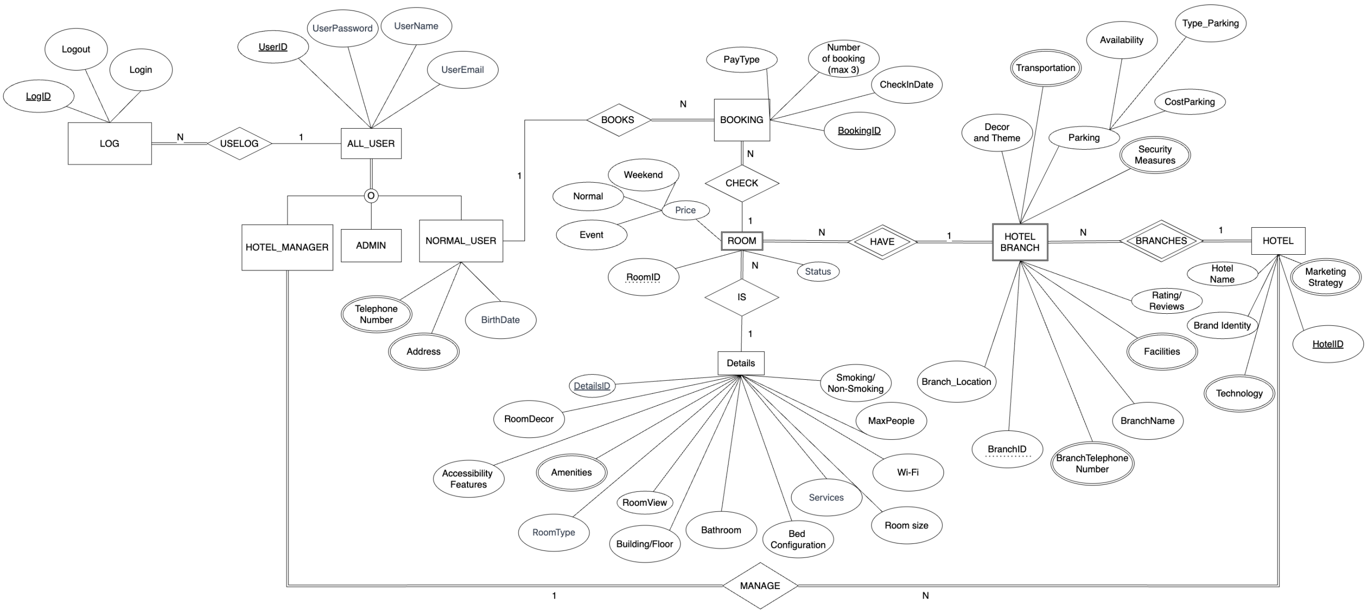
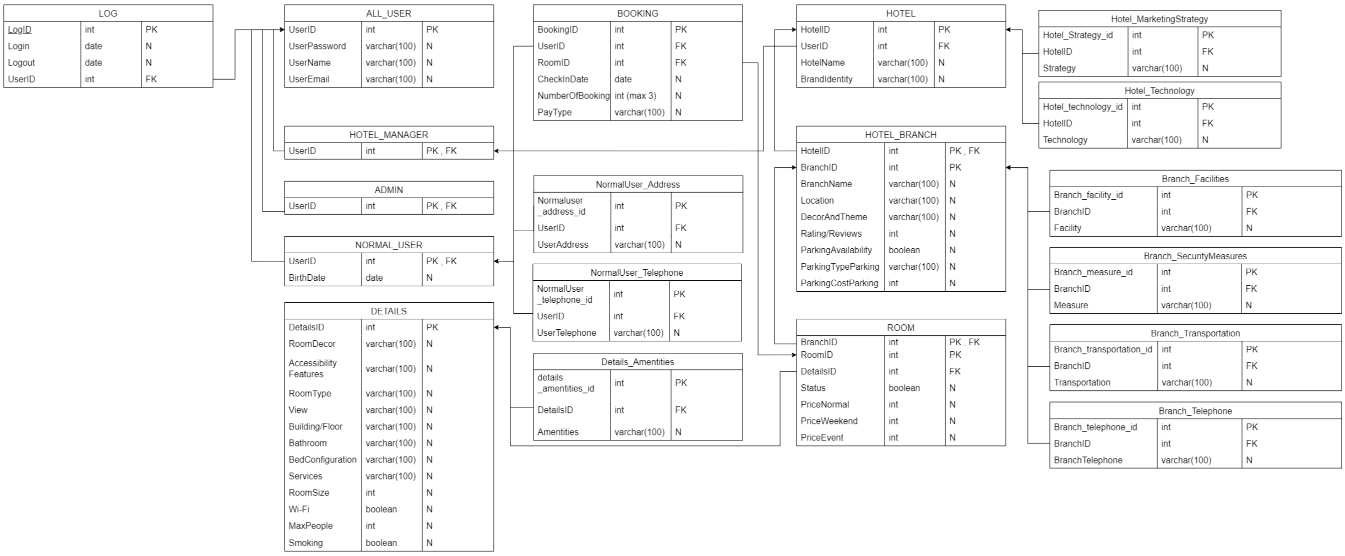
# Hotel Booking

## 1. ER Diagram (Chen's notation)



## 2. Schema diagram with referential integrity



| LOG | | |
|---|---|---|
| LogID | int | PK |
| Login | date | N |
| Logout | date | N |
| UserID | int | FK |

| ALL_USER | | |
|---|---|---|
| UserID | int | PK |
| UserPassword | varchar(100) | N |
| UserName | varchar(100) | N |
| UserEmail | varchar(100) | N |

| HOTEL_MANAGER | | |
|---|---|---|
| UserID | int | PK , FK |

| ADMIN | | |
|---|---|---|
| UserID | int | PK , FK |

| NORMAL_USER | | |
|---|---|---|
| UserID | int | PK , FK |
| BirthDate | date | N |

| DETAILS | | |
|---|---|---|
| DetailsID | int | PK |
| RoomDecor | varchar(100) | N |
| Accessibility Features | varchar(100) | N |
| RoomType | varchar(100) | N |
| View | varchar(100) | N |
| Building/Floor | varchar(100) | N |
| Bathroom | varchar(100) | N |
| BedConfiguration | varchar(100) | N |
| Services | varchar(100) | N |
| RoomSize | int | N |
| Wi-Fi | boolean | N |
| MaxPeople | int | N |
| Smoking | boolean | N |

| BOOKING | | |
|---|---|---|
| BookingID | int | PK |
| UserID | int | FK |
| RoomID | int | FK |
| CheckInDate | date | N |
| NumberOfBooking | int (max 3) | N |
| PayType | varchar(100) | N |

| NormalUser_Address | | |
|---|---|---|
| Normaluser _address_id | int | PK |
| UserID | int | FK |
| UserAddress | varchar(100) | N |

| NormalUser_Telephone | | |
|---|---|---|
| NormalUser _telephone_id | int | PK |
| UserID | int | FK |
| UserTelephone | varchar(100) | N |

| Details_Amentities | | |
|---|---|---|
| details _amentities_id | int | PK |
| DetailsID | int | FK |
| Amentities | varchar(100) | N |

| HOTEL | | |
|---|---|---|
| HotelID | int | PK |
| UserID | int | FK |
| HotelName | varchar(100) | N |
| BrandIdentity | varchar(100) | N |

| HOTEL_BRANCH | | |
|---|---|---|
| HotelID | int | PK , FK |
| BranchID | int | PK |
| BranchName | varchar(100) | N |
| Location | varchar(100) | N |
| DecorAndTheme | varchar(100) | N |
| Rating/Reviews | int | N |
| ParkingAvailability | boolean | N |
| ParkingTypeParking | varchar(100) | N |
| ParkingCostParking | int | N |

| ROOM | | |
|---|---|---|
| BranchID | int | PK , FK |
| RoomID | int | PK |
| DetailsID | int | FK |
| Status | boolean | N |
| PriceNormal | int | N |
| PriceWeekend | int | N |
| PriceEvent | int | N |

| Hotel_MarketingStrategy | | |
|---|---|---|
| Hotel_Strategy_id | int | PK |
| HotelID | int | FK |
| Strategy | varchar(100) | N |

| Hotel_Technology | | |
|---|---|---|
| Hotel_technology_id | int | PK |
| HotelID | int | FK |
| Technology | varchar(100) | N |

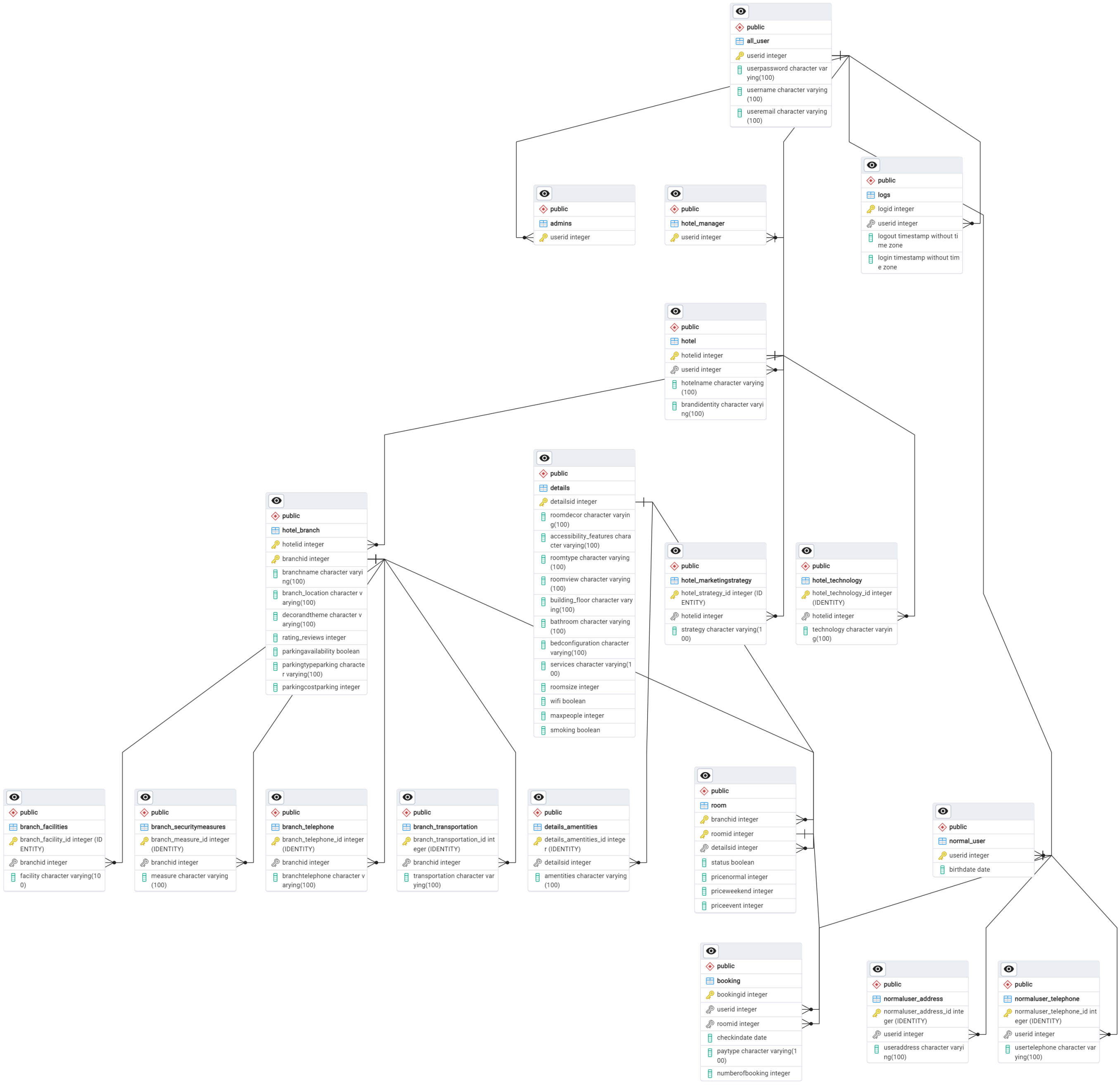
| Branch_Facilities | | |
|---|---|---|
| Branch_facility_id | int | PK |
| BranchID | int | FK |
| Facility | varchar(100) | N |

| Branch_SecurityMeasures | | |
|---|---|---|
| Branch_measure_id | int | PK |
| BranchID | int | FK |
| Measure | varchar(100) | N |

| Branch_Transportation | | |
|---|---|---|
| Branch_transportation_id | int | PK |
| BranchID | int | FK |
| Transportation | varchar(100) | N |

| Branch_Telephone | | |
|---|---|---|
| Branch_telephone_id | int | PK |
| BranchID | int | FK |
| BranchTelephone | varchar(100) | N |

//


public
all_user
userid integer
userpassword character varying(100)
username character varying (100)
useremail character varying (100)
public
admins
userid integer
public
hotel_manager
userid integer
public
logs
logid integer
userid integer
logout timestamp without time zone
login timestamp without time zone
public
hotel
hotelid integer
userid integer
hotelname character varying (100)
brandidentity character varying(100)
public
details
detailsid integer
roomdecor character varying(100)
accessibility_features character varying(100)
roomtype character varying (100)
roomview character varying (100)
building_floor character varying(100)
bathroom character varying (100)
bedconfiguration character varying(100)
services character varying(100)
roomsize integer
wifi boolean
maxpeople integer
smoking boolean
public
hotel_branch
hotelid integer
branchid integer
branchname character varying(100)
branch_location character varying(100)
decorandtheme character varying(100)
rating_reviews integer
parkingavailability boolean
parkingtypeparking character varying(100)
parkingcostparking integer
public
hotel_marketingstrategy
hotel_strategy_id integer (IDENTITY)
hotelid integer
strategy character varying(100)
public
hotel_technology
hotel_technology_id integer (IDENTITY)
hotelid integer
technology character varying(100)
public
branch_facilities
branch_facility_id integer (IDENTITY)
branchid integer
facility character varying(100)
public
branch_securitymeasures
branch_measure_id integer (IDENTITY)
branchid integer
measure character varying (100)
public
branch_telephone
branch_telephone_id integer (IDENTITY)
branchid integer
branchtelephone character varying(100)
public
branch_transportation
branch_transportation_id integer (IDENTITY)
branchid integer
transportation character varying(100)
public
details_amentities
details_amentities_id integer (IDENTITY)
detailsid integer
amentities character varying (100)
public
room
branchid integer
roomid integer
detailsid integer
status boolean
pricenormal integer
priceweekend integer
priceevent integer
public
normal_user
userid integer
birthdate date
public
booking
bookingid integer
userid integer
roomid integer
checkindate date
paytype character varying(100)
numberofbooking integer
public
normaluser_address
normaluser_address_id integer (IDENTITY)
userid integer
useraddress character varying(100)
public
normaluser_telephone
normaluser_telephone_id integer (IDENTITY)
userid integer
usertelephone character varying(100)

# Projet Funtion Requirement

1. The system shall allow a user to register by specifying the name, telephone number, email, and password.
   - cmd1_register_all_user
2. After registration, the user becomes a registered user, and the system shall allow the user to log in to use the system by specifying the email and password. The system shall allow a registered user to log out.
   - cmd2_login_user
3. After login, the system shall allow the registered user to book up to 3 nights by specifying the date and the preferred hotel. The hotel list is also provided to the user. A hotel information includes the hotel name, address, and telephone number.
   - cmd3_1_insert_booking_with_user_and_room
   - cmd3_2_user_view_all_room
4. The system shall allow the registered user to view his hotel bookings.

- cmd4_user_view_bookings

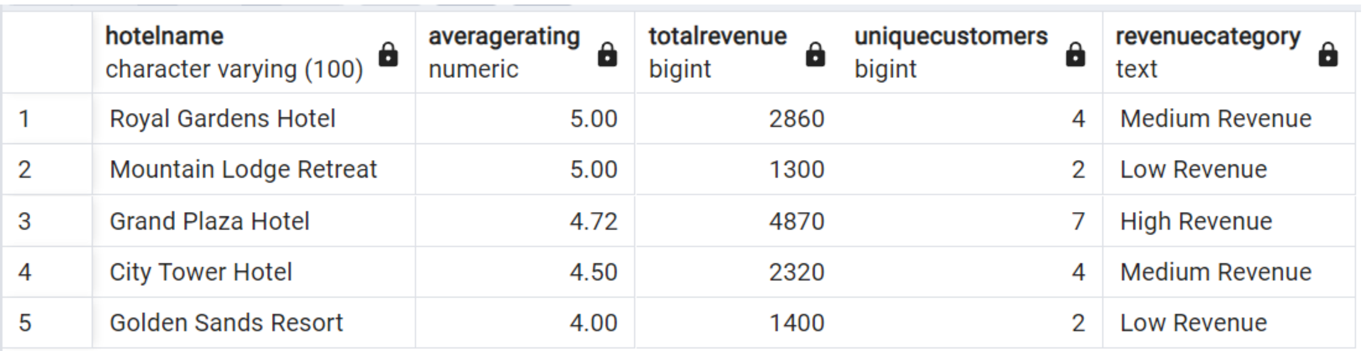
5. The system shall allow the registered user to edit his hotel bookings.
    - cmd5_user_edit_booking
6. The system shall allow the registered user to delete his hotel bookings.
    - cmd6_user_delete_booking
7. The system shall allow the admin to view any hotel bookings.
    - cmd7_admin_view_bookings
8. The system shall allow the admin to edit any hotel bookings.
    - cmd8_admin_edit_booking
9. The system shall allow the admin to delete any hotel bookings
    - cmd9_admin_delete_booking

## SQL Complex

ต้องการหา 5 อันดับโรงแรมที่มีเรตติ้งอย่างน้อย 4 โดยเรียงจากเรตติ้งจากสูงไปต่ำ หากเท่ากันจะจากรายได้สูงไปต่ำ และต้องการทราบจำนวนผู้ใช้บริการของโรงแรมนั้น

```sql
-- top 5 hotels with an average rating of at least 4,
-- categorizing them by revenue level based on booking amounts and counting the number of unique customers per hotel
SELECT
    "hotel"."hotelname",
    ROUND(AVG("hotel_branch"."rating_reviews"), 2) AS "averagerating",
    SUM("room"."pricenormal" * "booking"."numberofbooking") AS "totalrevenue",
    COUNT(DISTINCT "booking"."userid") AS "uniquecustomers",
    CASE
        WHEN SUM("room"."pricenormal" * "booking"."numberofbooking") > 4000 THEN 'High Revenue'
        WHEN SUM("room"."pricenormal" * "booking"."numberofbooking") > 1500 THEN 'Medium Revenue'
        ELSE 'Low Revenue'
    END AS "revenuecategory"
FROM
    "hotel"
JOIN
    "hotel_branch" ON "hotel"."hotelid" = "hotel_branch"."hotelid"
JOIN
    "room" ON "hotel_branch"."branchid" = "room"."branchid"
JOIN
    "booking" ON "room"."roomid" = "booking"."roomid"
GROUP BY
    "hotel"."hotelname"
HAVING
    AVG("hotel_branch"."rating_reviews") >= 4
ORDER BY
    "averagerating" DESC, "totalrevenue" DESC
LIMIT 5;
```

ตัวอย่าง Ouput



| | hotelname<br>character varying (100) | averagerating<br>numeric | totalrevenue<br>bigint | uniquecustomers<br>bigint | revenuecategory<br>text |
|---|---|---|---|---|---|
| 1 | Royal Gardens Hotel | 5.00 | 2860 | 4 | Medium Revenue |
| 2 | Mountain Lodge Retreat | 5.00 | 1300 | 2 | Low Revenue |
| 3 | Grand Plaza Hotel | 4.72 | 4870 | 7 | High Revenue |
| 4 | City Tower Hotel | 4.50 | 2320 | 4 | Medium Revenue |
| 5 | Golden Sands Resort | 4.00 | 1400 | 2 | Low Revenue |

## Document-based design schema

```
{
    "title": "BOOKING",
    "required": [
        "BookingID",
        "UserID",
        "RoomID",
        "CheckInDate",
        "PayType",
        "NumberOfBooking"
    ],
    "properties": {
        "BookingID": {
            "bsonType": "int",
            "description": "The unique identifier for each booking",
            "unique": true
        },
        "UserID": {
            "bsonType": "int",
            "description": "The unique identifier for each user"
        },
        "RoomID": {
            "bsonType": "int",
            "description": "The unique identifier for each room"
        },
        "CheckInDate": {
            "bsonType": "date",
            "description": "The date when the booking is made"
        },
        "PayType": {
            "bsonType": "string",
            "description": "The payment type for the booking"
        },
        "NumberOfBooking": {
            "bsonType": "int",
            "description": "The number night of bookings made by the user",
            "maximum": 3
        }
    }
}
```