

# **HOTEL DATABASE**

**TERM PROJECT**

**SUBMITTED BY:**

**NAVJOT SINGH**

**(0758059)**



**AMOD-5450H-A-S01-2023GS-PTBO**

**INTRODUCTION TO DATABASES**

**TRENT UNIVERSITY**

**AUGUST, 2023**

## I. Conceptual Model

We have identified some entities for the hotel database along with the relationships among them. A brief description of the data model is given below.

- Employees are identified using their employee id (emp\_id), and their names, office location, and joining date (working\_since) are recorded. An employee can manage multiple properties. Some employees (associates) report to other employees (manager).
- A property is identified by the property id (prop\_id). For each property, its name and address are also stored. Each property is managed by exactly one employee. Properties are specialized into subclasses, namely hotels and homes. Number of rooms (n\_rooms) in a hotel as well as the number of beds (n\_beds) and host for a home are also recorded.
- Customers are identified by their customer id (cust\_id). The name, email id and contact number of customers are recorded. Customers can book a property for a specified date range (start\_date and end\_date).
- Each property offers multiple amenities. Each amenity can be offered by multiple properties. An amenity is identified by its name, and is described using some other attributes like the type of amenity (amenity\_type) and whether it is considered luxurious or not (is\_luxurious).
- Properties can apply restrictions. Restrictions are identified using the restriction description (restriction\_desc) along with the property id (prop\_id) of the property on which the restriction is applied. Some other information about a restriction like effective date and fine amount are also recorded.

The entity sets and attributes identified from this description are as follows:

Employee: emp\_id, emp\_name, working\_since, office\_location

- Each employee has a unique emp\_id.

Property: prop\_id, prop\_name, address

- Each property has a unique prop\_id.

Hotel: prop\_id, n\_rooms

- Hotel is a specialized entity of Property entity.
- Each hotel has a unique prop\_id.

Home: prop\_id, n\_beds, host

- Home is a specialized entity of Property entity.
- Each home has a unique prop\_id.

Customer: cust\_id, cust\_name, email, contact\_no

- Each customer has a unique cust\_id.

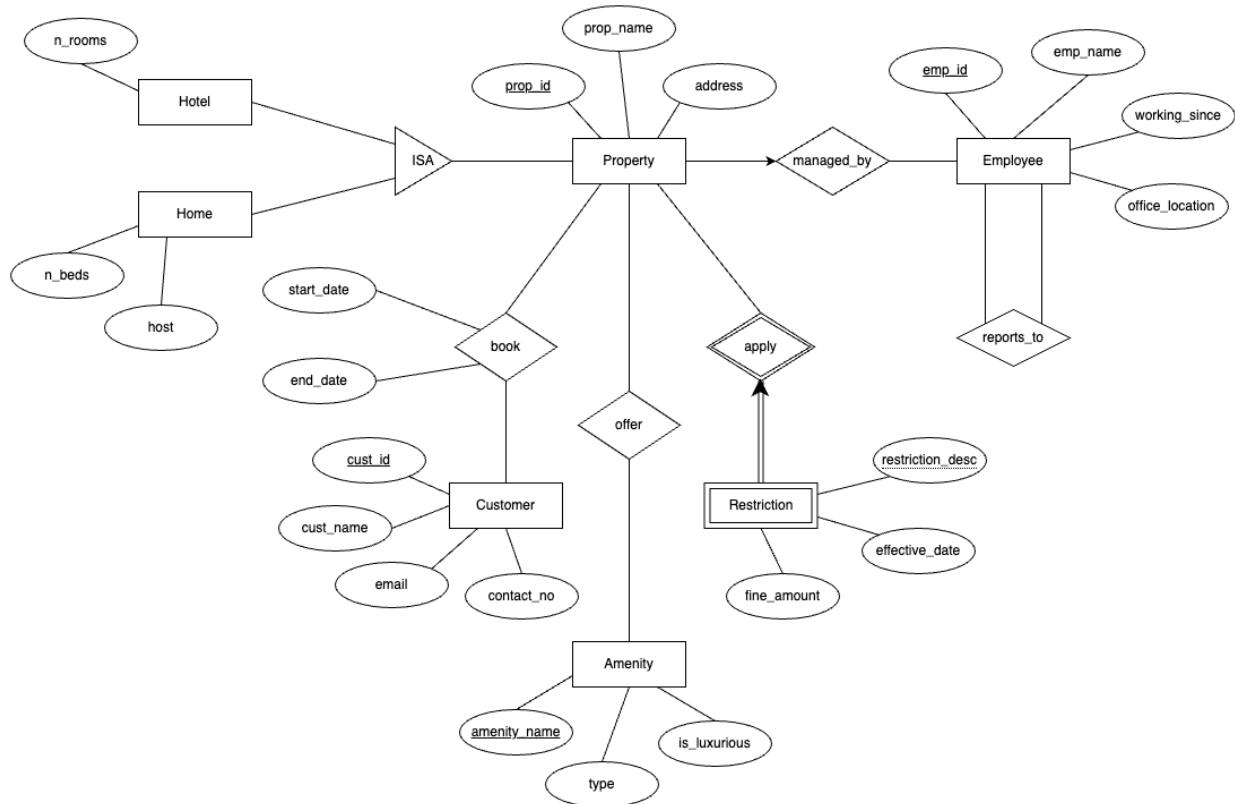
Amenity: amenity\_name, type, is\_luxurious

- Each amenity has a unique amenity\_name.

Restriction: restriction\_desc, effective\_date, fine\_amount

- Restriction is a weak entity, dependent on Property
- Restriction\_desc is unique within each property

The Entity Relationship diagram for this data model is given below:



## II. Relational Data Model

Relational Data Model for the Hotel Database is given below. SQL create statements are also included to describe the relations or tables and define the datatype for attributes of each relation. Relevant data is added to the tables using the insert queries provided. Screenshots of the final data in these tables are also included.

### 1. Entity ‘Employee’

Schema: Employee(emp\_id, emp\_name, working\_since, office\_location)  
- emp\_id is the primary key

SQL Create Statement:

```
CREATE TABLE Employee(
    emp_id int,
    emp_name varchar(100),
    working_since date,
    office_location varchar(100),
    PRIMARY KEY(emp_id)
)
```

SQL Insert Statement:

```
INSERT INTO Employee
VALUES (8202847, 'Navjot Singh', '2012-05-13', 'Peterborough'),
(8202848, 'Connor Paolo', '2013-12-26', 'Waterloo'),
(8202849, 'Daisy Fernandes', '2014-03-21', 'Kitchener'),
(8202850, 'Henry Ford', '2012-01-14', 'Peterborough'),
(8202851, 'Charles Darwin', '2022-11-22', 'Brampton'),
(8202852, 'Michael Jackson', '2010-07-02', 'Scarborough'),
(8202853, 'Jack Fernandes', '2021-08-07', 'Markham'),
(8202854, 'Noah Wyle', '2020-10-19', 'Brampton'),
(8202855, 'Emma Thompson', '2017-06-01', 'Oshawa'),
(8202856, 'James Parker', '2022-01-27', 'Mississauga'),
(8202857, 'Olivia Taylor', '2009-10-01', 'Toronto'),
(8202858, 'Robert Anderson', '2008-05-12', 'Toronto'),
(8202859, 'Ava Martinez', '2008-08-08', 'Toronto'),
(8202860, 'James Wilson', '2005-03-24', 'Toronto')
```

Screenshot:

The screenshot shows a MySQL query results page. At the top, a green bar indicates "Showing rows 0 - 13 (14 total, Query took 0.0004 seconds.)". Below this is a toolbar with the SQL query: "SELECT \* FROM Employee;". There are buttons for Profiling, Edit inline, Edit, Explain SQL, Create PHP code, and Refresh. The main area displays a table with 14 rows of employee data. The columns are emp\_id, emp\_name, working\_since, and office\_location. Each row includes edit, copy, and delete icons. The data is as follows:

	emp_id	emp_name	working_since	office_location
<input type="checkbox"/>	8202847	Navjot Singh	2012-05-13	Peterborough
<input type="checkbox"/>	8202848	Connor Paolo	2013-12-26	Waterloo
<input type="checkbox"/>	8202849	Daisy Fernandes	2014-03-21	Kitchener
<input type="checkbox"/>	8202850	Henry Ford	2012-01-14	Peterborough
<input type="checkbox"/>	8202851	Charles Darwin	2022-11-22	Brampton
<input type="checkbox"/>	8202852	Michael Jackson	2010-07-02	Scarborough
<input type="checkbox"/>	8202853	Jack Fernandes	2021-08-07	Markham
<input type="checkbox"/>	8202854	Noah Wyle	2020-10-19	Brampton
<input type="checkbox"/>	8202855	Emma Thompson	2017-06-01	Oshawa
<input type="checkbox"/>	8202856	James Parker	2022-01-27	Mississauga
<input type="checkbox"/>	8202857	Olivia Taylor	2009-10-01	Toronto
<input type="checkbox"/>	8202858	Robert Anderson	2008-05-12	Toronto
<input type="checkbox"/>	8202859	Ava Martinez	2008-08-08	Toronto
<input type="checkbox"/>	8202860	James Wilson	2005-03-24	Toronto

## 2. Relationship ‘reports\_to’

Schema: reports\_to(manager\_id, associate\_id)

- (manager\_id, associate\_id) is the primary key
- manager\_id is a foreign key referencing emp\_id in Employee
- associate\_id is a foreign key referencing emp\_id in Employee

SQL Create Statement:

```
CREATE TABLE reports_to(
    manager_id int,
    associate_id int,
    PRIMARY KEY(manager_id, associate_id),
    FOREIGN KEY(manager_id) REFERENCES Employee(emp_id),
    FOREIGN KEY(associate_id) REFERENCES Employee(emp_id)
)
```

## SQL Insert Statement:

```
INSERT INTO reports_to (associate_id, manager_id)
VALUES (8202847, 8202857),
(8202848, 8202857),
(8202849, 8202857),
(8202850, 8202858),
(8202851, 8202858),
(8202852, 8202859),
(8202853, 8202859),
(8202854, 8202860),
(8202855, 8202860),
(8202856, 8202860)
```

## Screenshot:

The screenshot shows a database interface with the following details:

- Query Result:** A green bar at the top indicates "Showing rows 0 - 9 (10 total, Query took 0.0003 seconds.)".
- SQL Query:** The query displayed is "SELECT \* FROM reports\_to;".
- Table Headers:** The table has columns labeled "manager\_id" and "associate\_id".
- Data Rows:** There are 10 rows of data, each containing a pair of IDs. The rows are:
  - 8202857, 8202847
  - 8202857, 8202848
  - 8202857, 8202849
  - 8202858, 8202850
  - 8202858, 8202851
  - 8202859, 8202852
  - 8202859, 8202853
  - 8202860, 8202854
  - 8202860, 8202855
  - 8202860, 8202856
- Table Tools:** A toolbar at the bottom of the table includes icons for Edit, Copy, and Delete.
- Navigation:** Buttons for "Show all" (unchecked), "Number of rows" (set to 25), "Filter rows" (Search this table), and "Sort by key" (None).
- Extra Options:** A button labeled "Extra options" is visible.

### 3. Entity ‘Property’

Schema: Property(prop\_id, prop\_name, address, *emp\_id*)

- prop\_id is the primary key
- emp\_id is a foreign key referencing emp\_id in Employee

SQL Create Statement:

```
CREATE TABLE Property(
    prop_id int,
    prop_name varchar(100),
    address varchar(100),
    emp_id int,
    PRIMARY KEY(prop_id),
    FOREIGN KEY(emp_id) REFERENCES Employee(emp_id)
)
```

SQL Insert Statement:

```
INSERT INTO Property
VALUES (101, 'Beautiful Villa', '123 Main Street, Peterborough', 8202847),
(102, 'Cozy Apartment', '456 Park Avenue, Waterloo', 8202848),
(103, 'Spacious House', '789 Oak Lane, Kitchener', 8202849),
(104, 'Luxury Condo', '321 Elm Road, Peterborough', 8202850),
(105, 'Riverside Cottage', '654 River Street, Brampton', 8202851),
(106, 'Mountain Chalet', '987 Mountain View, Scarborough', 8202852),
(107, 'Seaside Retreat', '654 Beachfront, Markham', 8202853),
(108, 'Urban Loft', '321 Loft Lane, Brampton', 8202854),
(109, 'Country Farmhouse', '789 Farm Road, Oshawa', 8202855),
(110, 'Garden Oasis', '111 Green Gardens, Mississauga', 8202856)
```

Screenshot:

Showing rows 0 - 9 (10 total, Query took 0.0005 seconds.)

`SELECT * FROM Property;`

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	prop_id	prop_name	address	emp_id
<input type="checkbox"/>	101	Beautiful Villa	123 Main Street, Peterborough	8202847
<input type="checkbox"/>	102	Cozy Apartment	456 Park Avenue, Waterloo	8202848
<input type="checkbox"/>	103	Spacious House	789 Oak Lane, Kitchener	8202849
<input type="checkbox"/>	104	Luxury Condo	321 Elm Road, Peterborough	8202850
<input type="checkbox"/>	105	Riverside Cottage	654 River Street, Brampton	8202851
<input type="checkbox"/>	106	Mountain Chalet	987 Mountain View, Scarborough	8202852
<input type="checkbox"/>	107	Seaside Retreat	654 Beachfront, Markham	8202853
<input type="checkbox"/>	108	Urban Loft	321 Loft Lane, Brampton	8202854
<input type="checkbox"/>	109	Country Farmhouse	789 Farm Road, Oshawa	8202855
<input type="checkbox"/>	110	Garden Oasis	111 Green Gardens, Mississauga	8202856

#### 4. Entity ‘Hotel’

Schema: Hotel(prop\_id, n\_rooms)

- prop\_id is the primary key
- prop\_id is a foreign key referencing prop\_id in Property

SQL Create Statement:

```
CREATE TABLE Hotel(
    prop_id int,
    n_rooms int,
    PRIMARY KEY(prop_id),
    FOREIGN KEY(prop_id) REFERENCES Property(prop_id)
)
```

SQL Insert Statement:

```
INSERT INTO Hotel
VALUES (101, 20),
(102, 25),
(103, 30),
(104, 15),
```

(105, 30),  
(106, 40)

Screenshot:

The screenshot shows a MySQL database interface. At the top, a green bar indicates "Showing rows 0 - 5 (6 total, Query took 0.0003 seconds.)". Below this, the SQL query "SELECT \* FROM Hotel;" is displayed. A toolbar below the query includes options for Profiling, Edit inline, Edit, Explain SQL, Create PHP code, and Refresh. Further down are filters for Show all (unchecked), Number of rows (set to 25), Filter rows, Search this table, and Sort by key (set to None). An "Extra options" button is also present. The main area displays a table with columns prop\_id and n\_rooms, containing six rows of data.

	prop_id	n_rooms
<input type="checkbox"/> Edit  Copy  Delete	101	20
<input type="checkbox"/> Edit  Copy  Delete	102	25
<input type="checkbox"/> Edit  Copy  Delete	103	30
<input type="checkbox"/> Edit  Copy  Delete	104	15
<input type="checkbox"/> Edit  Copy  Delete	105	30
<input type="checkbox"/> Edit  Copy  Delete	106	40

## 5. Entity 'Home'

Schema: Home(prop\_id, n\_beds, host)

- prop\_id is the primary key
- prop\_id is a foreign key referencing prop\_id in Property

SQL Create Statement:

```
CREATE TABLE Home(
    prop_id int,
    n_beds int,
    host varchar(100),
    PRIMARY KEY(prop_id),
    FOREIGN KEY(prop_id) REFERENCES Property(prop_id)
)
```

SQL Insert Statement:

```
INSERT INTO Home
VALUES (107, 5, 'John Smith'),
(108, 8, 'Jane Doe'),
(109, 6, 'Michael Johnson'),
(110, 4, 'Emily Brown')
```

Screenshot:

The screenshot shows a MySQL database interface. At the top, a green bar displays a success message: "Showing rows 0 - 3 (4 total, Query took 0.0002 seconds.)". Below this is a query editor with the SQL command: "SELECT \* FROM Home;". Underneath the query are several action buttons: "Profiling", "Edit inline", "Edit", "Explain SQL", "Create PHP code", and "Refresh". A toolbar below the editor includes options for "Show all" (unchecked), "Number of rows: 25", "Filter rows: Search this table", "Sort by key: None", and a dropdown menu. An "Extra options" button is also present. The main area displays a table named "Home" with four rows of data:

	prop_id	n_beds	host
<input type="checkbox"/>	107	5	John Smith
<input type="checkbox"/>	108	8	Jane Doe
<input type="checkbox"/>	109	6	Michael Johnson
<input type="checkbox"/>	110	4	Emily Brown

## 6. Entity ‘Customer’

Schema: Customer(cust\_id, cust\_name, email, contact\_no)

- cust\_id is the primary key

SQL Create Statement:

```
CREATE TABLE Customer(
    cust_id int,
    cust_name varchar(100),
    email varchar(20),
    contact_no varchar(15),
    PRIMARY KEY(cust_id)
)
```

SQL Insert Statement:

```
INSERT INTO Customer
VALUES (9185737, 'John Doe', 'john@gmail.com', '123-456-7890'),
(9185738, 'Jane Smith', 'jane@gmail.com', '987-654-3210'),
(9185739, 'Michael Johnson', 'michael@gmail.com', '555-123-4567'),
(9185740, 'Emily Brown', 'emily@gmail.com', '222-555-8888'),
(9185741, 'Robert Lee', 'robert@gmail.com', '444-999-7777'),
(9185742, 'Sarah Johnson', 'sarah@gmail.com', '111-222-3333'),
(9185743, 'David Wilson', 'david@gmail.com', '888-777-6666'),
(9185744, 'Emma Davis', 'emma@gmail.com', '555-888-9999'),
(9185745, 'James Anderson', 'james@gmail.com', '123-789-4560'),
(9185746, 'Olivia Thomas', 'olivia@gmail.com', '777-555-2222')
```

Screenshot:

The screenshot shows a database query results page. At the top, a green bar indicates "Showing rows 0 - 9 (10 total, Query took 0.0003 seconds.)". Below this is a toolbar with the SQL query: "SELECT \* FROM Customer;". There are buttons for Profiling, Edit inline, Explain SQL, Create PHP code, and Refresh. Below the toolbar are filtering options: "Show all" (unchecked), "Number of rows: 25", "Filter rows: Search this table", and "Sort by key: None". A "Extra options" button is also present. The main area displays a table with 10 rows of customer data:

	cust_id	cust_name	email	contact_no
<input type="checkbox"/>	9185737	John Doe	john@gmail.com	123-456-7890
<input type="checkbox"/>	9185738	Jane Smith	jane@gmail.com	987-654-3210
<input type="checkbox"/>	9185739	Michael Johnson	michael@gmail.com	555-123-4567
<input type="checkbox"/>	9185740	Emily Brown	emily@gmail.com	222-555-8888
<input type="checkbox"/>	9185741	Robert Lee	robert@gmail.com	444-999-7777
<input type="checkbox"/>	9185742	Sarah Johnson	sarah@gmail.com	111-222-3333
<input type="checkbox"/>	9185743	David Wilson	david@gmail.com	888-777-6666
<input type="checkbox"/>	9185744	Emma Davis	emma@gmail.com	555-888-9999
<input type="checkbox"/>	9185745	James Anderson	james@gmail.com	123-789-4560
<input type="checkbox"/>	9185746	Olivia Thomas	olivia@gmail.com	777-555-2222

## 7. Relationship 'book'

Schema: book(cust\_id, prop\_id, start\_date, end\_date)

- (cust\_id, prop\_id) is the primary key
- cust\_id is a foreign key referencing cust\_id in Customer
- prop\_id is a foreign key referencing prop\_id in Property

SQL Create Statement:

```
CREATE TABLE book(
    cust_id int,
    prop_id int,
    start_date date,
    end_date date,
    PRIMARY KEY(cust_id, prop_id),
    FOREIGN KEY(cust_id) REFERENCES Customer(cust_id),
    FOREIGN KEY(prop_id) REFERENCES Property(prop_id)
)
```

SQL Insert Statement:

```
INSERT INTO book
VALUES (9185737, 103, '2022-07-25', '2022-07-27'),
(9185738, 102, '2022-08-01', '2022-08-03'),
(9185739, 105, '2022-08-15', '2022-08-18'),
(9185740, 108, '2022-09-01', '2022-09-02'),
(9185741, 106, '2022-09-10', '2022-09-13'),
(9185742, 101, '2022-10-01', '2022-10-02'),
(9185743, 107, '2022-10-17', '2022-10-18'),
(9185744, 110, '2022-11-02', '2022-11-04'),
(9185745, 109, '2022-11-13', '2022-11-14'),
(9185746, 104, '2022-12-04', '2022-12-06')
```

Screenshot:

Showing rows 0 - 9 (10 total, Query took 0.0004 seconds.)

```
SELECT * FROM book;
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 ▾ Filter rows: Search this table Sort by key: None ▾

Extra options

		cust_id	prop_id	start_date	end_date
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	9185737	103	2022-07-25	2022-07-27
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	9185738	102	2022-08-01	2022-08-03
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	9185739	105	2022-08-15	2022-08-18
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	9185740	108	2022-09-01	2022-09-02
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	9185741	106	2022-09-10	2022-09-13
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	9185742	101	2022-10-01	2022-10-02
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	9185743	107	2022-10-17	2022-10-18
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	9185744	110	2022-11-02	2022-11-04
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	9185745	109	2022-11-13	2022-11-14
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a>	9185746	104	2022-12-04	2022-12-06

## 8. Entity ‘Amenity’

Schema: Amenity(amenity\_name, type, is\_luxurious)

- amenity\_name is the primary key

SQL Create Statement:

```
CREATE TABLE Amenity(
    amenity_name varchar(100),
    type varchar(20),
    is_luxurious varchar(5),
    PRIMARY KEY(amenity_name)
)
```

SQL Insert Statement:

```
INSERT INTO Amenity
VALUES ('Swimming Pool', 'Property', 'No'),
('Gym', 'Property', 'Yes'),
('Spa', 'Property', 'Yes'),
('Parking', 'Property', 'No'),
```

```
('Library', 'Property', 'No'),
('Garden', 'Property', 'No'),
('Wifi', 'Property', 'No'),
('Air Conditioning', 'Room', 'No'),
('Mini Fridge', 'Room', 'No'),
('Coffee Maker', 'Room', 'No'),
('Television', 'Room', 'No'),
('Jacuzzi', 'Room', 'Yes')
```

Screenshot:

The screenshot shows a database interface with the following details:

- Query Result:** Shows rows 0 - 11 (12 total, Query took 0.0005 seconds.)
- SQL Query:** SELECT \* FROM Amenity;
- Toolbar Buttons:** Profiling, Edit inline, Edit, Explain SQL, Create PHP code, Refresh.
- Table Headers:** amenity\_name, type, is\_luxurious.
- Table Data:** A list of 12 rows with columns: amenity\_name, type, is\_luxurious.

	amenity_name	type	is_luxurious			
<input type="checkbox"/>	Edit	Copy	Delete	Air Conditioning	Room	No
<input type="checkbox"/>	Edit	Copy	Delete	Coffee Maker	Room	No
<input type="checkbox"/>	Edit	Copy	Delete	Garden	Property	No
<input type="checkbox"/>	Edit	Copy	Delete	Gym	Property	Yes
<input type="checkbox"/>	Edit	Copy	Delete	Jacuzzi	Room	Yes
<input type="checkbox"/>	Edit	Copy	Delete	Library	Property	No
<input type="checkbox"/>	Edit	Copy	Delete	Mini Fridge	Room	No
<input type="checkbox"/>	Edit	Copy	Delete	Parking	Property	No
<input type="checkbox"/>	Edit	Copy	Delete	Spa	Property	Yes
<input type="checkbox"/>	Edit	Copy	Delete	Swimming Pool	Property	No
<input type="checkbox"/>	Edit	Copy	Delete	Television	Room	No
<input type="checkbox"/>	Edit	Copy	Delete	Wifi	Property	No

## 9. Relationship ‘offer’

Schema: offer(prop\_id, amenity\_name)

- (prop\_id, amenity\_name) is the primary key
- prop\_id is a foreign key referencing prop\_id in Property
- amenity\_name is a foreign key referencing amenity\_name in Amenity

SQL Create Statement:

```
CREATE TABLE offer(
    prop_id int,
    amenity_name varchar(100),
    PRIMARY KEY(prop_id, amenity_name),
    FOREIGN KEY(prop_id) REFERENCES Property(prop_id),
    FOREIGN KEY(amenity_name) REFERENCES
    Amenity(amenity_name)
)
```

SQL Insert Statement:

```
INSERT INTO offer (prop_id, amenity_name)
VALUES (101, 'Spa'), (101, 'Gym'), (101, 'Jacuzzi'), (101, 'Television'),
(101, 'Air Conditioning'),
(102, 'Swimming Pool'), (102, 'Parking'), (102, 'Mini Fridge'),
(102, 'Television'),
(103, 'Wifi'), (103, 'Parking'), (103, 'Coffee Maker'), (103, 'Television'),
(104, 'Garden'), (104, 'Library'), (104, 'Jacuzzi'), (104, 'Air Conditioning'),
(105, 'Spa'), (105, 'Gym'), (105, 'Mini Fridge'), (105, 'Television'),
(105, 'Air Conditioning'),
(106, 'Mini Fridge'), (106, 'Television'), (106, 'Wifi'), (106, 'Parking'),
(107, 'Spa'), (107, 'Gym'), (107, 'Jacuzzi'), (107, 'Television'),
(107, 'Air Conditioning'),
(108, 'Coffee Maker'), (108, 'Television'), (108, 'Garden'), (108, 'Library'),
(109, 'Wifi'), (109, 'Garden'), (109, 'Coffee Maker'),
(109, 'Mini Fridge'), (109, 'Air Conditioning'),
(110, 'Spa'), (110, 'Gym'), (110, 'Jacuzzi'),
(110, 'Television'), (110, 'Parking')
```

Screenshot:

Showing rows 0 - 24 (45 total, Query took 0.0003 seconds.)

```
SELECT * FROM offer;
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

	prop_id	amenity_name
<input type="checkbox"/>	101	Air Conditioning
<input type="checkbox"/>	101	Gym
<input type="checkbox"/>	101	Jacuzzi
<input type="checkbox"/>	101	Spa
<input type="checkbox"/>	101	Television
<input type="checkbox"/>	102	Mini Fridge
<input type="checkbox"/>	102	Parking
<input type="checkbox"/>	102	Swimming Pool
<input type="checkbox"/>	102	Television
<input type="checkbox"/>	103	Coffee Maker
<input type="checkbox"/>	103	Parking
<input type="checkbox"/>	103	Television
<input type="checkbox"/>	103	Wifi
<input type="checkbox"/>	104	Air Conditioning
<input type="checkbox"/>	104	Garden
<input type="checkbox"/>	104	Jacuzzi
<input type="checkbox"/>	104	Library
<input type="checkbox"/>	105	Air Conditioning
<input type="checkbox"/>	105	Gym
<input type="checkbox"/>	105	Mini Fridge
<input type="checkbox"/>	105	Spa
<input type="checkbox"/>	105	Television
<input type="checkbox"/>	106	Mini Fridge
<input type="checkbox"/>	106	Parking
<input type="checkbox"/>	106	Television

## 10. Weak Entity ‘Restriction’

Schema: Restriction(restriction\_desc, prop\_id, effective\_date, fine\_amount)

- (restriction\_desc, prop\_id) is the primary key
- prop\_id is a foreign key referencing prop\_id in Property

SQL Create Statement:

```
CREATE TABLE Restriction(
    restriction_desc varchar(100),
    effective_date date,
    fine_amount float(24),
    prop_id int,
    PRIMARY KEY(restriction_desc, prop_id),
    FOREIGN KEY(prop_id) REFERENCES Property(prop_id)
)
```

SQL Insert Statement:

```
INSERT INTO Restriction (restriction_desc, effective_date, fine_amount,
prop_id)
VALUES ('No Pets Allowed', '2022-07-25', 100.00, 101),
('No Loud Music 10 PM to 7 AM', '2022-08-11', 200.00, 101),
('No Smoking', '2022-08-17', 400.00, 102),
('Maximum Occupancy 2', '2022-09-21', 60.00, 102),
('Check-in Time 3 PM', '2022-10-14', 50.00, 103),
('Check-out Time 11 AM', '2022-10-05', 50.00, 103),
('No Outside Visitors', '2022-11-28', 100.00, 103),
('No Loud Music', '2021-01-19', 300.00, 105),
('No Swimming After Dark', '2021-02-11', 150.00, 105),
('Local ID Not Allowed', '2021-04-23', 330.00, 106),
('No Pets Allowed', '2021-05-31', 300.00, 106),
('Maximum Occupancy 3', '2022-09-01', 60.00, 108),
('No Smoking', '2022-08-15', 500.00, 110)
```

Screenshot:

Showing rows 0 - 12 (13 total, Query took 0.0004 seconds.)

SELECT \* FROM Restriction;

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows:

25

Filter rows:

Search this table

Sort by key:

None

Extra options

	restriction_desc	effective_date	fine_amount	prop_id
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a> Check-in Time 3 PM	2022-10-14	50	103
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a> Check-out Time 11 AM	2022-10-05	50	103
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a> Local ID Not Allowed	2021-04-23	330	106
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a> Maximum Occupancy 2	2022-09-21	60	102
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a> Maximum Occupancy 3	2022-09-01	60	108
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a> No Loud Music	2021-01-19	300	105
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a> No Loud Music 10 PM to 7 AM	2022-08-11	200	101
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a> No Outside Visitors	2022-11-28	100	103
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a> No Pets Allowed	2022-07-25	100	101
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a> No Pets Allowed	2021-05-31	300	106
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a> No Smoking	2022-08-17	400	102
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a> No Smoking	2022-08-15	500	110
<input type="checkbox"/>	<a href="#">Edit</a> <a href="#">Copy</a> <a href="#">Delete</a> No Swimming After Dark	2021-02-11	150	105

### III. SQL Queries

Queries for one table

- SQL query to report all luxurious amenities.

Query: `SELECT * FROM Amenity  
WHERE is_luxurious = 'Yes'`

Output:

The screenshot shows a MySQL query results page. At the top, a green bar indicates "Showing rows 0 - 2 (3 total, Query took 0.0006 seconds.)". Below this is the SQL query: `SELECT * FROM Amenity WHERE is_luxurious = 'Yes';`. A toolbar below the query includes options like Profiling, Edit inline, Explain SQL, Create PHP code, and Refresh. The main area displays a table with three rows of data:

	amenity_name	type	is_luxurious	
<input type="checkbox"/>	Edit Copy Delete	Gym	Property	Yes
<input type="checkbox"/>	Edit Copy Delete	Jacuzzi	Room	Yes
<input type="checkbox"/>	Edit Copy Delete	Spa	Property	Yes

- SQL query to print the office location of the oldest employee(s).

Query: `SELECT DISTINCT(office_location) FROM Employee  
WHERE working_since = (SELECT MIN(working_since)  
FROM Employee)`

Output:

The screenshot shows a MySQL query results page. At the top, a green bar indicates "Showing rows 0 - 0 (1 total, Query took 0.0006 seconds.)". Below this is the SQL query: `SELECT DISTINCT(office_location) FROM Employee WHERE working_since = (SELECT MIN(working_since) FROM Employee);`. A toolbar below the query includes options like Profiling, Edit inline, Explain SQL, Create PHP code, and Refresh. The main area displays a table with one row of data:

	office_location
<input type="checkbox"/>	Edit Copy Delete Toronto

- SQL query to print the name of the host of the home with the maximum number of beds.

Query: `SELECT DISTINCT(host) FROM Home  
WHERE n_beds = (SELECT MAX(n_beds)  
FROM Home)`

Output:

The screenshot shows a database query results page. At the top, a green bar indicates "Showing rows 0 - 0 (1 total, Query took 0.0006 seconds.)". Below this is the SQL query: `SELECT DISTINCT(host) FROM Home WHERE n_beds = (SELECT MAX(n_beds) FROM Home);`. A "Profiling" link is available below the query. The main area displays a table with one row. The table has a header "host" and a data row for "Jane Doe". Below the table are "Edit", "Copy", and "Delete" buttons, along with a "Jane Doe" label. Navigation buttons for the table are also present.

host
Jane Doe

## Queries for two tables

- SQL query to print the property id of properties which offer all luxurious amenities.

Query: `SELECT DISTINCT(O.prop_id) FROM offer O  
WHERE NOT EXISTS`

```
((SELECT amenity_name FROM Amenity  
WHERE is_luxurious = 'Yes')  
EXCEPT  
(SELECT O2.amenity_name  
FROM offer O2  
WHERE O2.prop_id = O.prop_id));
```

## Output:

Showing rows 0 - 2 (3 total, Query took 0.0009 seconds.)

```
SELECT DISTINCT(o.prop_id) FROM offer o WHERE NOT EXISTS ((SELECT amenity_name FROM Amenity WHERE is_luxurious = 'Yes') EXCEPT (SELECT o2.amenity_name FROM offer o2 WHERE o2.prop_id = o.prop_id));
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

prop_id
101
107
110

- SQL query to print the employee id of the managers with at least one of the subordinates working from the Peterborough office.

Query: `SELECT DISTINCT(R.manager_id)`

```
FROM Employee E, reports_to R  
WHERE E.emp_id = R.associate_id  
AND E.office_location = 'Peterborough'
```

## Output:

Showing rows 0 - 1 (2 total, Query took 0.0008 seconds.)

```
SELECT DISTINCT(R.manager_id) FROM Employee E, reports_to R WHERE E.emp_id = R.associate_id AND E.office_location = 'Peterborough';
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

manager_id
8202857
8202858

- SQL query to print the number of beds available in the “Urban loft” home.

Query: `SELECT H.n_beds`

```
FROM Home H, Property P  
WHERE H.prop_id = P.prop_id  
AND P.prop_name = 'Urban Loft'
```

## Output:

Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

```
SELECT H.n_beds FROM Home H, Property P WHERE H.prop_id = P.prop_id AND P.prop_name = 'Urban Loft';
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

	n_beds
<input type="checkbox"/>	8

Edit Copy Delete

## Queries for three tables

- SQL query to print the name of property managers for properties with no restrictions.

Query: `SELECT E.emp_name  
FROM Property P, Employee E  
WHERE P.emp_id = E.emp_id  
AND NOT EXISTS  
(SELECT * FROM Restriction R  
WHERE R.prop_id = P.prop_id)`

Output:

Showing rows 0 - 2 (3 total, Query took 0.0006 seconds.)

```
SELECT E.emp_name FROM Property P, Employee E WHERE P.emp_id = E.emp_id AND NOT EXISTS (SELECT * FROM Restriction R WHERE R.prop_id = P.prop_id);
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

	emp_name
<input type="checkbox"/>	Henry Ford
<input type="checkbox"/>	Jack Fernandes
<input type="checkbox"/>	Emma Thompson

Edit Copy Delete

- SQL query to print the name of customers who have booked “Mountain Chalet” property.

Query: `SELECT C.cust_name FROM Customer C, Property P, book B  
WHERE C.cust_id = B.cust_id AND P.prop_id = B.prop_id  
AND P.prop_name = 'Mountain Chalet'`

## Output:

Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

```
SELECT C.cust_name FROM Customer C, Property P, book B WHERE C.cust_id = B.cust_id AND P.prop_id = B.prop_id AND P.prop_name = 'Mountain Chalet';
```

Profile [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

cust_name
Robert Lee

Edit Copy Delete Robert Lee

- SQL query to print the names of properties that offer at least one luxurious and at least one non-luxurious amenity.

Query: `SELECT DISTINCT(P.prop_name) FROM Property P, offer O, Amenity A  
WHERE P.prop_id = O.prop_id AND O.amenity_name = A.amenity_name  
AND A.is_luxurious = 'Yes'  
AND EXISTS  
(SELECT *  
FROM offer O2, Amenity A2  
WHERE O2.amenity_name = A2.amenity_name  
AND O.prop_id = O2.prop_id  
AND A2.is_luxurious = 'No');`

## Output:

Showing rows 0 - 4 (5 total, Query took 0.0007 seconds.)

```
SELECT DISTINCT(P.prop_name) FROM Property P, offer O, Amenity A WHERE P.prop_id = O.prop_id AND O.amenity_name = A.amenity_name AND A.is_luxurious = 'Yes' AND EXISTS (SELECT * FROM offer O2, Amenity A2 WHERE O2.amenity_name = A2.amenity_name AND O.prop_id = O2.prop_id AND A2.is_luxurious = 'No');
```

Profile [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

prop_name
Beautiful Villa
Riverside Cottage
Seaside Retreat
Garden Oasis
Luxury Condo

#### **IV. Conclusion**

We have successfully created a Hotel Database and also presented some of the sample queries for the database. Major entities in the hotel database are property, hotel, home, customer, employee, amenity and restriction. Hotel and Home are specialized entities derived from Property. The relationship between these entities is explained in the ER diagram. The relational data model is also given to understand minute implementations for the hotel database more clearly. New entities and relationships can be easily added to the existing database without major changes in the existing data model. The sample queries provide an example of how the database can be used to extract important information.