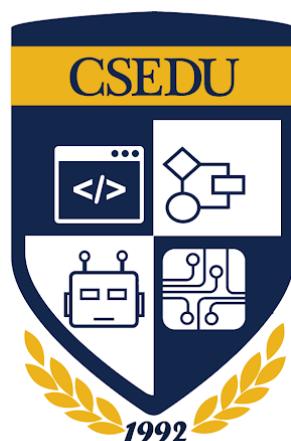


University of Dhaka  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



**Database and Management System I**

---

Project Report

# Hotel Management System

---

Advisor: Abu Ahmed Ferdous, Mosarrat Jahan

DHAKA, JANUARY 2023



## Member list

No.	Full name	Roll No
1	Afser Adil Olin	AE-47
2	Anika Tabassum	RK-61



## Contents

<b>1</b>	<b>Overview</b>	<b>4</b>
<b>2</b>	<b>Entity Relationship Diagram</b>	<b>5</b>
<b>3</b>	<b>Schema Diagram</b>	<b>6</b>
<b>4</b>	<b>SQL DDL Codes Snippet :</b>	<b>7</b>
4.1	Creating tables : . . . . .	7
4.2	Creating views : . . . . .	13
4.3	Queries : . . . . .	14
<b>5</b>	<b>Snapshots of the instances:</b>	<b>20</b>
5.1	Data of table : addresses . . . . .	20
5.2	Data of table : bookings . . . . .	21
5.3	Data of table : department . . . . .	22
5.4	Data of table : employees . . . . .	23
5.5	Data of table : guests . . . . .	24
5.6	Data of table : hotel . . . . .	25
5.7	Data of table : hotel_chain . . . . .	26
5.8	Data of table : hotel_chain_has_hotel . . . . .	27
5.9	Data of table : hotel_services . . . . .	28
5.10	Data of table : hotel_services_used_by_guests . . . . .	29
5.11	Data of table : room_rate_discount . . . . .	30
5.12	Data of table : room_type . . . . .	31
5.13	Data of table : rooms . . . . .	32
5.14	Data of table : rooms_booked . . . . .	33
5.15	Data of table : star_ratings . . . . .	34
<b>6</b>	<b>Non-trivial Functional Dependencies</b>	<b>35</b>
<b>7</b>	<b>Frontend Tools</b>	<b>36</b>
<b>8</b>	<b>Conclusions</b>	<b>37</b>



## 1 Overview

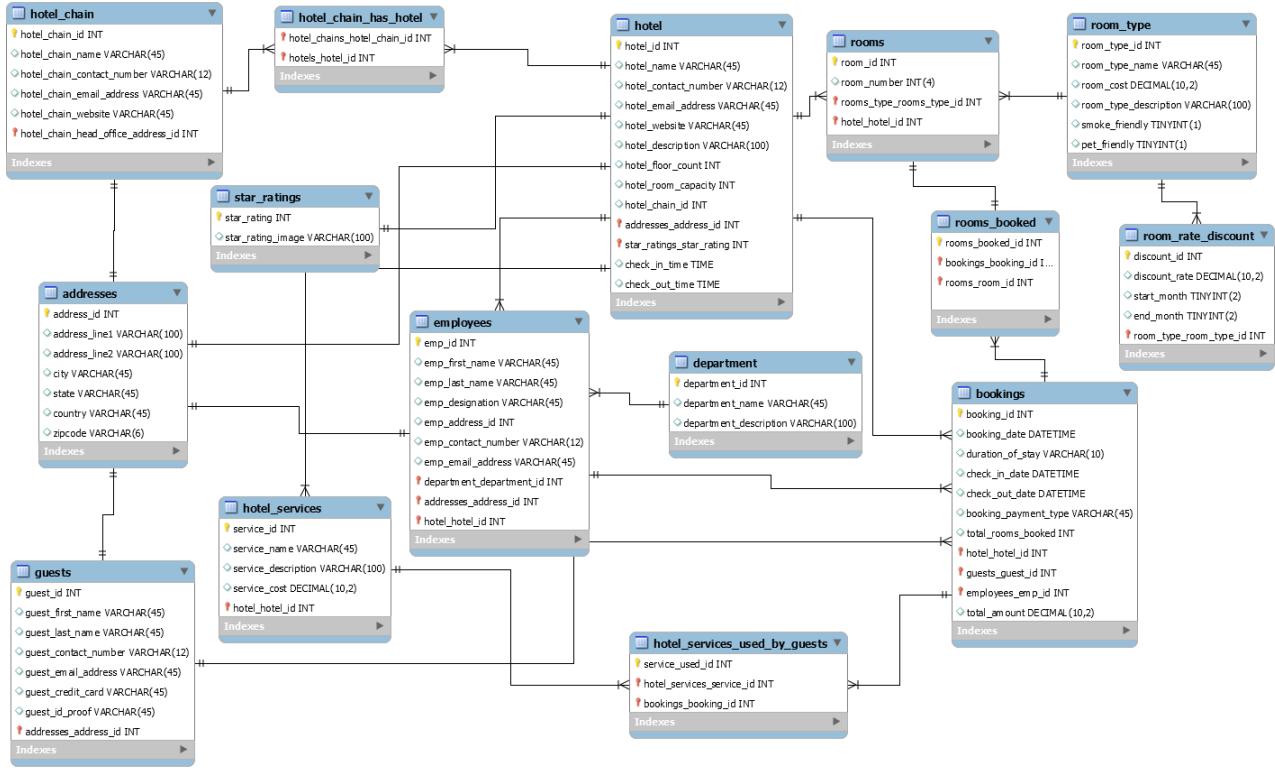
The main objective of this project is to create a database management system for a hotel. The hotel can have multiple chains, which can further have multiple hotels. Therefore, we need an organised management system, which can easily manage all the operations and data of the hotel chains and hotels respectively. We will be managing the below areas of the hotel database management system.

- The hotel chains, their details.
- The hotels in each chain and their details and other information like the rooms and their description and discounts, etc.
- Information about employees and departments they work in.
- Information about guests.
- Managing bookings and other services used by the guests.



## 2 Entity Relationship Diagram

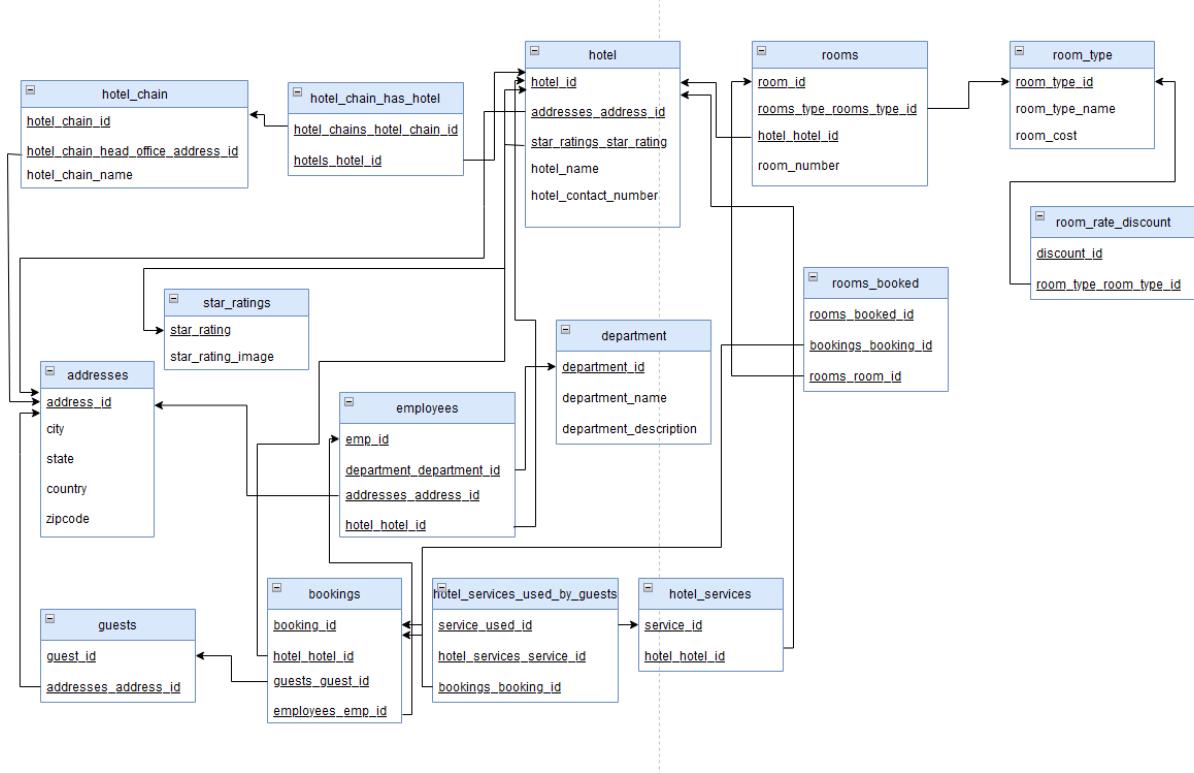
The ERR diagram created is as below:





### 3 Schema Diagram

The schema of the database is given below:





listings color

## 4 SQL DDL Codes Snippet :

### 4.1 Creating tables :

createing addresses table

```
CREATE TABLE IF NOT EXISTS "hotel_database"."addresses" (
  "address_id" INT NOT NULL,
  "address_line1" VARCHAR(100) NULL,
  "address_line2" VARCHAR(100) NULL,
  "city" VARCHAR(45) NULL,
  "state" VARCHAR(45) NULL,
  "country" VARCHAR(45) NULL,
  "zipcode" VARCHAR(8) NULL,
  PRIMARY KEY ("address_id"))
```

createing table : hotel\_chain

```
CREATE TABLE IF NOT EXISTS "hotel_database"."hotel_chain" (
  "hotel_chain_id" INT NOT NULL,
  "hotel_chain_name" VARCHAR(45) NULL,
  "hotel_chain_contact_number" VARCHAR(12) NULL,
  "hotel_chain_email_address" VARCHAR(45) NULL,
  "hotel_chain_website" VARCHAR(45) NULL,
  "hotel_chain_head_office_address_id" INT NOT NULL,
  PRIMARY KEY ("hotel_chain_id", "hotel_chain_head_office_address_id"),
  CONSTRAINT "fk_hotel_chains_addresses1"
    FOREIGN KEY ("hotel_chain_head_office_address_id")
    REFERENCES "hotel_database"."addresses" ("address_id")
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
```

createing table : star\_ratings

```
CREATE TABLE IF NOT EXISTS "hotel_database"."star_ratings" (
  "star_rating" INT NOT NULL,
  "star_rating_image" VARCHAR(100) NULL,
  PRIMARY KEY ("star_rating"))
```

createing table : hotel

```
CREATE TABLE IF NOT EXISTS "hotel_database"."hotel" (
```



```
"hotel_id" INT NOT NULL,  
"hotel_name" VARCHAR(45) NULL,  
"hotel_contact_number" VARCHAR(12) NULL,  
"hotel_email_address" VARCHAR(45) NULL,  
"hotel_website" VARCHAR(45) NULL,  
"hotel_description" VARCHAR(100) NULL,  
"hotel_floor_count" INT NULL,  
"hotel_room_capacity" INT NULL,  
"hotel_chain_id" INT NULL,  
"addresses_address_id" INT NOT NULL,  
"star_ratings_star_rating" INT NOT NULL,  
"check_in_time" TIME NULL,  
"check_out_time" TIME NULL,  
PRIMARY KEY ("hotel_id", "addresses_address_id", "star_ratings_star_rating"),  
CONSTRAINT "fk_hotels_addresses1"  
FOREIGN KEY ("addresses_address_id")  
REFERENCES "hotel_database"."addresses" ("address_id")  
ON DELETE NO ACTION  
ON UPDATE NO ACTION,  
CONSTRAINT "fk_hotel_star_ratings1"  
FOREIGN KEY ("star_ratings_star_rating")  
REFERENCES "hotel_database"."star_ratings" ("star_rating")  
ON DELETE NO ACTION  
ON UPDATE NO ACTION)
```

createing table : room\_type

```
CREATE TABLE IF NOT EXISTS "hotel_database"."room_type" (  
"room_type_id" INT NOT NULL,  
"room_type_name" VARCHAR(45) NULL,  
"room_cost" DECIMAL(10,2) NULL,  
"room_type_description" VARCHAR(100) NULL,  
PRIMARY KEY ("room_type_id"))
```

createing table : rooms

```
CREATE TABLE IF NOT EXISTS "hotel_database"."rooms" (  
"room_id" INT NOT NULL,  
"room_number" INT NULL,  
"rooms_type_rooms_type_id" INT NOT NULL,  
"hotel_hotel_id" INT NOT NULL,  
PRIMARY KEY ("room_id", "rooms_type_rooms_type_id", "hotel_hotel_id"),  
CONSTRAINT "fk_rooms_rooms_type1"  
FOREIGN KEY ("rooms_type_rooms_type_id")
```



```
REFERENCES "hotel_database"."room_type" ("room_type_id")
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT "fk_rooms_hotel1"
FOREIGN KEY ("hotel_hotel_id")
REFERENCES "hotel_database"."hotel" ("hotel_id")
ON DELETE NO ACTION
ON UPDATE NO ACTION)
```

createing table : guests

```
CREATE TABLE IF NOT EXISTS "hotel_database"."guests" (
"guest_id" INT NOT NULL,
"guest_first_name" VARCHAR(45) NULL,
"guest_last_name" VARCHAR(45) NULL,
"guest_contact_number" VARCHAR(12) NULL,
"guest_email_address" VARCHAR(45) NULL,
"guest_credit_card" VARCHAR(45) NULL,
"guest_id_proof" VARCHAR(45) NULL,
"addresses_address_id" INT NOT NULL,
PRIMARY KEY ("guest_id", "addresses_address_id"),
CONSTRAINT "fk_guests_addresses1"
FOREIGN KEY ("addresses_address_id")
REFERENCES "hotel_database"."addresses" ("address_id")
ON DELETE NO ACTION
ON UPDATE NO ACTION)
```

createing table : department

```
CREATE TABLE IF NOT EXISTS "hotel_database"."department" (
"department_id" INT NOT NULL,
"department_name" VARCHAR(45) NULL,
"department_description" VARCHAR(100) NULL,
PRIMARY KEY ("department_id"))
```

createing table : employees

```
CREATE TABLE IF NOT EXISTS "hotel_database"."employees" (
"emp_id" INT NOT NULL,
"emp_first_name" VARCHAR(45) NULL,
"emp_last_name" VARCHAR(45) NULL,
"emp_designation" VARCHAR(45) NULL,
"emp_contact_number" VARCHAR(12) NULL,
"emp_email_address" VARCHAR(45) NULL,
```



```
"department_department_id" INT NOT NULL,  
"addresses_address_id" INT NOT NULL,  
"hotel_hotel_id" INT NOT NULL,  
PRIMARY KEY ("emp_id", "department_department_id", "addresses_address_id", "hotel_hotel_id"),  
CONSTRAINT "fk_employees_services1"  
FOREIGN KEY ("department_department_id")  
REFERENCES "hotel_database"."department" ("department_id")  
ON DELETE NO ACTION  
ON UPDATE NO ACTION,  
CONSTRAINT "fk_employees_addresses1"  
FOREIGN KEY ("addresses_address_id")  
REFERENCES "hotel_database"."addresses" ("address_id")  
ON DELETE NO ACTION  
ON UPDATE NO ACTION,  
CONSTRAINT "fk_employees_hotel1"  
FOREIGN KEY ("hotel_hotel_id")  
REFERENCES "hotel_database"."hotel" ("hotel_id")  
ON DELETE NO ACTION  
ON UPDATE NO ACTION)
```

createing table : bookings

```
CREATE TABLE IF NOT EXISTS "hotel_database"."bookings" (  
"booking_id" INT NOT NULL,  
"booking_date" DATE NULL,  
"duration_of_stay" VARCHAR(10) NULL,  
"check_in_date" DATE NULL,  
"check_out_date" DATE NULL,  
"booking_payment_type" VARCHAR(45) NULL,  
"total_rooms_booked" INT NULL,  
"hotel_hotel_id" INT NOT NULL,  
"guests_guest_id" INT NOT NULL,  
"employees_emp_id" INT NOT NULL,  
"total_amount" DECIMAL(10,2) NULL,  
PRIMARY KEY ("booking_id", "hotel_hotel_id", "guests_guest_id", "employees_emp_id"),  
CONSTRAINT "fk_bookings_hotel1"  
FOREIGN KEY ("hotel_hotel_id")  
REFERENCES "hotel_database"."hotel" ("hotel_id")  
ON DELETE NO ACTION  
ON UPDATE NO ACTION,  
CONSTRAINT "fk_bookings_guests1"  
FOREIGN KEY ("guests_guest_id")  
REFERENCES "hotel_database"."guests" ("guest_id")
```



```
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT "fk_bookings_employees1"
FOREIGN KEY ("employees_emp_id" )
REFERENCES "hotel_database"."employees" ("emp_id" )
ON DELETE NO ACTION
ON UPDATE NO ACTION)
```

createing table : hotel\_chain\_has\_hotel

```
CREATE TABLE IF NOT EXISTS "hotel_database"."hotel_chain_has_hotel" (
"hotel_chains_hotel_chain_id" INT NOT NULL,
"hotels_hotel_id" INT NOT NULL,
PRIMARY KEY ("hotel_chains_hotel_chain_id", "hotels_hotel_id"),
CONSTRAINT "fk_hotel_chains_has_hotels_hotel_chains1"
FOREIGN KEY ("hotel_chains_hotel_chain_id")
REFERENCES "hotel_database"."hotel_chain" ("hotel_chain_id" )
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT "fk_hotel_chains_has_hotels_hotels1"
FOREIGN KEY ("hotels_hotel_id")
REFERENCES "hotel_database"."hotel" ("hotel_id")
ON DELETE NO ACTION
ON UPDATE NO ACTION)
```

createing table : room\_rate\_discount

```
CREATE TABLE IF NOT EXISTS "hotel_database"."room_rate_discount" (
"discount_id" INT NOT NULL,
"discount_rate" DECIMAL(10,2) NULL,
"start_month" INT NULL,
"end_month" INT NULL,
"room_type_room_type_id" INT NOT NULL,
PRIMARY KEY ("discount_id", "room_type_room_type_id"),
CONSTRAINT "fk_room_rate_discount_room_type1"
FOREIGN KEY ("room_type_room_type_id")
REFERENCES "hotel_database"."room_type" ("room_type_id")
ON DELETE NO ACTION
ON UPDATE NO ACTION)
```

createing table : rooms\_booked

```
CREATE TABLE IF NOT EXISTS "hotel_database"."rooms_booked" (
```



```
"rooms_booked_id" INT NOT NULL,
"bookings_booking_id" INT NOT NULL,
"rooms_room_id" INT NOT NULL,
PRIMARY KEY ("rooms_booked_id", "bookings_booking_id", "rooms_room_id"),
CONSTRAINT "fk_rooms_booked_bookings1"
FOREIGN KEY ("bookings_booking_id")
REFERENCES "hotel_database"."bookings" ("booking_id")
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT "fk_rooms_booked_rooms1"
FOREIGN KEY ("rooms_room_id")
REFERENCES "hotel_database"."rooms" ("room_id")
ON DELETE NO ACTION
ON UPDATE NO ACTION)
```

createing table : hotel\_services

```
CREATE TABLE IF NOT EXISTS "hotel_database"."hotel_services" (
"service_id" INT NOT NULL,
"service_name" VARCHAR(45) NULL,
"service_description" VARCHAR(100) NULL,
"service_cost" DECIMAL(10,2) NULL,
"hotel_hotel_id" INT NOT NULL,
PRIMARY KEY ("service_id", "hotel_hotel_id"),
CONSTRAINT "fk_hotel_services_hotel1"
FOREIGN KEY ("hotel_hotel_id")
REFERENCES "hotel_database"."hotel" ("hotel_id")
ON DELETE NO ACTION
ON UPDATE NO ACTION)
```

createing table : hotel\_services\_used\_by\_guests

```
CREATE TABLE IF NOT EXISTS "hotel_database"."hotel_services_used_by_guests" (
"service_used_id" INT NOT NULL,
"hotel_services_service_id" INT NOT NULL,
"bookings_booking_id" INT NOT NULL,
PRIMARY KEY ("service_used_id", "hotel_services_service_id", "bookings_booking_id"),
CONSTRAINT "fk_hotel_services_has_bookings_hotel_services1"
FOREIGN KEY ("hotel_services_service_id")
REFERENCES "hotel_database"."hotel_services" ("service_id")
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT "fk_hotel_services_has_bookings_bookings1"
FOREIGN KEY ("bookings_booking_id")
```



```
REFERENCES "hotel_database"."bookings" ("booking_id")
ON DELETE NO ACTION
ON UPDATE NO ACTION)
```

#### 4.2 Creating views :

createing view : booking\_summary\_by\_payment\_type

```
CREATE VIEW booking_summary_by_payment_type AS
SELECT b.booking_payment_type,
       count(rb.rooms_booked_id) AS total_rooms_booked,
       sum(b.total_amount) AS total_earnings
  FROM hotel_database.bookings b
       JOIN hotel_database.rooms_booked rb ON b.booking_id = rb.rooms_booked_id
 GROUP BY b.booking_payment_type;
```

createing view : hotel\_booking\_summary

```
CREATE VIEW hotel_booking_summary AS
SELECT h.hotel_name,
       count(b.booking_id) AS total_bookings,
       sum(b.total_amount) AS total_earnings
  FROM hotel_database.hotel h
       JOIN hotel_database.bookings b ON h.hotel_id = b.hotel_hotel_id
 GROUP BY h.hotel_name;
```

createing view : hotel\_employees

```
SELECT employees.emp_first_name AS "First Name",
       employees.emp_last_name AS "Last Name",
       employees.emp_email_address AS "Email Address",
       employees.emp_contact_number AS "Contact Number",
       department.department_name AS "Department"
  FROM hotel_database.employees
       JOIN hotel_database.department ON department.department_id =
           employees.department_department_id;
```

createing view : hotel\_guests

```
SELECT guests.guest_first_name AS "First Name",
       guests.guest_last_name AS "Last Name",
       guests.guest_email_address AS "Email Address",
       guests.guest_contact_number AS "Contact Number",
       addresses.country,
```



```
addresses.state,  
addresses.zipcode  
FROM hotel_database.guests  
JOIN hotel_database.addresses ON addresses.address_id = guests.addresses_address_id  
WHERE (guests.guest_id IN ( SELECT DISTINCT bookings.guests_guest_id  
FROM hotel_database.bookings  
WHERE bookings.hotel_hotel_id = 1));
```

createing view : hotels\_in\_chain\_with\_head\_office\_info

```
CREATE VIEW hotels_in_chain_with_head_office_info AS  
SELECT h.hotel_name,  
hc.hotel_chain_contact_number,  
hc.hotel_chain_email_address  
FROM hotel_database.hotel h  
JOIN hotel_database.hotel_chain_has_hotel hchh ON h.hotel_id = hchh.hotels_hotel_id  
JOIN hotel_database.hotel_chain hc ON hchh.hotel_chains_hotel_chain_id = hc.hotel_chain_id;
```

### 4.3 Queries :

How many distinct guest have made bookings for a particular month?

```
SELECT guest_first_name, guest_last_name, guest_contact_number  
FROM hotel_database.guests  
WHERE guest_id IN  
( SELECT distinct guests_guest_id  
FROM hotel_database.bookings  
WHERE EXTRACT(MONTH FROM check_in_date) = 8);
```

#### RA Expression :

$$\Pi_{guest\_first\_name, guest\_last\_name, guest\_contact\_number} (\sigma_{guest\_id \text{ IN } (\Pi_{guests\_guest\_id} (bookings))} (guests))$$

How many distinct guest have made bookings for a particular month?

```
SELECT guest_first_name, guest_last_name, guest_contact_number  
FROM hotel_database.guests  
WHERE guest_id IN  
( SELECT distinct guests_guest_id  
FROM hotel_database.bookings  
WHERE EXTRACT(MONTH FROM booking_date) = 8);
```

#### RA Expression :

$$\Pi_{guest\_first\_name, guest\_last\_name, guest\_contact\_number} (\sigma_{guest\_id \text{ IN } (\Pi_{guests\_guest\_id} (bookings))} (guests))$$



$(\sigma_{EXTRACT(MONTH FROM booking\_date)=8(bookings)}))$

How many hotels are in a hotel chain?

```
||| SELECT count(*) AS "Total Hotels"  
|||   FROM hotel_database.hotel_chain_has_hotel  
||| WHERE hotel_chains_hotel_chain_id = 1;
```

**RA Expression :**

$\Pi_{TotalHotels}(hotel\_chain\_has\_hotel)$

Find all the hotels in a particular city

```
||| SELECT hotel_name  
|||   FROM hotel_database.hotel  
|||   JOIN hotel_database.addresses ON addresses.address_id = hotel.addresses_address_id  
||| WHERE city = 'Mumbai';
```

**RA Expression :**

$\Pi_{hotel\_name}(hotel \bowtie \sigma_{city='Mumbai'}(addresses))$

Find the names and email addresses of all guests who have booked a room for more than 7 days

```
||| SELECT guest_first_name, guest_last_name, guest_email_address  
|||   FROM hotel_database.guests  
|||   JOIN hotel_database.bookings ON guests.guest_id = bookings.guests_guest_id  
||| WHERE CAST(duration_of_stay AS INTEGER) > 7;
```

**RA Expression :**

$\Pi_{guest\_first\_name, guest\_last\_name, guest\_email\_address}(guests \bowtie \sigma_{CAST(duration\_of\_stay AS INTEGER) > 7}(bookings))$

Find the names of all guests who have used the 'Laundry' service

```
||| SELECT guest_first_name, guest_last_name  
|||   FROM hotel_database.guests  
|||   JOIN hotel_database.bookings ON guests.guest_id = bookings.guests_guest_id  
|||   JOIN hotel_database.hotel_services_used_by_guests ON bookings.booking_id =  
|||           hotel_services_used_by_guests.bookings_booking_id  
|||   JOIN hotel_database.hotel_services ON hotel_services_used_by_guests.hotel_services_service_id =  
|||           hotel_services.service_id  
||| WHERE service_name = 'Laundry';
```

**RA Expression :**

$\Pi_{guest\_first\_name, guest\_last\_name}(guests \bowtie bookings \bowtie hotel\_services\_used\_by\_guests \bowtie  
||| \sigma_{service\_name='Laundry'}(hotel\_services))$

Find the names and contact numbers of all employees who work at the hotel with ID '1'

```
||| SELECT emp_first_name, emp_last_name, emp_contact_number
```



```
||| FROM hotel_database.employees
||| WHERE hotel_hotel_id = '1';
```

**RA Expression :**

$$\Pi_{emp\_first\_name, emp\_last\_name, emp\_contact\_number}(\sigma_{hotel\_id='1'}(employees))$$

Find the total number of rooms booked for each month in 2018

```
||| SELECT EXTRACT(MONTH FROM booking_date) as month, SUM(total_rooms_booked) as total_rooms_booked
||| FROM hotel_database.bookings
||| WHERE EXTRACT(YEAR FROM booking_date) = 2018
||| GROUP BY month;
```

**RA Expression :**

$$month \mathcal{G}_{SUM(total\_rooms\_booked)}(\Pi_{month, total\_rooms\_booked}(\sigma_{EXTRACT(YEAR FROM booking\_date)=2018}(bookings)))$$

Find the names of all hotels that are part of the 'China Town Hotels' chain

```
||| SELECT hotel_name
||| FROM hotel_database.hotel
||| JOIN hotel_database.hotel_chain_has_hotel ON hotel.hotel_id = hotel_chain_has_hotel.hotels_hotel_id
||| JOIN hotel_database.hotel_chain ON hotel_chain_has_hotel.hotel_chains_hotel_chain_id =
|||     hotel_chain.hotel_chain_id
||| WHERE hotel_chain_name = 'China Town Hotels';
```

**RA Expression :**

$$\Pi_{hotel\_name}(hotel \bowtie hotel\_chain\_has\_hotel \bowtie \sigma_{hotel\_chain\_name='China Town Hotels'}(hotel\_chain))$$

Listing:

**RA Expression :**

Find the names of all guests who booked a room in a hotel located in a specific city

```
||| SELECT g.guest_first_name, g.guest_last_name
||| FROM hotel_database.guests AS g
||| CROSS JOIN hotel_database.bookings AS b
||| CROSS JOIN hotel_database.hotel AS h
||| CROSS JOIN hotel_database.addresses AS a
||| WHERE h.hotel_id = b.hotel_hotel_id AND b.guests_guest_id = g.guest_id AND h.addresses_address_id =
|||     a.address_id AND a.city = 'Surrey';
```

**RA Expression :**

$$\Pi_{guests.guest\_first\_name, guests.guest\_last\_name}(guests \times bookings \times hotel \times \sigma_{addresses.city='Surrey'}(addresses))$$

Find all hotel chains and the hotels they are associated with, along with the address of the head office for each chain



```
SELECT hotel_chain.*, hotel_chain_has_hotel.*, addresses.*  
FROM hotel_database.hotel_chain  
LEFT JOIN hotel_database.hotel_chain_has_hotel ON hotel_chain.hotel_chain_id =  
    hotel_chain_has_hotel.hotel_chains_hotel_chain_id  
LEFT JOIN hotel_database.addresses ON hotel_chain.hotel_chain_head_office_address_id =  
    addresses.address_id;
```

#### RA Expression :

$$\Pi_{\text{hotel\_chain}.*,\text{hotel\_chain\_has\_hotel}.*,\text{addresses}.*}(\text{hotel\_chain} \bowtie \text{hotel\_chain\_has\_hotel} \bowtie \text{addresses})$$

Find all bookings and the payment type, along with the name and contact information of the employee who made the booking, even if the booking was not made by an employee, and the guest's contact information and the address of their permanent residence

```
SELECT bookings.*, employees.*, guests.*, addresses.*  
FROM hotel_database.bookings  
RIGHT JOIN hotel_database.employees ON bookings.employees_emp_id = employees.emp_id  
LEFT JOIN hotel_database.guests ON bookings.guests_guest_id = guests.guest_id  
LEFT JOIN hotel_database.addresses ON guests.addresses_address_id = addresses.address_id;
```

#### RA Expression :

$$\Pi_{\text{bookings}.*,\text{employees}.*,\text{guests}.*,\text{addresses}.*}(\text{bookings} \bowtie \text{bookings.employees\_emp\_id} = \text{employees.emp\_id} (\text{employees} \bowtie \text{guests.addresses\_address\_id} = \text{addresses.address\_id}) \bowtie \text{bookings.guests\_guest\_id} = \text{guests.guest\_id} (\text{addresses}))$$

Find the names and contact numbers of guests who have booked rooms in a hotel located in a city whose name starts with 'S' and have used the hotel's 'Laundry' service

```
SELECT guest_first_name, guest_last_name, guest_contact_number  
FROM hotel_database.guests  
WHERE guest_id IN  
    (SELECT guests_guest_id FROM hotel_database.bookings  
     WHERE hotel_hotel_id IN  
         (SELECT hotel_id FROM hotel_database.hotel  
          WHERE addresses_address_id IN  
              (SELECT address_id FROM hotel_database.addresses  
               WHERE city LIKE 'S%'))  
     AND booking_id IN  
         (SELECT bookings_booking_id FROM hotel_database.hotel_services_used_by_guests  
          WHERE hotel_services_service_id IN  
              (SELECT service_id FROM hotel_database.hotel_services  
               WHERE service_name = 'Laundry')))
```



#### RA Expression :

$$\Pi_{guest\_first\_name, guest\_last\_name, guest\_contact\_number} (guests \bowtie \sigma_{guest\_id} \text{ IN } (\Pi_{guests\_guest\_id} (bookings \bowtie hotel \bowtie addresses \bowtie \sigma_{service\_name='Laundry'} (hotel\_services))))$$

Find the names and contact numbers of guests who have booked rooms in a hotel located in a city whose name starts with 'S' or 'C'

```
SELECT guest_first_name, guest_last_name, guest_contact_number
FROM hotel_database.guests
WHERE guest_id IN
  (SELECT guests_guest_id FROM hotel_database.bookings
  WHERE hotel_hotel_id IN
    (SELECT hotel_id FROM hotel_database.hotel
    WHERE addresses_address_id IN
      (SELECT address_id FROM hotel_database.addresses
      WHERE city LIKE ANY (array['S%', 'C%']))))
```

#### RA Expression :

$$\Pi_{guest\_first\_name, guest\_last\_name, guest\_contact\_number} (guests \bowtie \sigma_{guest\_id} \text{ IN } (\Pi_{guests\_guest\_id} (bookings \bowtie hotel \bowtie addresses \bowtie \sigma_{city \text{ LIKE } 'S%' \text{ OR } city \text{ LIKE } 'C%'} (addresses))))$$

Find all the guests who have used the service "Laundry" during their stay at the hotel

```
SELECT g./*
FROM hotel_database.guests g
WHERE EXISTS (
  SELECT 1
  FROM hotel_database.hotel_services_used_by_guests hsug
  JOIN hotel_database.hotel_services hs ON hsug.hotel_services_service_id = hs.service_id
  WHERE hs.service_name = 'Laundry'
  AND hsug.bookings_booking_id IN (
    SELECT b.booking_id
    FROM hotel_database.bookings b
    WHERE b.guests_guest_id = g.guest_id
  )
);
```

#### RA Expression :

$$\Pi_{g.*} (guests \bowtie \sigma_{\text{EXISTS} (hotel\_services\_used\_by\_guests \bowtie \sigma_{service\_name='Laundry'} (hotel\_services)) \bowtie bookings \bowtie \sigma_{guests\_guest\_id=g.guest\_id} (g.guest\_id))}$$

Find all the guests who have used a unique service at least once during their stay at the hotel

```
SELECT DISTINCT g./*
FROM hotel_database.guests g
WHERE EXISTS (
```



```
||| SELECT 1
||| FROM hotel_database.hotel_services_used_by_guests hsug
||| WHERE hsug.bookings_booking_id IN (
|||     SELECT b.booking_id
|||     FROM hotel_database.bookings b
|||     WHERE b.guests_guest_id = g.guest_id
||| )
||| GROUP BY hsug.bookings_booking_id
||| HAVING COUNT(DISTINCT hsug.hotel_services_service_id) = 1
||| );
```

#### RA Expression :

$$\Pi_{guests.*}(guests \bowtie \sigma_{\text{EXISTS}(\Pi_1(hotel\_services\_used\_by\_guests \bowtie bookings \bowtie guests \bowtie \text{COUNT}(DISTINCT hotel\_services\_service\_id)=1))})$$

Set the check-out time for all hotels to 11:00 AM

```
||| UPDATE hotel_database.hotel
||| SET chek_out_time = '11:00:00';
```

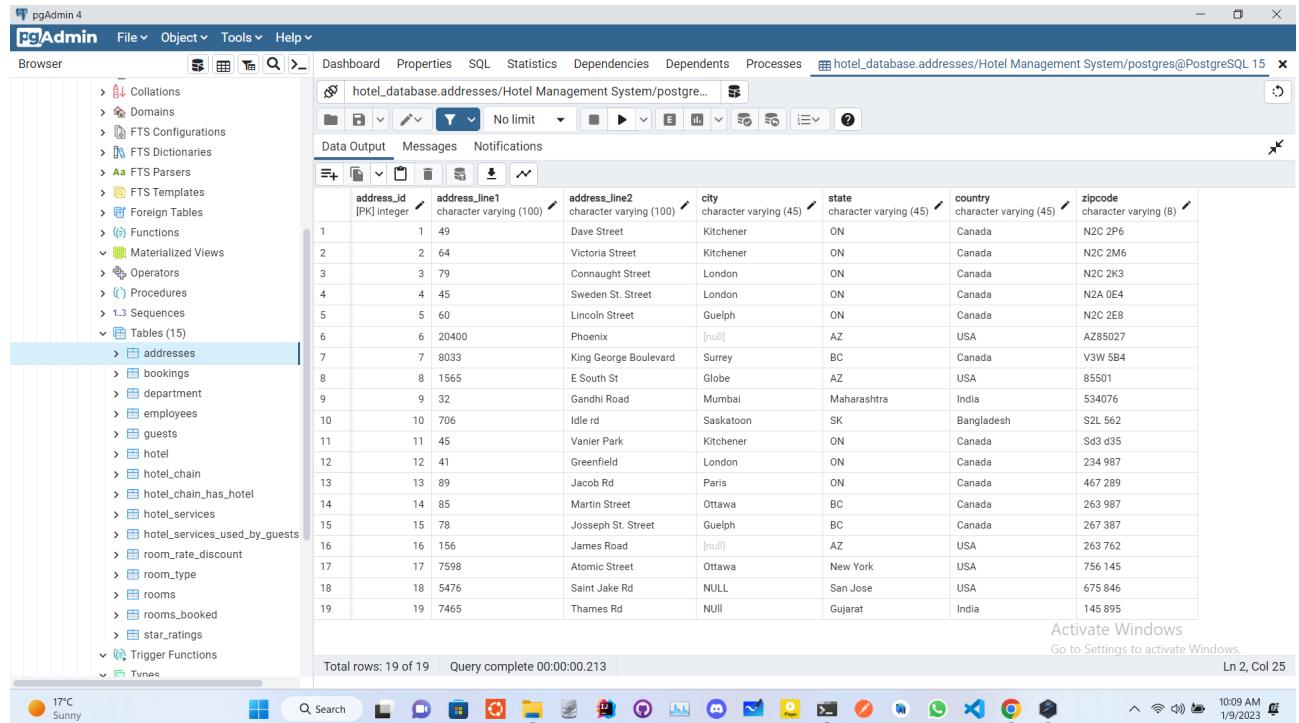
Delete all hotel chains that don't have any hotels associated with them

```
||| DELETE
||| FROM hotel_database.hotel_chain
||| WHERE hotel_chain_id NOT IN (
|||     SELECT hotel_chains_hotel_chain_id
|||     FROM hotel_database.hotel_chain_has_hotel);
```



## 5 Snapshots of the instances:

### 5.1 Data of table : addresses

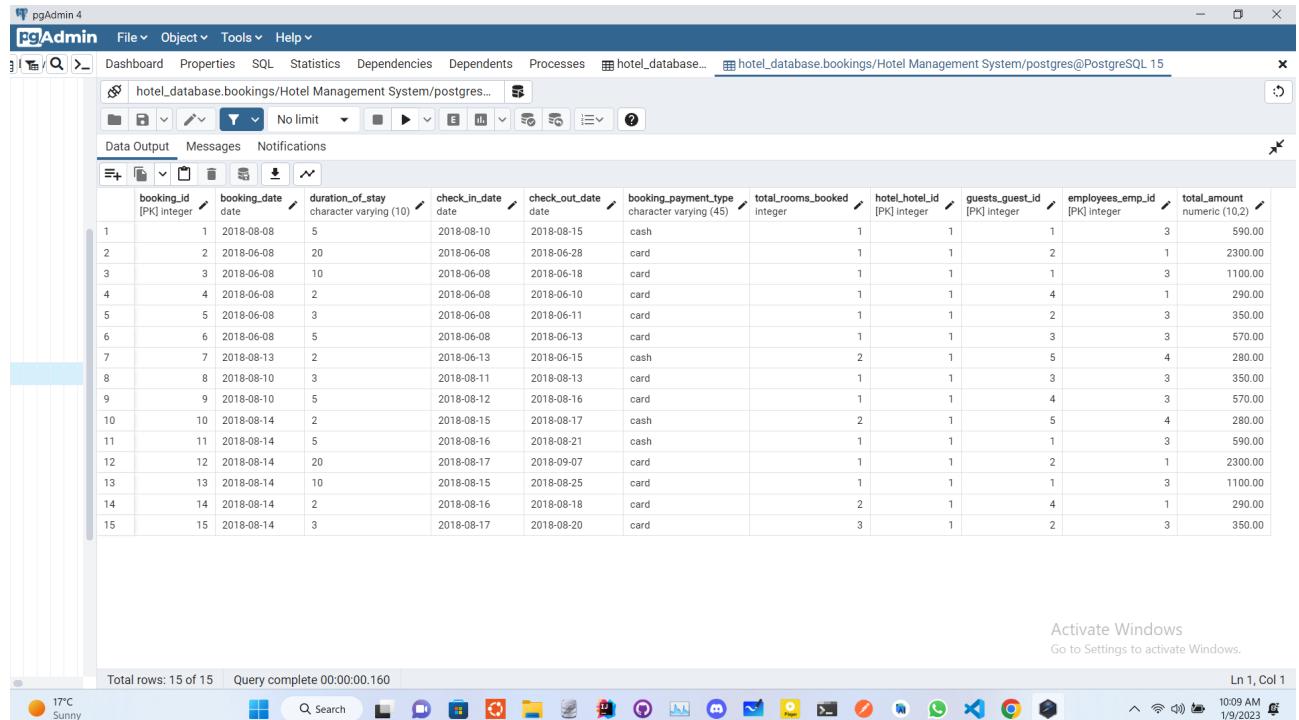


The screenshot shows the pgAdmin 4 interface with the 'addresses' table selected in the left sidebar. The main pane displays the table structure and data. The table has columns: address\_id, address\_line1, address\_line2, city, state, country, and zipcode. The data shows 19 rows of address information from around the world, including Canada, USA, and India.

	address_id	address_line1	address_line2	city	state	country	zipcode
	[PK] integer	character varying (100)	character varying (100)	character varying (45)	character varying (45)	character varying (45)	character varying (8)
1	1	49	Dave Street	Kitchener	ON	Canada	N2C 2P6
2	2	64	Victoria Street	Kitchener	ON	Canada	N2C 2M6
3	3	79	Connaught Street	London	ON	Canada	N2C 2K3
4	4	45	Sweden St. Street	London	ON	Canada	N2A 0E4
5	5	60	Lincoln Street	Guelph	ON	Canada	N2C 2E8
6	6	20400	Phoenix	[null]	AZ	USA	AZ85027
7	7	8033	King George Boulevard	Surrey	BC	Canada	V3W 5B4
8	8	1565	E South St	Globe	AZ	USA	85501
9	9	32	Gandhi Road	Mumbai	Maharashtra	India	534076
10	10	706	Idle rd	Saskatoon	SK	Bangladesh	S2L 562
11	11	45	Vanier Park	Kitchener	ON	Canada	Sd3 d35
12	12	41	Greenfield	London	ON	Canada	234 987
13	13	89	Jacob Rd	Paris	ON	Canada	467 289
14	14	85	Martin Street	Ottawa	BC	Canada	263 987
15	15	78	Joseph St. Street	Guelph	BC	Canada	267 387
16	16	156	James Road	[null]	AZ	USA	263 762
17	17	7598	Atomic Street	Ottawa	New York	USA	756 145
18	18	5476	Saint Jake Rd	NULL	San Jose	USA	675 846
19	19	7465	Thames Rd	NULL	Gujarat	India	145 895



## 5.2 Data of table : bookings



The screenshot shows the pgAdmin 4 interface with the 'bookings' table selected. The table has 15 rows and 12 columns. The columns are: booking\_id, booking\_date, duration\_of\_stay, check\_in\_date, check\_out\_date, booking\_payment\_type, total\_rooms\_booked, hotel\_hotel\_id, guests\_guest\_id, employees\_emp\_id, and total\_amount. The data shows various booking details, including dates, payment types (cash or card), and guest counts.

	booking_id [PK] integer	booking_date date	duration_of_stay character varying (10)	check_in_date date	check_out_date date	booking_payment_type character varying (45)	total_rooms_booked integer	hotel_hotel_id [PK] integer	guests_guest_id [PK] integer	employees_emp_id [PK] integer	total_amount numeric (10,2)
1	1	2018-08-08	5	2018-08-10	2018-08-15	cash	1	1	1	3	590.00
2	2	2018-06-08	20	2018-06-08	2018-06-28	card	1	1	2	1	2300.00
3	3	2018-06-08	10	2018-06-08	2018-06-18	card	1	1	1	3	1100.00
4	4	2018-06-08	2	2018-06-08	2018-06-10	card	1	1	4	1	290.00
5	5	2018-06-08	3	2018-06-08	2018-06-11	card	1	1	2	3	350.00
6	6	2018-06-08	5	2018-06-08	2018-06-13	card	1	1	3	3	570.00
7	7	2018-08-13	2	2018-08-13	2018-08-15	cash	2	1	5	4	280.00
8	8	2018-08-10	3	2018-08-11	2018-08-13	card	1	1	3	3	350.00
9	9	2018-08-10	5	2018-08-12	2018-08-16	card	1	1	4	3	570.00
10	10	2018-08-14	2	2018-08-15	2018-08-17	cash	2	1	5	4	280.00
11	11	2018-08-14	5	2018-08-16	2018-08-21	cash	1	1	1	3	590.00
12	12	2018-08-14	20	2018-08-17	2018-09-07	card	1	1	2	1	2300.00
13	13	2018-08-14	10	2018-08-15	2018-08-25	card	1	1	1	3	1100.00
14	14	2018-08-14	2	2018-08-16	2018-08-18	card	2	1	4	1	290.00
15	15	2018-08-14	3	2018-08-17	2018-08-20	card	3	1	2	3	350.00



### 5.3 Data of table : department

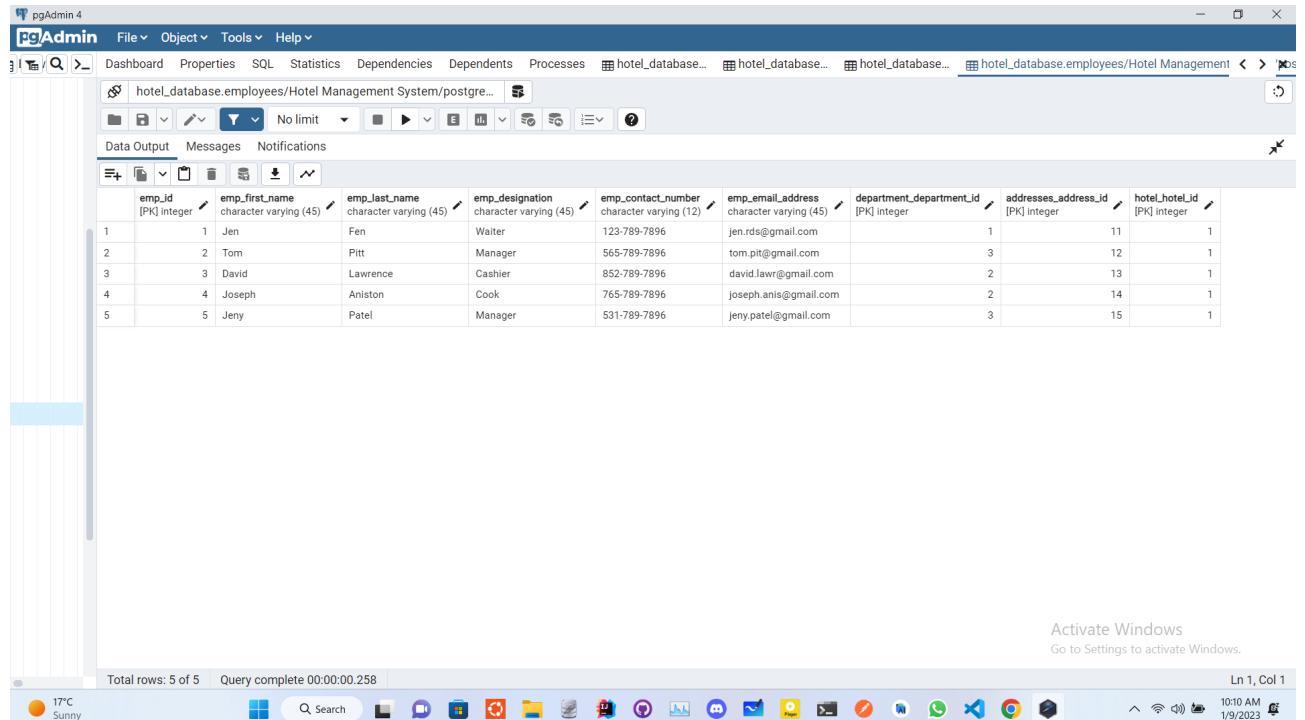
The screenshot shows the pgAdmin 4 interface. The left sidebar is a tree view of database objects, with 'Tables (15)' expanded and 'department' selected. The main pane displays the 'department' table with the following data:

department_id	department_name	department_description
1	Kitchen	cooking
2	Cleaning	sweep and mop
3	Front Staff	handle bookings and query resolution
4	Management	handles customer and resolve complaints
5	Commute	pick up and drop

At the bottom of the pgAdmin window, there is a status bar with 'Total rows: 5 of 5' and 'Query complete 00:00:00.213'. The system tray at the bottom of the screen shows the date and time as '1/9/2023 10:13 AM'.



## 5.4 Data of table : employees



The screenshot shows the pgAdmin 4 interface with the 'Data Output' tab selected. The title bar indicates the connection is to 'hotel\_database.employees/Hotel Management System/postgre...'. The main area displays the 'employees' table with the following data:

emp_id	emp_first_name	emp_last_name	emp_designation	emp_contact_number	emp_email_address	department_department_id	addresses_address_id	hotel_hotel_id
1	1	Jen	Fen	Walter	123-789-7896	jen.rds@gmail.com	1	11
2	2	Tom	Pitt	Manager	565-789-7896	tom.pit@gmail.com	3	12
3	3	David	Lawrence	Cashier	852-789-7896	david.lawr@gmail.com	2	13
4	4	Joseph	Aniston	Cook	765-789-7896	joseph.anis@gmail.com	2	14
5	5	Jeny	Patel	Manager	531-789-7896	jeny.patel@gmail.com	3	15

Total rows: 5 of 5    Query complete 00:00:00.258    Ln 1, Col 1

Activate Windows  
Go to Settings to activate Windows.

17°C Sunny

10:10 AM 1/9/2023

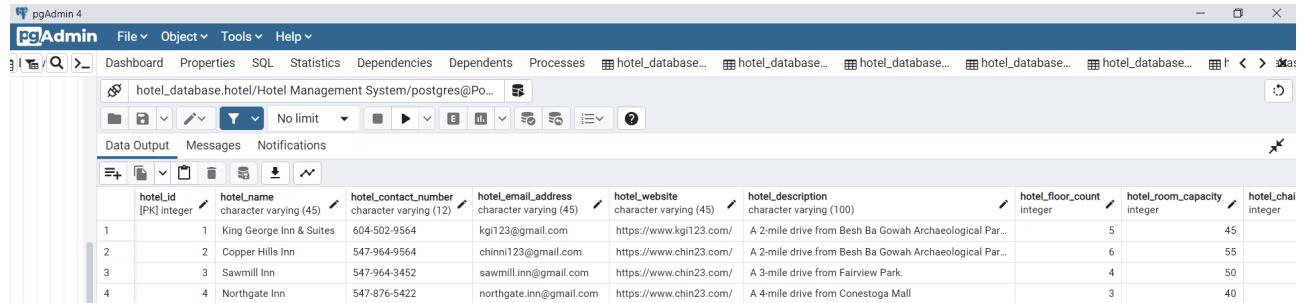


## 5.5 Data of table : guests

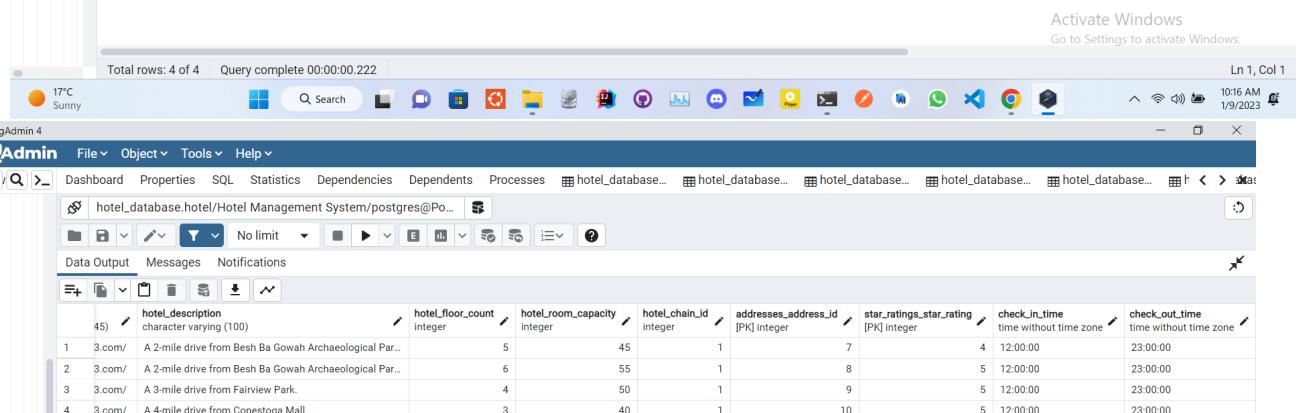
guest_id	guest_first_name	guest_last_name	guest_contact_number	guest_email_address	guest_credit_card	guest_id_proof	addresses_address_id
1	Jane	Doe	132-456-8564	jane.doe@gmail.com	[null]	/images/drivingLicense1023	1
2	Jerry	Thacher	564-896-4752	jerry.ytsg@gmail.com	[null]	/images/pasport45612	2
3	Rihanna	Perry	745-986-7451	rih.vfdj89@gmail.com	[null]	/images/drivingLicense4889	3
4	Mathew	Jose	489-624-8633	mathew.jose@gmail.com	[null]	/images/drivingLicense8945	4
5	Jessica	Smith	487-956-8963	jess.smith@gmail.com	[null]	/images/passport7896	5



## 5.6 Data of table : hotel



hotel_id	hotel_name	hotel_contact_number	hotel_email_address	hotel_website	hotel_description	hotel_floor_count	hotel_room_capacity	hotel_chain_id
1	1 King George Inn & Suites	604-502-9564	kgi123@gmail.com	https://www.kgi123.com/	A 2-mile drive from Besh Ba Gowah Archaeological Par...	5	45	
2	2 Copper Hills Inn	547-964-9564	chinni123@gmail.com	https://www.chin23.com/	A 2-mile drive from Besh Ba Gowah Archaeological Par...	6	55	
3	3 Sawmill Inn	547-964-3452	sawmill.inn@gmail.com	https://www.chin23.com/	A 3-mile drive from Fairview Park.	4	50	
4	4 Northgate Inn	547-876-5422	northgate.inn@gmail.com	https://www.chin23.com/	A 4-mile drive from Conestoga Mall	3	40	



hotel_id	hotel_name	hotel_contact_number	hotel_email_address	hotel_website	hotel_description	hotel_floor_count	hotel_room_capacity	hotel_chain_id
1	1 King George Inn & Suites	604-502-9564	kgi123@gmail.com	https://www.kgi123.com/	A 2-mile drive from Besh Ba Gowah Archaeological Par...	5	45	
2	2 Copper Hills Inn	547-964-9564	chinni123@gmail.com	https://www.chin23.com/	A 2-mile drive from Besh Ba Gowah Archaeological Par...	6	55	
3	3 Sawmill Inn	547-964-3452	sawmill.inn@gmail.com	https://www.chin23.com/	A 3-mile drive from Fairview Park.	4	50	
4	4 Northgate Inn	547-876-5422	northgate.inn@gmail.com	https://www.chin23.com/	A 4-mile drive from Conestoga Mall	3	40	



hotel_id	hotel_name	hotel_contact_number	hotel_email_address	hotel_website	hotel_description	hotel_floor_count	hotel_room_capacity	hotel_chain_id
1	1 King George Inn & Suites	604-502-9564	kgi123@gmail.com	https://www.kgi123.com/	A 2-mile drive from Besh Ba Gowah Archaeological Par...	5	45	
2	2 Copper Hills Inn	547-964-9564	chinni123@gmail.com	https://www.chin23.com/	A 2-mile drive from Besh Ba Gowah Archaeological Par...	6	55	
3	3 Sawmill Inn	547-964-3452	sawmill.inn@gmail.com	https://www.chin23.com/	A 3-mile drive from Fairview Park.	4	50	
4	4 Northgate Inn	547-876-5422	northgate.inn@gmail.com	https://www.chin23.com/	A 4-mile drive from Conestoga Mall	3	40	



## 5.7 Data of table : hotel\_chain

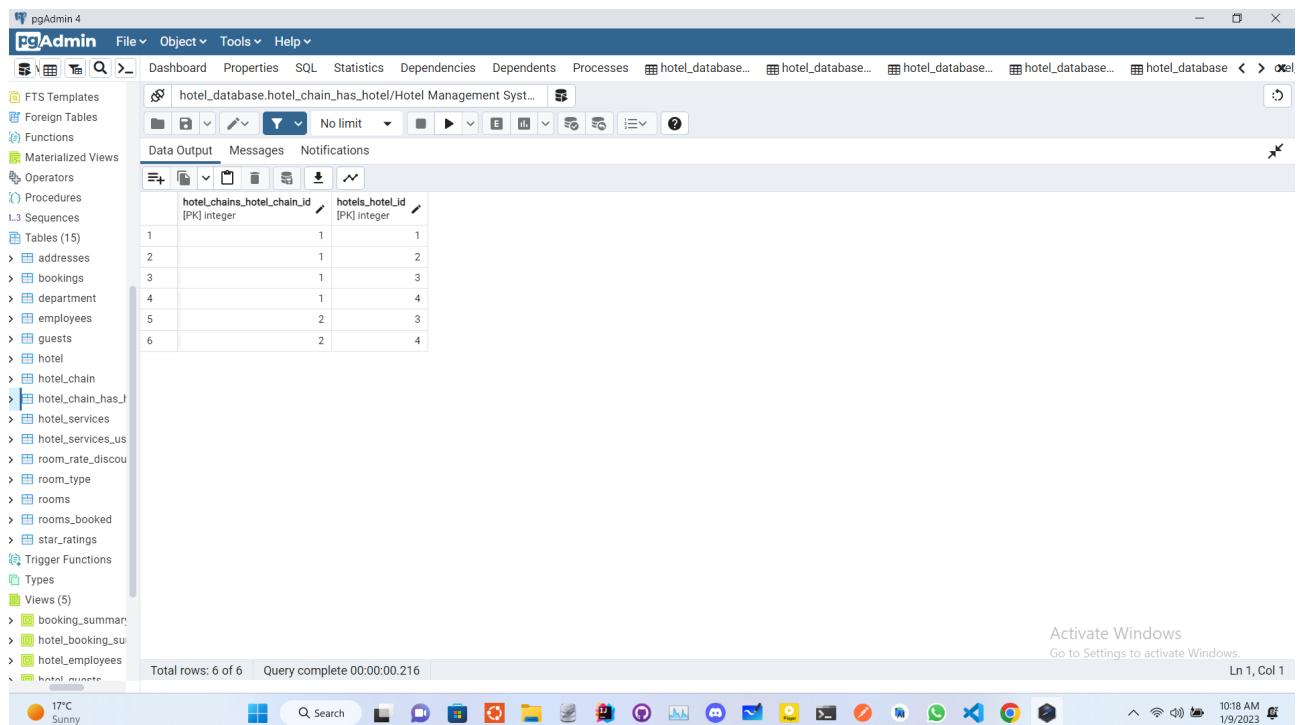
The screenshot shows the pgAdmin 4 interface with the 'hotel\_database...' database selected. The 'Tables' section is open, and the 'hotel\_chain' table is selected. The table has the following structure:

	hotel_chain_id	hotel_chain_name	hotel_chain_contact_number	hotel_chain_email_address	hotel_chain_website	hotel_chain_head_office_address_id
1	1	Best Western Hotels	456-865-8956	bw123@gmail.com	https://www.bestwestern.com/	6
2	2	China Town Hotels	110-526-5647	chinatown123@gmail.com	https://www.chinatown.com/	16

Below the table, the status bar shows 'Total rows: 2 of 2' and 'Query complete 00:00:00.194'. The system tray at the bottom indicates the date and time as '1/9/2023 10:17 AM'.



## 5.8 Data of table : hotel\_chain\_has\_hotel



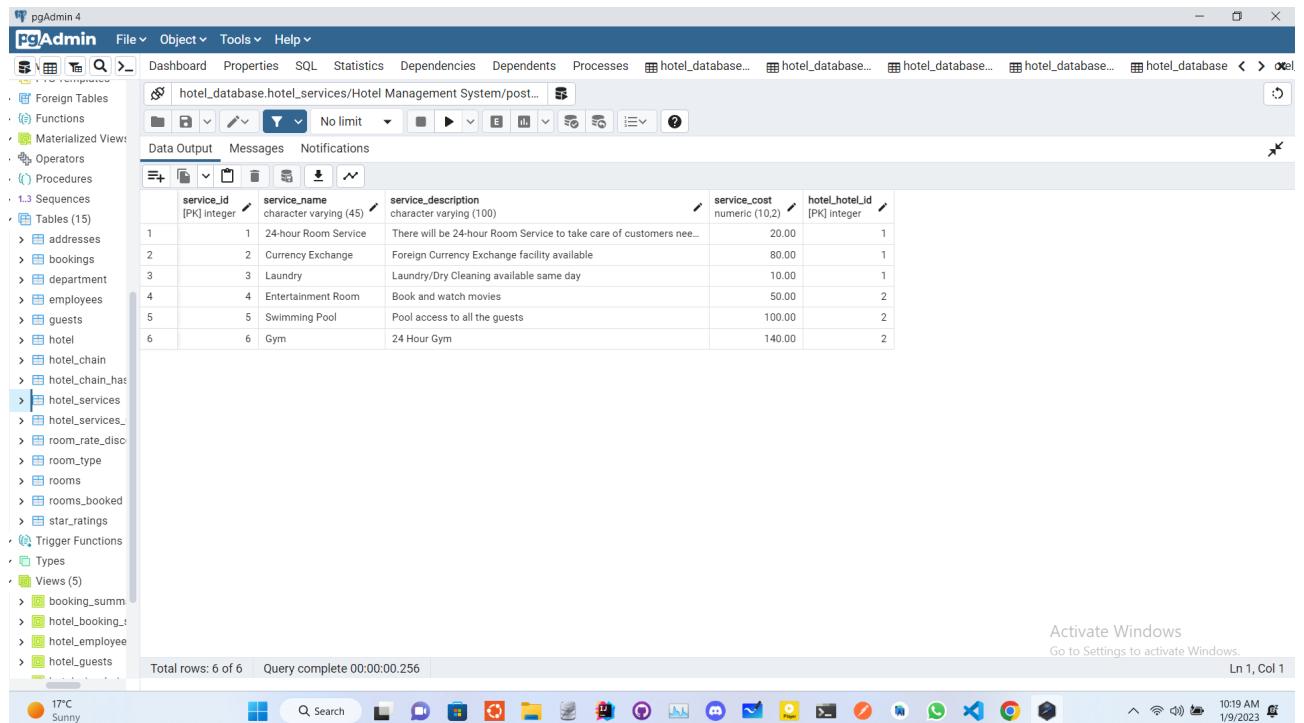
The screenshot shows the pgAdmin 4 interface with the 'hotel\_database...' database selected. The 'Tables (15)' section is expanded, and the 'hotel\_chain\_has\_hotel' table is selected. The table has two columns: 'hotel\_chains\_hotel\_chain\_id' and 'hotels\_hotel\_id', both of type [PK] integer. The data shows six rows:

	hotel_chains_hotel_chain_id	hotels_hotel_id
1	1	1
2	1	2
3	1	3
4	1	4
5	2	3
6	2	4

Total rows: 6 of 6    Query complete 00:00:00.216    Ln 1, Col 1



## 5.9 Data of table : hotel\_services



The screenshot shows the pgAdmin 4 interface with the 'hotel\_database' database selected. The 'Tables' section is expanded, and the 'hotel\_services' table is selected. The table has the following structure and data:

service_id	service_name	service_description	service_cost	hotel_hotel_id
1	1	24-hour Room Service	20.00	1
2	2	Currency Exchange	80.00	1
3	3	Laundry	10.00	1
4	4	Entertainment Room	50.00	2
5	5	Swimming Pool	100.00	2
6	6	Gym	140.00	2

Below the table, a message reads: 'Activate Windows  
Go to Settings to activate Windows.' The status bar at the bottom shows 'Total rows: 6 of 6' and 'Query complete 00:00:00.256'.



## 5.10 Data of table : hotel\_services\_used\_by\_guests

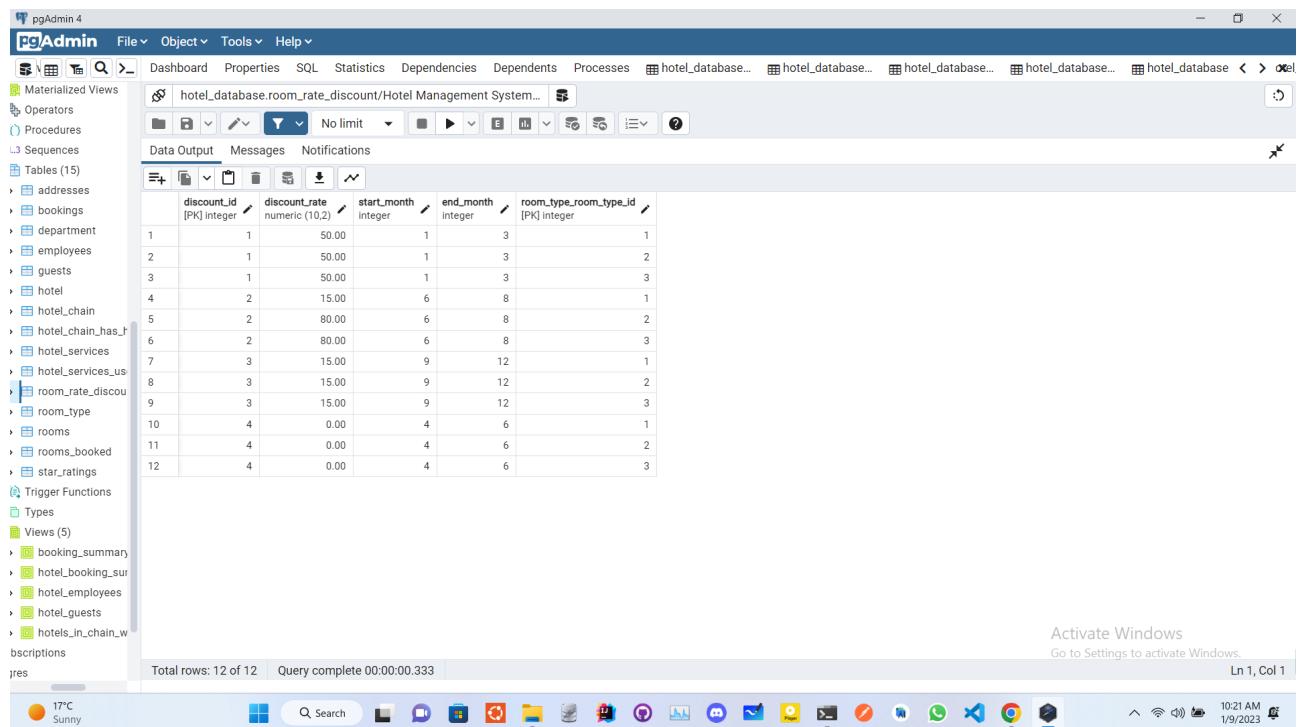
The screenshot shows the pgAdmin 4 interface with the 'hotel\_database.hotel\_services\_used\_by\_guests' connection selected. The 'Tables' node in the left sidebar is expanded, and the 'hotel\_services\_used\_by\_guests' table is selected. The table data is displayed in a grid format with the following columns: service\_used\_id, hotel\_services\_service\_id, and bookings\_booking\_id. The data consists of three rows:

service_used_id	hotel_services_service_id	bookings_booking_id
1	1	1
2	2	2
3	3	2

Below the table, the status bar shows 'Total rows: 3 of 3' and 'Query complete 00:00:00.272'. The system tray at the bottom of the screen shows the date and time as '1/9/2023 10:20 AM'.



## 5.11 Data of table : room\_rate\_discount

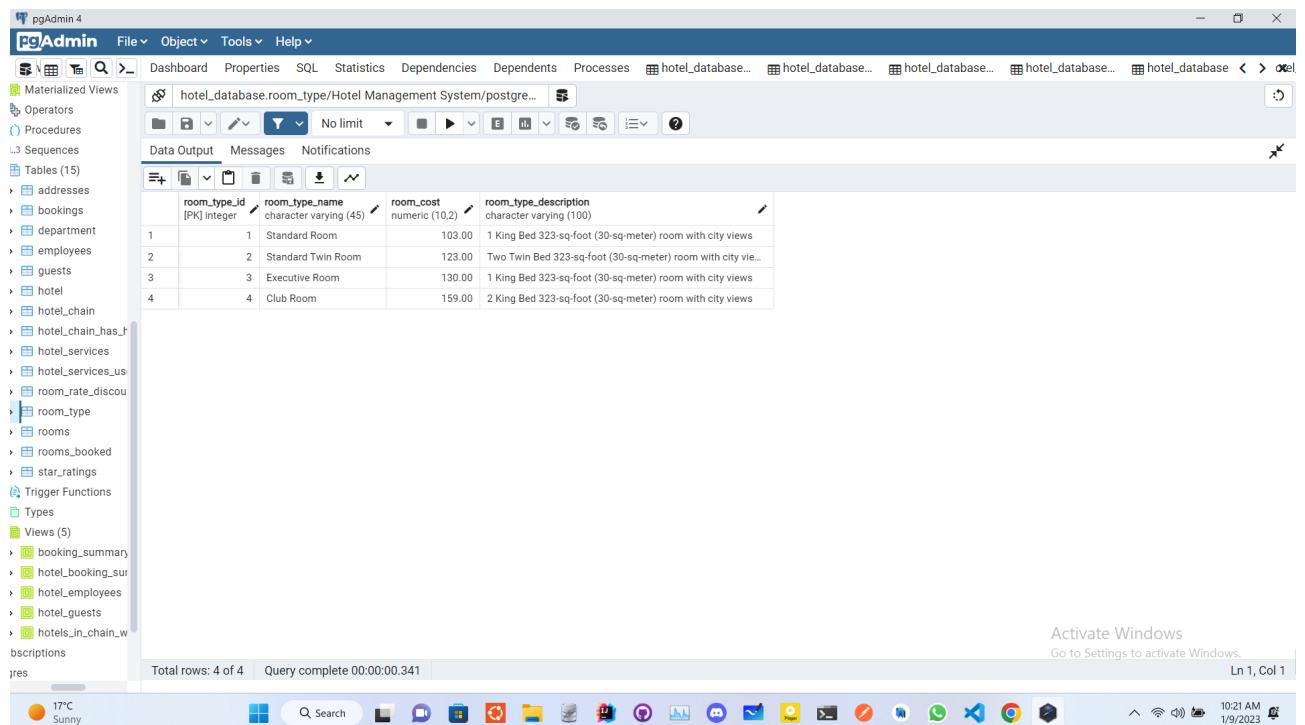


The screenshot shows the pgAdmin 4 interface with the 'room\_rate\_discount' table selected in the 'Tables (15)' section of the left sidebar. The table has four columns: discount\_id, discount\_rate, start\_month, and end\_month, with room\_type\_id as a foreign key. The data shows 12 rows of discount information for different room types.

	discount_id	discount_rate	start_month	end_month	room_type_id
1	1	50.00	1	3	1
2	1	50.00	1	3	2
3	1	50.00	1	3	3
4	2	15.00	6	8	1
5	2	80.00	6	8	2
6	2	80.00	6	8	3
7	3	15.00	9	12	1
8	3	15.00	9	12	2
9	3	15.00	9	12	3
10	4	0.00	4	6	1
11	4	0.00	4	6	2
12	4	0.00	4	6	3



## 5.12 Data of table : room\_type

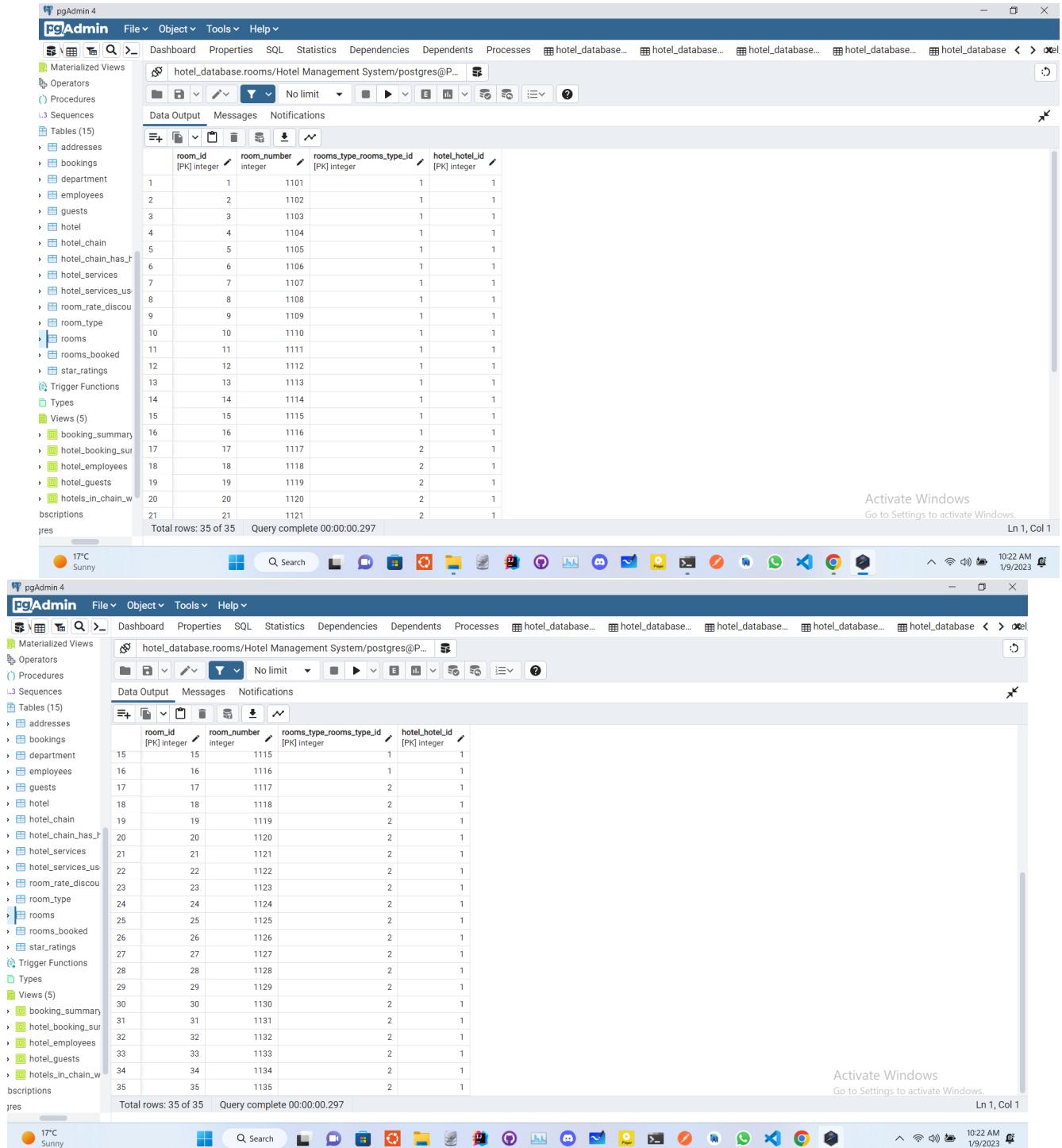


The screenshot shows the pgAdmin 4 interface with the 'hotel\_database' database selected. The 'Tables' section on the left shows 15 tables, including 'room\_type'. The 'room\_type' table is selected, and its data is displayed in a grid. The table has four columns: room\_type\_id, room\_type\_name, room\_cost, and room\_type\_description. The data is as follows:

room_type_id	room_type_name	room_cost	room_type_description
1	Standard Room	103.00	1 King Bed 323-sqfoot (30-sq-meter) room with city views
2	Standard Twin Room	123.00	Two Twin Bed 323-sqfoot (30-sq-meter) room with city views
3	Executive Room	130.00	1 King Bed 323-sqfoot (30-sq-meter) room with city views
4	Club Room	159.00	2 King Bed 323-sqfoot (30-sq-meter) room with city views



### 5.13 Data of table : rooms



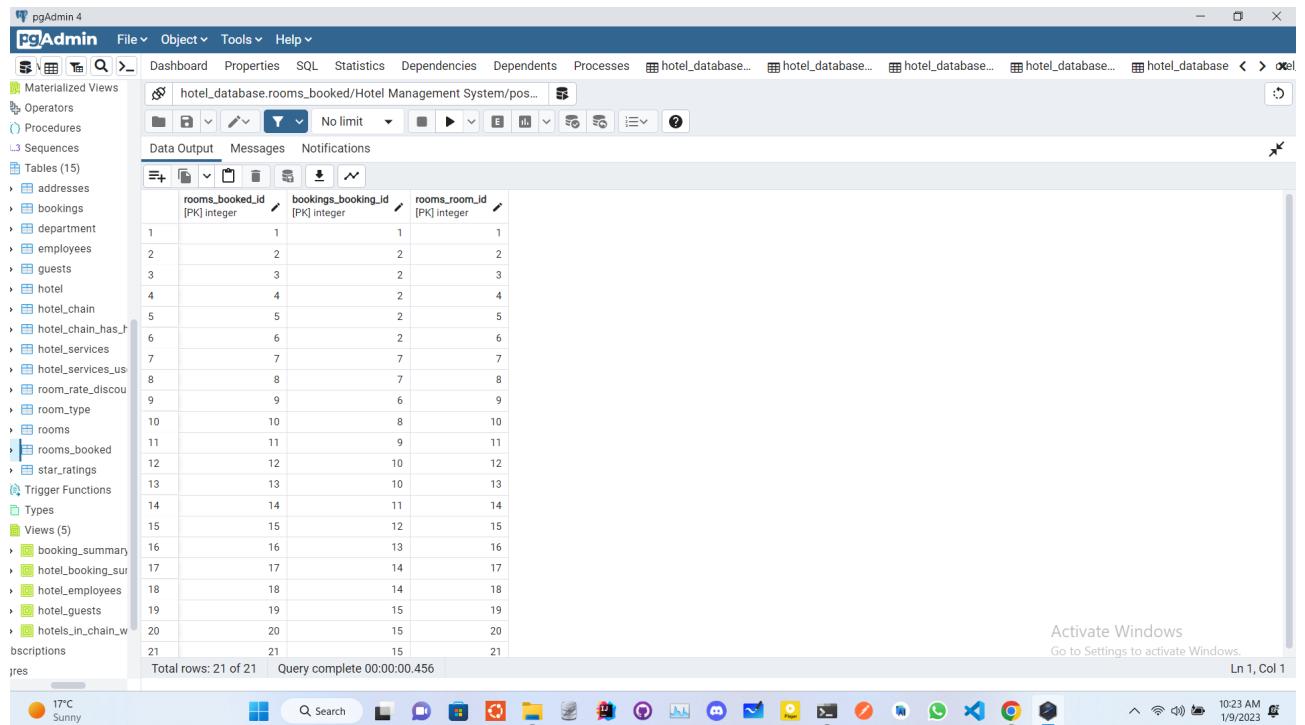
The screenshot shows two sessions of pgAdmin 4 displaying the data from the 'rooms' table in a PostgreSQL database. The table structure is as follows:

room_id	room_number	rooms_type_rooms_type_id	hotel_hotel_id
1	1101	1	1
2	1102	1	1
3	1103	1	1
4	1104	1	1
5	1105	1	1
6	1106	1	1
7	1107	1	1
8	1108	1	1
9	1109	1	1
10	1110	1	1
11	1111	1	1
12	1112	1	1
13	1113	1	1
14	1114	1	1
15	1115	1	1
16	1116	1	1
17	1117	2	1
18	1118	2	1
19	1119	2	1
20	1120	2	1
21	1121	2	1

Both sessions show 35 rows of data, with the query completed at 00:00:00.297. The system status bar indicates 17°C, Sunny, and the date/time as 1/9/2023, 10:22 AM.



## 5.14 Data of table : rooms\_booked



The screenshot shows the pgAdmin 4 interface with the 'Tables (15)' section selected. The 'rooms\_booked' table is highlighted. The table structure and data are as follows:

	rooms_booked_id	bookings_booking_id	rooms_room_id
1	1	1	1
2	2	2	2
3	3	2	3
4	4	2	4
5	5	2	5
6	6	2	6
7	7	7	7
8	8	7	8
9	9	6	9
10	10	8	10
11	11	9	11
12	12	10	12
13	13	10	13
14	14	11	14
15	15	12	15
16	16	13	16
17	17	14	17
18	18	14	18
19	19	15	19
20	20	15	20
21	21	15	21

Total rows: 21 of 21    Query complete 00:00:00.456



## 5.15 Data of table : star\_ratings

The screenshot shows the pgAdmin 4 interface for a PostgreSQL database named 'hotel\_database'. The 'Tables' node is expanded, showing 15 tables. The 'star\_ratings' table is selected and its data is displayed in a grid. The table has two columns: 'star\_rating' (integer, PK) and 'star\_rating\_image' (character varying(100)). The data shows five rows with star ratings 1 through 5 and their corresponding image paths: one\_star.jpg, two\_star.jpg, three\_star.jpg, four\_star.jpg, and five\_star.jpg.

star_rating	star_rating_image
1	/images/one_star.jpg
2	/images/two_star.jpg
3	/images/three_star.jpg
4	/images/four_star.jpg
5	/images/five_star.jpg

Total rows: 5 of 5    Query complete 00:00:00.327    Ln 1, Col 1



## 6 Non-trivial Functional Dependencies

Database normalization is the process of restructuring a relational database in accordance with a series of so-called normal forms in order to reduce data redundancy and improve data integrity. Generally, if a database is normalized until third normal form then it is considered to be normalized. We tried to normalize the database until 3NF(third normal form) and BCNF(Boyce Codd Normal Form).

### Proof of 3NF :

**Definition :** A relation is in 3NF if at least one of the following condition holds in every non-trivial function dependency  $X \rightarrow Y$  :

1.  $X$  is a super key.
2.  $Y$  is a prime attribute (each element of  $Y$  is part of some candidate key).

### Proof :

Suppose we have the set of all candidate keys  $S$ . It follows that this set will contain a composition of some primary keys and primary keys themselves too. Thus for every primary key  $X$ ,  $X \in S$ . Since all of our functional dependencies are trivial it follows that they are a composition of some primary keys. Therefore for any functional dependency  $X \rightarrow Y$  we have  $Y$  as a composition of primary keys i.e.  $Y \in P(K) \neq \emptyset$  where  $K = \{X : X \text{ is a Primary Key}\}$  and  $P(K)$  is the power-set of  $K$  containing all subsets of the set  $K$  including the empty set. This implies  $Y \subseteq C$  where  $C \in S$ . Thus it satisfies the 2<sup>nd</sup> condition and the proof is done

### Proof of BCNF :

**Definition :** A relational schema  $R$  is in Boyce–Codd normal form if and only if for every one of its dependencies  $X \rightarrow Y$ , at least one of the following conditions hold:

1.  $X \rightarrow Y$  is a trivial functional dependency ( $Y \subseteq X$ ),
2.  $X$  is a superkey for schema  $R$ .

### Proof :

Since all of our functional dependencies are trivial, the proof is trivial as it follows directly from the definition of BCNF itself.



## 7 Frontend Tools

For the frontend we have decided to use Budibase, as it allows for the deployment of quick and easy CRUD applications. It meets our needs well and has helped us visualize all the data that we have easier into different sections we have. In the frontend it is possible for there to be normal "Basic" Users and "Admin" Users. Each different type of user will see different pages as per their ability to change data/only view data. As an example an admin would be able to see the data and change/add new data whereas someone with basic access would only be able to see the data.



## 8 Conclusions

This was an attempt to create a database management system for hotel where a DBA can easily manage the hotels, rooms, bookings, guests, employees, departments, services, etc. and other things as well, easily and quickly. Overall, it is huge area and we tried to cover few of the parts of it.