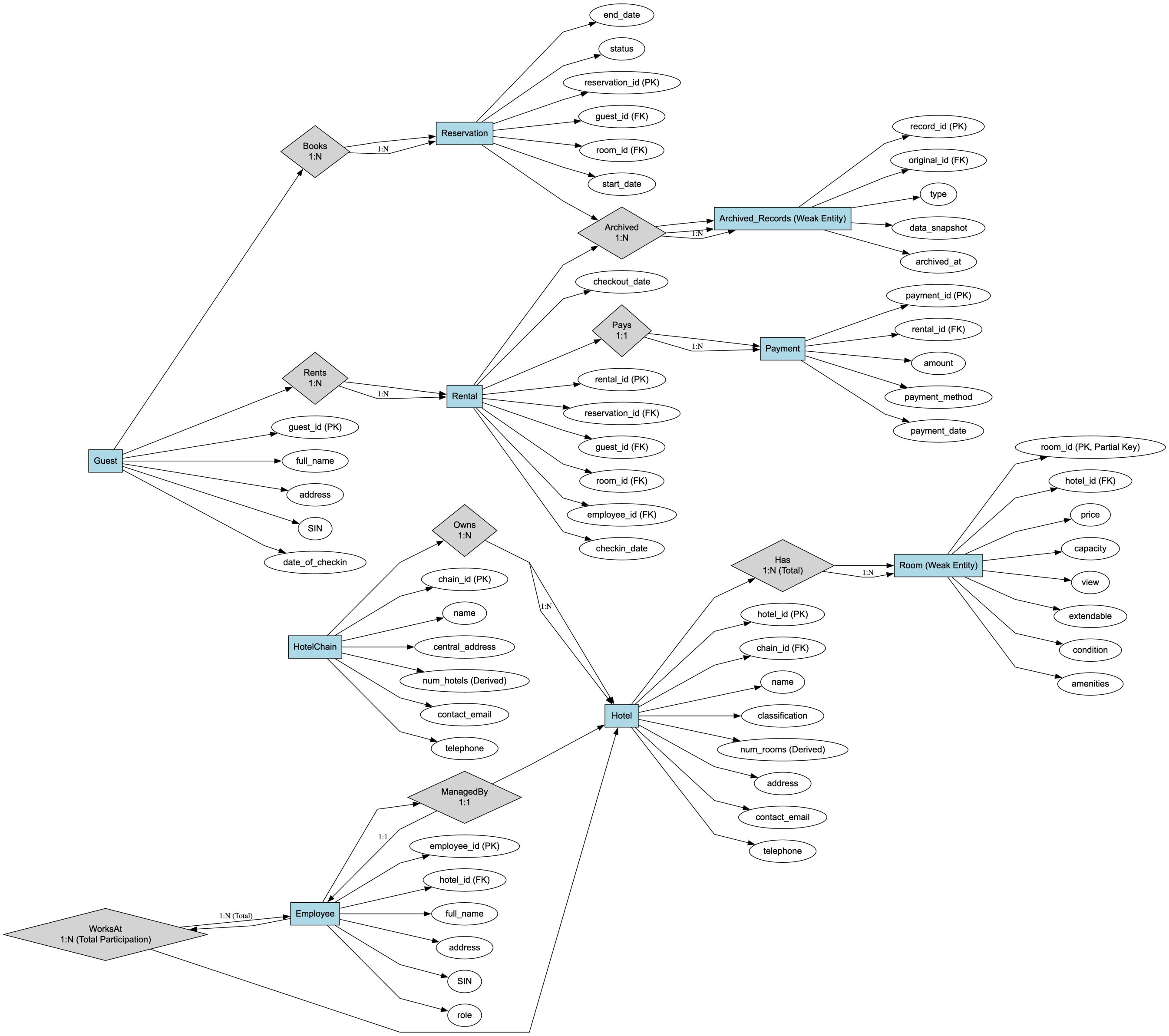
Project Report

Name: Hamed Tava
Student Number: 300321356
Class: CSI 2532
Semester: Winter 25
Date: 21 Mar, 2025

Contents

1 E-R Diagram



1.1 Key Design Considerations

- **Hotel Chains own multiple Hotels** → `1:N relationship`: A hotel chain can own many hotels, but each hotel is owned by only one hotel chain.
- **Hotels contain multiple Rooms** → `1:N relationship`: A hotel consists of multiple rooms, and each room belongs to one hotel.
- **Rooms are Weak Entities** → Rooms cannot exist without a Hotel: Rooms are dependent on the existence of a hotel. A room cannot be created unless a hotel is present.
- **Guests make multiple Reservations** → `1:N relationship`: A guest can make multiple reservations, but each reservation is linked to one guest.
- **Reservations get converted into Rentals** → `1:N relationship`: A reservation can convert into multiple rentals, and each rental is associated with one reservation.

- **Employees work at Hotels** → `1:N relationship`: Employees work at specific hotels, and a hotel can have multiple employees.

- **Hotels have one Manager** → `1:1 relationship` (subset of Employees): Each hotel has exactly one manager, who is also considered an employee of the hotel.

- **Archived Records store old Reservations and Rentals** → To preserve history: Archived records retain historical data of past reservations and rentals.

- **Rentals have Payments** → `1:N relationship`: A rental can have multiple payments (e.g., installments), and each payment is linked to one rental.

1.1.1 Weak Entities and Relationships

- **Room is a Weak Entity**: A room cannot exist independently and must always be associated with a hotel. Hence, the **Room** entity is a weak entity.

- **Hotel is strongly dependent on the Hotel Chain**: A hotel is considered strongly dependent on the hotel chain because every hotel must belong to a hotel chain.

- **Reservation depends on Guest and Room, but remains a regular entity**: A reservation is dependent on both a guest and a room, but it is not a weak entity since it can exist independently.

- **Rental has Payment**: Each rental can have multiple payments (e.g., in installments), but the payment is always linked to one rental.

1.1.2 Participation Constraints

- **Every Hotel must belong to a Hotel Chain** → `Total Participation`: Every hotel must be associated with a hotel chain. There can be no hotel without a hotel chain.

- **Every Room must belong to a Hotel** → `Total Participation`: Each room must be part of a specific hotel. A room cannot exist without being assigned to a hotel.

- **Every Rental must come from a Reservation** → `Total Participation`: A rental cannot exist without having an associated reservation.

- **Every Employee must belong to a Hotel** → `Total Participation`: An employee must always be assigned to a hotel, ensuring that each hotel has employees working at it.

- **Every Hotel must have at least one Manager** → `Total Participation`: A hotel is required to have a manager, ensuring that each hotel has one and only one manager.

- **Not every Guest has to make a Reservation** → `Partial Participation`: While many guests will make reservations, not every guest is required to make one. Some guests may only inquire without making any bookings.

1.2 Relationships

- **Hotel Chains** own **Hotels** (1:N) – A hotel chain can own multiple hotels, but each hotel is owned by exactly one hotel chain.

- **Hotels** have **Rooms** (1:N, Weak Entity) – Each hotel has multiple rooms, and a room can only exist in one hotel.

- **Guests** make **Reservations** (1:N) – A guest can make multiple reservations, but each reservation is linked to exactly one guest.

- **Employees** work at **Hotels** (1:N, Total Participation) – Employees work at a specific hotel, and each hotel must have employees working at it (total participation).

- A **Hotel** has exactly one **Manager** (1:1 Relationship) – Each hotel has one and only one manager, and a manager manages exactly one hotel.
- **Reservations** become **Rentals** (1:N Conversion) – A reservation can evolve into a rental (a booking for accommodation), and a rental corresponds to a specific reservation.
- **Archived Records** store past **Reservations/Rentals** – Archived records track past reservations or rentals, which may have been completed.
- **Rentals have Payments** (1:N) – Each rental can have multiple payments, but each payment is associated with exactly one rental.

1.3 Justification of Constraints

1.3.1 Primary Key Constraints

- **HotelChain**: `chain_id` is the primary key.
  - **Justification**: Uniquely identifies each hotel chain.
- **Hotel**: `hotel_id` is the primary key.
  - **Justification**: Uniquely identifies each hotel within a hotel chain.
- **Room**: `room_id` is the primary key.
  - **Justification**: Uniquely identifies each room in the hotel.
- **Guest**: `guest_id` is the primary key.
  - **Justification**: Uniquely identifies each guest.
- **Employee**: `employee_id` is the primary key.
  - **Justification**: Uniquely identifies each employee.
- **Reservation**: `reservation_id` is the primary key.
  - **Justification**: Uniquely identifies each reservation made by guests.
- **Rental**: `rental_id` is the primary key.
  - **Justification**: Uniquely identifies each rental (the actual transaction after check-in).
- **Archived_Reservation**: `archived_reservation_id` is the primary key.
  - **Justification**: Uniquely identifies each archived reservation.
- **Payment**: `payment_id` is the primary key.
  - **Justification**: Uniquely identifies each payment made for a rental.

1.3.2 Foreign Key Constraints

- **HotelChain → Hotel** (1:N relationship): `hotel_chain_id` in `Hotel` references `hotel_chain_id` in `HotelChain`.
  - **Justification**: Ensures that a hotel belongs to an existing hotel chain.
- **Hotel → Room** (1:N relationship): `hotel_id` in `Room` references `hotel_id` in `Hotel`.
  - **Justification**: Ensures that a room belongs to an existing hotel.
- **Room → Reservation** (1:N relationship): `room_id` in `Reservation` references `room_id` in `Room`.

  - **Justification**: Ensures that reservations are made for available rooms.

- **Guest → Reservation** (1:N relationship): `guest_id` in `Reservation` references `guest_id` in `Guest`.
  - **Justification**: Ensures that each reservation is associated with a guest.

- **Employee → Hotel** (1:1 relationship, specialized): `employee_id` in `Hotel` references `employee_id` in `Employee`.
  - **Justification**: Each hotel has a manager (employee), enforcing a one-to-one relationship.

- **Reservation → Rental** (1:1 relationship): `reservation_id` in `Rental` references `reservation_id` in `Reservation`.
  - **Justification**: A reservation must have a corresponding rental once the guest checks in.

- **HotelChain → Archived_Reservation** (1:N relationship): `hotel_chain_id` in `Archived_Reservation` references `hotel_chain_id` in `HotelChain`.
  - **Justification**: Ensures that archived reservations are linked back to a hotel chain.

1.3.3 Domain Constraints (Valid Values & Ranges)

- **Room price**: `price` in `Room` must be greater than or equal to 0.
  - **Justification**: Ensures no room can have a negative price.

- **Room condition**: `room_condition` in `Room` must be one of {"Damaged", "In Repair", "Good Condition"}.
  - **Justification**: Ensures only valid conditions are recorded.

- **Room capacity**: `room_capacity` in `Room` must be one of {"Single", "Double", "Suite"}.
  - **Justification**: Enforces correct room classification.

- **Guest SIN**: `sin` in `Guest` must be a valid SIN format (9 digits).
  - **Justification**: Ensures that social security numbers are properly formatted.

- **Employee role**: `role` in `Employee` must be one of {"Manager", "Receptionist", "Housekeeper"}.
  - **Justification**: Enforces valid role assignments for employees.

- **Hotel stars classification**: `hotel_stars` in `Hotel` must be an integer between 1 and 5.
  - **Justification**: Enforces a valid rating system for hotels.

- **Booking status**: `status` in `Reservation` must be one of {"Booked", "Checked-in", "Cancelled"}.
  - **Justification**: Ensures reservations are in valid states.

- **View options**: `view` in `Room` must be one of {"Sea View", "Mountain View", "City View"}.
  - **Justification**: Ensures only valid view options are recorded.

1.3.4 Referential Integrity Constraints

- **HotelChain → Hotel**: A hotel cannot exist without being associated with a hotel chain.
  - **Action on Deletion**: `ON DELETE CASCADE` ensures that when a hotel chain is deleted, all associated hotels are deleted.
- **Hotel → Room**: A room cannot exist without being associated with a hotel.
  - **Action on Deletion**: `ON DELETE CASCADE` ensures rooms are deleted when a hotel is deleted.
- **Room → Reservation**: A room cannot be reserved if it does not exist.
  - **Action on Deletion**: `ON DELETE CASCADE` ensures reservations are deleted if the room is deleted.
- **Reservation → Rental**: A reservation must be converted into a rental once the guest checks in.
  - **Action on Deletion**: `ON DELETE CASCADE` ensures that rentals linked to a deleted reservation are deleted.
- **Guest → Reservation**: A guest cannot make a reservation without being registered in the system.

1.3.5 User-Defined Constraints

- **Manager for every hotel**: Each hotel must have at least one employee with the role of ”Manager”.
  - **Justification**: Enforces that each hotel must have one and only one manager.
  - **Action on Deletion**: If a manager is deleted, their associated hotel must not be left without a manager.
  - **Implementation**: This could be a constraint ensuring `role='Manager'` and `employee_count > 0`.
- **No overlapping reservations for the same room**: A room cannot be double-booked.
  - **Justification**: Ensures that the same room cannot be booked by multiple guests at the same time.
- **Only one reservation per guest per room at a time**: A guest can only have one active reservation for each room at any time.
  - **Justification**: Prevents double-booking a room for the same guest.
- **No over-booking of rooms**: Ensure the room count and reservation count match at the time of booking.
  - **Justification**: Prevents booking more rooms than are available.

1.3.6 Historical Data and Archival Constraints

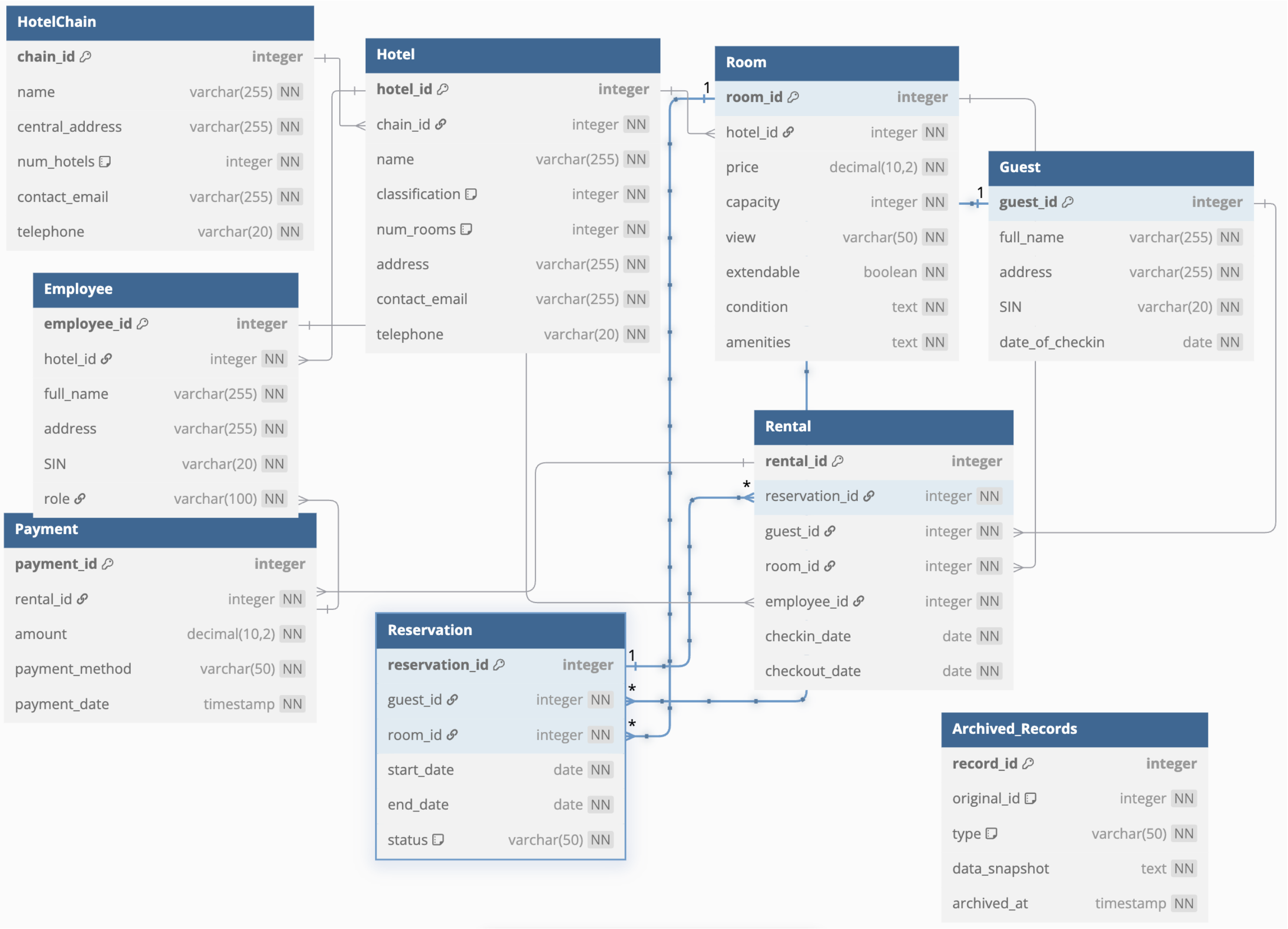
- **Archived Reservations**: Once a reservation or rental is completed, it should be archived.
  - **Justification**: Keeps historical data even after the reservation/rental ends.
- **Archived_Reservation**: Data about rooms that no longer exist should be preserved for historical reference.
  - **Justification**: Maintains a historical record even for outdated room types or deleted hotels.
- **Integrity of archived data**: Archived data must maintain links to the original reservation and room even if the room is deleted.
  - **Justification**: Ensures that even deleted rooms maintain a historical link to reservations and rentals.

1.4 DOT Code

```
digraph ER_Diagram {
    rankdir=LR;
    graph [splines=polyline];

    // Entity Nodes
    node [shape=box, style=filled, fillcolor=lightblue, fontname="Arial"];
    HotelChain [label="HotelChain"];
    Hotel [label="Hotel"];
    Room [label="Room (Weak Entity)"];
    Guest [label="Guest"];
    Employee [label="Employee"];
    Reservation [label="Reservation"];
    Rental [label="Rental"];
    Archived_Records [label="Archived_Records (Weak Entity)"];
    Payment [label="Payment"];

    // Relationship Nodes
    node [shape=diamond, style=filled, fillcolor=lightgray, fontname="Arial"];
    Owns [label="Owns\n1:N"];
    HasRoom [label="Has\n1:N (Total)"];
    Books [label="Books\n1:N"];
    Rents [label="Rents\n1:N"];
    WorksAt [label="WorksAt\n1:N (Total Participation)"];
    ManagedBy [label="ManagedBy\n1:1"];
    Archived [label="Archived\n1:N"];
    Pays [label="Pays\n1:1"];

    // Attributes
    node [shape=ellipse, style=filled, fillcolor=white, fontname="Arial"];

    // HotelChain Attributes
    chain_id [label="chain_id (PK)"];
    chain_name [label="name"];
    central_address [label="central_address"];
    num_hotels [label="num_hotels (Derived)"];
    contact_email [label="contact_email"];
    telephone [label="telephone"];

    HotelChain -> chain_id;
    HotelChain -> chain_name;
    HotelChain -> central_address;
    HotelChain -> num_hotels;
    HotelChain -> contact_email;
    HotelChain -> telephone;

    // Hotel Attributes
    hotel_id [label="hotel_id (PK)"];
    chain_id_fk [label="chain_id (FK)"];
    hotel_name [label="name"];
    classification [label="classification"];
    num_rooms [label="num_rooms (Derived)"];
    hotel_address [label="address"];
    hotel_contact_email [label="contact_email"];
    hotel_telephone [label="telephone"];

    Hotel -> hotel_id;
    Hotel -> chain_id_fk;
    Hotel -> hotel_name;
    Hotel -> classification;
    Hotel -> num_rooms;
    Hotel -> hotel_address;
    Hotel -> hotel_contact_email;
    Hotel -> hotel_telephone;

    // Room (Weak Entity) Attributes
    room_id [label="room_id (PK, Partial Key)"];
    hotel_id_fk [label="hotel_id (FK)"];
    price [label="price"];
    capacity [label="capacity"];
    view [label="view"];
    extendable [label="extendable"];
    condition [label="condition"];
    amenities [label="amenities"];

    Room -> room_id;
    Room -> hotel_id_fk;
    Room -> price;
    Room -> capacity;
    Room -> view;
    Room -> extendable;
    Room -> condition;
    Room -> amenities;

    // Guest Attributes
    guest_id [label="guest_id (PK)"];
    guest_name [label="full_name"];
    guest_address [label="address"];
    SIN [label="SIN"];
    date_of_checkin [label="date_of_checkin"];

    Guest -> guest_id;
    Guest -> guest_name;
    Guest -> guest_address;
    Guest -> SIN;
    Guest -> date_of_checkin;

    // Employee Attributes
    employee_id [label="employee_id (PK)"];
    hotel_id_fk_emp [label="hotel_id (FK)"];
    emp_name [label="full_name"];
    emp_address [label="address"];
    emp_SIN [label="SIN"];
    role [label="role"];

    Employee -> employee_id;
    Employee -> hotel_id_fk_emp;
    Employee -> emp_name;
    Employee -> emp_address;
    Employee -> emp_SIN;
    Employee -> role;

    // Reservation Attributes
    reservation_id [label="reservation_id (PK)"];
    guest_id_fk [label="guest_id (FK)"];
    room_id_fk [label="room_id (FK)"];
    start_date [label="start_date"];
    end_date [label="end_date"];
    status [label="status"];

    Reservation -> reservation_id;
    Reservation -> guest_id_fk;
    Reservation -> room_id_fk;
    Reservation -> start_date;
    Reservation -> end_date;
    Reservation -> status;

    // Rental Attributes
    rental_id [label="rental_id (PK)"];
    reservation_id_fk [label="reservation_id (FK)"];
    guest_id_fk_rental [label="guest_id (FK)"];
    room_id_fk_rental [label="room_id (FK)"];
    employee_id_fk [label="employee_id (FK)"];
    checkin_date [label="checkin_date"];
    checkout_date [label="checkout_date"];

    Rental -> rental_id;
    Rental -> reservation_id_fk;
    Rental -> guest_id_fk_rental;
    Rental -> room_id_fk_rental;
    Rental -> employee_id_fk;
    Rental -> checkin_date;
    Rental -> checkout_date;

    // Archived Records (Weak Entity) Attributes
    record_id [label="record_id (PK)"];
    original_id [label="original_id (FK)"];
    type [label="type"];
    data_snapshot [label="data_snapshot"];
    archived_at [label="archived_at"];

    Archived_Records -> record_id;
    Archived_Records -> original_id;
    Archived_Records -> type;
    Archived_Records -> data_snapshot;
    Archived_Records -> archived_at;

    // Payment Attributes
    payment_id [label="payment_id (PK)"];
    rental_id_fk [label="rental_id (FK)"];
    amount [label="amount"];
    payment_method [label="payment_method"];
    payment_date [label="payment_date"];

    Payment -> payment_id;
    Payment -> rental_id_fk;
    Payment -> amount;
    Payment -> payment_method;
    Payment -> payment_date;

    // Relationships with Cardinalities
    HotelChain -> Owns -> Hotel;
    Hotel -> HasRoom -> Room;
    Guest -> Books -> Reservation;
    Guest -> Rents -> Rental;
    Employee -> WorksAt -> Hotel;
    Employee -> ManagedBy -> Hotel;
    Reservation -> Archived -> Archived_Records;
    Rental -> Archived -> Archived_Records;
    Rental -> Pays -> Payment;

    // Relationship Constraints
    edge [labeldistance=1.5, labelangle=45, fontsize=12];

    Owns -> Hotel [label="1:N"];
    HasRoom -> Room [label="1:N"];
    Books -> Reservation [label="1:N"];
    Rents -> Rental [label="1:N"];
    WorksAt -> Employee [label="1:N (Total)"];
    ManagedBy -> Employee [label="1:1"];
    Archived -> Archived_Records [label="1:N"];
    Pays -> Payment [label="1:N"];
}
```

2 Relational Database Schema

2.1 Schema Diagram Representation



HotelChain		
chain_id	integer	
name	varchar(255)	NN
central_address	varchar(255)	NN
num_hotels	integer	NN
contact_email	varchar(255)	NN
telephone	varchar(20)	NN

Hotel		
hotel_id	integer	
chain_id	integer	NN
name	varchar(255)	NN
classification	integer	NN
num_rooms	integer	NN
address	varchar(255)	NN
contact_email	varchar(255)	NN
telephone	varchar(20)	NN

Room		
room_id	integer	
hotel_id	integer	NN
price	decimal(10,2)	NN
capacity	integer	NN
view	varchar(50)	NN
extendable	boolean	NN
condition	text	NN
amenities	text	NN

Guest		
guest_id	integer	
full_name	varchar(255)	NN
address	varchar(255)	NN
SIN	varchar(20)	NN
date_of_checkin	date	NN

Employee		
employee_id	integer	
hotel_id	integer	NN
full_name	varchar(255)	NN
address	varchar(255)	NN
SIN	varchar(20)	NN
role	varchar(100)	NN

Rental		
rental_id	integer	
reservation_id	integer	NN
guest_id	integer	NN
room_id	integer	NN
employee_id	integer	NN
checkin_date	date	NN
checkout_date	date	NN

Payment		
payment_id	integer	
rental_id	integer	NN
amount	decimal(10,2)	NN
payment_method	varchar(50)	NN
payment_date	timestamp	NN

Reservation		
reservation_id	integer	
guest_id	integer	NN
room_id	integer	NN
start_date	date	NN
end_date	date	NN
status	varchar(50)	NN

Archived_Records		
record_id	integer	
original_id	integer	NN
type	varchar(50)	NN
data_snapshot	text	NN
archived_at	timestamp	NN

2.2 Schema Description

- HotelChains: Stores information about the hotel chains.
- Hotels: Contains data about the hotels.
- Rooms: A table representing rooms in a hotel, with foreign keys linking to the Hotels table.
- Guests: Stores data about the guests.
- Reservations: Includes data on reservations made by guests, with foreign keys linking to Guests and Hotels.
- Employees: Contains employee data, with foreign keys linking to Hotels.
- Managers: A table linking each hotel to its manager, ensuring the 1:1 relationship.
- Rentals: A table representing rentals, which are derived from reservations.
- ArchivedRecords: Stores past reservations and rentals.

2.3 DBML Code

```
Table HotelChain {
    chain_id integer [primary key]
    name varchar(255) [not null]
    central_address varchar(255) [not null]
    num_hotels integer [not null, note: "Derived"]
    contact_email varchar(255) [not null, unique]
    telephone varchar(20) [not null]
}

Table Hotel {
    hotel_id integer [primary key]
    chain_id integer [not null, ref: > HotelChain.chain_id]
    name varchar(255) [not null]
    classification integer [not null, note: "1 to 5-star classification"]
    num_rooms integer [not null, note: "Derived"]
    address varchar(255) [not null]
    contact_email varchar(255) [not null, unique]
    telephone varchar(20) [not null]
}

Table Room {
    room_id integer [primary key]
    hotel_id integer [not null, ref: > Hotel.hotel_id]
    price decimal(10,2) [not null]
    capacity integer [not null]
    view varchar(50) [not null]
    extendable boolean [not null]
    condition text [not null]
    amenities text [not null]
}

Table Guest {
    guest_id integer [primary key]
    full_name varchar(255) [not null]
    address varchar(255) [not null]
    SIN varchar(20) [not null, unique]
    date_of_checkin date [not null]
}

Table Employee {
    employee_id integer [primary key]
    hotel_id integer [not null, ref: > Hotel.hotel_id]
    full_name varchar(255) [not null]
    address varchar(255) [not null]
    SIN varchar(20) [not null, unique]
    role varchar(100) [not null]
}

Table Reservation {
    reservation_id integer [primary key]
    guest_id integer [not null, ref: > Guest.guest_id]
    room_id integer [not null, ref: > Room.room_id]
    start_date date [not null]
    end_date date [not null]
    status varchar(50) [not null, note: "Pending, Confirmed, Canceled"]
}

Table Rental {
    rental_id integer [primary key]
    reservation_id integer [not null, ref: > Reservation.reservation_id]
    guest_id integer [not null, ref: > Guest.guest_id]
    room_id integer [not null, ref: > Room.room_id]
    employee_id integer [not null, ref: > Employee.employee_id]
    checkin_date date [not null]
    checkout_date date [not null]
}
```

```

Table Archived_Records {
    record_id integer [primary key]
    original_id integer [not null, note: "References old reservation/rental"]
    type varchar(50) [not null, note: "Reservation or Rental"]
    data_snapshot text [not null]
    archived_at timestamp [not null]
}

Table Payment {
    payment_id integer [primary key]
    rental_id integer [not null, ref: > Rental.rental_id]
    amount decimal(10,2) [not null]
    payment_method varchar(50) [not null]
    payment_date timestamp [not null]
}
```

3 SQL Database Schema

```
-- Creating tables
CREATE TABLE HotelChain (
    chain_id SERIAL PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    central_address VARCHAR(255) NOT NULL,
    num_hotels INTEGER NOT NULL,
    contact_email VARCHAR(255) NOT NULL UNIQUE,
    telephone VARCHAR(20) NOT NULL
);

CREATE TABLE Hotel (
    hotel_id SERIAL PRIMARY KEY,
    chain_id INTEGER NOT NULL,
    name VARCHAR(255) NOT NULL,
    classification INTEGER NOT NULL CHECK (classification BETWEEN 1 AND 5),
    num_rooms INTEGER NOT NULL,
    address VARCHAR(255) NOT NULL,
    contact_email VARCHAR(255) NOT NULL UNIQUE,
    telephone VARCHAR(20) NOT NULL,
    FOREIGN KEY (chain_id) REFERENCES HotelChain(chain_id) ON DELETE CASCADE
);

CREATE TABLE Room (
    room_id SERIAL PRIMARY KEY,
    hotel_id INTEGER NOT NULL,
    price DECIMAL(10,2) NOT NULL CHECK (price >= 0),
    capacity INTEGER NOT NULL CHECK (capacity IN (1, 2, 3)),
    view VARCHAR(50) NOT NULL CHECK (view IN (’Sea View’, ’Mountain View’, ’City View’)),
    extendable BOOLEAN NOT NULL,
    condition VARCHAR(50) NOT NULL CHECK (condition IN (’Damaged’, ’In Repair’, ’Good Condition’)),
    amenities TEXT NOT NULL,
    FOREIGN KEY (hotel_id) REFERENCES Hotel(hotel_id) ON DELETE CASCADE
);

CREATE TABLE Guest (
    guest_id SERIAL PRIMARY KEY,
    full_name VARCHAR(255) NOT NULL,
    address VARCHAR(255) NOT NULL,
    SIN VARCHAR(20) NOT NULL UNIQUE CHECK (SIN ~ ’^\d{9}$’),
    date_of_checkin DATE NOT NULL
);

CREATE TABLE Employee (
    employee_id SERIAL PRIMARY KEY,
    hotel_id INTEGER NOT NULL,
    full_name VARCHAR(255) NOT NULL,
    address VARCHAR(255) NOT NULL,
    SIN VARCHAR(20) NOT NULL UNIQUE,
    role VARCHAR(100) NOT NULL CHECK (role IN ('Manager', 'Receptionist', 'Housekeeper')),
    FOREIGN KEY (hotel_id) REFERENCES Hotel(hotel_id) ON DELETE CASCADE
);

CREATE TABLE Reservation (
    reservation_id SERIAL PRIMARY KEY,
    guest_id INTEGER NOT NULL,
    room_id INTEGER NOT NULL,
    start_date DATE NOT NULL,
    end_date DATE NOT NULL,
    status VARCHAR(50) NOT NULL CHECK (status IN ('Booked', 'Checked-in', 'Cancelled')),
    FOREIGN KEY (guest_id) REFERENCES Guest(guest_id),
    FOREIGN KEY (room_id) REFERENCES Room(room_id)
);

CREATE TABLE Rental (
    rental_id SERIAL PRIMARY KEY,
    reservation_id INTEGER NOT NULL,
    guest_id INTEGER NOT NULL,
    room_id INTEGER NOT NULL,
    employee_id INTEGER NOT NULL,
    checkin_date DATE NOT NULL,
    checkout_date DATE NOT NULL,
    FOREIGN KEY (reservation_id) REFERENCES Reservation(reservation_id) ON DELETE CASCADE,
    FOREIGN KEY (guest_id) REFERENCES Guest(guest_id),
    FOREIGN KEY (room_id) REFERENCES Room(room_id),
    FOREIGN KEY (employee_id) REFERENCES Employee(employee_id)
);

CREATE TABLE Archived_Records (
    record_id SERIAL PRIMARY KEY,
    original_id INTEGER NOT NULL,
    type VARCHAR(50) NOT NULL CHECK (type IN ('Reservation', 'Rental')),
    data_snapshot TEXT NOT NULL,
    archived_at TIMESTAMP NOT NULL
);

CREATE TABLE Payment (
    payment_id SERIAL PRIMARY KEY,
    rental_id INTEGER NOT NULL,
    amount DECIMAL(10,2) NOT NULL CHECK (amount > 0),
    payment_method VARCHAR(50) NOT NULL CHECK (payment_method IN ('Credit Card', 'Debit Card', 'Cash', 'Bank Transfer')),
    payment_date TIMESTAMP NOT NULL,
    FOREIGN KEY (rental_id) REFERENCES Rental(rental_id) ON DELETE CASCADE
);
```

4 DBMS and Programming Languages Used

The application uses a combination of a relational database management system (RDBMS), PostgreSQL, and modern web development frameworks such as Express.js and Next.js, along with JavaScript and TypeScript. Below is a detailed breakdown of each of these technologies:

4.1 PostgreSQL (DBMS)

PostgreSQL is the relational database management system used in this application for storing and managing data. It is an open-source, object-relational database that supports complex queries and transactions.

4.2 Express.js (Backend Framework)

Express.js is a web framework for Node.js, providing a robust set of features for web and mobile applications. It is used here to build the RESTful API for interacting with the database.

4.3 Next.js (Frontend Framework)

Next.js is a React framework for building static and dynamic websites. In this project, Next.js is used to implement the user interface for interacting with the backend API.

4.4 JavaScript / TypeScript

- **JavaScript** is the primary language for the backend and frontend.
- **TypeScript**, a superset of JavaScript, is utilized to add static typing, making the code more maintainable, scalable, and less error-prone.

—

5 Installation Guide

This section provides the necessary steps to set up and run the application locally. You have two options:

5.1 Step 0: Install Software

You have two options:

1. Use Docker containers **This method is strongly recommended**

   ```
   docker build . -t "hotel-app"
   docker run -d -p 3000:3000 -p 5010:5010 hotel-app
   ```

   Continue to Step 5

2. Install on local machine

   - **Install PostgreSQL**: If PostgreSQL is not installed, download and install it from the official website.
   - **Install Node.js & npm**: If Node.js or npm is not installed, download and install it from the official website.

5.2 Step 1: Create PostgreSQL Database

1. Open a terminal and connect to the PostgreSQL command-line interface:

   ```
   psql -U postgres
   ```

2. Create a database called ’hotel’ and a user ’root’ with no password:

   ```
   CREATE DATABASE hotel;
   CREATE USER root WITH PASSWORD ’’;
   ALTER ROLE root SET client_encoding TO ’utf8’;
   ALTER ROLE root SET default_transaction_isolation TO ’read committed’;
   ALTER ROLE root SET timezone TO ’UTC’;
   GRANT ALL PRIVILEGES ON DATABASE hotel TO root;
   ```

3. Exit the PostgreSQL CLI by typing `\q`.

4. **Update Connection String (Optional)**: If you’re using a different database URI, update the `server.js` file:

   ```
   const connectionString = ’postgresql://root@localhost:5432/hotel’;
   ```

5.3 Step 2: Run SQL Setup Files

You have two options:

1. Setup using the dump file Use the database dump to quicky setup and prefill the database with demo data:
   (a) Navigate to the `project` folder:

   ```
   cd /path/to/project/
   ```

   (b) Setup Database (`database.sql`):

   ```
   psql -U root -d hotel -f ./sql/database.sql
   ```

2. Setup by configuration
   (a) Navigate to the `project` folder:

   ```
   cd /path/to/project/
   ```

   (b) Setup Tables (`setup.sql`):

   ```
   psql -U root -d hotel -f ./sql/setup.sql
   ```

   (c) Setup Triggers (`triggers.sql`):

   ```
   psql -U root -d hotel -f ./sql/triggers.sql
   ```

   (d) Setup Views (`views.sql`):

   ```
   psql -U root -d hotel -f ./sql/views.sql
   ```

   (e) Insert Sample Data (`sample.sql`):

   ```
   psql -U root -d hotel -f ./sql/sample.sql
   ```

5.4 Step 3: Setup and Start API Server

1. Navigate to the `project` folder:

   ```
   cd /path/to/project/
   ```

2. Install dependencies for the backend API:

   ```
   cd ./server
   npm install
   ```

3. Start the API server:

   ```
   node server.js
   ```

   The server should now be running on `http://localhost:5010` by default.

5.5 Step 4: Setup and Start User Client (Next.js)

1. Navigate to the `project` folder:

   ```
   cd /path/to/project/
   ```

2. Install dependencies for the Next.js client:

   ```
   cd app-v2
   npm install --force
   ```

3. Start the Next.js application:

   ```
   npm run dev
   ```

   The frontend should now be available at `http://localhost:3000`.

5.6 Step 5: Access the Web Application

1. Open your browser and navigate to `http://localhost:3000` to interact with the user interface.

User Guide: Once the app is running, you can navigate through the web application to perform operations such as room search, reservation creation, rental management, etc. Use the intuitive interface along with the help of the user guide to interact with the application.

UI Navigation & Usage Manual

6 Main Navigation

6.1 Navigation Bar

The main navigation bar is located at the top of every page and contains the following elements:

- **Logo**: Click on the ”E Hotel” logo to return to the homepage
- **Main Menu Links**:
  - Home: Navigate to the landing page
  - Search: Access the room search functionality
  - Hotels: Browse available hotels
  - My Bookings: View your current and past reservations
- **Mode Toggle**: Switch between ”Customer Mode” and ”Employee Mode”
- **Employee Portal**: Access to employee hotel management functions
- **Theme Toggle**: Switch between light and dark themes (Experimental)
- **User Menu**: To be implemented

6.2 Mobile

On smaller screens, the main menu collapses into a hamburger menu:

1. Tap the hamburger icon () in the top-left corner
2. A slide-out drawer will appear with all navigation options
3. Tap any menu item to navigate to that section
4. Tap outside the drawer or the X button to close it

7 Home Page

7.1 Main Section

- **Welcome Banner**: Displays the main promotional message
- **Quick Action Buttons**:
  - ”Search Rooms”: Directs to the search page
  - ”Employee Portal”: Access to employee & hotel management functions

7.2 Search Widget (Experimental)

The search widget allows for quick room searches:

1. **Date Selection**:
   - Click ”Check-in Date” to open a calendar picker
   - Click ”Check-out Date” to select your departure
2. **Guest Count**: Select the number of guests from the dropdown
3. **Location**: Enter your desired destination
4. **Search Button**: Click to view matching results

8 Search and Filtering

8.1 Advanced Search Page

The search page provides comprehensive filtering options:

1. **Left Sidebar Filters**:
   - Dates: Select check-in and check-out dates
   - Room Capacity: Choose the number of guests
   - Price Range: Adjust the minimum and maximum price using the slider
   - Hotel Chain: Filter by specific hotel chains
   - Room Features:
     - View: Filter by room view (Sea, Mountain, City)
     - Condition: Filter by room condition
     - Extendable: Toggle to show only extendable rooms
   - Amenities: Check specific amenities (WiFi, TV, etc.) (Experimental)

2. **Active Filters**:
   - View currently applied filters at the top of the results
   - Click the X next to any filter to remove it
   - Click ”Clear all” to reset all filters

3. **Search Results**:
   - Results display as cards with room information
   - Each card shows price, capacity, view, and key amenities
   - Click ”Reserve Room” to proceed with booking

8.2 Hotels Page

The Hotels page allows browsing by hotel rather than individual rooms:

1. **View Toggles**:
   - List View: Shows hotels in a list format
   - Map View: Displays hotels on a map (To be implemented)

2. **Hotel Filters**:
   - Similar to room filters but focused on hotel properties
   - Filter by classification (star rating), location, and amenities

3. **Hotel Cards**:
   - Each card shows the hotel name, location, and classification
   - Click ”View Details” to see rooms and more information
   - Click ”View Hotels” to see all properties in a chain

9 Room Details and Booking

9.1 Room Details Page

When viewing a specific room:

1. **Room Information**:
   - Photos of the room at the top
   - Room name and hotel information
   - Price per night displayed prominently

2. **Detail Tabs**:
   - Room Details: Basic information about the room
   - Amenities: List of available amenities
   - Policies: Cancellation policy and other rules

3. **Booking Widget**:
   - Located on the right side
   - Select dates for your stay
   - Choose number of guests
   - View total price calculation
   - Click ”Book Now” to proceed

9.2 Booking Process

To complete a booking:

1. **Reservation Modal**:
   - Confirm room details and dates
   - Enter guest information
   - Review total price
2. **Confirmation**:
   - After booking, you'll be redirected to a confirmation page
   - Booking details and reference number will be displayed
   - Options to view all bookings or return home

10 My Bookings

The My Bookings page allows you to manage your reservations:

1. **Booking Tabs**:
   - Upcoming: Shows future reservations
   - Past: Displays completed stays
   - Cancelled: Lists cancelled bookings
2. **Booking Cards**:
   - Each card shows booking details including dates, hotel, and price
   - For upcoming bookings, options to modify or cancel
   - For past bookings, option to book again

11 Employee Portal

The Employee Portal is accessible from the home page or navigation bar using a button:

11.1 Dashboard

The main dashboard provides an overview:

- **Quick Access Cards**:
  - Hotel Management
  - Rental Management
  - Guest Management
  - Employee Management
  - Payment Management
  - Archived Records
  - Room Availability
  - Room Capacity

11.2 Rental Management

For managing active rentals and reservations:

1. **Tabs**:
   - Active Rentals: Currently occupied rooms
   - Today's Check-outs: Guests checking out today
   - Pending Reservations: Upcoming reservations
   - Today's Check-ins: Guests checking in today
   - Payments: To be implemented
2. **Actions**:
   - Check In: Convert a reservation to an active rental
   - Check Out: Complete a rental and archive the record
   - Payment: Process payments for a rental

11.3 Guest Management

For managing guest information:

1. **Guest List**:
   - View all registered guests
   - Search by name, ID, or other criteria
2. **Actions**:
   - Add Guest: Create a new guest record
   - Edit: Modify guest information
   - Delete: Remove a guest record

11.4 Hotel Management

For managing hotel properties:

1. **Tabs**:
   - Hotel Chains: Manage parent companies
   - Hotels: Manage individual properties
   - Rooms: Manage specific rooms
2. **Actions**:
   - Add: Create new records
   - Edit: Modify existing information
   - Delete: Remove records

11.5 Archives

For viewing historical data:

1. **Filter Options**:
   - Record Type: Filter by reservation or rental
   - Search: Find specific records
2. **Record Details**:
   - View complete information about archived records
   - Access the original data snapshot

12 SQL Query Visualizer

The SQL Query Visualizer is a developer tool that captures and displays database queries:

12.1 Accessing the Visualizer

1. Click the database icon () on the right side of the screen
2. The sidebar will slide in from the right

12.2 Visualizer Features

1. **Query List**:
   - Queries are grouped by endpoint
   - Each query shows its type (SELECT, INSERT, etc.)
   - Click on a query to expand and see details
2. **Filtering Options**:
   - Search box to find specific queries
   - Tabs to filter by query type (SELECT, INSERT, UPDATE, DELETE)
3. **Query Details**:
   - SQL statement with syntax highlighting
   - Parameters used in the query
   - Execution time and row count
   - Timestamp of when the query was executed
4. **Actions**:
   - Copy: Copy the query to clipboard
   - Download: Export all queries as JSON
   - Clear: Remove all captured queries

13 Error Handling

13.1 Form Validation

When required information is missing:

1. Fields will be highlighted in red
2. Error messages will appear below the fields
3. Submit buttons will be disabled until errors are resolved

Please note that this is an experimental feature and in certain scenarios, no indication might be given.

13.2 Error Messages

When operations fail:

- Toast notifications will appear with error details

14 Use Cases

14.1 Booking a Room

1. Navigate to the Search page
2. Enter your desired location, dates, and guest count
3. Apply additional filters as needed
4. Browse the results and select a room
5. Review room details and policies
6. Enter guest information in the booking form
7. Confirm the reservation

14.2 Managing Reservations (Employee)

1. Switch to Employee Mode
2. Navigate to the Rental Management section
3. View pending reservations
4. Process check-ins for arriving guests
5. Handle payments
6. Process check-outs for departing guests
7. Archive completed rentals

14.3 SQL (Education/Developer)

1. Open the SQL Query Visualizer
2. Perform operations in the application
3. View captured queries
4. Analyze query structure
5. Export queries for further analysis

—

15 DDLs Used

15.1 Setup SQL Files

- **setup.sql**: This file contains DDLs for creating the necessary tables such as `Hotel`, `Room`, `Reservation`, `Rental`, `Payment`, etc.
  SQL Code
- **triggers.sql**: Defines triggers that are set to execute in response to specific actions, like data changes in the `Reservation` table.

```
-- Triggers:
-- Archive a Reservation When Deleted
CREATE OR REPLACE FUNCTION archive_reservation_on_delete()
RETURNS TRIGGER AS $$
BEGIN
    INSERT INTO Archived_Records (original_id, type, data_snapshot, archived_at)
    VALUES (OLD.reservation_id, 'Reservation',
           ROW(OLD.guest_id, OLD.room_id, OLD.start_date, OLD.end_date, OLD.status)::TEXT, NOW
    ());
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_archive_reservation_on_delete
AFTER DELETE ON Reservation
FOR EACH ROW
EXECUTE FUNCTION archive_reservation_on_delete();

-- Archive a Rental When Deleted
CREATE OR REPLACE FUNCTION archive_rental_on_delete()
RETURNS TRIGGER AS $$
BEGIN
    INSERT INTO Archived_Records (original_id, type, data_snapshot, archived_at)
    VALUES (OLD.rental_id, 'Rental',
           ROW(OLD.reservation_id, OLD.guest_id, OLD.room_id, OLD.employee_id, OLD.
    checkin_date, OLD.checkout_date)::TEXT, NOW());
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_archive_rental_on_delete
AFTER DELETE ON Rental
FOR EACH ROW
EXECUTE FUNCTION archive_rental_on_delete();

-- Prevent Deletion of Hotel Chain If Hotels Exist
CREATE OR REPLACE FUNCTION prevent_delete_hotel_chain()
RETURNS TRIGGER AS $$
BEGIN
    IF EXISTS (SELECT 1 FROM Hotel WHERE chain_id = OLD.chain_id) THEN
```

```
        RAISE EXCEPTION ’Cannot delete hotel chain with associated hotels.’;
    END IF;
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_prevent_delete_hotel_chain
BEFORE DELETE ON HotelChain
FOR EACH ROW
EXECUTE FUNCTION prevent_delete_hotel_chain();

-- Archive a Hotel When Deleted
CREATE OR REPLACE FUNCTION archive_hotel_on_delete()
RETURNS TRIGGER AS $$
BEGIN
    INSERT INTO Archived_Records (original_id, type, data_snapshot, archived_at)
    VALUES (OLD.hotel_id, ’Hotel’,
           ROW(OLD.chain_id, OLD.name, OLD.classification, OLD.num_rooms, OLD.address, OLD.contact_email, OLD.telephone)::TEXT, NOW());
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_archive_hotel_on_delete
AFTER DELETE ON Hotel
FOR EACH ROW
EXECUTE FUNCTION archive_hotel_on_delete();

-- Archive a Room When Deleted
CREATE OR REPLACE FUNCTION archive_room_on_delete()
RETURNS TRIGGER AS $$
BEGIN
    INSERT INTO Archived_Records (original_id, type, data_snapshot, archived_at)
    VALUES (OLD.room_id, ’Room’,
           ROW(OLD.hotel_id, OLD.price, OLD.capacity, OLD.view, OLD.extendable, OLD.condition, OLD.amenities)::TEXT, NOW());
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_archive_room_on_delete
AFTER DELETE ON Room
FOR EACH ROW
EXECUTE FUNCTION archive_room_on_delete();
```

- **views.sql**: This file defines views for simplifying complex queries such as room availability by view type and room capacity by hotel.

```
-- Views:
CREATE VIEW Available_Rooms_By_Area AS
SELECT h.address AS hotel_address, COUNT(r.room_id) AS available_rooms
FROM Hotel h
JOIN Room r ON h.hotel_id = r.hotel_id
WHERE r.condition = ’Good Condition’
GROUP BY h.address;

CREATE VIEW Room_Capacity_By_Hotel AS
SELECT hotel_id, capacity, COUNT(*) AS count
FROM Room
GROUP BY hotel_id, capacity
ORDER BY hotel_id, capacity;
```

- **sample.sql**: Inserts sample data into the tables for demonstration purposes.

```
INSERT INTO HotelChain (name, central_address, num_hotels, contact_email, telephone)
VALUES
(’Oceanic Hotels’, ’123 Ocean Ave, Miami, FL’, 8, ’contact@oceanic.com’, ’305-555-1234’),
```

```sql
('Mountain Resorts', '456 Mountain Rd, Denver, CO', 8, 'info@mountainresorts.com', '720-555-5678'),
('Cityscape Hotels', '789 City Blvd, New York, NY', 8, 'support@cityscape.com', '212-555-8765'),
('Luxury Escapes', '101 Luxury St, Los Angeles, CA', 8, 'service@luxuryescapes.com', '310-555-4321'),
('Budget Inns', '202 Budget Ave, Austin, TX', 8, 'contact@budgetinns.com', '512-555-9876');

-- Oceanic Hotels
INSERT INTO Hotel (chain_id, name, classification, num_rooms, address, contact_email, telephone)
VALUES
(11, 'Oceanic Bay Resort', 5, 10, '123 Ocean Bay, Miami, FL', 'contact@oceanicbay.com', '305-555-1001'),
(11, 'Oceanic Harbor Hotel', 4, 12, '456 Harbor St, Miami, FL', 'contact@oceanicharbor.com', '305-555-1002'),
(11, 'Oceanic Paradise', 5, 8, '789 Paradise Rd, Miami, FL', 'contact@oceanicparadise.com', '305-555-1003'),
(11, 'Oceanic Shores Inn', 3, 10, '101 Shores Blvd, Miami, FL', 'contact@oceanicshores.com', '305-555-1004'),
(11, 'Oceanic Breeze Resort', 4, 9, '102 Breeze Dr, Miami, FL', 'contact@oceanicbreeze.com', '305-555-1005'),
(11, 'Oceanic Sunset Retreat', 3, 6, '103 Sunset Blvd, Miami, FL', 'contact@oceanicsunset.com', '305-555-1006'),
(11, 'Oceanic Coastline', 4, 7, '104 Coastline Ave, Miami, FL', 'contact@oceaniccoastline.com', '305-555-1007'),
(11, 'Oceanic Vista Hotel', 5, 8, '105 Vista Rd, Miami, FL', 'contact@oceanicvista.com', '305-555-1008');

-- Mountain Resorts
INSERT INTO Hotel (chain_id, name, classification, num_rooms, address, contact_email, telephone)
VALUES
(12, 'Mountain Retreat', 5, 10, '123 Mountain Rd, Denver, CO', 'info@mountainretreat.com', '720-555-2001'),
(12, 'Alpine Summit Resort', 4, 12, '456 Summit Rd, Denver, CO', 'info@alpinesummit.com', '720-555-2002'),
(12, 'Rocky Peak Inn', 4, 8, '789 Peak Rd, Denver, CO', 'info@rockypeakinn.com', '720-555-2003'),
(12, 'Cedar Woods Lodge', 3, 7, '101 Cedar Ln, Denver, CO', 'info@cedarwoods.com', '720-555-2004'),
(12, 'Snowfall Lodge', 4, 9, '102 Snowfall Dr, Denver, CO', 'info@snowfalllodge.com', '720-555-2005'),
(12, 'Silver Pine Inn', 3, 6, '103 Pine Rd, Denver, CO', 'info@silverpineinn.com', '720-555-2006'),
(12, 'Summit View Hotel', 5, 11, '104 Summit View, Denver, CO', 'info@summitview.com', '720-555-2007'),
(12, 'Evergreen Resort', 5, 10, '105 Evergreen St, Denver, CO', 'info@evergreenresort.com', '720-555-2008');

-- Cityscape Hotels
INSERT INTO Hotel (chain_id, name, classification, num_rooms, address, contact_email, telephone)
VALUES
(13, 'Skyline Tower', 5, 15, '123 Skyline St, New York, NY', 'support@skylinetower.com', '212-555-3001'),
(13, 'Central Park Inn', 4, 10, '456 Park Ave, New York, NY', 'support@centralparkinn.com', '212-555-3002'),
(13, 'City Plaza Hotel', 4, 12, '789 Plaza Rd, New York, NY', 'support@cityplazahotel.com', '212-555-3003'),
(13, 'Downtown Luxury', 5, 9, '101 Downtown St, New York, NY', 'support@downtownluxury.com', '212-555-3004'),
(13, 'Grand City Hotel', 5, 11, '102 Grand Blvd, New York, NY', 'support@grandcityhotel.com', '212-555-3005'),
(13, 'Urban Heights Hotel', 3, 8, '103 Heights Ave, New York, NY', 'support@urbanheights.com', '212-555-3006'),
(13, 'Empire City Inn', 4, 10, '104 Empire Rd, New York, NY', 'support@empirecityinn.com', '212-555-3007'),
(13, 'Metropolitan Suites', 5, 13, '105 Metropolitan St, New York, NY', 'support@metropolitansuites.com', '212-555-3008');

-- Luxury Escapes
INSERT INTO Hotel (chain_id, name, classification, num_rooms, address, contact_email, telephone)
VALUES
(14, 'Golden Sands Resort', 5, 10, '123 Golden Sands St, Los Angeles, CA', 'service@goldensands.com', '310-555-4001'),
(14, 'Majestic Palms Hotel', 4, 12, '456 Majestic Blvd, Los Angeles, CA', 'service@majesticpalms.com', '310-555-4002'),
(14, 'Regal Heights Hotel', 5, 15, '789 Regal Ave, Los Angeles, CA', 'service@regalheights.com', '310-555-4003'),
(14, 'Opulent View Resort', 4, 11, '101 Opulent View Dr, Los Angeles, CA', 'service@opulentview.com', '310-555-4004'),
(14, 'Royal Oceanside', 5, 10, '102 Royal Blvd, Los Angeles, CA', 'service@royaloceanside.com', '310-555-4005'),
(14, 'Diamond Retreat', 5, 12, '103 Diamond Dr, Los Angeles, CA', 'service@diamondretreat.com', '310-555-4006'),
(14, 'Prestige Towers', 4, 9, '104 Prestige Rd, Los Angeles, CA', 'service@prestigetowers.com', '310-555-4007'),
(14, 'Crown Ridge Resort', 5, 8, '105 Crown Ridge, Los Angeles, CA', 'service@crownridge.com', '310-555-4008');

-- Budget Inns
INSERT INTO Hotel (chain_id, name, classification, num_rooms, address, contact_email, telephone)
VALUES
(15, 'Inexpensive Stay Inn', 2, 10, '123 Inexpensive St, Austin, TX', 'contact@inexpensivestay.com', '512-555-5001'),
(15, 'Budget Bunkhouse', 3, 12, '456 Budget Rd, Austin, TX', 'contact@budgetbunkhouse.com', '512-555-5002'),
(15, 'Cheap Suites Hotel', 2, 8, '789 Cheap Ave, Austin, TX', 'contact@cheapsuites.com', '512-555-5003'),
(15, 'Low-Cost Inn', 1, 6, '101 Low-Cost Dr, Austin, TX', 'contact@lowcostinn.com', '512-555-5004'),
(15, 'Affordable Rest', 3, 7, '102 Affordable St, Austin, TX', 'contact@affordablerest.com', '512-555-5005'),
(15, 'Economy Stay Inn', 2, 10, '103 Economy Ave, Austin, TX', 'contact@economystay.com', '512-555-5006'),
(15, 'Budget Beachfront Hotel', 3, 9, '104 Beachfront Rd, Austin, TX', 'contact@budgetbeachfront.com', '512-555-5007'),
(15, 'Value Park Inn', 3, 8, '105 Value Park St, Austin, TX', 'contact@valueparkinn.com', '512-555-5008');

-- Oceanic Bay Resort (hotel 1)
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities)
VALUES
(11, 200.00, 1, 'Sea View', TRUE, 'Good Condition', 'AC, TV, Wi-Fi'),
(11, 250.00, 2, 'Mountain View', TRUE, 'Good Condition', 'AC, TV, Wi-Fi, Mini Bar'),
(11, 300.00, 3, 'City View', FALSE, 'Good Condition', 'AC, TV, Wi-Fi, Balcony'),
(11, 180.00, 1, 'Sea View', TRUE, 'In Repair', 'AC, TV'),
(11, 150.00, 2, 'Mountain View', FALSE, 'Damaged', 'AC, TV');

-- Oceanic Harbor Hotel (hotel 2)
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities)
VALUES
(12, 220.00, 1, 'Mountain View', TRUE, 'Good Condition', 'AC, TV, Wi-Fi'),
(12, 270.00, 2, 'City View', TRUE, 'Good Condition', 'AC, TV, Wi-Fi, Mini Bar'),
(12, 330.00, 3, 'Sea View', FALSE, 'Good Condition', 'AC, TV, Wi-Fi, Balcony'),
(12, 200.00, 1, 'City View', TRUE, 'In Repair', 'AC, TV'),
(12, 180.00, 2, 'Mountain View', FALSE, 'Damaged', 'AC, TV');

-- Oceanic Paradise (hotel 3)
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities)
VALUES
(13, 210.00, 1, 'Sea View', TRUE, 'Good Condition', 'AC, TV, Wi-Fi'),
(13, 260.00, 2, 'Mountain View', TRUE, 'Good Condition', 'AC, TV, Wi-Fi, Mini Bar'),
(13, 290.00, 3, ’City View’, FALSE, ’Good Condition’, ’AC, TV, Wi-Fi, Balcony’),
(13, 190.00, 1, ’Sea View’, TRUE, ’In Repair’, ’AC, TV’),
(13, 170.00, 2, ’Mountain View’, FALSE, ’Damaged’, ’AC, TV’);

-- Oceanic Shores Inn (hotel 4)
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities)
VALUES
(14, 180.00, 1, ’Sea View’, TRUE, ’Good Condition’, ’AC, TV, Wi-Fi’),
(14, 230.00, 2, ’Mountain View’, TRUE, ’Good Condition’, ’AC, TV, Wi-Fi, Mini Bar’),
(14, 270.00, 3, ’City View’, FALSE, ’Good Condition’, ’AC, TV, Wi-Fi, Balcony’),
(14, 160.00, 1, ’Sea View’, TRUE, ’In Repair’, ’AC, TV’),
(14, 140.00, 2, ’Mountain View’, FALSE, ’Damaged’, ’AC, TV’);

-- Oceanic Breeze Resort (hotel 5)
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities)
VALUES
(15, 210.00, 1, ’Sea View’, TRUE, ’Good Condition’, ’AC, TV, Wi-Fi’),
(15, 260.00, 2, ’Mountain View’, TRUE, ’Good Condition’, ’AC, TV, Wi-Fi, Mini Bar’),
(15, 300.00, 3, ’City View’, FALSE, ’Good Condition’, ’AC, TV, Wi-Fi, Balcony’),
(15, 190.00, 1, ’Sea View’, TRUE, ’In Repair’, ’AC, TV’),
(15, 170.00, 2, ’Mountain View’, FALSE, ’Damaged’, ’AC, TV’);


INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (16, 184, 2, ’City View’, true, ’Damaged’, ’TV, Balcony, AC, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (16, 250, 1, ’Sea View’, false, ’In Repair’, ’Balcony, Mini Bar’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (16, 330, 1, ’Mountain View’, true, ’In Repair’, ’Wi-Fi, Balcony, Mini Bar’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (16, 152, 1, ’Mountain View’, false, ’Good Condition’, ’AC, Mini Bar, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (16, 242, 3, ’Mountain View’, false, ’Good Condition’, ’TV, Balcony’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (17, 196, 2, ’Mountain View’, true, ’In Repair’, ’Mini Bar, TV’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (17, 154, 1, ’City View’, false, ’Damaged’, ’Mini Bar, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (17, 151, 2, ’Mountain View’, true, ’In Repair’, ’AC, Wi-Fi, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (17, 269, 3, ’City View’, false, ’In Repair’, ’Balcony, AC, Mini Bar’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (17, 283, 3, ’City View’, true, ’Good Condition’, ’Wi-Fi, Mini Bar, TV, AC’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (18, 288, 2, ’City View’, true, ’Damaged’, ’Mini Bar, Wi-Fi, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (18, 279, 2, ’Sea View’, true, ’Damaged’, ’TV, AC, Mini Bar, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (18, 195, 2, ’Sea View’, true, ’Good Condition’, ’Mini Bar, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (18, 194, 3, ’Sea View’, false, ’In Repair’, ’Balcony, AC, TV’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (18, 201, 3, ’Sea View’, true, ’In Repair’, ’AC, Balcony’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (19, 165, 2, ’Mountain View’, true, ’Damaged’, ’TV, AC, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (19, 157, 3, ’City View’, false, ’Good Condition’, ’Wi-Fi, TV’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (19, 140, 3, ’Sea View’, false, ’Good Condition’, ’Balcony, AC, TV, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (19, 145, 1, ’Sea View’, false, ’Damaged’, ’Wi-Fi, Mini Bar, AC, TV’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (19, 192, 3, ’City View’, true, ’Damaged’, ’Balcony, TV’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
```

```
    (20, 174, 1, ’Sea View’, false, ’In Repair’, ’Balcony, AC, Wi-Fi, Mini Bar’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (20, 292, 2, ’City View’, true, ’Damaged’, ’AC, Wi-Fi, Balcony, TV’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (20, 276, 3, ’City View’, true, ’Damaged’, ’Balcony, TV’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (20, 283, 1, ’Sea View’, true, ’Good Condition’, ’Balcony, Mini Bar, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (20, 271, 3, ’Sea View’, false, ’Damaged’, ’TV, Balcony, AC, Mini Bar’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (21, 308, 2, ’Mountain View’, false, ’In Repair’, ’Wi-Fi, Balcony, AC’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (21, 247, 2, ’Sea View’, false, ’Damaged’, ’Balcony, Mini Bar’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (21, 249, 1, ’City View’, true, ’In Repair’, ’Mini Bar, Balcony, TV, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (21, 270, 3, ’Mountain View’, false, ’Good Condition’, ’Mini Bar, Balcony, AC, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (21, 142, 3, ’Mountain View’, true, ’Damaged’, ’Balcony, Mini Bar, TV, AC’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (22, 263, 3, ’City View’, true, ’Good Condition’, ’TV, Balcony, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (22, 241, 2, ’Mountain View’, false, ’Good Condition’, ’Balcony, TV’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (22, 271, 3, ’City View’, false, ’Damaged’, ’Balcony, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (22, 145, 3, ’Sea View’, false, ’In Repair’, ’Balcony, AC, TV’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (22, 275, 2, ’City View’, false, ’Damaged’, ’AC, Wi-Fi, Mini Bar, TV’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (23, 248, 3, ’City View’, true, ’Good Condition’, ’Mini Bar, AC’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (23, 151, 1, ’Mountain View’, true, ’In Repair’, ’Mini Bar, TV’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (23, 180, 3, ’Sea View’, true, ’In Repair’, ’TV, AC, Balcony, Mini Bar’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (23, 197, 1, ’Mountain View’, false, ’Good Condition’, ’Mini Bar, AC’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (23, 292, 2, ’City View’, true, ’Damaged’, ’Mini Bar, AC, Wi-Fi, Balcony’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (24, 142, 1, ’City View’, false, ’Damaged’, ’Wi-Fi, Balcony, Mini Bar, AC’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (24, 191, 1, ’Sea View’, false, ’Good Condition’, ’Wi-Fi, Mini Bar, TV, AC’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (24, 272, 1, ’Mountain View’, true, ’Good Condition’, ’Wi-Fi, TV, AC, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (24, 159, 2, ’City View’, true, ’Good Condition’, ’Mini Bar, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (24, 179, 2, ’City View’, true, ’Damaged’, ’AC, Wi-Fi, Mini Bar, TV’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (25, 288, 3, ’City View’, true, ’Good Condition’, ’AC, Balcony, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (25, 267, 3, ’City View’, true, ’Damaged’, ’Mini Bar, Wi-Fi, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (25, 245, 2, ’Mountain View’, true, ’Good Condition’, ’Wi-Fi, Mini Bar’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (25, 248, 3, ’Sea View’, true, ’Good Condition’, ’Wi-Fi, AC, TV, Mini Bar’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (25, 311, 1, ’Sea View’, false, ’In Repair’, ’TV, AC’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (26, 173, 2, ’City View’, false, ’Damaged’, ’TV, Mini Bar, AC’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
```

```
    (26, 275, 2, ’Mountain View’, true, ’Damaged’, ’AC, TV, Balcony, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (26, 151, 1, ’Mountain View’, false, ’Good Condition’, ’Wi-Fi, Mini Bar, AC’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (26, 311, 2, ’City View’, false, ’Damaged’, ’Mini Bar, AC, Wi-Fi, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (26, 266, 1, ’Sea View’, true, ’Good Condition’, ’Balcony, AC, TV, Mini Bar’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (27, 224, 2, ’Sea View’, false, ’Damaged’, ’Balcony, TV, Mini Bar, AC’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (27, 169, 2, ’Mountain View’, false, ’In Repair’, ’AC, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (27, 144, 2, ’Sea View’, true, ’Good Condition’, ’TV, AC’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (27, 324, 2, ’Sea View’, true, ’Damaged’, ’TV, Mini Bar, AC, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (27, 167, 2, ’Sea View’, true, ’In Repair’, ’Wi-Fi, Balcony, AC’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (28, 183, 2, ’Sea View’, true, ’Damaged’, ’Mini Bar, TV, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (28, 288, 3, ’Sea View’, true, ’In Repair’, ’Wi-Fi, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (28, 172, 3, ’Mountain View’, true, ’Damaged’, ’Wi-Fi, TV’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (28, 282, 2, ’Sea View’, true, ’In Repair’, ’Wi-Fi, Mini Bar, TV’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (28, 189, 2, ’Sea View’, true, ’Good Condition’, ’AC, Wi-Fi’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (29, 304, 3, ’City View’, false, ’Good Condition’, ’TV, AC, Mini Bar’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (29, 319, 1, ’City View’, false, ’In Repair’, ’TV, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (29, 324, 2, ’Sea View’, false, ’Damaged’, ’Mini Bar, Wi-Fi, AC, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (29, 313, 2, ’Sea View’, true, ’In Repair’, ’TV, Mini Bar, Balcony, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (29, 277, 3, ’Mountain View’, false, ’In Repair’, ’Wi-Fi, Mini Bar’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (30, 157, 3, ’Sea View’, false, ’Good Condition’, ’TV, AC’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (30, 144, 1, ’City View’, true, ’Damaged’, ’TV, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (30, 233, 1, ’City View’, false, ’Good Condition’, ’AC, Mini Bar’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (30, 259, 3, ’City View’, false, ’Good Condition’, ’AC, Mini Bar, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (30, 218, 3, ’City View’, false, ’In Repair’, ’TV, Balcony, Wi-Fi, AC’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (31, 143, 2, ’Sea View’, false, ’Good Condition’, ’Wi-Fi, AC’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (31, 308, 3, ’City View’, false, ’Good Condition’, ’Balcony, TV, Mini Bar’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (31, 209, 3, ’City View’, false, ’Good Condition’, ’Wi-Fi, AC’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (31, 166, 1, ’Mountain View’, true, ’Good Condition’, ’Balcony, Mini Bar, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (31, 198, 1, ’City View’, false, ’Damaged’, ’Wi-Fi, Mini Bar, AC’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (32, 217, 2, ’Sea View’, true, ’Good Condition’, ’Balcony, Wi-Fi, AC’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (32, 208, 3, ’Sea View’, false, ’Good Condition’, ’Balcony, AC, Mini Bar, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
```

```
    (32, 164, 2, 'City View', true, 'Good Condition', 'AC, TV, Wi-Fi, Balcony');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (32, 270, 2, 'Sea View', true, 'Good Condition', 'Wi-Fi, AC, TV');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (32, 192, 3, 'City View', false, 'In Repair', 'AC, TV, Balcony, Wi-Fi');

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (33, 157, 2, 'City View', true, 'Damaged', 'TV, Wi-Fi, Balcony');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (33, 228, 2, 'Sea View', true, 'Damaged', 'TV, Balcony, Mini Bar, Wi-Fi');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (33, 247, 1, 'Mountain View', true, 'Damaged', 'Balcony, Mini Bar, AC');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (33, 242, 1, 'Mountain View', true, 'In Repair', 'Balcony, Wi-Fi, Mini Bar, TV');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (33, 314, 2, 'City View', false, 'Damaged', 'Mini Bar, Wi-Fi, TV, Balcony');

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (34, 282, 2, 'Mountain View', false, 'In Repair', 'Balcony, AC, Wi-Fi, Mini Bar');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (34, 281, 2, 'Mountain View', true, 'Good Condition', 'Mini Bar, Balcony');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (34, 308, 2, 'City View', false, 'Good Condition', 'TV, Mini Bar, Wi-Fi');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (34, 223, 1, 'Sea View', true, 'Damaged', 'Wi-Fi, AC, TV');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (34, 176, 3, 'City View', false, 'Damaged', 'Balcony, Mini Bar, Wi-Fi');

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (35, 278, 1, 'Mountain View', false, 'Good Condition', 'TV, Balcony, AC');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (35, 221, 2, 'City View', true, 'Damaged', 'TV, AC');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (35, 168, 2, 'Mountain View', false, 'In Repair', 'Balcony, TV, Mini Bar');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (35, 293, 2, 'Sea View', true, 'In Repair', 'AC, Wi-Fi, Balcony');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (35, 210, 3, 'City View', true, 'In Repair', 'Mini Bar, TV, AC, Balcony');

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (36, 298, 3, 'Mountain View', false, 'Damaged', 'Mini Bar, Balcony, TV, Wi-Fi');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (36, 157, 2, 'Mountain View', true, 'In Repair', 'Mini Bar, AC, Wi-Fi');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (36, 164, 1, 'Sea View', true, 'Damaged', 'AC, Mini Bar, Wi-Fi');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (36, 248, 2, 'City View', false, 'Damaged', 'TV, Mini Bar');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (36, 155, 1, 'Mountain View', true, 'Damaged', 'Mini Bar, Wi-Fi, Balcony');

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (37, 277, 3, 'Sea View', false, 'In Repair', 'AC, TV, Mini Bar, Balcony');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (37, 178, 2, 'City View', false, 'In Repair', 'TV, Wi-Fi, AC, Mini Bar');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (37, 150, 2, 'Mountain View', false, 'In Repair', 'Balcony, Mini Bar');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (37, 253, 1, 'City View', false, 'Damaged', 'Balcony, AC, TV');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (37, 167, 3, 'Mountain View', true, 'In Repair', 'Wi-Fi, AC, Balcony');

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (38, 317, 2, 'Mountain View', false, 'In Repair', 'Mini Bar, AC, Wi-Fi');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (38, 273, 3, 'Sea View', false, 'Damaged', 'Mini Bar, Wi-Fi, AC, TV');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (38, 142, 1, 'Mountain View', false, 'Damaged', 'Balcony, AC, Wi-Fi');
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
```

```
    (38, 194, 2, ’Mountain View’, false, ’Damaged’, ’AC, Wi-Fi, Mini Bar, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (38, 271, 1, ’Sea View’, true, ’In Repair’, ’AC, TV, Balcony, Mini Bar’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (39, 196, 2, ’City View’, false, ’Good Condition’, ’Wi-Fi, AC, Mini Bar, TV’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (39, 194, 3, ’Mountain View’, false, ’In Repair’, ’TV, AC, Mini Bar, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (39, 278, 3, ’Mountain View’, false, ’Good Condition’, ’AC, TV, Mini Bar, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (39, 158, 1, ’City View’, true, ’In Repair’, ’Wi-Fi, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (39, 295, 1, ’Mountain View’, true, ’In Repair’, ’AC, Balcony’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (40, 144, 3, ’City View’, true, ’Damaged’, ’Balcony, TV, Mini Bar’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (40, 312, 1, ’City View’, true, ’In Repair’, ’Wi-Fi, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (40, 244, 1, ’Mountain View’, false, ’Good Condition’, ’Wi-Fi, AC, TV, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (40, 289, 3, ’City View’, true, ’Good Condition’, ’TV, Wi-Fi, Mini Bar’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (40, 239, 3, ’Sea View’, false, ’Damaged’, ’TV, Mini Bar, AC’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (41, 289, 3, ’Mountain View’, true, ’Good Condition’, ’Mini Bar, Wi-Fi, AC, TV’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (41, 164, 2, ’Mountain View’, false, ’Good Condition’, ’TV, Balcony, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (41, 279, 2, ’City View’, true, ’In Repair’, ’Wi-Fi, TV, AC, Mini Bar’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (41, 277, 2, ’Mountain View’, false, ’Good Condition’, ’Mini Bar, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (41, 169, 1, ’City View’, true, ’In Repair’, ’Wi-Fi, AC’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (42, 272, 1, ’Mountain View’, false, ’Good Condition’, ’Balcony, Wi-Fi, AC, TV’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (42, 261, 1, ’City View’, true, ’In Repair’, ’Wi-Fi, Balcony, AC’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (42, 297, 3, ’Sea View’, false, ’Damaged’, ’AC, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (42, 295, 1, ’City View’, false, ’Good Condition’, ’AC, TV, Wi-Fi, Mini Bar’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (42, 258, 2, ’Mountain View’, true, ’In Repair’, ’Mini Bar, Wi-Fi, TV, Balcony’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (43, 213, 2, ’Sea View’, false, ’In Repair’, ’AC, Balcony, TV’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (43, 258, 2, ’Sea View’, false, ’Good Condition’, ’Mini Bar, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (43, 291, 3, ’Sea View’, false, ’In Repair’, ’Wi-Fi, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (43, 163, 2, ’Sea View’, false, ’In Repair’, ’Balcony, TV, Mini Bar, AC’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (43, 273, 2, ’Mountain View’, false, ’Good Condition’, ’AC, Mini Bar, Wi-Fi’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (44, 187, 3, ’City View’, false, ’Damaged’, ’Wi-Fi, AC’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (44, 286, 3, ’Sea View’, false, ’In Repair’, ’TV, Wi-Fi, Mini Bar, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (44, 223, 2, ’Sea View’, false, ’Good Condition’, ’Balcony, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (44, 236, 2, ’Mountain View’, false, ’In Repair’, ’TV, AC, Wi-Fi, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
```

```
    (44, 223, 3, ’Mountain View’, false, ’Damaged’, ’TV, Mini Bar, AC, Wi-Fi’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (45, 151, 2, ’City View’, false, ’In Repair’, ’Balcony, TV’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (45, 159, 2, ’City View’, true, ’Good Condition’, ’Wi-Fi, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (45, 256, 2, ’Sea View’, false, ’In Repair’, ’AC, Wi-Fi, Mini Bar’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (45, 275, 2, ’Sea View’, true, ’In Repair’, ’Balcony, AC, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (45, 169, 3, ’Mountain View’, true, ’In Repair’, ’Mini Bar, TV, AC, Wi-Fi’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (46, 304, 2, ’City View’, true, ’Damaged’, ’TV, Mini Bar, AC, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (46, 230, 2, ’City View’, true, ’Good Condition’, ’AC, Mini Bar’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (46, 161, 1, ’Sea View’, true, ’Damaged’, ’Wi-Fi, Balcony, Mini Bar, TV’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (46, 183, 3, ’Sea View’, true, ’Damaged’, ’AC, Mini Bar, TV’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (46, 269, 3, ’Sea View’, true, ’Damaged’, ’Balcony, Wi-Fi, Mini Bar’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (47, 306, 1, ’City View’, true, ’Good Condition’, ’Wi-Fi, AC’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (47, 252, 2, ’City View’, true, ’In Repair’, ’AC, Mini Bar’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (47, 276, 2, ’City View’, true, ’In Repair’, ’AC, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (47, 195, 1, ’City View’, true, ’Good Condition’, ’AC, Wi-Fi, TV’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (47, 321, 1, ’Mountain View’, true, ’In Repair’, ’Balcony, TV, Wi-Fi, AC’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (48, 251, 1, ’City View’, false, ’In Repair’, ’Wi-Fi, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (48, 151, 3, ’Sea View’, false, ’In Repair’, ’TV, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (48, 264, 1, ’Sea View’, false, ’In Repair’, ’Balcony, Wi-Fi, Mini Bar’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (48, 245, 1, ’Mountain View’, false, ’Good Condition’, ’Wi-Fi, Mini Bar, AC’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (48, 316, 1, ’Sea View’, true, ’In Repair’, ’Balcony, Mini Bar, AC, TV’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (49, 215, 1, ’Mountain View’, true, ’In Repair’, ’AC, TV, Wi-Fi, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (49, 175, 3, ’Sea View’, true, ’In Repair’, ’Balcony, TV, Mini Bar, AC’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (49, 271, 1, ’Mountain View’, false, ’Good Condition’, ’Mini Bar, Balcony, AC’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (49, 158, 1, ’City View’, true, ’In Repair’, ’Wi-Fi, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (49, 205, 1, ’Sea View’, true, ’In Repair’, ’AC, Balcony, Wi-Fi, TV’);

INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (50, 228, 3, ’Mountain View’, true, ’In Repair’, ’Balcony, AC, TV’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (50, 149, 2, ’Mountain View’, true, ’Good Condition’, ’Mini Bar, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (50, 236, 2, ’Sea View’, false, ’In Repair’, ’Wi-Fi, Mini Bar, AC, Balcony’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (50, 145, 2, ’Mountain View’, false, ’Damaged’, ’Mini Bar, Wi-Fi’);
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities) VALUES
    (50, 306, 2, ’Mountain View’, true, ’Good Condition’, ’Balcony, TV, Wi-Fi, AC’);

```

```
-- Oceanic Bay Resort (guests)
INSERT INTO Guest (full_name, address, SIN, date_of_checkin)
VALUES
(’John Doe’, ’123 Ocean Ave, Miami, FL’, ’123456789’, ’2025-03-01’),
(’Jane Smith’, ’456 Coral St, Miami, FL’, ’987654321’, ’2025-03-02’);

-- Mountain Retreat (guests)
INSERT INTO Guest (full_name, address, SIN, date_of_checkin)
VALUES
(’Alice Johnson’, ’123 Mountain Rd, Denver, CO’, ’111223344’, ’2025-03-03’),
(’Bob Brown’, ’456 Aspen Ln, Denver, CO’, ’112233445’, ’2025-03-04’);

-- Skyline Tower (guests)
INSERT INTO Guest (full_name, address, SIN, date_of_checkin)
VALUES
(’Charlie White’, ’789 Broadway, New York, NY’, ’223344556’, ’2025-03-05’),
(’David Green’, ’101 5th Ave, New York, NY’, ’334455667’, ’2025-03-06’);

-- Golden Sands Resort (guests)
INSERT INTO Guest (full_name, address, SIN, date_of_checkin)
VALUES
(’Eva Davis’, ’123 Sunset Blvd, Los Angeles, CA’, ’445566778’, ’2025-03-07’),
(’Frank Miller’, ’456 Palm St, Los Angeles, CA’, ’556677889’, ’2025-03-08’);

-- Inexpensive Stay Inn (guests)
INSERT INTO Guest (full_name, address, SIN, date_of_checkin)
VALUES
(’Grace Wilson’, ’123 Low-Cost Rd, Austin, TX’, ’667788990’, ’2025-03-09’),
(’Henry Lee’, ’789 Budget Ln, Austin, TX’, ’778899001’, ’2025-03-10’);

-- Oceanic Bay Resort (employees)
INSERT INTO Employee (hotel_id, full_name, address, SIN, role)
VALUES
(11, ’Alice Fisher’, ’123 Ocean Ave, Miami, FL’, ’223344556’, ’Manager’),
(11, ’Bob Martinez’, ’456 Coral St, Miami, FL’, ’334455667’, ’Receptionist’);

-- Mountain Retreat (employees)
INSERT INTO Employee (hotel_id, full_name, address, SIN, role)
VALUES
(12, ’Carla Wood’, ’123 Mountain Rd, Denver, CO’, ’445566778’, ’Manager’),
(12, ’Dan Walker’, ’456 Aspen Ln, Denver, CO’, ’556677889’, ’Housekeeper’);

-- Skyline Tower (employees)
INSERT INTO Employee (hotel_id, full_name, address, SIN, role)
VALUES
(13, ’Eve Green’, ’789 Broadway, New York, NY’, ’667788990’, ’Manager’),
(13, ’Frank Harris’, ’101 5th Ave, New York, NY’, ’778899001’, ’Receptionist’);

-- Golden Sands Resort (employees)
INSERT INTO Employee (hotel_id, full_name, address, SIN, role)
VALUES
(14, ’Grace Brown’, ’123 Sunset Blvd, Los Angeles, CA’, ’889900112’, ’Manager’),
(14, ’Harry Wilson’, ’456 Palm St, Los Angeles, CA’, ’990011223’, ’Housekeeper’);

-- Inexpensive Stay Inn (employees)
INSERT INTO Employee (hotel_id, full_name, address, SIN, role)
VALUES
(15, ’Isaac Walker’, ’123 Low-Cost Rd, Austin, TX’, ’112233445’, ’Manager’),
(15, ’Jack Black’, ’789 Budget Ln, Austin, TX’, ’223344559’, ’Receptionist’);

-- Oceanic Bay Resort (reservations)
INSERT INTO Reservation (guest_id, room_id, start_date, end_date, status)
VALUES
(1, 21, ’2025-03-01’, ’2025-03-03’, ’Booked’),
(2, 22, ’2025-03-02’, ’2025-03-04’, ’Booked’);

-- Mountain Retreat (reservations)
INSERT INTO Reservation (guest_id, room_id, start_date, end_date, status)
```

```
VALUES
(3, 23, ’2025-03-03’, ’2025-03-05’, ’Booked’),
(4, 24, ’2025-03-04’, ’2025-03-06’, ’Booked’);

-- Skyline Tower (reservations)
INSERT INTO Reservation (guest_id, room_id, start_date, end_date, status)
VALUES
(5, 25, ’2025-03-05’, ’2025-03-07’, ’Booked’),
(6, 26, ’2025-03-06’, ’2025-03-08’, ’Booked’);

-- Golden Sands Resort (reservations)
INSERT INTO Reservation (guest_id, room_id, start_date, end_date, status)
VALUES
(7, 27, ’2025-03-07’, ’2025-03-09’, ’Booked’),
(8, 28, ’2025-03-08’, ’2025-03-10’, ’Booked’);

-- Inexpensive Stay Inn (reservations)
INSERT INTO Reservation (guest_id, room_id, start_date, end_date, status)
VALUES
(9, 29, ’2025-03-09’, ’2025-03-11’, ’Booked’),
(10, 30, ’2025-03-10’, ’2025-03-12’, ’Booked’);

-- Oceanic Bay Resort (rentals)
INSERT INTO Rental (reservation_id, guest_id, room_id, employee_id, checkin_date, checkout_date)
VALUES
(1, 1, 1, 1, ’2025-03-01’, ’2025-03-03’),
(2, 2, 2, 2, ’2025-03-02’, ’2025-03-04’);

-- Mountain Retreat (rentals)
INSERT INTO Rental (reservation_id, guest_id, room_id, employee_id, checkin_date, checkout_date)
VALUES
(3, 3, 3, 3, ’2025-03-03’, ’2025-03-05’),
(4, 4, 4, 4, ’2025-03-04’, ’2025-03-06’);

-- Skyline Tower (rentals)
INSERT INTO Rental (reservation_id, guest_id, room_id, employee_id, checkin_date, checkout_date)
VALUES
(5, 5, 5, 5, ’2025-03-05’, ’2025-03-07’),
(6, 6, 6, 6, ’2025-03-06’, ’2025-03-08’);

-- Golden Sands Resort (rentals)
INSERT INTO Rental (reservation_id, guest_id, room_id, employee_id, checkin_date, checkout_date)
VALUES
(7, 7, 7, 7, ’2025-03-07’, ’2025-03-09’),
(8, 8, 8, 8, ’2025-03-08’, ’2025-03-10’);

-- Inexpensive Stay Inn (rentals)
INSERT INTO Rental (reservation_id, guest_id, room_id, employee_id, checkin_date, checkout_date)
VALUES
(9, 9, 9, 9, ’2025-03-09’, ’2025-03-11’),
(10, 10, 10, 10, ’2025-03-10’, ’2025-03-12’);

CREATE INDEX idx_room_price ON Room(price);
CREATE INDEX idx_reservation_status ON Reservation(status);
CREATE INDEX idx_hotel_name ON Hotel(name);
```

15.2 API DDLs (SQL Endpoints)

The following API endpoints are used to manage various aspects of the hotel management system.

15.2.1 Room Management

- **Get all rooms**:

```
SELECT Room.*, Hotel.name AS
hotel_name, Hotel.classification AS hotel_classification, HotelChain.name AS chain_name
FROM Room
JOIN Hotel ON Room.hotel_id = Hotel.hotel_id
LEFT JOIN HotelChain ON Hotel.chain_id = HotelChain.chain_id
WHERE 1=1 [+ additional filters]
```

- **Create a new room**:

```
INSERT INTO Room (hotel_id, price, capacity, view, extendable, condition, amenities)
VALUES ($1, $2, $3, $4, $5, $6, $7) RETURNING room_id
```

- **Delete a room**:

```
DELETE FROM Room WHERE room_id = $1 RETURNING *
```

15.2.2 Reservation Management

- **Create a new reservation**:

```
INSERT INTO Reservation (guest_id, room_id, start_date, end_date, status)
VALUES ($1, $2, $3, $4, $5) RETURNING *
```

- **Get all reservations**:

```
SELECT * FROM Reservation
```

- **Delete a reservation**:

```
DELETE FROM Reservation WHERE reservation_id = $1 RETURNING *
```

15.2.3 Rental Management

- **Convert reservation to rental**:

```
INSERT INTO Rental (reservation_id, guest_id, room_id, employee_id, checkin_date, checkout_date)
SELECT reservation_id, guest_id, room_id, $1, $2, $3 FROM Reservation WHERE reservation_id = $4
```

- **Get all rentals with associated payments**:

```
SELECT rental.*, room.price AS totalamount, COALESCE(json_agg(...)) AS payments
FROM Rental rental
JOIN Room room ON room.room_id = rental.room_id
LEFT JOIN Payment payment ON payment.rental_id = rental.rental_id
GROUP BY rental.rental_id, room.price
```

- **Delete a rental**:

```
DELETE FROM Rental WHERE rental_id = $1 RETURNING *
```

15.2.4 Hotel Chain Management

- **Create a new hotel chain**:

```
INSERT INTO HotelChain (name, central_address, num_hotels, contact_email, telephone)
VALUES ($1, $2, $3, $4, $5) RETURNING *
```

- **Update a hotel chain**:

```
UPDATE HotelChain SET
name = $1, central_address = $2, num_hotels = $3, contact_email = $4, telephone = $5
WHERE chain_id = $6 RETURNING *
```

- **Delete a hotel chain**:

```
DELETE FROM HotelChain WHERE chain_id = $1
```

15.2.5 Hotel Management

- **Get all hotels**:

```
SELECT * FROM Hotel;
```

- **Get all hotel addresses**:

```
SELECT address FROM Hotel;
```

- **Create a new hotel**:

```
INSERT INTO Hotel (chain_id, name, classification, num_rooms, address, contact_email, telephone)
VALUES ($1, $2, $3, $4, $5, $6, $7) RETURNING *;
```

- **Delete a hotel**:

```
DELETE FROM Hotel WHERE hotel_id = $1 RETURNING *;
```

15.2.6 Payment Management

- **Create a payment**:

```
INSERT INTO Payment (rental_id, amount, payment_method, payment_date)
VALUES ($1, $2, $3, $4) RETURNING payment_id
```

- **Get all payments for a rental**:

```
SELECT * FROM Payment WHERE rental_id = $1;
```

- **Delete a payment**:

```
DELETE FROM Payment WHERE payment_id = $1 RETURNING *;
```

15.2.7 Guest Management

- **Create a new guest**:

```
INSERT INTO Guest (first_name, last_name, email, phone_number, address)
VALUES ($1, $2, $3, $4, $5) RETURNING guest_id;
```

- **Get all guests**:

```
SELECT * FROM Guest;
```

- **Delete a guest**:

```
DELETE FROM Guest WHERE guest_id = $1 RETURNING *;
```

- **Update a guest's details**:

```
UPDATE Guest SET first_name = $1, last_name = $2, email = $3, phone_number = $4, address = $5
WHERE guest_id = $6 RETURNING *;
```

15.2.8 Employee Management

- **Create a new employee**:

```
INSERT INTO Employee (first_name, last_name, position, hire_date, salary)
VALUES ($1, $2, $3, $4, $5) RETURNING employee_id;
```

- **Get all employees**:

```
SELECT * FROM Employee;
```

- **Delete an employee**:

```
DELETE FROM Employee WHERE employee_id = $1 RETURNING *;
```

- **Update an employee's details**:

```
UPDATE Employee SET first_name = $1, last_name = $2, position = $3, hire_date = $4, salary = $5
WHERE employee_id = $6 RETURNING *;
```

15.2.9 Hotel Amenities Management

- **Create a new amenity**:

```
INSERT INTO Amenity (name, description)
VALUES ($1, $2) RETURNING amenity_id;
```

- **Get all amenities**:

```
SELECT * FROM Amenity;
```

- **Delete an amenity**:

```
DELETE FROM Amenity WHERE amenity_id = $1 RETURNING *;
```

15.2.10 Room Amenity Assignment

- **Assign an amenity to a room**:

```
INSERT INTO Room_Amenity (room_id, amenity_id)
VALUES ($1, $2) RETURNING *;
```

- **Remove an amenity from a room**:

```
DELETE FROM Room_Amenity WHERE room_id = $1 AND amenity_id = $2 RETURNING *;
```

15.2.11 Hotel Statistics

- **Get the average price of rooms by hotel**:

```
SELECT Hotel.name, AVG(Room.price) AS average_price
FROM Hotel
JOIN Room ON Room.hotel_id = Hotel.hotel_id
GROUP BY Hotel.hotel_id;
```

- **Get the total number of rooms by hotel**:

```
SELECT Hotel.name, COUNT(Room.room_id) AS total_rooms
FROM Hotel
JOIN Room ON Room.hotel_id = Hotel.hotel_id
GROUP BY Hotel.hotel_id;
```

- **Get the total revenue from rentals by hotel**:

```
SELECT Hotel.name, SUM(Room.price) AS total_revenue
FROM Rental
JOIN Room ON Rental.room_id = Room.room_id
JOIN Hotel ON Room.hotel_id = Hotel.hotel_id
GROUP BY Hotel.hotel_id;
```