

# **CSI 2132 – Databases I**

**Winter 2024**

**University of Ottawa**

## **Course Project - Deliverable 2**

Course Instructor: Verena Kantere



Carl Cheng (300309128)  
Bradley Gao (300282778)  
Penghao Yuan (300283368)

Submission Date: 31 March

## **DBMS and the Programming Languages Used:**

DBMS Used: PostgreSQL

Programming Languages:

- Java
- HTML
- Apache Tomcat

## **Steps to Install the Application:**

The following steps require the user to install the application on a windows computer. Java must be already installed beforehand.

1. Download project from BrightSpace
2. Load Project Folder “CSI2132Project” with IntelliJ IDE
3. Create a new Apache Tomcat run configuration to change Catalina base to correct directory
4. Open ConnectionDB.java and change dbServerPort, dbName, dbusername and dbpassword to match personal database
5. Copy Query information from txt file or report to create and populate database tables

### **GitHub link:**

<https://github.com/Cchegas/CSI-2132-DATABASE-PROJECT->

### **Video link:**

[ink](https://drive.google.com/file/d/1MJmHjPROkPLRUQhDBplB0Bdd5sboHzcZ/view?usp=drive_l)

*Table 1: Contents of the Video*

Requirement	Start Timestamp
1 - technologies	00:00
2 - schema	00:14
3 - integrity constraints	00:31
4 - data population	01:26
5 - queries	02:08
6 - triggers	03:04
7 - indexes	04:17
8 - views	04:59
9 - UI	05:34

## DDLs Used to Create the Database:

### DDLs for creating all the tables:

```
CREATE TABLE employees (  
    employeeID SERIAL PRIMARY KEY,  
    firstName VARCHAR(50) NOT NULL,  
    lastName VARCHAR(50) NOT NULL,  
    address VARCHAR(255),  
    sinNumber VARCHAR(15) UNIQUE,  
    role VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE customer (  
    customerID SERIAL PRIMARY KEY,  
    firstname VARCHAR(30),  
    lastname VARCHAR(30),  
    dateofregistration DATE NOT NULL,  
    IDtype VARCHAR(30),  
    IDnumber VARCHAR(30)  
);
```

```
CREATE TABLE hotelChain (  
    chainID SERIAL PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,
```

```
    centralOfficeAddress VARCHAR(255) ,  
    numberOfHotels INT ,  
    contactEmail VARCHAR(255) ,  
    phoneNumber VARCHAR(20)  
);
```

```
CREATE TABLE hotels (  
    hotelID SERIAL PRIMARY KEY,  
    chainID INT NOT NULL,  
    name VARCHAR(100) NOT NULL,  
    category INT NOT NULL,  
    numberOfRooms INT NOT NULL,  
    address VARCHAR(255) ,  
    area VARCHAR(20) ,  
    contactEmail VARCHAR(255) ,  
    phoneNumber VARCHAR(20),  
    FOREIGN KEY (chainID) REFERENCES  
    hotelChain(chainID)  
);
```

```
CREATE TABLE rooms (  
    roomID SERIAL PRIMARY KEY,  
    hotelID INT NOT NULL,
```

```

roomNumber VARCHAR(20) NOT NULL,
price DECIMAL(10, 2) NOT NULL,
capacity INT NOT NULL,
views VARCHAR(20) ,
problems BOOLEAN ,
petFriendly BOOLEAN ,
FOREIGN KEY (hotelID) REFERENCES hotels(hotelID)
);

```

```

CREATE TABLE booking (
    bookingID SERIAL PRIMARY KEY,
    roomID INT NOT NULL,
    customerID INT NOT NULL,
    startDate DATE NOT NULL,
    endDate DATE NOT NULL,
    depositAmount DECIMAL(10, 2)
);

```

```

CREATE TABLE renting (
    rentingID SERIAL PRIMARY KEY,
    roomID INT NOT NULL,
    customerID INT NOT NULL,
    startDate DATE NOT NULL,
    endDate DATE NOT NULL,
    paymentMethod VARCHAR(30),
    paymentAmount DECIMAL(10, 2)
);

```

### DDLs for inserting hotel chains:

```

INSERT INTO hotelchain (name, centralofficeaddress,
    numberofhotels, contactemail, phonenumber) VALUES
('Beachy Hotels', 'Beachy 101 street', 8,
    'beach@hotels.com', '617-432-5235'),
('Sky Hotels', 'Sky 301 street', 8, 'skys@hotels.com',
    '617-124-5235'),
('Train Hotels', 'Train 101 street', 8, 'train@hotels.com',
    '533-432-2354'),
('Craine Hotels', 'Craine 11251 street', 8,
    'Craine@hotels.com', '613-432-1245'),
('Penacony Hotels', 'Penacony 101 street', 8,
    'Penacony@hotels.com', '913-932-5235');

```

### DDLs for inserting hotels:

```

INSERT INTO hotels (chainid, name, category,
    numberofrooms, address, area, contactemail, phonenumber)
VALUES
(1, 'Beach Hotel 1', 5, 25, '123 Beach Avenue', 'Ottawa',
    'beachhotel1@gmail.com', '713-565-1111'),
(1, 'Beach Hotel 2', 4, 30, '456 Seaside Blvd', 'Ottawa',
    'beachhotel2@gmail.com', '653-558-1112'),
--Two hotels in the same area
(1, 'Beach Hotel 3', 3, 40, '789 Shoreline Street', 'Guelph',
    'beachhotel3@gmail.com', '414-595-2221'),
(1, 'Beach Hotel 4', 4, 35, '1011 Coastal Road', 'Guelph',
    'beachhotel4@gmail.com', '613-9555-2222'),

```

-- Continuing with unique addresses but ensuring variety in ratings

```

(1, 'Beach Hotel 5', 5, 45, '1213 Sandy Lane', 'Halifax',
    'beachhotel5@gmail.com', '613-575-3341'),

```

```

(1, 'Beach Hotel 6', 4, 50, '1415 Palm Street', 'Vancouver',
    'beachhotel6@gmail.com', '613-559-4241'),

```

```

(1, 'Beach Hotel 7', 3, 20, '1617 Wave Crest Avenue',
    'Calgary', 'beachhotel7@gmail.com', '613-545-5551'),

```

```

(1, 'Beach Hotel 8', 5, 30, '1819 Ocean Drive', 'Montreal',
    'beachhotel8@gmail.com', '613-255-3661'),

```

-- hotelchain 2

```

(2, 'Sky Hotel 1', 5, 25, '123 Cloud Street', 'Ottawa',
    'Skyhotel1@gmail.com', '123-565-1111'),

```

```

(2, 'Sky Hotel 2', 4, 30, '456 Skyscraper Avenue', 'Ottawa',
    'Skyhotel2@gmail.com', '123-558-1112'),

```

--Two hotels in the same area

```

(2, 'Sky Hotel 3', 3, 40, '789 Heavenly Court', 'Guelph',
    'Skyhotel3@gmail.com', '123-595-2221'),

```

```

(2, 'Sky Hotel 4', 4, 35, '1011 Celestial Boulevard', 'Guelph',
    'Skyhotel4@gmail.com', '123-9555-2222'),

```

-- Continuing with unique addresses but ensuring variety in ratings

```

(2, 'Sky Hotel 5', 5, 45, '1213 Starlight Avenue', 'Halifax',
    'Skyhotel5@gmail.com', '123-575-3341'),

```

```

(2, 'Sky Hotel 6', 4, 50, '1415 Moonbeam Drive', 'Vancouver',
    'Skyhotel6@gmail.com', '123-559-4241'),

```

```

(2, 'Sky Hotel 7', 3, 20, '1617 Aurora Street', 'Calgary',
    'Skyhotel7@gmail.com', '123-545-5551'),

```

```

(2, 'Sky Hotel 8', 5, 30, '1819 Galaxy Lane', 'Montreal',
    'Skyhotel8@gmail.com', '123-255-3661'),

```

--hotel chain 3

```

(3, 'Train Hotel 1', 5, 25, '123 Railway Road', 'Ottawa',
    'trainhotel1@gmail.com', '345-876-1111'),

```

```

(3, 'Train Hotel 2', 4, 30, '456 Station Street', 'Ottawa',
    'trainhotel2@gmail.com', '345-876-1112'),

```

--Two hotels in the same area

```

(3, 'Train Hotel 3', 3, 40, '789 Express Lane', 'Guelph',
    'trainhotel3@gmail.com', '345-876-2221'),

```

```

(3, 'Train Hotel 4', 4, 35, '1011 Locomotive Avenue', 'Guelph',
    'trainhotel4@gmail.com', '345-876-2222'),

```

-- Continuing with unique addresses but ensuring variety in ratings

```

(3, 'Train Hotel 5', 5, 45, '1213 Freight Street', 'Halifax',
    'train5@gmail.com', '345-876-3341'),

```

```

(3, 'Train Hotel 6', 4, 50, '1415 Railroad Drive', 'Vancouver',
    'trainhotel6@gmail.com', '345-876-4241'),

```

```

(3, 'Train Hotel 7', 3, 20, '1617 Crossing Avenue', 'Calgary',
    'trainhotel7@gmail.com', '345-876-5551'),

```

```

(3, 'Train Hotel 8', 5, 30, '1819 Depot Street', 'Montreal',
    'trainhotel8@gmail.com', '345-876-3661'),

```

-- hotel chain 4

```

(4, 'Craine Hotel 1', 5, 25, '123 Crane Avenue', 'Ottawa',
    'Crainehotel1@gmail.com', '543-876-1111'),

```

```

(4, 'Craine Hotel 2', 4, 30, '456 Lift Lane', 'Ottawa',
    'Crainehotel2@gmail.com', '345-876-1112'),

```

--Two hotels in the same area

```

(4, 'Craine Hotel 3', 3, 40, '789 Elevator Street', 'Guelph',
    'Crainenhotel3@gmail.com', '543-876-2221'),

```

```

(4, 'Craine Hotel 4', 4, 35, '1011 Hoist Avenue', 'Guelph',
    'Crainehotel4@gmail.com', '543-876-2222'),

```

-- Continuing with unique addresses but ensuring variety in ratings

(4, 'Craine Hotel 5', 5, 45, '1213 Pulley Road', 'Halifax', 'Craine15@gmail.com', '543-876-3341'),  
(4, 'Craine Hotel 6', 4, 50, '1415 Winch Boulevard', 'Vancouver', 'Crainehotel6@gmail.com', '543-876-4241'),  
(4, 'Craine Hotel 7', 3, 20, '1617 Gear Street', 'Calgary', 'Crainehotel7@gmail.com', '543-876-5551'),  
(4, 'Craine Hotel 8', 5, 30, '1819 Pulley Lane', 'Montreal', 'Crainehotel8@gmail.com', '543-876-3661'),  
-- hotel chain 5

(5, 'Penacony Hotel 1', 5, 25, '123 Ocean View Drive', 'Ottawa', 'Penaconyhotel1@gmail.com', '765-876-1111'),  
(5, 'Penacony Hotel 2', 4, 30, '456 Bayshore Avenue', 'Ottawa', 'Penaconyhotel2@gmail.com', '765-876-1112'),  
--Two hotels in the same area

(5, 'Penacony Hotel 3', 3, 40, '789 Coastal Street', 'Guelph', 'Penaconynhotel3@gmail.com', '765-876-2221'),  
(5, 'Penacony Hotel 4', 4, 35, '1011 Seaview Road', 'Guelph', 'Penaconyhotel4@gmail.com', '765-876-2222'),

-- Continuing with unique addresses but ensuring variety in ratings

(5, 'Penacony Hotel 5', 5, 45, '1213 Beachside Drive', 'Halifax', 'Penaconyl5@gmail.com', '765-876-3341'),  
(5, 'Penacony Hotel 6', 4, 50, '1415 Cliffside Avenue', 'Vancouver', 'Penaconyhotel6@gmail.com', '765-876-4241'),  
(5, 'Penacony Hotel 7', 3, 20, '1617 Cove Court', 'Calgary', 'Penaconyhotel7@gmail.com', '765-876-5551'),  
(5, 'Penacony Hotel 8', 5, 30, '1819 Oceanfront Lane', 'Montreal', 'Penaconyhotel8@gmail.com', '765-876-3661');

## DDLs for inserting rooms:

INSERT INTO rooms (hotelID, roomNumber, price, capacity, views, problems, petFriendly) VALUES

(1, 'A3245', 200.0, 2, 'seaView', FALSE, TRUE),  
(1, 'B4672', 100.0, 3, 'mountainView', TRUE, TRUE),  
(1, 'C1234', 200.0, 4, 'mountainView', FALSE, FALSE),  
(1, 'D9876', 50.0, 5, 'seaView', FALSE, TRUE),  
(1, 'E5643', 75.0, 6, 'mountainView', FALSE, TRUE),

(2, 'F2341', 200.0, 2, 'seaView', FALSE, TRUE),  
(2, 'G8765', 100.0, 3, 'mountainView', FALSE, TRUE),  
(2, 'H9321', 200.0, 4, 'mountainView', FALSE, TRUE),  
(2, 'I5678', 50.0, 5, 'seaView', FALSE, TRUE),  
(2, 'J4321', 75.0, 6, 'mountainView', FALSE, TRUE),

(3, 'K4321', 200.0, 2, 'seaView', FALSE, TRUE),  
(3, 'L9876', 100.0, 3, 'mountainView', FALSE, TRUE),  
(3, 'M6789', 200.0, 4, 'mountainView', FALSE, TRUE),  
(3, 'N3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(3, 'O8765', 75.0, 6, 'mountainView', FALSE, TRUE),

(4, 'P8765', 200.0, 2, 'seaView', FALSE, TRUE),  
(4, 'Q0987', 100.0, 3, 'mountainView', FALSE, TRUE),  
(4, 'R4567', 200.0, 4, 'mountainView', FALSE, TRUE),  
(4, 'S7654', 50.0, 5, 'seaView', FALSE, TRUE),  
(4, 'T5432', 75.0, 6, 'mountainView', FALSE, TRUE),

(5, 'U5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(5, 'V5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(5, 'W2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(5, 'X3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(5, 'Y1098', 75.0, 6, 'mountainView', FALSE, TRUE),

(6, 'Z5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(6, 'A5432', 100.0, 3, 'mountainView', TRUE, TRUE),  
(6, 'B2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(6, 'C3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(6, 'D1098', 75.0, 6, 'mountainView', FALSE, TRUE),

(7, 'E5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(7, 'F5432', 100.0, 3, 'mountainView', TRUE, TRUE),  
(7, 'G2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(7, 'H3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(7, 'I1098', 75.0, 6, 'mountainView', FALSE, TRUE),

(8, 'J5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(8, 'K5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(8, 'L2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(8, 'M3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(8, 'N1098', 75.0, 6, 'mountainView', FALSE, TRUE),

(9, 'T5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(9, 'U5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(9, 'V2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(9, 'W3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(9, 'X1098', 75.0, 6, 'mountainView', FALSE, TRUE),

(10, 'Y5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(10, 'Z5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(10, 'A2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(10, 'B3456', 50.0, 5, 'seaView', TRUE, TRUE),  
(10, 'C1098', 75.0, 6, 'mountainView', FALSE, TRUE),

(11, 'D5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(11, 'E5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(11, 'F2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(11, 'G3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(11, 'H1098', 75.0, 6, 'mountainView', FALSE, TRUE),

(12, 'I5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(12, 'J5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(12, 'K2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(12, 'L3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(12, 'M1098', 75.0, 6, 'mountainView', FALSE, TRUE),

(13, 'K6678', 200.0, 2, 'seaView', FALSE, TRUE),  
(13, 'O6432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(13, 'K2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(13, 'L3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(13, 'M1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 14

(14, 'N5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(14, 'O5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(14, 'P2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(14, 'Q3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(14, 'R1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 15

(15, 'S5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(15, 'T5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(15, 'U2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(15, 'V3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(15, 'W1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 16

(16, 'X5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(16, 'Y5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(16, 'Z2109', 200.0, 4, 'mountainView', TRUE, TRUE),  
(16, 'A3456', 50.0, 5, 'seaView', TRUE, TRUE),  
(16, 'B1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 17

(17, 'C5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(17, 'D5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(17, 'E2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(17, 'F3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(17, 'G1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 18

(18, 'H5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(18, 'I5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(18, 'J2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(18, 'K3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(18, 'L1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 19

(19, 'M5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(19, 'N5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(19, 'O2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(19, 'P3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(19, 'Q1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 20

(20, 'R5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(20, 'S5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(20, 'T2109', 200.0, 4, 'mountainView', TRUE, TRUE),  
(20, 'U3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(20, 'V1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 21

(21, 'W5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(21, 'X5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(21, 'Y2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(21, 'Z3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(21, 'A1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 22

(22, 'B5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(22, 'C5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(22, 'D2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(22, 'E3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(22, 'F1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 23

(23, 'G5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(23, 'H5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(23, 'I2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(23, 'J3456', 50.0, 5, 'seaView', FALSE, TRUE),

(23, 'K1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 24

(24, 'L5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(24, 'M5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(24, 'N2109', 200.0, 4, 'mountainView', TRUE, TRUE),  
(24, 'O3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(24, 'P1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 25

(25, 'Q5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(25, 'R5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(25, 'S2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(25, 'T3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(25, 'U1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 26

(26, 'V5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(26, 'W5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(26, 'X2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(26, 'Y3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(26, 'Z1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 27

(27, 'A5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(27, 'B5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(27, 'C2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(27, 'D3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(27, 'E1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 28

(28, 'F5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(28, 'G5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(28, 'H2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(28, 'I3456', 50.0, 5, 'seaView', TRUE, TRUE),  
(28, 'J1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 29

(29, 'K5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(29, 'L5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(29, 'M2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(29, 'N3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(29, 'O1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 30

(30, 'P5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(30, 'Q5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(30, 'R2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(30, 'S3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(30, 'T1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 31

(31, 'U5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(31, 'V5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(31, 'W2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(31, 'X3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(31, 'Y1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 32

(32, 'Z5678', 200.0, 2, 'seaView', FALSE, TRUE),

(32, 'A5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(32, 'B2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(32, 'C3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(32, 'D1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 33

(33, 'E5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(33, 'F5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(33, 'G2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(33, 'H3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(33, 'I1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 34

(34, 'J5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(34, 'K5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(34, 'L2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(34, 'M3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(34, 'N1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 35

(35, 'O5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(35, 'P5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(35, 'Q2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(35, 'R3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(35, 'S1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 36

(36, 'T5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(36, 'U5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(36, 'V2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(36, 'W3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(36, 'X1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 37

(37, 'Y5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(37, 'Z5432', 100.0, 3, 'mountainView', TRUE, TRUE),  
(37, 'A2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(37, 'B3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(37, 'C1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 38

(38, 'D5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(38, 'E5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(38, 'F2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(38, 'G3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(38, 'H1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 39

(39, 'I5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(39, 'J5432', 100.0, 3, 'mountainView', FALSE, TRUE),  
(39, 'K2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(39, 'L3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(39, 'M1098', 75.0, 6, 'mountainView', FALSE, TRUE),

--hotel 40

(40, 'N5678', 200.0, 2, 'seaView', FALSE, TRUE),  
(40, 'O5432', 100.0, 3, 'mountainView', TRUE, TRUE),  
(40, 'P2109', 200.0, 4, 'mountainView', FALSE, TRUE),  
(40, 'Q3456', 50.0, 5, 'seaView', FALSE, TRUE),  
(40, 'R1098', 75.0, 6, 'mountainView', FALSE, TRUE);

### DDLs for inserting employees:

INSERT INTO employees (firstName, lastName, address, sinNumber, role) VALUES  
( 'John', 'Doe', '123 Main St', '123456789', 'Manager'),  
( 'Jane', 'Smith', '456 Oak St', '987654321', 'Receptionist'),  
( 'Michael', 'Johnson', '789 Elm St', '456789123', 'Housekeeper'),  
( 'Emily', 'Brown', '101 Maple St', '654321987', 'Maintenance'),  
( 'David', 'Wilson', '234 Pine St', '321987654', 'Chef'),  
( 'Sarah', 'Anderson', '567 Cedar St', '789123456', 'Waiter'),  
( 'Daniel', 'Taylor', '890 Birch St', '159263487', 'Security'),  
( 'Jessica', 'Martinez', '111 Spruce St', '357159486', 'Concierge'),  
( 'Ryan', 'Garcia', '222 Fir St', '258147369', 'Bellboy'),  
( 'Laura', 'Lopez', '333 Walnut St', '369258147', 'Valet');

### DDLs for inserting Customers:

INSERT INTO customer (firstname, lastname, dateofregistration, IDtype, IDnumber) VALUES  
( 'Alice', 'Johnson', '2024-01-01', 'Passport', 'AB123456'),  
( 'Bob', 'Smith', '2024-01-05', 'Driver License', 'CD789012'),  
( 'Eva', 'Williams', '2024-01-10', 'Passport', 'EF345678'),  
( 'Jack', 'Brown', '2024-01-15', 'Passport', 'GH901234'),  
( 'Olivia', 'Taylor', '2024-02-01', 'Driver License', 'IJ567890'),  
( 'Sophia', 'Miller', '2024-02-05', 'Passport', 'KL123456'),  
( 'Liam', 'Davis', '2024-02-10', 'Passport', 'MN789012'),  
( 'Emma', 'Wilson', '2024-02-15', 'Driver License', 'OP345678'),  
( 'Noah', 'Moore', '2024-03-01', 'Passport', 'QR901234'),  
( 'Ava', 'Garcia', '2024-03-05', 'Passport', 'ST567890');

## 2c) Database Queries

**Query 1:** Retrieves the hotel chains along with the number of hotels they have and sort them by the number of hotels in descending order:

```
SELECT hc.name AS chain_name, hc.numberOfHotels
FROM hotelChain hc
JOIN (
    SELECT chainID, COUNT(hotelID) AS numberOfHotels
    FROM hotels
    GROUP BY chainID
) AS h ON hc.chainID = h.chainID
ORDER BY numberOfHotels DESC;
```

**Query 2:** retrieves the average price for each hotel category (rating):

```
SELECT h.category, AVG(r.price) AS average_price_per_room
FROM hotels h
JOIN rooms r ON h.hotelID = r.hotelID
GROUP BY h.category;
```

**Query 3, nested:** Retrieve the names of hotels that have at least one room with a capacity greater than 4.

```
SELECT DISTINCT h.name AS hotel_name
FROM hotels h
JOIN rooms r ON h.hotelID = r.hotelID
WHERE r.capacity > 4;
```

**Query 4, aggregation:** Retrieve the total number of rooms and the average capacity of rooms for each hotel chain.

```
SELECT hc.name AS hotel_chain_name, COUNT(r.roomID) AS total_rooms, AVG(r.capacity)
AS avg_room_capacity
FROM hotelChain hc
JOIN hotels h ON hc.chainID = h.chainID
JOIN rooms r ON h.hotelID = r.hotelID
GROUP BY hc.name;
```

## 2d) Database modifications

**Trigger 1, overlapping booking prevention.** The trigger ensures that a new booking / renting does not overlap with existing bookings / rentings in the database.



```

CREATE OR REPLACE FUNCTION prevent_overlapping_bookings()
RETURNS TRIGGER AS $$
BEGIN
    -- Check for any overlapping bookings
    IF EXISTS (
        SELECT 1 FROM booking
        WHERE roomID = NEW.roomID
        AND NOT(NEW.endDate <= startDate OR NEW.startDate >= endDate)
    ) THEN
        RAISE EXCEPTION 'This room is already booked for the given date range.';
    END IF;

    -- Check for any overlapping rentings
    IF EXISTS (
        SELECT 1 FROM renting
        WHERE roomID = NEW.roomID
        AND NOT(NEW.endDate <= startDate OR NEW.startDate >= endDate)
    ) THEN
        RAISE EXCEPTION 'This room is already rented for the given date range.';
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER trigger_prevent_overlapping_bookings
BEFORE INSERT ON booking
FOR EACH ROW EXECUTE FUNCTION prevent_overlapping_bookings();

```

**Trigger 2: problematic room renting prevention.** The trigger ensures that a room with a problem (problem = 'true') cannot be rented for good measure.

```

CREATE OR REPLACE FUNCTION prevent_renting_problematic_rooms()
RETURNS TRIGGER AS $$
BEGIN
    -- Check if the room to be rented has been marked with problems
    IF EXISTS (
        SELECT 1 FROM rooms
        WHERE roomID = NEW.roomID AND problems = TRUE
    ) THEN
        RAISE EXCEPTION 'Cannot rent a room with reported problems.';
    END IF;
    RETURN NEW;

```

```
END;  
$$ LANGUAGE plpgsql;  
  
CREATE TRIGGER trigger_prevent_renting_problematic_rooms  
BEFORE INSERT ON renting  
FOR EACH ROW EXECUTE FUNCTION prevent_renting_problematic_rooms();
```

## 2e) Database Indexes

**Index 1:** Frequent queries on booking dates to check room availability or list active bookings within a certain period.

```
CREATE INDEX idx_booking_dates ON booking(startDate, endDate);
```

**Index 2:** For queries filtering rooms by hotel and whether they have problems (maintenance issues), especially useful for maintenance schedules and room availability checks.

```
CREATE INDEX idx_rooms_hotelID_problems ON rooms(hotelID, problems);
```

**Index 3:** Queries on renting dates to find active rentings or check room availability will benefit from this index.

```
CREATE INDEX idx_renting_dates ON renting(startDate, endDate);
```

## 2f) Database Views

**View 1: number of available rooms in an area:**

**Implemented this:**

```
CREATE VIEW available_rooms_per_area AS  
SELECT h.address AS area, COUNT(r.roomID) AS available_rooms  
FROM hotels h  
LEFT JOIN rooms r ON h.hotelID = r.hotelID  
GROUP BY h.address;
```

**To see the view of all hotels:** SELECT \* FROM available\_rooms\_per\_area;

**Of a specific hotel:** SELECT \* FROM available\_rooms\_per\_area

WHERE area = '1213 Sandy Lane';

**View 2: Aggregated capacity of all the rooms of a specific hotel:**

**Implemented this:**

```
CREATE VIEW hotel_room_capacity AS
SELECT h.hotelID, h.name AS hotel_name, SUM(r.capacity) AS total_capacity
FROM hotels h
JOIN rooms r ON h.hotelID = r.hotelID
GROUP BY h.hotelID, h.name;
```

**To see the aggregated capacity of a specific hotel:** SELECT \* FROM hotel\_room\_capacity  
WHERE hotelID = 1;

**Of all hotels:** SELECT \* FROM hotel\_room\_capacity;