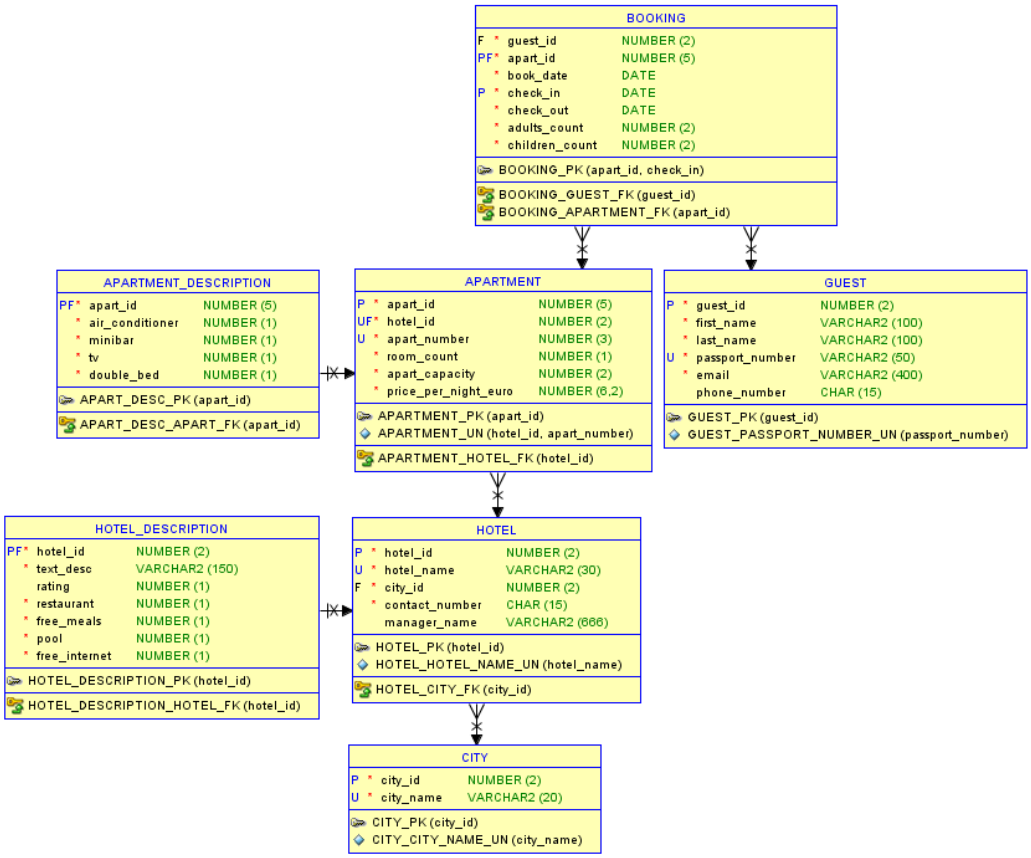

Figure 1: Diagrama ER.

Entitatea BOOKING și relațiile one-to-many afterente ei au fost obținute în urma normalizării relației many-to-many dintre entitățile GUEST și APARTMENT.

4 Constrângeri

Am folosit id-uri ca primary key pentru a facilita actualizarea entităților. Entitatea BOOKING nu are un id drept primary key, deoarece nu există foreign key-uri la niciuna din atributele sale.

Constrângerile de tip unique au fost folosite pentru a putea realiza toate operațiile fără a folosi explicit id-urile, care sunt redundante.

Foreign key-urile sunt folosite pentru a lega entitățile între ele.

Atributele air_conditioner, minibar, tv, double_bed, restaurant, free_meals, pool și free_internet pot avea doar valorile 0 sau 1 prin care e subînțeles False sau, respectiv, True, deoarece sunt folosite doar pentru a confirma existența acestor facilități.

Apart_capacity și adults_count pot lua valori în intervalul 1 - 20 inclusiv, iar children_count: 0 - 19, deoarece copii trebuie însoțiți. De asemenea, există constrângerea children_count + adults_count <= apart_capacity care e singura constrângere care nu e asigurată de baza de date.

Rating poate lua valori doar de la 1 la 5, deoarece e o modalitate de apreciere a hotelelor aproape universală, sau poate fi null în cazul în care nu obținut o apreciere oficială.

Atributele contact_number și phone_number trebuie să înceapă cu '+' și să conțină doar cifre după acesta pentru a asigura uniformitatea numerelor de telefon și a reduce confuzia.

Manager_name, first_name și last_name trebuie să conțină cel puțin un caracter și pot fi introduse doar litere din alfabetul latin sau spații. În plus, manager_name poate fi și null, în cazul în care hotelul nu are un manager la moment.

Passport_number poate conține doar cifre și litere majuscule din alfabetul latin, deoarece nu am găsit vreo regulă universală pentru numărul unui pașaport, iar majoritatea folosesc doar aceste caractere.

Atributul email trebuie să aibă structura generală ___@___.___.

Rezevările se efectuează pentru cel puțin o noapte, deci check_out - check_in > 0, iar rezervarea nu poate fi făcută începând cu o dată din trecut: check_in - book_date >= 0, unde book_date e data curentă obținută prin interogarea bazei de date, atunci când e confirmată rezervarea.

Disjuncția intervalelor [check_in. check_out) pentru același apartament e asigurată, de asemenea, de baza de date la momentul adăugării sau modificării rezervărilor.

5 Conectarea la baza de date

Conectarea și accesul la baza de date e realizată prin modulul Python cx_Oracle.
Conectare: self.db_connection = cx_Oracle.connect('tema', 'vlad', 'localhost/xe')

Interogările au forma:

```
with self.db_connection.cursor() as cursor:
    cursor.execute(interogare)
```

Închiderea conexiunii e realizată într-un block finally pentru a o asigura în cât mai multe cazuri.

6 Git repository

https://github.com/veeyslaw/hotelDB

7 Interfața aplicației și exemple din cod

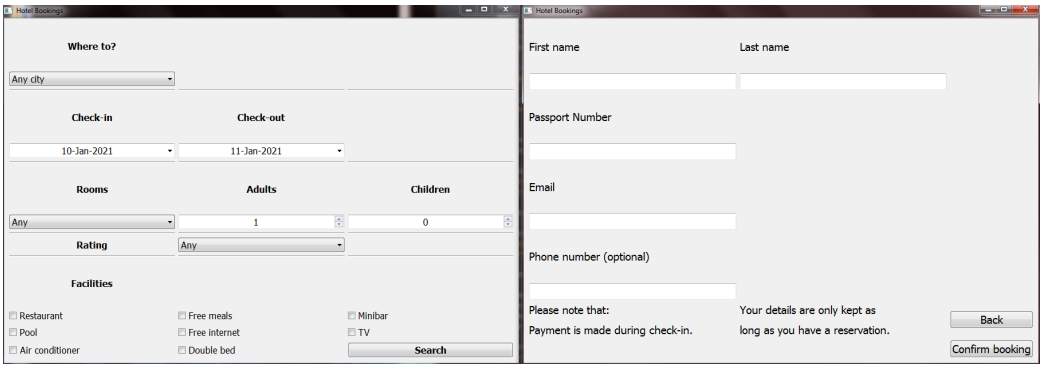

Figure 2: Interfață client.

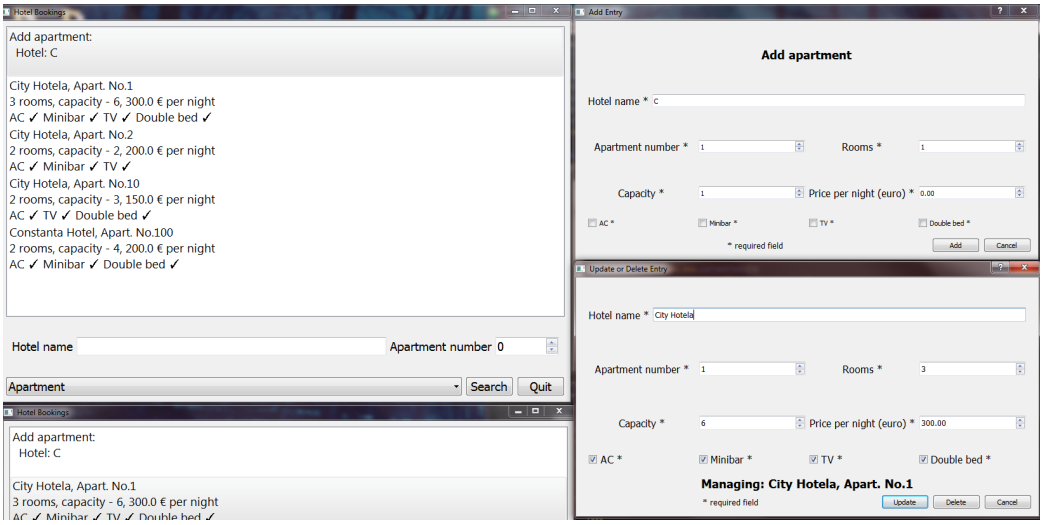

Figure 3: Interfață administrator.

```
query = f"""
        SELECT
            city_name,
            hotel_name,
            text_desc, rating, restaurant, free_meals, pool, free_internet,
            apart_number, room_count, price_per_night_euro,
            air_conditioner, minibar, tv, double_bed
        FROM
            CITY c,
            HOTEL h,
            HOTEL_DESCRIPTION hd,
            (SELECT *
             FROM APARTMENT inner_a
             WHERE (SELECT COUNT(*)
                    FROM BOOKING b
                    WHERE b.apart_id = inner_a.apart_id
                        AND b.check_out > TO_DATE('{self.check_in}', 'YYYY-MM-DD')
                        AND b.check_in < TO_DATE('{self.check_out}', 'YYYY-MM-DD')) = 0)
            a,
            APARTMENT_DESCRIPTION ad
        WHERE
            c.city_id = h.city_id
            AND h.hotel_id = hd.hotel_id
            AND a.hotel_id = h.hotel_id
            AND ad.apart_id = a.apart_id
            AND restaurant >= {restaurant}
            AND free_meals >= {free_meals}
            AND pool >= {pool}
            AND free_internet >= {free_internet}
            AND air_conditioner >= {ac}
            AND minibar >= {minibar}
            AND tv >= {tv}
            AND double_bed >= {double_bed}
            AND apart_capacity >= {self.adults_count + self.children_count}
            {f"AND city_name = '{city}'" if city is not None else ''}
            {f'AND rating >= {rating}' if rating is not None else ''}
            {f'AND room_count >= {room_count}' if room_count is not None else ''}
        """

with self.db_connection.cursor() as cursor:
    cursor.execute(query)
    offers = [row for row in cursor]
if len(offers) == 0:
    self.error('No available offers. Try a broader search.')
else:
    self.go_to_offers_page(offers)
```

Figure 4: Exemplu interogare: căutare.