

### Question 1:

**Design a database schema that can handle the operations of a room reservation system for a global hotel chain. Ensure the schema is scalable, maintainable, and supports necessary business operations.**

Considering all the requirements, I have finalized the approach with the following solution, which should appropriately meet the necessary needs.

I will include the following tables: Countries, Cities, Hotels, Staff, Rooms, Customers, Reservations, Payments, and RoomTypes.

**Countries Table** – Stores information about countries where the hotels are located.

Column Name	Data Type	Constraints
country_id	INT	PRIMARY KEY
country_name	VARCHAR(100)	NOT NULL, UNIQUE
country_code	VARCHAR(10)	NOT NULL, UNIQUE

**Cities Table** – Stores information about cities where the hotels are located.

Column Name	Data Type	Constraints
city_id	INT	PRIMARY KEY
city_name	VARCHAR(100)	NOT NULL
state	VARCHAR(100)	NOT NULL
country_id	INT	FOREIGN KEY (country_id) REFERS Countries (country_id)

**Hotels Table** – Contains information about the hotels in the chain.

Column Name	Data Type	Constraints
hotel_id	INT	PRIMARY KEY
name	VARCHAR(100)	NOT NULL
address	VARCHAR(255)	NOT NULL
city_id	INT	FOREIGN KEY (city_id) REFERS Cities (city_id)
postal_code	VARCHAR(20)	NOT NULL
phone_number	VARCHAR(20)	NOT NULL
email	VARCHAR(20)	
starrating	INT	

### Staffs Table

Column Name	Data Type	Constraints
employee_id	INT	PRIMARY KEY
hotel_id	INT	FOREIGN KEY (hotel_id) REFERS Hotels (hotel_id)
first_name	VARCHAR(50)	NOT NULL
last_name	VARCHAR(50)	NOT NULL
position	VARCHAR(50)	NOT NULL
email	VARCHAR(100)	NOT NULL, UNIQUE
phone_number	VARCHAR (20)	NOT NULL
date_of_birth	DATE	NOT NULL
salary	DECIMAL(10, 2)	NOT NULL

### Rooms Table – Contains information about the rooms in each hotel.

Column Name	Data Type	Constraints
room_number	INT	PRIMARY KEY
hotel_id	INT	FOREIGN KEY (hotel_id) REFERS Hotels (hotel_id)
room_number	VARCHAR(10)	NOT NULL
room_type	VARCHAR(50)	FOREIGN KEY (room_type) REFERS Room Types (room_type)
capacity	INT	NOT NULL
price_per_night	DECIMAL(10, 2)	NOT NULL
is_available	BOOLEAN	DEFAULT TRUE

### RoomTypes Table - Store types of rooms available.

Column Name	Data Type	Constraints
room_type	INT	PRIMARY KEY
type_name	VARCHAR(50)	NOT NULL, UNIQUE
description	TEXT	

### Customers Table - Stores information about the customers.

Column Name	Data Type	Constraints
customer_id	INT	PRIMARY KEY
first_name	VARCHAR(50)	NOT NULL
last_name	VARCHAR(50)	NOT NULL
email	VARCHAR(100)	NOT NULL, UNIQUE
phone_number	VARCHAR (20)	NOT NULL
date_of_birth	DATE	NOT NULL

address	VARCHAR(255)	NOT NULL
---------	--------------	----------

**Reservations Table** - Stores information about room reservations.

Column Name	Data Type	Constraints
reservation_id	INT	PRIMARY KEY
customer_id	INT	FOREIGN KEY (customer_id) REFERS Customers (customer_id)
hotel_id	INT	FOREIGN KEY (hotel_id) REFERS Hotels (hotel_id)
room_number	INT	FOREIGN KEY (room_id) REFERS Rooms(room_id)
check_in_date	DATE	NOT NULL
check_out_date	DATE	NOT NULL
total_amount	DECIMAL(10, 2)	NOT NULL
reservation_status	VARCHAR(50)	NOT NULL

**Payments Table** - Stores payment information for reservations.

Column Name	Data Type	Constraints
payment_id	INT	PRIMARY KEY
reservation_id	INT	FOREIGN KEY (reservation_id) REFERS Reservations (reservation_id)
payment_date	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP
amount	DECIMAL(10, 2)	NOT NULL
payment_method	VARCHAR(50)	NOT NULL
payment_status	VARCHAR(50)	NOT NULL
amount	DECIMAL(10, 2)	NOT NULL
update_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP

### Relationships:

1. **Countries to Cities:** One-to-Many (A country has many cities)
2. **Cities to Hotels:** One-to-Many (A city has many hotels)
3. **Hotels to Rooms:** One-to-Many (A hotel has many rooms)
4. **Hotels to Staff:** One-to-Many (A hotel employs multiple staff)
5. **Rooms to Reservations:** One-to-Many (A room can have many reservations)
6. **Customers to Reservations:** One-to-Many (A customer can make many reservations)
7. **Reservations to Payments:** One-to-Many (Each reservation might have multiple payments)
8. **RoomTypes to Rooms:** One-to-Many (A room type can be assigned to many rooms)

For a better understanding the below is the ER diagram.

