CSI2132 Deliverable 2 eHotel Web-App/Database

Group 72
Andrew Horgan 300199610
Alperen Akin 300290090
Dipak Kumar 300
Professor: Verena Kantera

DBMS/Programming Languages Used

- DBMS: postgreSQL pgadmin 4
- Web App: backend - Java, frontend - Java and HTML
- Server/Servlet: Apache TomCat (local)

Guide/Installation Steps

Database Set-up

1. Open pgAdmin 4 and create a new Database
2. Go to the query tool and copy and the schema.sql folder found in the "HotelDatabase72" project file for the new database created
3. Go to the query tool and copy the queries.sql folder found in the "HotelDatabase72" project file. Be sure to avoid the select statements at the bottom of the file and to also copy the triggers.
4. Your Database is populated enough to run the web app.
5. NOTE: To create employees and customers, you can do so quickly in the web app, by going to employee portal and clicking "edit customers" or "edit employees"

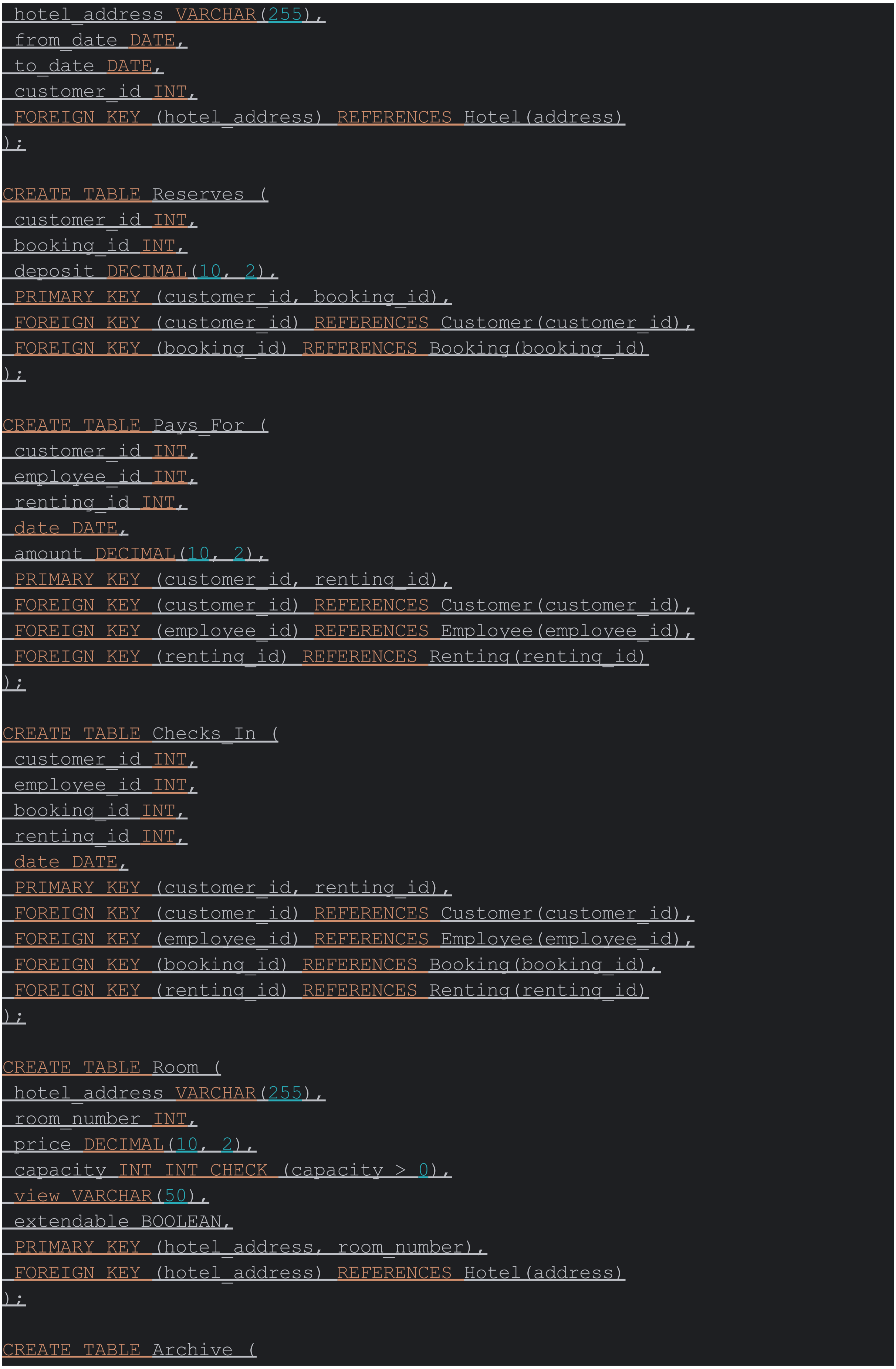
Web-App Set-up

1. Open the "HotelDatabase72" project folder in IntelliJ IDEA IDE
2. Go to the "ConectionDB.java" class in the src folder and change all ConnectionDB attributes: ipAddress, ServerPort, databasename, username and password to values corresponding to your database in pg admin 4.
3. Press "run-tomcat" on the top right and click the localhost link that appears in the console to run the web app.

SQL Schema/Querries (Also attached in submission folder, queries.sql and schema.sql)

File: schema.sql

```
CREATE TABLE Customer (
 customer_id INT PRIMARY KEY,
 id VARCHAR(50),
 first_name VARCHAR(50),
 last_name VARCHAR(50),
 address VARCHAR(255),
 date_registered DATE
);

CREATE TABLE Employee (
   employee_id INT PRIMARY KEY,
   ssn VARCHAR(255),
   first_name VARCHAR(255),
   last_name VARCHAR(255),
   address VARCHAR(255)
);

-- Hotel_Chain Table
CREATE TABLE Hotel_Chain (
   name VARCHAR(255) PRIMARY KEY,
   office_address VARCHAR(255)
);

-- Hotel Table
CREATE TABLE Hotel (
   address VARCHAR(255) PRIMARY KEY,
   stars INT CHECK (star_rating BETWEEN 1 AND 5),
   contact_email VARCHAR(255),
   num_rooms INT,
   hotel_chain_name VARCHAR(255),
   FOREIGN KEY (hotel_chain_name) REFERENCES Hotel_Chain(name)
);

CREATE TABLE Manages (
   hotel_address VARCHAR(255),
   employee_id INT,
   FOREIGN KEY (hotel_address) REFERENCES Hotel(address),
   FOREIGN KEY (employee_id) REFERENCES Employee(employee_id),
   PRIMARY KEY (hotel_address, employee_id)
);

CREATE TABLE Works_For (
   hotel_address VARCHAR(255),
   employee_id INT,
   role VARCHAR(255),
   FOREIGN KEY (hotel_address) REFERENCES Hotel(address),
   FOREIGN KEY (employee_id) REFERENCES Employee(employee_id),
   PRIMARY KEY (hotel_address, employee_id)
);

CREATE TABLE Booking (
 booking_id INT PRIMARY KEY,
 room_number INT,
 hotel_address VARCHAR(255),
 from_date DATE,
 to_date DATE,
 customer_id INT,
 FOREIGN KEY (hotel_address) REFERENCES Hotel(address)
);

CREATE TABLE Renting (
 renting_id INT PRIMARY KEY,
 room_number INT,
```

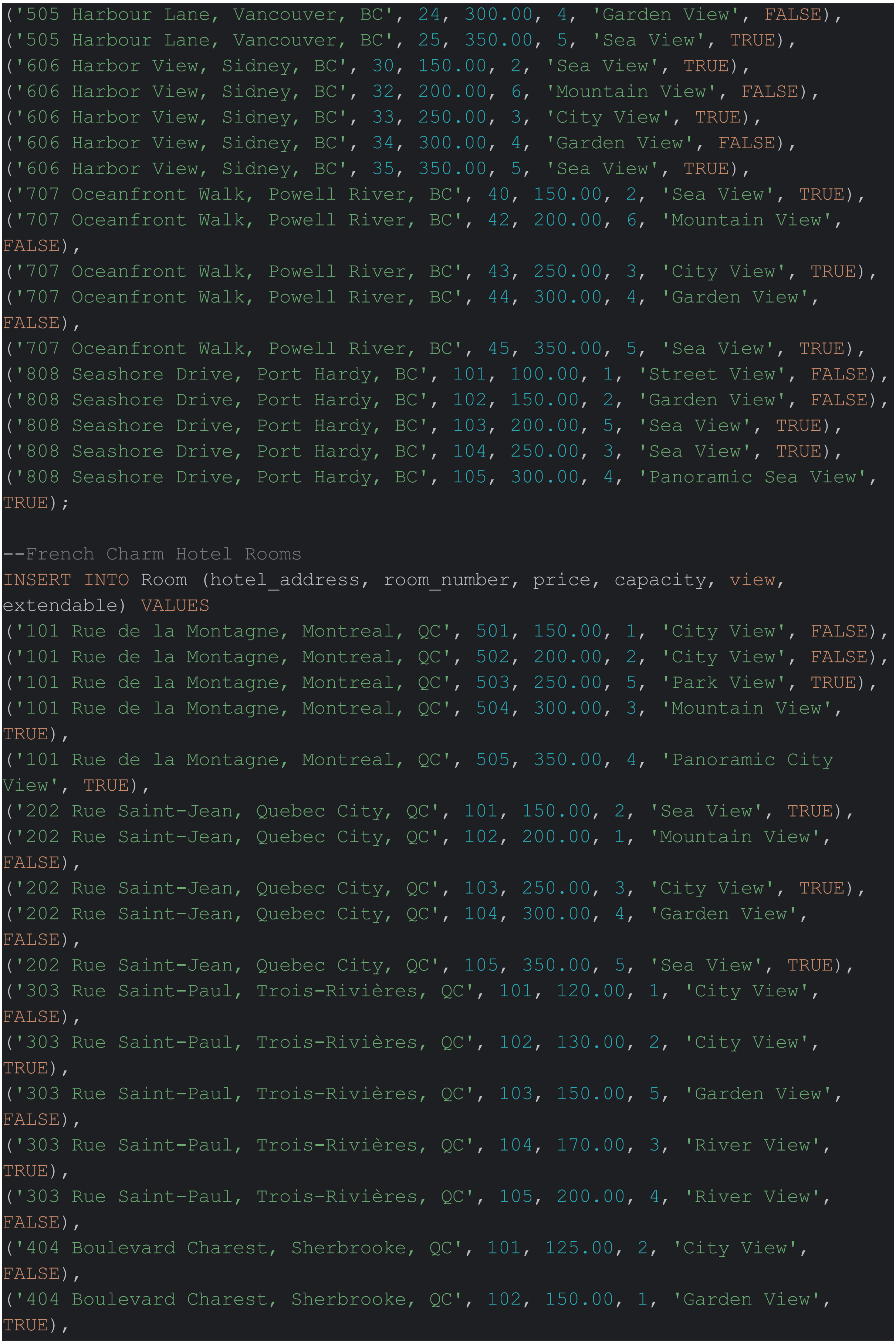


```
 hotel_address VARCHAR(255),
 from_date DATE,
 to_date DATE,
 customer_id INT,
 FOREIGN KEY (hotel_address) REFERENCES Hotel(address)
);

CREATE TABLE Reserves (
 customer_id INT,
 booking_id INT,
 deposit DECIMAL(10, 2),
 PRIMARY KEY (customer_id, booking_id),
 FOREIGN KEY (customer_id) REFERENCES Customer(customer_id),
 FOREIGN KEY (booking_id) REFERENCES Booking(booking_id)
);

CREATE TABLE Pays_For (
 customer_id INT,
 employee_id INT,
 renting_id INT,
 date DATE,
 amount DECIMAL(10, 2),
 PRIMARY KEY (customer_id, renting_id),
 FOREIGN KEY (customer_id) REFERENCES Customer(customer_id),
 FOREIGN KEY (employee_id) REFERENCES Employee(employee_id),
 FOREIGN KEY (renting_id) REFERENCES Renting(renting_id)
);

CREATE TABLE Checks_In (
 customer_id INT,
 employee_id INT,
 booking_id INT,
 renting_id INT,
 date DATE,
 PRIMARY KEY (customer_id, renting_id),
 FOREIGN KEY (customer_id) REFERENCES Customer(customer_id),
 FOREIGN KEY (employee_id) REFERENCES Employee(employee_id),
 FOREIGN KEY (booking_id) REFERENCES Booking(booking_id),
 FOREIGN KEY (renting_id) REFERENCES Renting(renting_id)
);

CREATE TABLE Room (
 hotel_address VARCHAR(255),
 room_number INT,
 price DECIMAL(10, 2),
 capacity INT INT CHECK (capacity > 0),
 view VARCHAR(50),
 extendable BOOLEAN,
 PRIMARY KEY (hotel_address, room_number),
 FOREIGN KEY (hotel_address) REFERENCES Hotel(address)
);

CREATE TABLE Archive (
```

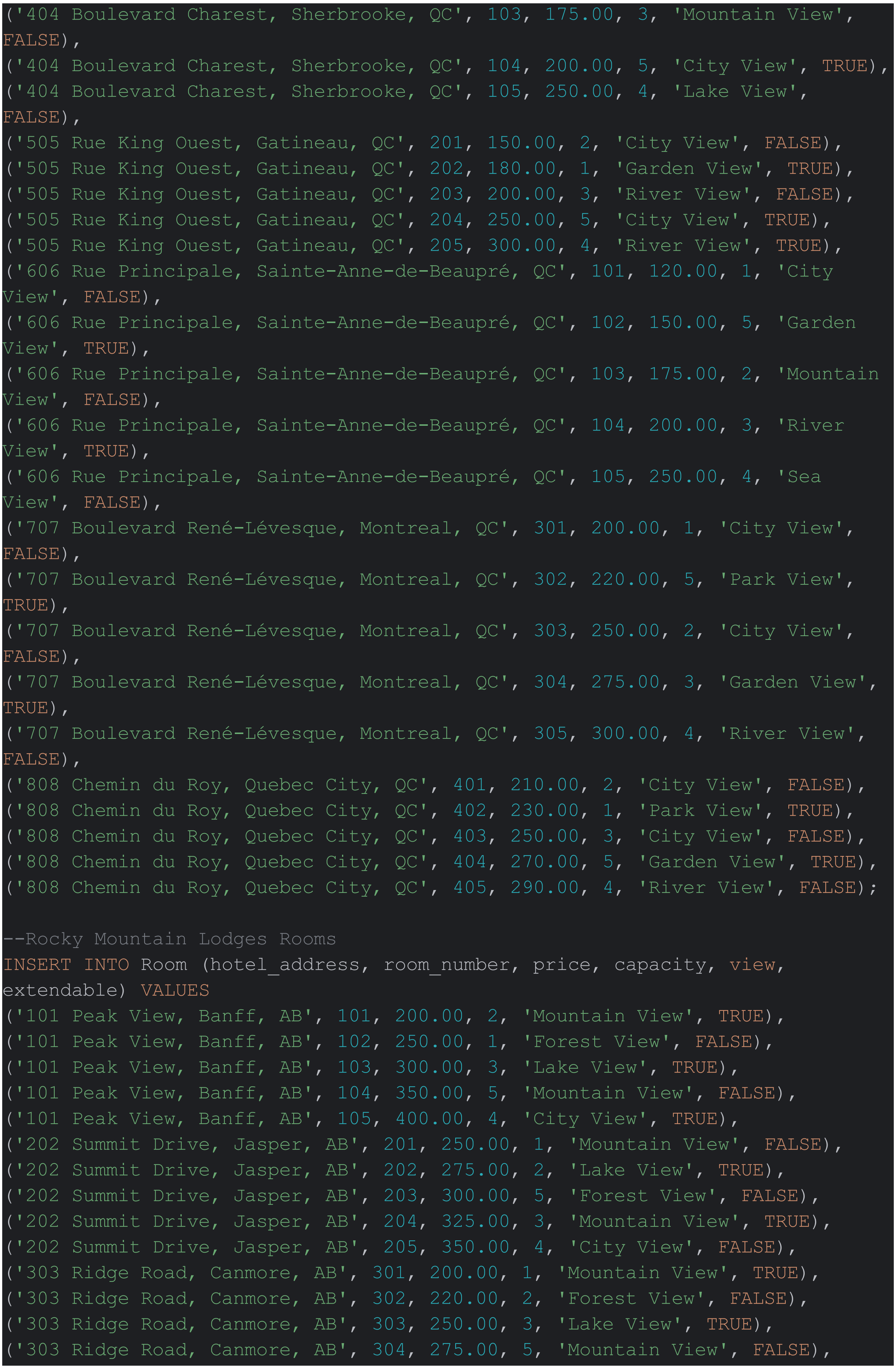
```sql
 address VARCHAR(255),
 record_date DATE,
 booking_id INT,
 renting_id INT,
 PRIMARY KEY (booking_id, renting_id),
 FOREIGN KEY (booking_id) REFERENCES Booking(booking_id),
 FOREIGN KEY (renting_id) REFERENCES Renting(renting_id)
);
```

File: queries.sql

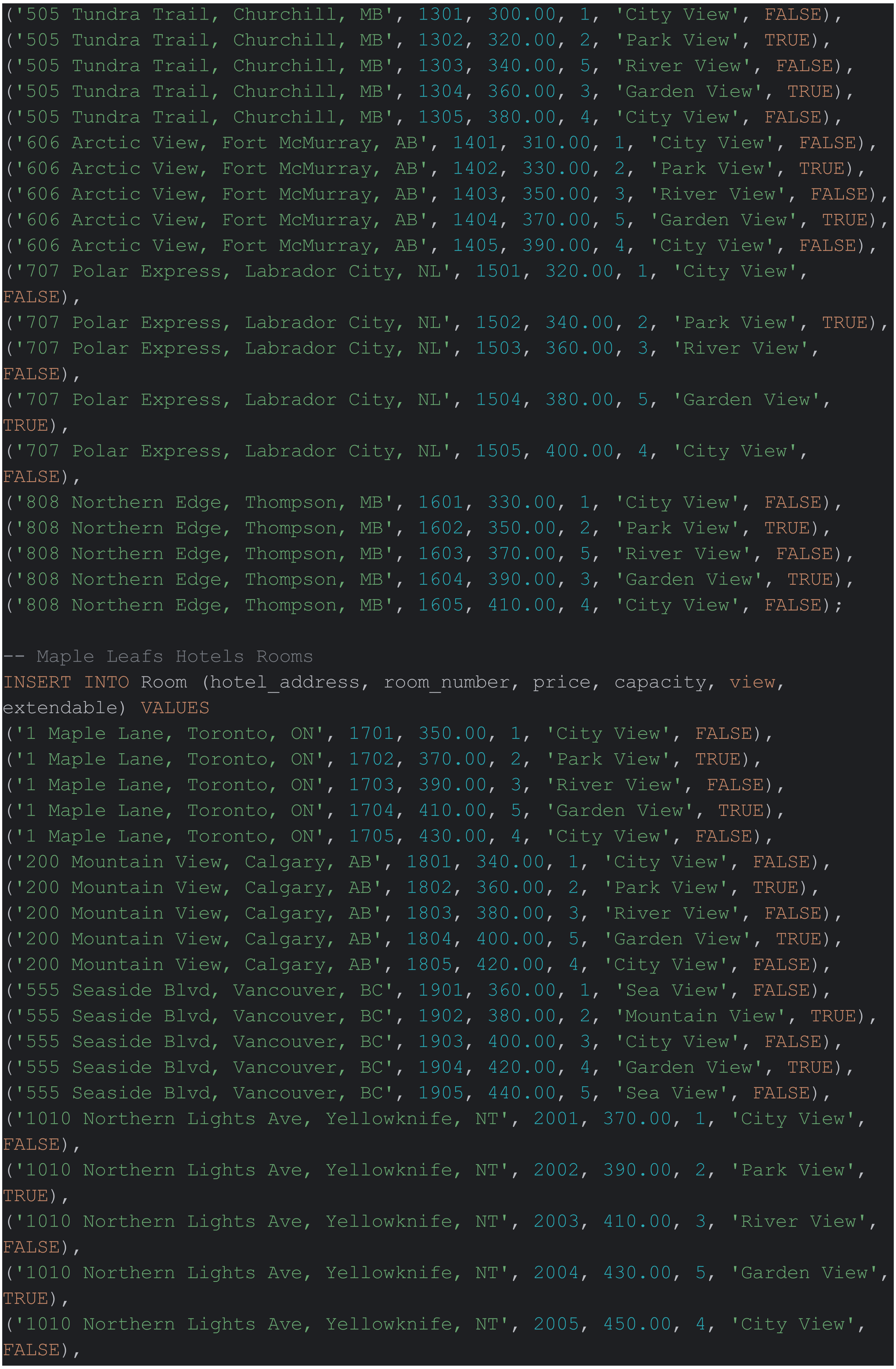
```sql
--The 5 hotel chains
INSERT INTO Hotel_Chain (name, office_address) VALUES
('Maple Leaf Hotels', '123 Maple St, Toronto, ON'),
('Northern Lights Resorts', '100 Aurora Ave, Yellowknife, NT'),
('Rocky Mountain Lodges', '200 Mountain Rd, Banff, AB'),
('Pacific Coast Inns', '400 Ocean Blvd, Vancouver, BC'),
('French Charm Hotels', '500 Rue Saint, Quebec City, QC');

--8 hotels from Maple Leafs Hotels
INSERT INTO Hotel (address, stars, contact_email, num_rooms,
hotel_chain_name) VALUES
('1 Maple Lane, Toronto, ON', 4, 'info@mapleleafhotels.com', 120, 'Maple Leaf
Hotels'),
('200 Mountain View, Calgary, AB', 5, 'contact@mapleleafhotels.com', 150,
'Maple Leaf Hotels'),
('555 Seaside Blvd, Vancouver, BC', 3, 'booking@mapleleafhotels.com', 100,
'Maple Leaf Hotels'),
('1010 Northern Lights Ave, Yellowknife, NT', 4,
'reservations@mapleleafhotels.com', 80, 'Maple Leaf Hotels'),
('768 East Coast Rd, Halifax, NS', 3, 'service@mapleleafhotels.com', 90,
'Maple Leaf Hotels'),
('909 Maple Street, Ottawa, ON', 4, 'ottawa@mapleleafhotels.com', 110, 'Maple
Leaf Hotels'),
('1010 Maple Avenue, Winnipeg, MB', 3, 'winnipeg@mapleleafhotels.com', 70,
'Maple Leaf Hotels'),
('1111 Maple Road, Regina, SK', 5, 'regina@mapleleafhotels.com', 130, 'Maple
Leaf Hotels');


--8 hotels from Northern Lights Resorts
INSERT INTO Hotel (address, stars, contact_email, num_rooms,
hotel_chain_name) VALUES
('101 Aurora Street, Whitehorse, YT', 4, 'info@northernlightsresorts.com',
100, 'Northern Lights Resorts'),
('202 Polar Road, Yellowknife, NT', 5, 'contact@northernlightsresorts.com',
120, 'Northern Lights Resorts'),
('303 Iceberg Lane, Iqaluit, NU', 3, 'booking@northernlightsresorts.com', 80,
'Northern Lights Resorts'),
('404 Northern Ave, Edmonton, AB', 4,
'reservations@northernlightsresorts.com', 90, 'Northern Lights Resorts'),
('505 Tundra Trail, Churchill, MB', 3, 'service@northernlightsresorts.com',
70, 'Northern Lights Resorts'),
('606 Arctic View, Fort McMurray, AB', 4,
'arcticview@northernlightsresorts.com', 90, 'Northern Lights Resorts'),
('707 Polar Express, Labrador City, NL', 5,
'polarexpress@northernlightsresorts.com', 100, 'Northern Lights Resorts'),
('808 Northern Edge, Thompson, MB', 3,
'northernedge@northernlightsresorts.com', 80, 'Northern Lights Resorts');



--8 hotels from Rocky Mountain Lodges
INSERT INTO Hotel (address, stars, contact_email, num_rooms,
hotel_chain_name) VALUES
('101 Peak View, Banff, AB', 5, 'info@rockymountainlodges.com', 150, 'Rocky
Mountain Lodges'),
('202 Summit Drive, Jasper, AB', 4, 'contact@rockymountainlodges.com', 100,
'Rocky Mountain Lodges'),
('303 Ridge Road, Canmore, AB', 4, 'booking@rockymountainlodges.com', 120,
'Rocky Mountain Lodges'),
('404 Mountain Trail, Revelstoke, BC', 3,
'reservations@rockymountainlodges.com', 80, 'Rocky Mountain Lodges'),
('505 Forest Parkway, Whistler, BC', 5, 'service@rockymountainlodges.com',
130, 'Rocky Mountain Lodges');
('909 Valley Road, Kananaskis, AB', 4, 'valley@rockymountainlodges.com', 85,
'Rocky Mountain Lodges'),
('1010 Highland Ave, Fernie, BC', 5, 'highland@rockymountainlodges.com', 95,
'Rocky Mountain Lodges'),
('1111 Mountain View, Kimberley, BC', 3,
'mountainview@rockymountainlodges.com', 75, 'Rocky Mountain Lodges');

--8 hotels from Pacific Coast Inns
INSERT INTO Hotel (address, stars, contact_email, num_rooms,
hotel_chain_name) VALUES
('101 Ocean Drive, Victoria, BC', 4, 'info@pacificcoastinns.com', 100,
'Pacific Coast Inns'),
('202 Seaside Blvd, Tofino, BC', 5, 'contact@pacificcoastinns.com', 120,
'Pacific Coast Inns'),
('303 Coastal Road, Prince Rupert, BC', 3, 'booking@pacificcoastinns.com',
80, 'Pacific Coast Inns'),
('404 Beachfront Ave, Nanaimo, BC', 4, 'reservations@pacificcoastinns.com',
90, 'Pacific Coast Inns'),
('505 Harbour Lane, Vancouver, BC', 3, 'service@pacificcoastinns.com', 70,
'Pacific Coast Inns'),
('606 Harbor View, Sidney, BC', 4, 'harborview@pacificcoastinns.com', 95,
'Pacific Coast Inns'),
('707 Oceanfront Walk, Powell River, BC', 3,
'oceanfront@pacificcoastinns.com', 85, 'Pacific Coast Inns'),
('808 Seashore Drive, Port Hardy, BC', 4, 'seashore@pacificcoastinns.com',
100, 'Pacific Coast Inns');



--8 hotels from French Charm Hotel
INSERT INTO Hotel (address, stars, contact_email, num_rooms,
hotel_chain_name) VALUES
('101 Rue de la Montagne, Montreal, QC', 4, 'info@frenchcharmhotels.com',
120, 'French Charm Hotels'),
('202 Rue Saint-Jean, Quebec City, QC', 5, 'contact@frenchcharmhotels.com',
150, 'French Charm Hotels'),
('303 Rue Saint-Paul, Trois-Rivières, QC', 3,
'booking@frenchcharmhotels.com', 90, 'French Charm Hotels'),
('404 Boulevard Charest, Sherbrooke, QC', 4,
'reservations@frenchcharmhotels.com', 100, 'French Charm Hotels'),
('505 Rue King Ouest, Gatineau, QC', 3, 'service@frenchcharmhotels.com', 80,
'French Charm Hotels'),
('606 Rue Principale, Sainte-Anne-de-Beaupré, QC', 4,
'stay@frenchcharmhotels.com', 85, 'French Charm Hotels'),
('707 Boulevard René-Lévesque, Montreal, QC', 5,
'luxury@frenchcharmhotels.com', 150, 'French Charm Hotels'),
('808 Chemin du Roy, Quebec City, QC', 3, 'heritage@frenchcharmhotels.com',
75, 'French Charm Hotels');

--ROOMS (5 for each hotel, soo 200 rooms total)
--Pacific Coast Inns Rooms
INSERT INTO Room (hotel_address, room_number, price, capacity, view,
extendable) VALUES
('101 Ocean Drive, Victoria, BC', 101, 150.00, 2, 'Sea View', TRUE),
('101 Ocean Drive, Victoria, BC', 102, 200.00, 6, 'Mountain View', FALSE),
('101 Ocean Drive, Victoria, BC', 103, 250.00, 3, 'City View', TRUE),
('101 Ocean Drive, Victoria, BC', 104, 300.00, 4, 'Garden View', FALSE),
('101 Ocean Drive, Victoria, BC', 105, 350.00, 5, 'Sea View', TRUE),
('202 Seaside Blvd, Tofino, BC', 1, 150.00, 2, 'Sea View', TRUE),
('202 Seaside Blvd, Tofino, BC', 2, 200.00, 6, 'Mountain View', FALSE),
('202 Seaside Blvd, Tofino, BC', 3, 250.00, 3, 'City View', TRUE),
('202 Seaside Blvd, Tofino, BC', 4, 300.00, 4, 'Garden View', FALSE),
('202 Seaside Blvd, Tofino, BC', 5, 350.00, 5, 'Sea View', TRUE),
('303 Coastal Road, Prince Rupert, BC', 10, 150.00, 2, 'Sea View', TRUE),
('303 Coastal Road, Prince Rupert, BC', 12, 200.00, 6, 'Mountain View',
FALSE),
('303 Coastal Road, Prince Rupert, BC', 13, 250.00, 3, 'City View', TRUE),
('303 Coastal Road, Prince Rupert, BC', 14, 300.00, 4, 'Garden View', FALSE),
('303 Coastal Road, Prince Rupert, BC', 15, 350.00, 5, 'Sea View', TRUE),
('404 Beachfront Ave, Nanaimo, BC', 20, 150.00, 2, 'Sea View', TRUE),
('404 Beachfront Ave, Nanaimo, BC', 22, 200.00, 6, 'Mountain View', FALSE),
('404 Beachfront Ave, Nanaimo, BC', 23, 250.00, 3, 'City View', TRUE),
('404 Beachfront Ave, Nanaimo, BC', 24, 300.00, 4, 'Garden View', FALSE),
('404 Beachfront Ave, Nanaimo, BC', 25, 350.00, 5, 'Sea View', TRUE),
('505 Harbour Lane, Vancouver, BC', 20, 150.00, 2, 'Sea View', TRUE),
('505 Harbour Lane, Vancouver, BC', 22, 200.00, 6, 'Mountain View', FALSE),
('505 Harbour Lane, Vancouver, BC', 23, 250.00, 3, 'City View', TRUE),
```



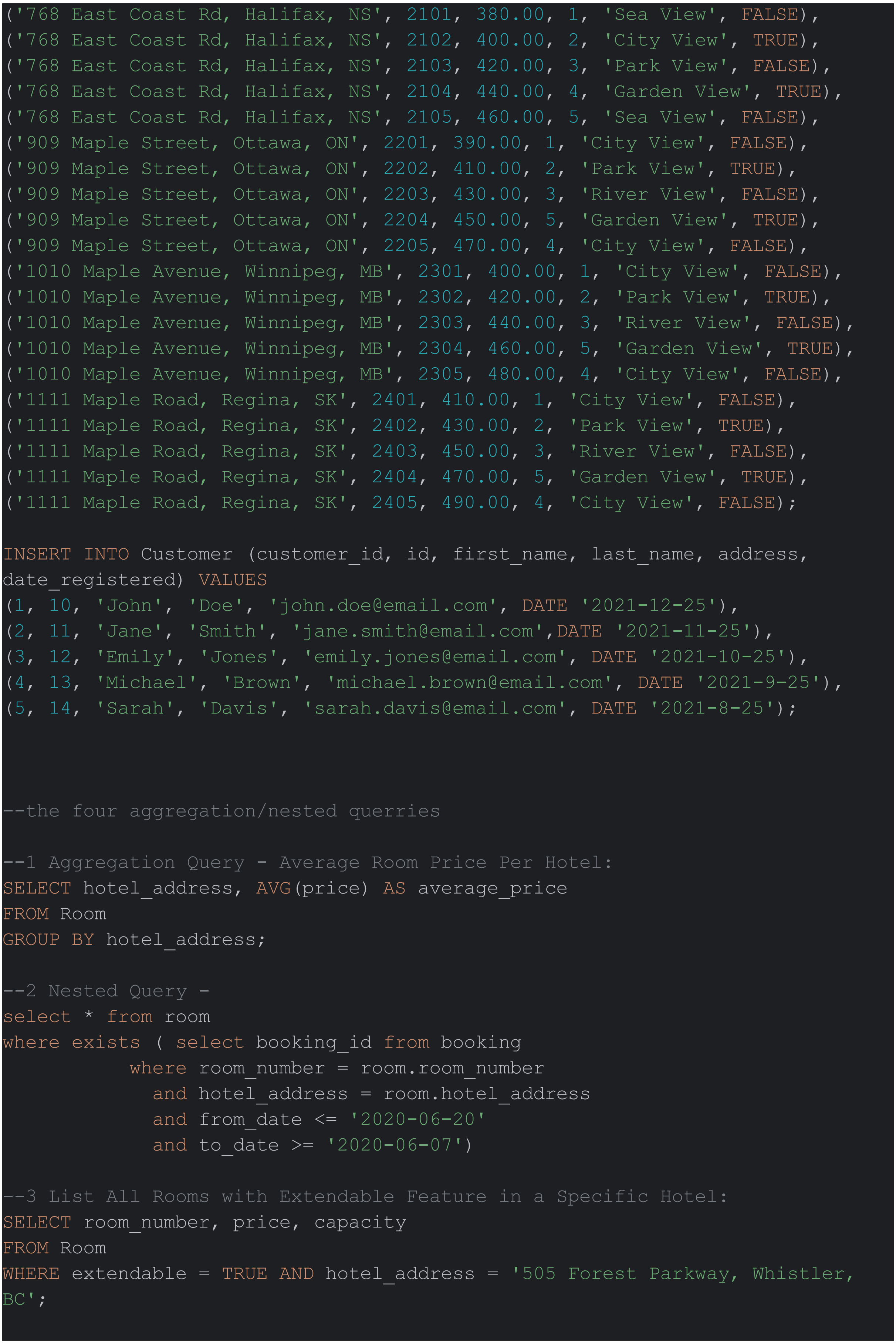
```
('505 Harbour Lane, Vancouver, BC', 24, 300.00, 4, 'Garden View', FALSE),
('505 Harbour Lane, Vancouver, BC', 25, 350.00, 5, 'Sea View', TRUE),
('606 Harbor View, Sidney, BC', 30, 150.00, 2, 'Sea View', TRUE),
('606 Harbor View, Sidney, BC', 32, 200.00, 6, 'Mountain View', FALSE),
('606 Harbor View, Sidney, BC', 33, 250.00, 3, 'City View', TRUE),
('606 Harbor View, Sidney, BC', 34, 300.00, 4, 'Garden View', FALSE),
('606 Harbor View, Sidney, BC', 35, 350.00, 5, 'Sea View', TRUE),
('707 Oceanfront Walk, Powell River, BC', 40, 150.00, 2, 'Sea View', TRUE),
('707 Oceanfront Walk, Powell River, BC', 42, 200.00, 6, 'Mountain View',
FALSE),
('707 Oceanfront Walk, Powell River, BC', 43, 250.00, 3, 'City View', TRUE),
('707 Oceanfront Walk, Powell River, BC', 44, 300.00, 4, 'Garden View',
FALSE),
('707 Oceanfront Walk, Powell River, BC', 45, 350.00, 5, 'Sea View', TRUE),
('808 Seashore Drive, Port Hardy, BC', 101, 100.00, 1, 'Street View', FALSE),
('808 Seashore Drive, Port Hardy, BC', 102, 150.00, 2, 'Garden View', FALSE),
('808 Seashore Drive, Port Hardy, BC', 103, 200.00, 5, 'Sea View', TRUE),
('808 Seashore Drive, Port Hardy, BC', 104, 250.00, 3, 'Sea View', TRUE),
('808 Seashore Drive, Port Hardy, BC', 105, 300.00, 4, 'Panoramic Sea View',
TRUE);

--French Charm Hotel Rooms
INSERT INTO Room (hotel_address, room_number, price, capacity, view,
extendable) VALUES
('101 Rue de la Montagne, Montreal, QC', 501, 150.00, 1, 'City View', FALSE),
('101 Rue de la Montagne, Montreal, QC', 502, 200.00, 2, 'City View', FALSE),
('101 Rue de la Montagne, Montreal, QC', 503, 250.00, 5, 'Park View', TRUE),
('101 Rue de la Montagne, Montreal, QC', 504, 300.00, 3, 'Mountain View',
TRUE),
('101 Rue de la Montagne, Montreal, QC', 505, 350.00, 4, 'Panoramic City
View', TRUE),
('202 Rue Saint-Jean, Quebec City, QC', 101, 150.00, 2, 'Sea View', TRUE),
('202 Rue Saint-Jean, Quebec City, QC', 102, 200.00, 1, 'Mountain View',
FALSE),
('202 Rue Saint-Jean, Quebec City, QC', 103, 250.00, 3, 'City View', TRUE),
('202 Rue Saint-Jean, Quebec City, QC', 104, 300.00, 4, 'Garden View',
FALSE),
('202 Rue Saint-Jean, Quebec City, QC', 105, 350.00, 5, 'Sea View', TRUE),
('303 Rue Saint-Paul, Trois-Rivières, QC', 101, 120.00, 1, 'City View',
FALSE),
('303 Rue Saint-Paul, Trois-Rivières, QC', 102, 130.00, 2, 'City View',
TRUE),
('303 Rue Saint-Paul, Trois-Rivières, QC', 103, 150.00, 5, 'Garden View',
FALSE),
('303 Rue Saint-Paul, Trois-Rivières, QC', 104, 170.00, 3, 'River View',
TRUE),
('303 Rue Saint-Paul, Trois-Rivières, QC', 105, 200.00, 4, 'River View',
FALSE),
('404 Boulevard Charest, Sherbrooke, QC', 101, 125.00, 2, 'City View',
FALSE),
('404 Boulevard Charest, Sherbrooke, QC', 102, 150.00, 1, 'Garden View',
TRUE),
```



```
('404 Boulevard Charest, Sherbrooke, QC', 103, 175.00, 3, 'Mountain View',
FALSE),
('404 Boulevard Charest, Sherbrooke, QC', 104, 200.00, 5, 'City View', TRUE),
('404 Boulevard Charest, Sherbrooke, QC', 105, 250.00, 4, 'Lake View',
FALSE),
('505 Rue King Ouest, Gatineau, QC', 201, 150.00, 2, 'City View', FALSE),
('505 Rue King Ouest, Gatineau, QC', 202, 180.00, 1, 'Garden View', TRUE),
('505 Rue King Ouest, Gatineau, QC', 203, 200.00, 3, 'River View', FALSE),
('505 Rue King Ouest, Gatineau, QC', 204, 250.00, 5, 'City View', TRUE),
('505 Rue King Ouest, Gatineau, QC', 205, 300.00, 4, 'River View', TRUE),
('606 Rue Principale, Sainte-Anne-de-Beaupré, QC', 101, 120.00, 1, 'City
View', FALSE),
('606 Rue Principale, Sainte-Anne-de-Beaupré, QC', 102, 150.00, 5, 'Garden
View', TRUE),
('606 Rue Principale, Sainte-Anne-de-Beaupré, QC', 103, 175.00, 2, 'Mountain
View', FALSE),
('606 Rue Principale, Sainte-Anne-de-Beaupré, QC', 104, 200.00, 3, 'River
View', TRUE),
('606 Rue Principale, Sainte-Anne-de-Beaupré, QC', 105, 250.00, 4, 'Sea
View', FALSE),
('707 Boulevard René-Lévesque, Montreal, QC', 301, 200.00, 1, 'City View',
FALSE),
('707 Boulevard René-Lévesque, Montreal, QC', 302, 220.00, 5, 'Park View',
TRUE),
('707 Boulevard René-Lévesque, Montreal, QC', 303, 250.00, 2, 'City View',
FALSE),
('707 Boulevard René-Lévesque, Montreal, QC', 304, 275.00, 3, 'Garden View',
TRUE),
('707 Boulevard René-Lévesque, Montreal, QC', 305, 300.00, 4, 'River View',
FALSE),
('808 Chemin du Roy, Quebec City, QC', 401, 210.00, 2, 'City View', FALSE),
('808 Chemin du Roy, Quebec City, QC', 402, 230.00, 1, 'Park View', TRUE),
('808 Chemin du Roy, Quebec City, QC', 403, 250.00, 3, 'City View', FALSE),
('808 Chemin du Roy, Quebec City, QC', 404, 270.00, 5, 'Garden View', TRUE),
('808 Chemin du Roy, Quebec City, QC', 405, 290.00, 4, 'River View', FALSE);

--Rocky Mountain Lodges Rooms
INSERT INTO Room (hotel_address, room_number, price, capacity, view,
extendable) VALUES
('101 Peak View, Banff, AB', 101, 200.00, 2, 'Mountain View', TRUE),
('101 Peak View, Banff, AB', 102, 250.00, 1, 'Forest View', FALSE),
('101 Peak View, Banff, AB', 103, 300.00, 3, 'Lake View', TRUE),
('101 Peak View, Banff, AB', 104, 350.00, 5, 'Mountain View', FALSE),
('101 Peak View, Banff, AB', 105, 400.00, 4, 'City View', TRUE),
('202 Summit Drive, Jasper, AB', 201, 250.00, 1, 'Mountain View', FALSE),
('202 Summit Drive, Jasper, AB', 202, 275.00, 2, 'Lake View', TRUE),
('202 Summit Drive, Jasper, AB', 203, 300.00, 5, 'Forest View', FALSE),
('202 Summit Drive, Jasper, AB', 204, 325.00, 3, 'Mountain View', TRUE),
('202 Summit Drive, Jasper, AB', 205, 350.00, 4, 'City View', FALSE),
('303 Ridge Road, Canmore, AB', 301, 200.00, 1, 'Mountain View', TRUE),
('303 Ridge Road, Canmore, AB', 302, 220.00, 2, 'Forest View', FALSE),
('303 Ridge Road, Canmore, AB', 303, 250.00, 3, 'Lake View', TRUE),
('303 Ridge Road, Canmore, AB', 304, 275.00, 5, 'Mountain View', FALSE),
('303 Ridge Road, Canmore, AB', 305, 300.00, 4, 'City View', TRUE),
('404 Mountain Trail, Revelstoke, BC', 401, 210.00, 1, 'Forest View', FALSE),
('404 Mountain Trail, Revelstoke, BC', 402, 230.00, 2, 'Mountain View',
TRUE),
('404 Mountain Trail, Revelstoke, BC', 403, 250.00, 5, 'Lake View', FALSE),
('404 Mountain Trail, Revelstoke, BC', 404, 270.00, 3, 'City View', TRUE),
('404 Mountain Trail, Revelstoke, BC', 405, 290.00, 4, 'Forest View', FALSE),
('505 Forest Parkway, Whistler, BC', 501, 220.00, 1, 'Mountain View', FALSE),
('505 Forest Parkway, Whistler, BC', 502, 240.00, 2, 'Forest View', TRUE),
('505 Forest Parkway, Whistler, BC', 503, 260.00, 3, 'Lake View', FALSE),
('505 Forest Parkway, Whistler, BC', 504, 280.00, 5, 'City View', TRUE),
('505 Forest Parkway, Whistler, BC', 505, 300.00, 4, 'Forest View', FALSE),
('909 Valley Road, Kananaskis, AB', 601, 230.00, 2, 'Mountain View', FALSE),
('909 Valley Road, Kananaskis, AB', 602, 250.00, 1, 'Forest View', TRUE),
('909 Valley Road, Kananaskis, AB', 603, 270.00, 3, 'Lake View', FALSE),
('909 Valley Road, Kananaskis, AB', 604, 290.00, 5, 'Valley View', TRUE),
('909 Valley Road, Kananaskis, AB', 605, 310.00, 4, 'Forest View', FALSE),
('1010 Highland Ave, Fernie, BC', 701, 240.00, 1, 'Mountain View', FALSE),
('1010 Highland Ave, Fernie, BC', 702, 260.00, 2, 'Forest View', TRUE),
('1010 Highland Ave, Fernie, BC', 703, 280.00, 5, 'Lake View', FALSE),
('1010 Highland Ave, Fernie, BC', 704, 300.00, 3, 'Valley View', TRUE),
('1010 Highland Ave, Fernie, BC', 705, 320.00, 4, 'Forest View', FALSE),
('1111 Mountain View, Kimberley, BC', 801, 250.00, 1, 'Mountain View',
FALSE),
('1111 Mountain View, Kimberley, BC', 802, 270.00, 2, 'Forest View', TRUE),
('1111 Mountain View, Kimberley, BC', 803, 290.00, 5, 'Lake View', FALSE),
('1111 Mountain View, Kimberley, BC', 804, 310.00, 3, 'City View', TRUE),
('1111 Mountain View, Kimberley, BC', 805, 330.00, 4, 'Mountain View',
FALSE);

--Northern Lights Resorts Rooms
INSERT INTO Room (hotel_address, room_number, price, capacity, view,
extendable) VALUES
('101 Aurora Street, Whitehorse, YT', 901, 260.00, 1, 'City View', FALSE),
('101 Aurora Street, Whitehorse, YT', 902, 280.00, 2, 'Mountain View', TRUE),
('101 Aurora Street, Whitehorse, YT', 903, 300.00, 3, 'Forest View', FALSE),
('101 Aurora Street, Whitehorse, YT', 904, 320.00, 4, 'Lake View', TRUE),
('101 Aurora Street, Whitehorse, YT', 905, 340.00, 5, 'City View', FALSE),
('202 Polar Road, Yellowknife, NT', 1001, 270.00, 1, 'City View', FALSE),
('202 Polar Road, Yellowknife, NT', 1002, 290.00, 2, 'Mountain View', TRUE),
('202 Polar Road, Yellowknife, NT', 1003, 310.00, 3, 'Forest View', FALSE),
('202 Polar Road, Yellowknife, NT', 1004, 330.00, 4, 'Lake View', TRUE),
('202 Polar Road, Yellowknife, NT', 1005, 350.00, 5, 'City View', FALSE),
('303 Iceberg Lane, Iqaluit, NU', 1101, 280.00, 1, 'City View', FALSE),
('303 Iceberg Lane, Iqaluit, NU', 1102, 300.00, 5, 'Mountain View', TRUE),
('303 Iceberg Lane, Iqaluit, NU', 1103, 320.00, 2, 'Forest View', FALSE),
('303 Iceberg Lane, Iqaluit, NU', 1104, 340.00, 3, 'Lake View', TRUE),
('303 Iceberg Lane, Iqaluit, NU', 1105, 360.00, 4, 'City View', FALSE),
('404 Northern Ave, Edmonton, AB', 1201, 290.00, 1, 'City View', FALSE),
('404 Northern Ave, Edmonton, AB', 1202, 310.00, 2, 'Park View', TRUE),
('404 Northern Ave, Edmonton, AB', 1203, 330.00, 3, 'River View', FALSE),
('404 Northern Ave, Edmonton, AB', 1204, 350.00, 5, 'Garden View', TRUE),
('404 Northern Ave, Edmonton, AB', 1205, 370.00, 4, 'City View', FALSE),
```



```
('505 Tundra Trail, Churchill, MB', 1301, 300.00, 1, 'City View', FALSE),
('505 Tundra Trail, Churchill, MB', 1302, 320.00, 2, 'Park View', TRUE),
('505 Tundra Trail, Churchill, MB', 1303, 340.00, 5, 'River View', FALSE),
('505 Tundra Trail, Churchill, MB', 1304, 360.00, 3, 'Garden View', TRUE),
('505 Tundra Trail, Churchill, MB', 1305, 380.00, 4, 'City View', FALSE),
('606 Arctic View, Fort McMurray, AB', 1401, 310.00, 1, 'City View', FALSE),
('606 Arctic View, Fort McMurray, AB', 1402, 330.00, 2, 'Park View', TRUE),
('606 Arctic View, Fort McMurray, AB', 1403, 350.00, 3, 'River View', FALSE),
('606 Arctic View, Fort McMurray, AB', 1404, 370.00, 5, 'Garden View', TRUE),
('606 Arctic View, Fort McMurray, AB', 1405, 390.00, 4, 'City View', FALSE),
('707 Polar Express, Labrador City, NL', 1501, 320.00, 1, 'City View',
FALSE),
('707 Polar Express, Labrador City, NL', 1502, 340.00, 2, 'Park View', TRUE),
('707 Polar Express, Labrador City, NL', 1503, 360.00, 3, 'River View',
FALSE),
('707 Polar Express, Labrador City, NL', 1504, 380.00, 5, 'Garden View',
TRUE),
('707 Polar Express, Labrador City, NL', 1505, 400.00, 4, 'City View',
FALSE),
('808 Northern Edge, Thompson, MB', 1601, 330.00, 1, 'City View', FALSE),
('808 Northern Edge, Thompson, MB', 1602, 350.00, 2, 'Park View', TRUE),
('808 Northern Edge, Thompson, MB', 1603, 370.00, 5, 'River View', FALSE),
('808 Northern Edge, Thompson, MB', 1604, 390.00, 3, 'Garden View', TRUE),
('808 Northern Edge, Thompson, MB', 1605, 410.00, 4, 'City View', FALSE);

-- Maple Leafs Hotels Rooms
INSERT INTO Room (hotel_address, room_number, price, capacity, view,
extendable) VALUES
('1 Maple Lane, Toronto, ON', 1701, 350.00, 1, 'City View', FALSE),
('1 Maple Lane, Toronto, ON', 1702, 370.00, 2, 'Park View', TRUE),
('1 Maple Lane, Toronto, ON', 1703, 390.00, 3, 'River View', FALSE),
('1 Maple Lane, Toronto, ON', 1704, 410.00, 5, 'Garden View', TRUE),
('1 Maple Lane, Toronto, ON', 1705, 430.00, 4, 'City View', FALSE),
('200 Mountain View, Calgary, AB', 1801, 340.00, 1, 'City View', FALSE),
('200 Mountain View, Calgary, AB', 1802, 360.00, 2, 'Park View', TRUE),
('200 Mountain View, Calgary, AB', 1803, 380.00, 3, 'River View', FALSE),
('200 Mountain View, Calgary, AB', 1804, 400.00, 5, 'Garden View', TRUE),
('200 Mountain View, Calgary, AB', 1805, 420.00, 4, 'City View', FALSE),
('555 Seaside Blvd, Vancouver, BC', 1901, 360.00, 1, 'Sea View', FALSE),
('555 Seaside Blvd, Vancouver, BC', 1902, 380.00, 2, 'Mountain View', TRUE),
('555 Seaside Blvd, Vancouver, BC', 1903, 400.00, 3, 'City View', FALSE),
('555 Seaside Blvd, Vancouver, BC', 1904, 420.00, 4, 'Garden View', TRUE),
('555 Seaside Blvd, Vancouver, BC', 1905, 440.00, 5, 'Sea View', FALSE),
('1010 Northern Lights Ave, Yellowknife, NT', 2001, 370.00, 1, 'City View',
FALSE),
('1010 Northern Lights Ave, Yellowknife, NT', 2002, 390.00, 2, 'Park View',
TRUE),
('1010 Northern Lights Ave, Yellowknife, NT', 2003, 410.00, 3, 'River View',
FALSE),
('1010 Northern Lights Ave, Yellowknife, NT', 2004, 430.00, 5, 'Garden View',
TRUE),
('1010 Northern Lights Ave, Yellowknife, NT', 2005, 450.00, 4, 'City View',
FALSE),
```



```
('768 East Coast Rd, Halifax, NS', 2101, 380.00, 1, 'Sea View', FALSE),
('768 East Coast Rd, Halifax, NS', 2102, 400.00, 2, 'City View', TRUE),
('768 East Coast Rd, Halifax, NS', 2103, 420.00, 3, 'Park View', FALSE),
('768 East Coast Rd, Halifax, NS', 2104, 440.00, 4, 'Garden View', TRUE),
('768 East Coast Rd, Halifax, NS', 2105, 460.00, 5, 'Sea View', FALSE),
('909 Maple Street, Ottawa, ON', 2201, 390.00, 1, 'City View', FALSE),
('909 Maple Street, Ottawa, ON', 2202, 410.00, 2, 'Park View', TRUE),
('909 Maple Street, Ottawa, ON', 2203, 430.00, 3, 'River View', FALSE),
('909 Maple Street, Ottawa, ON', 2204, 450.00, 5, 'Garden View', TRUE),
('909 Maple Street, Ottawa, ON', 2205, 470.00, 4, 'City View', FALSE),
('1010 Maple Avenue, Winnipeg, MB', 2301, 400.00, 1, 'City View', FALSE),
('1010 Maple Avenue, Winnipeg, MB', 2302, 420.00, 2, 'Park View', TRUE),
('1010 Maple Avenue, Winnipeg, MB', 2303, 440.00, 3, 'River View', FALSE),
('1010 Maple Avenue, Winnipeg, MB', 2304, 460.00, 5, 'Garden View', TRUE),
('1010 Maple Avenue, Winnipeg, MB', 2305, 480.00, 4, 'City View', FALSE),
('1111 Maple Road, Regina, SK', 2401, 410.00, 1, 'City View', FALSE),
('1111 Maple Road, Regina, SK', 2402, 430.00, 2, 'Park View', TRUE),
('1111 Maple Road, Regina, SK', 2403, 450.00, 3, 'River View', FALSE),
('1111 Maple Road, Regina, SK', 2404, 470.00, 5, 'Garden View', TRUE),
('1111 Maple Road, Regina, SK', 2405, 490.00, 4, 'City View', FALSE);

INSERT INTO Customer (customer_id, id, first_name, last_name, address,
date_registered) VALUES
(1, 10, 'John', 'Doe', 'john.doe@email.com', DATE '2021-12-25'),
(2, 11, 'Jane', 'Smith', 'jane.smith@email.com',DATE '2021-11-25'),
(3, 12, 'Emily', 'Jones', 'emily.jones@email.com', DATE '2021-10-25'),
(4, 13, 'Michael', 'Brown', 'michael.brown@email.com', DATE '2021-9-25'),
(5, 14, 'Sarah', 'Davis', 'sarah.davis@email.com', DATE '2021-8-25');



--the four aggregation/nested querries

--1 Aggregation Query - Average Room Price Per Hotel:
SELECT hotel_address, AVG(price) AS average_price
FROM Room
GROUP BY hotel_address;

--2 Nested Query -
select * from room
where exists ( select booking_id from booking
           where room_number = room.room_number
             and hotel_address = room.hotel_address
             and from_date <= '2020-06-20'
             and to_date >= '2020-06-07')

--3 List All Rooms with Extendable Feature in a Specific Hotel:
SELECT room_number, price, capacity
FROM Room
WHERE extendable = TRUE AND hotel_address = '505 Forest Parkway, Whistler,
BC';
```

```
--4 USED DURING SEARCH IN WEB APP(Nested): Rooms that are not booked between
a specific date from a specific hotel chain and a specific capacity:
SELECT * FROM room LEFT OUTER JOIN hotel ON room.hotel_address =
hotel.address
WHERE exists( select address from hotel h where h.address =
room.hotel_address and h.hotel_chain_name = 'Pacific Coast Inns')
AND capacity = 4 AND not exists( select booking_id from booking
where room_number = room.room_number and hotel_address = room.hotel_address
and from_date <= '2024-03-21' and to_date >= '2024-03-06')
8

--5 useful query, select all bookings by a specific customer
SELECT *
FROM Booking
WHERE customer_id = 1;



--TRIGGERS

--TRIGGER#1
--prevent deletion of a hotel if there are associated rooms
CREATE OR REPLACE FUNCTION prevent_hotel_deletion()
RETURNS TRIGGER AS $$
BEGIN
   -- Check if there are any rooms associated with the hotel
   IF EXISTS (SELECT 1 FROM Room WHERE hotel_address = OLD.address) THEN
       RAISE EXCEPTION 'Cannot delete hotel: there are rooms associated with
it.';
   END IF;
   RETURN OLD;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER check_hotel_deletion
BEFORE DELETE ON Hotel
FOR EACH ROW EXECUTE FUNCTION prevent_hotel_deletion();

--TRIGGER #2
--room capacity does not exceed a maximum limit
CREATE OR REPLACE FUNCTION check_room_capacity()
RETURNS TRIGGER AS $$
BEGIN
   -- Define the maximum allowed capacity which is 10
   IF NEW.capacity > 10 THEN
       RAISE EXCEPTION 'Room capacity exceeds the maximum limit of 10.';
   END IF;
   RETURN NEW;
END;
$$ LANGUAGE plpgsql;

-- Trigger for Room Insertion to check capacity
CREATE TRIGGER verify_room_capacity_before_insert
```

```
BEFORE INSERT ON Room
FOR EACH ROW EXECUTE FUNCTION check_room_capacity();

-- Trigger for Room Update to check capacity
CREATE TRIGGER verify_room_capacity_before_update
BEFORE UPDATE ON Room
FOR EACH ROW EXECUTE FUNCTION check_room_capacity();

--TRIGGER #3
-- prevent the deletion of a customer, if that customer has a booking
CREATE OR REPLACE FUNCTION prevent_customer_deletion_if_booking_exists()
RETURNS TRIGGER AS $$
BEGIN
   -- Check if the customer has any bookings
   IF EXISTS (SELECT 1 FROM Booking WHERE customer_id = OLD.customer_id) THEN
       RAISE EXCEPTION 'Cannot delete customer: they have existing
bookings.';
   END IF;
   RETURN OLD;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER check_customer_booking_before_deletion
BEFORE DELETE ON Customer
FOR EACH ROW EXECUTE FUNCTION prevent_customer_deletion_if_booking_exists();


--INDEXES
CREATE INDEX idx_room_hotel_address ON Room (hotel_address);
--help in speeding up queries that filter or join on the hotel_address.
--For example, in room_search.jsp page, when we search for rooms in a
specific hotel,
--the query will be faster if we have an index on hotel_address.
CREATE INDEX idx_booking_customer_id ON Booking (customer_id);
--useful for queries that involve searching, filtering, or joining on the
customer_id.
--For example, when we want to display all bookings made by a specific
customer, the query will be faster if we have an index on customer_id.
CREATE INDEX idx_booking_dates ON Booking (from_date, to_date);
--optimize queries involving date ranges, which is used in room_search when
we run the NOT EXIST (from_date) (to_date) query.


--VIEWS
--view #1 available rooms ber province
CREATE VIEW AvailableRoomsPerProvince AS
SELECT
   split_part(Hotel.address, ', ', 3) AS province,
   COUNT(*) AS available_rooms
FROM
   Hotel
JOIN
   Room ON Hotel.address = Room.hotel_address
```

```
GROUP BY
   province;

SELECT * FROM AvailableRoomsPerProvince WHERE province = 'BC';


--view #2 aggregated capacity of all rooms of a specific hotel
CREATE VIEW TotalRoomCapacityPerHotel AS
SELECT hotel_address, SUM(capacity) AS total_capacity
FROM Room
GROUP BY hotel_address;

SELECT * FROM TotalRoomCapacityPerHotel WHERE hotel_address = '1 Maple Lane,
Toronto, ON';

--view #3 list bookings and rentings together (use B and R to categorize them
in the view)

CREATE VIEW BookingAndRenting AS
SELECT b.booking_id as book_rent_id, 'B' as book_or_rent, b.customer_id,
b.room_number, b.hotel_address, b.from_date, b.to_date
 from booking as b
UNION
SELECT r.renting_id as book_rent_id, 'R' as book_or_rent, r.customer_id,
r.room_number, r.hotel_address, r.from_date, r.to_date
from renting as r

select * from bookingandrenting;
```

[redacted]