Lab 8 QUERIES

202012025-28_202018013-14

- Riya Dineshkumar Soni (202012025)
- Kakkan Anurag Kishor (202012026)
- Gandhi Viral Ashok (202012027)
- Sukhadia Rutvi Kumarpal (202012028)
- Shah Siddhant Alkeshbhai (202018013)
- Shah Nihar Shaileshbhai (202018014)

--SQL Queries and Relational Algebra

SET search_path TO Tourism_Management_System;

--1) Retrieve the packages within a particular price range

Relational Algebra:

$\sigma_{(amount >=15000 \text{ AND } amount <= 20000)}(package)$

SQL Query:

SELECT * FROM package WHERE amount >= 15000 and amount <= 20000;

postgres/postgres@PostgreSQL 13

Query Editor Query History

```
set search_path to tourism_management_system;
SELECT * FROM package WHERE amount >= 15000 and amount <= 20000;
```

Data Output Explain Messages Notifications

	packageid [PK] integer	title character varying (15)	duration integer	no_of_people integer	isactive boolean	amount double precision
1	2	Srinagar	4	2	true	19500
2	3	Varansi	4	4	true	15000
3	4	Kedarnath	6	2	true	20000

--2) Show the list of top 5 packages based on the number of users who selected it.

Relational Algebra:

r1 -> $\rho(bfp, {}_{packageid}\mathcal{F}_{COUNT(bid)}(Booking\_for\_package))$

r2 -> $\rho(p, package) \bowtie_{<p.packageid = bfp.packageid>}(r1)$

result -> $\Pi_{p.packageid,\ title,\ duration,\ no\_of\_people,\ amount}(r2)$

SQL Query:

SELECT p.packageid, title, duration, no_of_people, amount FROM package as p

JOIN

(SELECT COUNT(bid), packageid FROM Booking_for_package GROUP BY packageid) as bfp ON (bfp.packageID = p.packageID);

Query Editor Query History

```
set search_path to tourism_management_system;
SELECT p.packageid, title, duration, no_of_people, amount FROM package as p
JOIN
(SELECT COUNT(bid), packageid FROM Booking_for_package GROUP BY packageid) as bfp ON (bfp.packageID = p.p
```

Data Output Explain Messages Notifications

	packageid [PK] integer	title character varying (15)	duration integer	no_of_people integer	amount double precision
1	1	Manali Tour	9	8	64000
2	5	Taj Mahal	2	4	12000
3	9	Lonavla	2	6	12000

--3) View list of all the tourist spots at a particular location.

Relational Algebra:

r1 -> ρ(ts, tourist_spots) $\bowtie_{<ts.pincode = l.pincode>}$ ρ(l, location)

result -> Π(“Name”, season, ratings, CONCAT (ts.address,', ',l.city,', ',l.state, ' - ',ts.pincode) -> Address) (σ l.city=”kullu”(r1))

SQL Query:

```
SELECT ts."Name", ts.season, ts.ratings,
 CONCAT (ts.address,', ',l.city,', ',l.state, ' - ',ts.pincode) AS "Address"
FROM tourist_spots AS ts JOIN location AS l ON ts.pincode = l.pincode
where l.city = 'Kullu';
```

Query Editor Query History

```
1 set search_path to tourism_maSELECT ts."Name", ts.season, ts.ratings,
2 SELECT ts."Name", ts.season, ts.ratings,
3  CONCAT (ts.address,', ',l.city,', ',l.state, ' - ',ts.pincode) AS "Address"
4 FROM tourist_spots AS ts JOIN location AS l ON ts.pincode = l.pincode
5 where l.city = 'Kullu';
6
7
8
```

Data Output Explain Messages Notifications

	Name character varying (20)	season character varying (10)	ratings double precision	Address text
1	Manali	Winter	4.7	Rotang Pass...

--4) View the tourist spots included in “abc” package.

Relational Algebra:

r1 -> ρ(l, location) $\bowtie_{<l.pincode = ts.pincode>}$ ρ(ts, tourist_spots) $\bowtie_{<ts.spotid = pt.spotid>}$ ρ(pt, package_includes_spots) $\bowtie_{<pt.packageid = p.packageid>}$ ρ(p, package)

result -> Π(title, duration, no_of_people, amount, “Name”, rating, CONCAT (ts.address,', ',l.city,', ',l.state, ' - ',ts.pincode) -> Address) ($\sigma_{title=”Manali Tour”}$(r1))

SQL Query:

```
SELECT p.title, p.duration, p.no_of_people, p.amount, ts."Name", ts.season, ts.ratings,
CONCAT (ts.address,', ',l.city,', ',l.state, ' - ',ts.pincode) AS "Address"
FROM tourist_spots AS ts
JOIN package_includes_spots pt ON(pt.spotid = ts.spotid)
JOIN package AS p ON (p.packageid = pt.packageid)
JOIN location AS l ON ts.pincode = l.pincode
where p.title = 'Manali Tour';
```

Query Editor Query History Scratch Pad

```
1  set search_path to tourism_maSELECT ts."Name", ts.season, ts.ratings,
2  SELECT p.title, p.duration, p.no_of_people, p.amount, ts."Name", ts.season, ts.ratings,
3  CONCAT (ts.address,', ',l.city,', ',l.state, ' - ',ts.pincode) AS "Address"
4  FROM tourist_spots AS ts
5  JOIN package_includes_spots pt ON(pt.spotid = ts.spotid)
6  JOIN package AS p ON (p.packageid = pt.packageid)
7  JOIN location AS l ON ts.pincode = l.pincode
8  where p.title = 'Manali Tour';
9
10
11
12
```

Data Output Explain Messages Notifications

	title character varying (15)	duration integer	no_of_people integer	amount double precision	Name character varying (20)	season character varying (10)	ratings double precision	Address text
1	Manali Tour	9	8	64000	Manali	Winter	4.7	Rotang Pass...

--5) Retrieve the tourist spot with highest user ratings

Relational Algebra:

r1 -> $\mathcal{F}_{MAX(ratings)->ratings}$ (ρ(ts2, tourist_spots))

r2 -> r1 $\bowtie_{<ts2.ratings = ts1.ratings>}$ ρ(ts1, tourist_spots) $\bowtie_{<ts1.pincode = l.pincode>}$ ρ(l, location))

result -> $\Pi_{\text{"Name", season, ts2.ratings, address, t1.pincode, city, state}}$(r2)

SQL Query:

```
SELECT "Name", season, ts2.ratings,
CONCAT (ts1.address,', ', l.city,', ',l.state, ' - ',ts1.pincode) AS "Address"
FROM tourist_spots AS ts1
JOIN
(SELECT MAX (ratings) AS ratings FROM tourist_spots) AS ts2
ON (ts1.ratings = ts2.ratings)
JOIN Location as l
ON (ts1.pincode = l.pincode);
```

Query Editor Query History

```
set search_path to tourism_maSELECT ts."Name", ts.season, ts.ratings,
SELECT "Name", season, ts2.ratings,
CONCAT (ts1.address,', ', l.city,', ',l.state, ' - ',ts1.pincode) AS "Address"
FROM tourist_spots AS ts1
JOIN
(SELECT MAX (ratings) AS ratings FROM tourist_spots) AS ts2
ON (ts1.ratings = ts2.ratings)
JOIN Location as l
ON (ts1.pincode = l.pincode);
```

Data Output Explain Messages Notifications

	Name character varying (20)	season character varying (10)	ratings double precision	Address text
1	Golden	All	4.9	Golden Tem...
2	Taj	All	4.9	Dharmapuri, ...

--6) View all the restaurants that serve “only veg” food.

Relational Algebra:

result -> Π(“Name”, phone, foodtype, rating, CONCAT (r.address,', ', l.city, ', ', l.state, ' - ', r.pincode) -> Address) (σ foodtype=”Veg”(ρ(r, restaurant) $\bowtie$<r.pincode = l.pincode> ρ(l, location)))

SQL Query:

SELECT r."Name", r.phone, r.foodType, r.ratings,

CONCAT (r.address,', ', l.city, ', ', l.state, ' - ', r.pincode) AS "Address"

FROM restaurant AS r JOIN location AS l ON r.pincode = l.pincode

where r.foodType = 'VEG';

```
set search_path to tourism_management_system;
SELECT r."Name", r.phone, r.foodType, r.ratings,
CONCAT (r.address,', ', l.city, ', ', l.state, ' - ', r.pincode) AS "Address"
FROM restaurant AS r JOIN location AS l ON r.pincode = l.pincode
where r.foodType = 'VEG';
```

Data Output Explain Messages Notifications

	Name character varying (50)	phone numeric (10)	foodtype character varying (20)	ratings double precision	Address text
1	Jahanpanah	9898456721	VEG	4.5	E 23, Shoppi...
2	Huber &	9889855455	VEG	2.5	Shreekunj M...
3	Cryo Lab	9876543210	VEG	3	Ground Floo...
4	Subway	6826589432	VEG	4.5	207/53, Mah...
5	Open Hand	9898569825	VEG	3	B1/128-3, D...

--7) Retrieve list of all the restaurants at “abc” location.

Relational Algebra:

result -> $\Pi_{(\text{“Name”, phone, foodtype, rating, CONCAT (r.address,', ', l.city, ', ', l.state, ' - ', r.pincode) -> Address})}$ ($\sigma_{\text{city=”Ahmedabad”}}$(ρ(r, restaurant) $\bowtie_{<\text{r.pincode = l.pincode}>}$ ρ(l, location)))

SQL Query:

SELECT r."Name", r.phone, r.foodType, r.ratings,

CONCAT (r.address,', ', l.city, ', ', l.state, ' - ', r.pincode) AS "Address"

FROM restaurant AS r JOIN location AS l ON r.pincode = l.pincode

where l.city = 'Ahmedabad';

Query Editor Query History

```
set search_path to tourism_management_system;
SELECT r."Name", r.phone, r.foodType, r.ratings,
CONCAT (r.address,', ', l.city, ', ', l.state, ' - ', r.pincode) AS "Address"
FROM restaurant AS r JOIN location AS l ON r.pincode = l.pincode
where l.city = 'Ahmedabad';
```

Data Output Explain Messages Notifications

	Name character varying (50)	phone numeric (10)	foodtype character varying (20)	ratings double precision	Address text
1	Huber &	9889855455	VEG	2.5	Shreekunj M...
2	Cryo Lab	9876543210	VEG	3	Ground Floo...

--8) View all the restaurants that have “Chinese” cuisine included in their menu.

Relational Algebra:

r1 -> ρ(l, location) $\bowtie_{<l.pincode = r.pincode>}$ ρ(r, restaurant) $\bowtie_{<r.rid = rc.rid>}$ ρ(rc, restaurant_cuisines)

result -> $\Pi_{(\text{“Name”, phone, foodtype, ratings, cuisines, CONCAT (r.address,', ', l.city, ', ', l.state, ' - ', r.pincode) -> Address})}$ ($\sigma_{\text{cuisines=”Chinese”}}$(r1))

SQL Query:

```
SELECT r."Name", r.phone, r.foodType, r.ratings, rc.cuisines,
CONCAT (r.address,', ', l.city, ', ', l.state, ' - ', r.pincode) AS "Address"
FROM restaurant AS r JOIN location AS l ON r.pincode = l.pincode
JOIN restaurant_cuisines AS rc ON r.rid = rc.rid WHERE rc.cuisines = 'Chinese';
```

Query Editor Query History

```
1 set search_path to tourism_management_system;
2 SELECT r."Name", r.phone, r.foodType, r.ratings, rc.cuisines,
3 CONCAT (r.address,', ', l.city, ', ', l.state, ' - ', r.pincode) AS "Address"
4 FROM restaurant AS r JOIN location AS l ON r.pincode = l.pincode
5 JOIN restaurant_cuisines AS rc ON r.rid = rc.rid WHERE rc.cuisines = 'Chinese';
6
7
8
```

Data Output Explain Messages Notifications

	Name character varying (50)	phone numeric (10)	foodtype character varying (20)	ratings double precision	cuisines character varying (20)	Address text
1	Jahanpanah	9898456721	VEG	4.5	Chinese	E 23, Shoppi...
2	Huber &	9889855455	VEG	2.5	Chinese	Shreekunj M...
3	Tandoor	7954215885	NON-VEG	4	Chinese	17/33, Maha...
4	Three Dots	7878252364	NON-VEG	3.5	Chinese	840/1,100 F...
5	ECHOES	9465853246	BOTH	3	Chinese	44, 4th B Cr...

--9) Retrieve all the hotels that are situated at location “xyz”.

Relational Algebra:

result -> $\Pi_{(\text{“Name”, phone, foodtype, ratings, cuisines, CONCAT (h.address,', ', l.city,', ', l.state,' - ', h.pincode) -> Address})}$ $(\sigma_{\text{city=”Ahmedabad”}}$ $(\rho(h, \text{hotel}) \bowtie_{<h.pincode = l.pincode>} \rho(l, \text{location})))$

SQL Query:

SELECT h."Name", h.phone, h.foodType, h.ratings,

CONCAT (h.address,', ', l.city,', ', l.state,' - ', h.pincode) AS "Address"

FROM hotel AS h JOIN location AS l ON h.pincode = l.pincode WHERE l.city = 'Ahmedabad';

Query Editor Query History

```
set search_path to tourism_management_system;
SELECT h."Name", h.phone, h.foodType, h.ratings,
CONCAT (h.address,', ', l.city,', ', l.state,' - ', h.pincode) AS "Address"
FROM hotel AS h JOIN location AS l ON h.pincode = l.pincode WHERE  l.city = 'Ahmedabad';
```

Data Output Explain Messages Notifications

	Name character varying (50)	phone numeric (10)	foodtype character varying (20)	ratings double precision	Address text
1	Ahmedabad	9878456512	BOTH	4.5	Vastrapur,Ah...
2	Hyatt	8794561251	BOTH	4.8	Opp Ahmed...
3	The Metropole	7889451575	BOTH	4.2	Near R.T.O. ...

--10) Retrieve list of hotels that are providing “xyz” services.

Relational Algebra:

r1 -> $\rho(l, \text{location}) \bowtie_{<l.pincode = h.pincode>} \rho(h, \text{hotel}) \bowtie_{<h.hotelid = hs.hotelid>} \rho(hs, \text{hotel\_services})$

result -> $\Pi_{(\text{“Name”, phone, foodtype, ratings, services, CONCAT (h.address,', ', l.city,', ', l.state,' - ', h.pincode) -> Address})}$ $(\sigma_{\text{servies=”Gym”}}(r1))$

SQL Query:

SELECT h."Name", h.phone, h.foodType, h.ratings,hs.services,

CONCAT (h.address,', ', l.city,', ', l.state,' - ', h.pincode) AS "Address"

FROM hotel AS h

JOIN location AS l ON h.pincode = l.pincode

JOIN hotel_services AS hs ON h.hotelid = hs.hotelid

where hs.services = 'Gym';

Query Editor Query History

```
set search_path to tourism_management_system;
SELECT h."Name", h.phone, h.foodType, h.ratings,hs.services,
CONCAT (h.address,', ', l.city,', ', l.state,' - ', h.pincode) AS "Address"
FROM hotel AS h
JOIN location AS l ON h.pincode = l.pincode
JOIN hotel_services AS hs ON h.hotelid = hs.hotelid
where hs.services = 'Gym';
```

Data Output Explain Messages Notifications

	Name character varying (50)	phone numeric (10)	foodtype character varying (20)	ratings double precision	services character varying (50)	Address text
1	Hotel Thomas	9855004767	VEG	3.5	Gym	Simsa Villag...
2	Punjab Sindh	7858495615	BOTH	3	Gym	Main Market...
3	Jaisalkot - A	9116010801	VEG	4.6	Gym	Kuldhara Tu...
4	Statue of Unity	9797949494	VEG	4.5	Gym	Sardar Saro...
5	Aurick Hotel	6658498756	NON-VEG	3	Gym	15th Cross, ...

--11) Retrieve the hotel with highest user ratings

Relational Algebra:

r1 -> $\mathcal{F}_{MAX(ratings)}$ (hotel)

r2 -> $_{>}$ ρ(h, hotel) $\bowtie_{<h.pincode = l.pincode>}$ ρ(l, location)

result -> $\Pi_{(\text{“Name”, phone, foodtype, ratings, services, CONCAT (h.address,', ', l.city,', ', l.state,' - ', h.pincode) -> Address})}$ ($\sigma_{\text{ratings IN (r1)=”Gym”}}$(r2))

SQL Query:

SELECT h."Name", h.phone, h.foodType, h.ratings,

CONCAT (h.address,', ', l.city,', ', l.state,' - ', h.pincode) AS "Address"

FROM hotel AS h JOIN location AS l ON h.pincode = l.pincode

where h.ratings IN (SELECT max(ratings) from hotel);

Query Editor Query History

```
set search_path to tourism_management_system;
SELECT h."Name", h.phone, h.foodType, h.ratings,
CONCAT (h.address,', ', l.city,', ', l.state,' - ', h.pincode) AS "Address"
FROM hotel AS h JOIN location AS l ON h.pincode = l.pincode
where h.ratings IN (SELECT max(ratings) from hotel);
```

Data Output Explain Messages Notifications

	Name character varying (50)	phone numeric (10)	foodtype character varying (20)	ratings double precision	Address text
1	Hyatt	8794561251	BOTH	4.8	Opp Ahmed...

--12) Retrieve list of hotels sorted according to their user ratings.

<u>Relational Algebra:</u>

r1-> ρ(h, hotel) $\bowtie_{<h.pincode = l.pincode>}$ ρ(l, location)

result -> $\Pi_{(\text{“Name”, phone, foodtype, ratings, services, CONCAT (h.address,', ', l.city,', ', l.state,' - ', h.pincode) -> Address})}$ (σ ORDER BY $_{h.ratings}$ (r1))

<u>SQL Query:</u>

SELECT h."Name", h.phone, h.foodType, h.ratings,

CONCAT (h.address,', ', l.city,', ', l.state,' - ', h.pincode) AS "Address"

FROM hotel AS h JOIN location AS l ON h.pincode = l.pincode

ORDER BY h.ratings DESC;

```
set search_path to tourism_management_system;
SELECT h."Name", h.phone, h.foodType, h.ratings,
CONCAT (h.address,', ', l.city,', ', l.state,' - ', h.pincode) AS "Address"
FROM hotel AS h JOIN location AS l ON h.pincode = l.pincode
ORDER BY h.ratings DESC;
```

Data Output | Explain | Messages | Notifications

	Name character varying (50)	phone numeric (10)	foodtype character varying (20)	ratings double precision	Address text
1	Hyatt	8794561251	BOTH	4.8	Opp Ahmed...
2	ITC Grand	8978152345	BOTH	4.6	287, Dr, Dr B...
3	Jaisalkot - A	9116010801	VEG	4.6	Kuldhara Tu...
4	ITC Mughal, A	5624021700	BOTH	4.5	Itc Mughal, ...
5	Central Hotel	9894517223	BOTH	4.5	Ashirwad En...
6	Ahmedabad	9878456512	BOTH	4.5	Vastrapur,Ah...
7	Statue of Unity	9797949494	VEG	4.5	Sardar Saro...
8	The Imperial	7894556218	VEG	4.4	Dr Yagnik Rd...
9	The Metropole	7889451575	BOTH	4.2	Near R.T.O. ...
10	Cosset-Comfort	7859485625	BOTH	4.2	Mumbai Pun...
11	Meritas	9689521111	BOTH	4.2	Plot No. 13...
12	Hotel	8989456512	NON-VEG	4.1	6, VS Marg, ...
13	Ahdoos	7889464612	VEG	4.1	Residency r...
14	The Fern	2876225200	VEG	4.1	Talala Road, ...
15	Astoria Hotel	9726549956	BOTH	4	J. Tata Road...
16	Billberry Hotel	9865476521	NON-VEG	4	Ghat No. 17,...
17	Ramada by	1835025555	BOTH	3.9	117 Hall Baz...
18	Hotel	9878451532	VEG	3.8	No.74-A, Ka...
19	The White	6868626585	VEG	3.5	181/1, Oppo...
20	Hotel Thomas	9855004767	VEG	3.5	Simsa Villag...
21	Anmol Hotel	9898565844	VEG	3.5	8180 Street ...
22	Aurick Hotel	6658498756	NON-VEG	3	15th Cross, ...
23	BB Palace-A	8655855687	VEG	3	2638-2642 ...
24	Punjab Sindh	7858495615	BOTH	3	Main Market...
25	Chouki Dhani	9465813587	VEG	3	Near All Indi...
26	Hotel Sai	7889456124	VEG	2.9	Chandpur In...

--13) View list of hotel rooms starting from the Lowest Price to Highest Price.

Relational Algebra:

r1 -> ρ(h, hotel) $\bowtie_{<h.hotelid = r.hotelid>}$ ρ(r, room)

result -> Π(h. “Name”->Hotel_Name,r.room->Room_Number,r.Type->Room_Type,r.beds->No_of_beds,r.capacity->capacity, r.rate->price,r.status->Current_Status) (σ ORDER BY$_{r.rate,h.”name”,r.room_no}$ (r1))

SQL Query:

SELECT h."Name" As "Hotel_Name", r.room_no AS "Room_Number", r."Type" AS "Room_Type",

r.beds AS "No_of_Beds", r.capacity AS "Capacity", r.rate AS "Price", r.status As "Current_Status"

FROM hotel AS h JOIN room AS r ON h.hotelid = r.hotelid

ORDER BY r.rate,h."Name", r.room_no;

```
set search_path to tourism_management_system;
SELECT h."Name" As "Hotel_Name", r.room_no AS "Room_Number", r."Type" AS "Room_Type",
r.beds AS "No_of_Beds", r.capacity AS "Capacity", r.rate AS "Price", r.status As "Current_Status"
FROM hotel AS h JOIN room AS r ON h.hotelid = r.hotelid
ORDER BY r.rate,h."Name", r.room_no;
```

Data Output Explain Messages Notifications

	Hotel_Name character varying (50)	Room_Number numeric (3)	Room_Type character varying (6)	No_of_Beds integer	Capacity integer	Price double precision	Current_Status character varying (15)
1	Ahdoos	2	HEATER	1	2	1500	AVAILABLE
2	Ahmedabad	2	AC	1	2	1500	AVAILABLE
3	Astoria Hotel	2	AC	1	2	1500	AVAILABLE
4	Aurick Hotel	5	NONAC	1	2	1500	AVAILABLE
5	BB Palace-A	2	AC	1	2	1500	AVAILABLE
6	Billberry Hotel	5	NONAC	1	2	1500	AVAILABLE
7	Central Hotel	5	NONAC	1	2	1500	AVAILABLE
8	Chouki Dhani	2	AC	1	2	1500	AVAILABLE
9	Cosset-Comfort	2	AC	1	2	1500	AVAILABLE
10	Hotel Sai	2	AC	1	2	1500	AVAILABLE
11	Hotel Thomas	10	NONAC	1	2	1500	AVAILABLE
12	Hotel	5	NONAC	1	2	1500	AVAILABLE
13	Hotel	5	NONAC	1	2	1500	AVAILABLE
14	ITC Grand	5	NONAC	1	2	1500	AVAILABLE
15	ITC Mughal, A	5	NONAC	1	2	1500	AVAILABLE

--14) Retrieve list of hotel rooms that have “Cable TV” facility at a particular location.

Relational Algebra:

r1 -> ρ(h, hotel) $\bowtie_{<h.pincode=l.pincode>}$ ρ(l, location) $\bowtie_{<h.hotelid= r.hotelid>}$ ρ(r, room)

$\bowtie_{<\ h.hotelid=rf.hotelid\ and\ r.room\_no=rf.roomno\ >}$ ρ(rf, room_facilities)

Result -> $\Pi_{(h.\ \text{“Name”->Hotel\_Name,r.room->Room\_Number,r.Type->Room\_Type,r.beds->No\_of\_beds,r.capacity->capacity,}}$

$_{\text{r.rate->price,r.status->Current\_Status,rf.facility,l.city)}}$ ($\sigma_{\text{(rf.facility='Cable TV' and l.city = 'Amritsar')}}$

ORDER BY$_{\text{r.rate,h.”name”,r.room_no}}$ (r1))

SQL Query:

SELECT h."Name" As "Hotel_Name",r.room_no AS "Room_Number", r."Type" AS "Room_Type",

r.beds AS "No_of_Beds", r.capacity AS "Capacity", r.rate AS "Price", r.status As "Current_Status", rf.facility, l.city

FROM hotel AS h JOIN location AS l ON h.pincode = l.pincode

JOIN room AS r ON h.hotelid = r.hotelid

JOIN room_facilities AS rf ON (h.hotelid=rf.hotelid and r.room_no=rf.roomno)

WHERE rf.facility='Cable TV' and l.city = 'Amritsar'

ORDER BY h."Name", r.room_no;

Query Editor Query History Scratch Pad

```
set search_path to tourism_management_system;
SELECT h."Name" As "Hotel_Name",r.room_no AS "Room_Number", r."Type" AS "Room_Type",
r.beds AS "No_of_Beds", r.capacity AS "Capacity", r.rate AS "Price", r.status As "Current_Status", rf.fac
FROM hotel AS h JOIN location AS l ON h.pincode = l.pincode
JOIN room AS r ON h.hotelid = r.hotelid
JOIN room_facilities AS rf ON (h.hotelid=rf.hotelid and r.room_no=rf.roomno)
WHERE rf.facility='Cable TV' and l.city = 'Amritsar'
ORDER BY h."Name", r.room_no;
```

Data Output Explain Messages Notifications

	Hotel_Name character varying (50)	Room_Number numeric (3)	Room_Type character varying (6)	No_of_Beds integer	Capacity integer	Price double precision	Current_Status character varying (15)	facility character varying (50)	city character varyi
1	Ramada by	1	AC	2	2	3000	AVAILABLE	Cable TV	Amritsar

--15) Retrieve all the packages associated with a particular guide. (admin)

Relational Algebra:

r1 -> ρ(g, guide) $\bowtie_{<g.guideid = pg.guideid>}$ ρ(pg, guideid) $\bowtie_{<pg.packageid = p.packageid>}$ ρ(p, package)

result -> $\Pi_{(CONCAT\ (g.fname,'\ ',\ g.lname) \rightarrow Guide\_Name,\ p.title \rightarrow package\_name,\ p.duration \rightarrow Duration(in\ days),\ p.no\_of\_people \rightarrow No\_of\_people,\ p.amount \rightarrow Amount)}(\sigma_{g.fname='Sachin'\ and\ g.lname='Bhide'}(r1))$

SQL Query:

```
SELECT CONCAT (g.fname,' ', g.lname) AS "Guide_Name",
p.title AS "Package_Name", p.duration as "Duration (In Days)",
p.no_of_people AS "No_Of_People", p.amount AS "Amount"
FROM guide AS g JOIN package_includes_guides AS pg ON (g.guideid=pg.guideid)
JOIN package AS p ON (pg.packageid=p.packageid)
WHERE g.fname='Sachin' and g.lname='Bhide';
```

postgres/postgres@PostgreSQL 13

Query Editor | Query History

```
set search_path to tourism_management_system;
SELECT CONCAT (g.fname,' ', g.lname) AS "Guide_Name",
p.title AS "Package_Name", p.duration as "Duration (In Days)",
p.no_of_people AS "No_Of_People", p.amount AS "Amount"
FROM guide AS g JOIN package_includes_guides AS pg ON (g.guideid=pg.guideid)
JOIN package AS p ON (pg.packageid=p.packageid)
WHERE g.fname='Sachin' and g.lname='Bhide';
```

Data Output | Explain | Messages | Notifications

	Guide_Name text	Package_Name character varying (15)	Duration (In Days) integer	No_Of_People integer	Amount double precision
1	Sachin Bhide	Manali Tour	9	8	64000

--16) Retrieve the list of all package associated with a particular hotel.

Relational Algebra:

r1 -> ρ(h, hotel) $\bowtie_{<h.hotelid = ph.hotelid>}$ ρ(ph, package_includes_hotels) $\bowtie_{<pg.packageid = p.packageid>}$ ρ(p, package)

result -> Π(h."Name" -> Hotel_Name, p.title -> package_name, p.duration -> Duration(in days), p.no_of_people -> No_of_people, p.amount -> Amount)(σ h."Name"='Hotel Thomas Villa' (r1))

SQL Query:

```
SELECT h."Name" AS "Hotel_Name",
p.title AS "Package_Name", p.duration as "Duration (In Days)",
p.no_of_people AS "No_Of_People", p.amount AS "Amount"
FROM hotel AS h JOIN package_includes_hotels AS ph ON (h.hotelid=ph.hotelid)
JOIN package AS p ON (ph.packageid=p.packageid)
WHERE h."Name"='Hotel Thomas Villa';
```

postgres/postgres@laptop

Query Editor Query History

```
SELECT h."Name" AS "Hotel_Name",
p.title AS "Package_Name", p.duration as "Duration (In Days)",
p.no_of_people AS "No_Of_People", p.amount AS "Amount"
FROM hotel AS h JOIN package_includes_hotels AS ph ON (h.hotelid=ph.hotelid)
JOIN package AS p ON (ph.packageid=p.packageid)
WHERE h."Name"='Hotel Thomas Villa';
```

Data Output Explain Messages Notifications

	Hotel_Name character varying (50)	Package_Name character varying (15)	Duration (In Days) integer	No_Of_People integer	Amount double precision
1	Hotel Thomas Villa	Manali Tour	9	8	64000

--17) Retrieve all the packages which include “xyz” spots.

Relational Algebra:

r1 -> ρ(t, tourist_spots) $\bowtie_{<t.spotid = ps.spotid>}$ ρ(ps, package_includes_spots) $\bowtie_{<ps.packageid = p.packageid>}$ ρ(p, package)

result -> $\Pi_{(t.\text{”Name”} \to \text{Spot\_Name, p.title} \to \text{package\_name, p.duration} \to \text{Duration(in days), p.no\_of\_people} \to \text{No\_of\_people, p.amount} \to \text{Amount})}(\sigma_{\text{t."Name"='Manali'}}(r1))$

SQL Query:

SELECT t."Name" AS "Spot_Name",

p.title AS "Package_Name", p.duration as "Duration (In Days)",

p.no_of_people AS "No_Of_People", p.amount AS "Amount"

FROM tourist_spots AS t JOIN package_includes_spots AS ps ON (t.spotid=ps.spotid)

JOIN package AS p ON (ps.packageid=p.packageid)

WHERE t."Name"='Manali';

postgres/postgres@PostgreSQL 15

Query Editor Query History

```
1  set search_path to tourism_management_system;
2  SELECT t."Name" AS "Spot_Name",
3  p.title AS "Package_Name", p.duration as "Duration (In Days)",
4  p.no_of_people AS "No_Of_People", p.amount AS "Amount"
5  FROM tourist_spots AS t JOIN package_includes_spots AS ps ON (t.spotid=ps.spotid)
6  JOIN package AS p ON (ps.packageid=p.packageid)
7  WHERE t."Name"='Manali';
8
9
```

Data Output Explain Messages Notifications

	Spot_Name character varying (20)	Package_Name character varying (15)	Duration (In Days) integer	No_Of_People integer	Amount double precision
1	Manali	Manali Tour	9	8	64000

--18) Best tourist place to visit in “xyz” season.

Relational Algebra:

r1 -> ρ(ts, tourist_spots) $\bowtie_{<ts.pincode = l.pincode>}$ ρ(l, location)

result -> Π(t.”Name” -> Spot_Name, ts.season -> Season, ts.ratings -> Ratings,

CONCAT (ts.address,', ',l.city,', ',l.state, ' - ',ts.pincode) ->Address)) (σ ts.season "='Winter' (r1))

SQL Query:

```
SELECT ts."Name" AS "Spot_Name", ts.season AS "Season", ts.ratings AS "Ratings",
CONCAT (ts.address,', ',l.city,', ',l.state, ' - ',ts.pincode) AS "Address"
FROM tourist_spots AS ts JOIN "location" AS l
ON ts.pincode=l.pincode
WHERE ts.season='Winter';
```

Query Editor Query History

```
set search_path to tourism_management_system;
SELECT ts."Name" AS "Spot_Name", ts.season AS "Season", ts.ratings AS "Ratings",
CONCAT (ts.address,', ',l.city,', ',l.state, ' - ',ts.pincode) AS "Address"
FROM tourist_spots AS ts JOIN "location" AS l
ON ts.pincode=l.pincode
WHERE ts.season='Winter';
```

Data Output Explain Messages Notifications

	Spot_Name character varying (20)	Season character varying (10)	Ratings double precision	Address text
1	Manali	Winter	4.7	Rotang Pass...
2	Srinagar	Winter	4.7	Srinagar,J & ...
3	Dashashwamedh	Winter	4.6	Dashashwa...
4	Somnath	Winter	4.3	Somnath M...

--19) Name and address of hotels which provides rooms between specific price range.

Relational Algebra:

r1 -> ρ(h, hotel) $\bowtie_{<h.pincode=l.pincode>}$ ρ(l, location) $\bowtie_{<h.hotelid= r.hotelid>}$ ρ(r, room)

LEFT $\bowtie_{<\ h.hotelid=rf.hotelid\ and\ r.room\_no=rf.roomno\ >}$ ρ(rf, room_facilities)

Result -> Π(h. "Name"->Hotel_Name,r.room->Room_Number,r.Type->Room_Type,r.beds->No_of_beds,r.capacity->capacity,

r.rate->price,r.status->Current_Status,rf.facility, CONCAT (h.address,', ', l.city,', ', l.state,' - ', h.pincode) -> Address)

($\sigma_{(r.rate\ BETWEEN\ 1500\ and\ 2000)}$ ORDER BY$_{r.rate,h."name",r.room_no}$ (r1))

SQL Query:

SELECT h."Name" As "Hotel_Name", r.room_no AS "Room_Number", r."Type" AS "Room_Type",

r.beds AS "No_of_Beds", r.capacity AS "Capacity", r.rate AS "Price", r.status As "Current_Status",

rf.facility, CONCAT(h.address,', ', l.city,', ', l.state,' - ', h.pincode) AS "Address"

FROM hotel AS h JOIN location AS l ON h.pincode = l.pincode

JOIN room AS r ON h.hotelid = r.hotelid

LEFT JOIN room_facilities AS rf ON (h.hotelid=rf.hotelid and r.room_no=rf.roomno)

WHERE r.rate between 1500 and 2000

ORDER BY r.rate, h."Name", r.room_no;

Query Editor | Query History | Scratch Pad

```
set search_path to tourism_management_system;
SELECT h."Name" As "Hotel_Name", r.room_no AS "Room_Number", r."Type" AS "Room_Type",
r.beds AS "No_of_Beds", r.capacity AS "Capacity", r.rate AS "Price", r.status As "Current_Status",
rf.facility, CONCAT(h.address,', ', l.city,', ', l.state,' - ', h.pincode) AS "Address"
FROM hotel AS h JOIN location AS l ON h.pincode = l.pincode
JOIN room AS r ON h.hotelid = r.hotelid
LEFT JOIN room_facilities AS rf ON (h.hotelid=rf.hotelid and r.room_no=rf.roomno)
WHERE r.rate between 1500 and 2000
ORDER BY r.rate, h."Name", r.room_no;
```

Data Output | Explain | Messages | Notifications

	Hotel_Name character varying (50)	Room_Number numeric (3)	Room_Type character varying (6)	No_of_Beds integer	Capacity integer	Price double precision	Current_Status character varying (15)	facility character varying (50)	Address text
1	Ahdoos	2	HEATER	1	2	1500	AVAILABLE	[null]	Residency r...
2	Ahmedabad	2	AC	1	2	1500	AVAILABLE	[null]	Vastrapur,Ah.
3	Astoria Hotel	2	AC	1	2	1500	AVAILABLE	[null]	J. Tata Road.
4	Aurick Hotel	5	NONAC	1	2	1500	AVAILABLE	[null]	15th Cross, ...
5	BB Palace-A	2	AC	1	2	1500	AVAILABLE	[null]	2638-2642 ...
6	Billberry Hotel	5	NONAC	1	2	1500	AVAILABLE	[null]	Ghat No. 17,.
7	Central Hotel	5	NONAC	1	2	1500	AVAILABLE	[null]	Ashirwad En.
8	Chouki Dhani	2	AC	1	2	1500	AVAILABLE	[null]	Near All Indi..
9	Cosset-Comfort	2	AC	1	2	1500	AVAILABLE	[null]	Mumbai Pun.
10	Hotel Sai	2	AC	1	2	1500	AVAILABLE	[null]	Chandpur In..

--20) Retrieve list of all the guides which are not associated with any active packages.

Relational Algebra:

r1-> $\Pi_{(pg.guideid)}$ ($\sigma_{(p.isActive='TRUE')}$ (ρ(pg, package_include_guide) $\bowtie_{<pg.package.id=p.packageid>}$ ρ(p,package)))

r2 -> ρ(g,guide) $\bowtie_{<g.pincode=l.pincode>}$ ρ(l, location)

result-> $\Pi_{(CONCAT(g.fname,' ',g.lname)->Guide\_Name,g.email,g.phone,g.age,g.gender,(g.address,',',l.city,',',l.state,'-',g.pincode)->Address)}$ ($\sigma_{(g.guideid\ NOT\ IN\ (r1))}$ (r2))

SQL Query:

SELECT CONCAT (g.fname,' ', g.lname) AS "Guide_Name", g.email, g.phone, g.age, g.gender,

CONCAT (g.address,', ',l.city,', ',l,state,' - ',g.pincode) AS "Address"

FROM guide AS g JOIN "location" AS l ON (g.pincode=l.pincode)

WHERE g.guideid NOT IN

(SELECT guideid from package_includes_guides AS pg

JOIN (Select * from package where isActive='TRUE') AS p ON (pg.packageid=p.packageid));

Query Editor Query History

```
set search_path to tourism_management_system;
SELECT CONCAT (g.fname,' ', g.lname) AS "Guide_Name", g.email, g.phone, g.age, g.gender,
CONCAT (g.address,', ',l.city,', ',l,state,' - ',g.pincode) AS "Address"
FROM guide AS g JOIN "location" AS l ON (g.pincode=l.pincode)
WHERE g.guideid NOT IN
(SELECT guideid from package_includes_guides AS pg
JOIN (Select * from package where isActive='TRUE') AS p ON (pg.packageid=p.packageid));
```

Data Output Explain Messages Notifications

	Guide_Name text	email character varying (20)	phone numeric (10)	age integer	gender character (1)	Address text
1	Param Singh	psingh@gmail.com	6645789155	28	M	Gill Medical ...
2	Pooran Singh	theps4@gmail.com	9977884455	40	M	Dargah Hom...
3	Shanker Desai	shivd88@gmail.com	7984561534	56	M	Shree Muniv...
4	Karan Thakker	kt14@gmail.com	9988451601	30	M	Abhay Ghat, ...
5	Akshar Patel	akpatel45@gmail.com	7845561255	26	M	Aarogya Van...
6	Ganesh Gaitonde	gg0007@gmail.com	9988990007	35	M	128 ,pragati ...

--21) Which hotel have availability of room right now?

Relational Algebra:

r1 -> $\rho(hl, hotel) \bowtie_{<hl.hotelid = r.hotelid>} \rho(r, {}_{hotelid}\mathcal{F}_{COUNT(room\_no) \rightarrow rooms}(room))$

result -> $\Pi_{(hl.\text{"Name"}, rooms, hl.ratings, hl.address)}(\sigma_{rooms > 0 \text{ AND } hl.isActive=true}(r1))$

SQL Query:

SELECT hl."Name", rooms ,hl.ratings, hl.address

FROM hotel as hl JOIN

(SELECT hotelid, COUNT (room_no) AS rooms FROM room GROUP BY hotelid) as r

ON (r.hotelid = hl.hotelid) WHERE rooms > 0 AND hl.isactive = true;

```
set search_path to tourism_management_system;
SELECT hl."Name", rooms ,hl.ratings, hl.address
FROM hotel as hl JOIN
(SELECT hotelid, COUNT (room_no) AS rooms FROM room GROUP BY hotelid) as r
ON (r.hotelid = hl.hotelid) WHERE rooms > 0 AND hl.isactive = true;
```

Data Output Explain Messages Notifications

	Name character varying (50)	rooms bigint	ratings double precision	address character varying (100)
1	Ramada by	10	3.9	117 Hall Bazaar, Punjab
2	Hotel Thomas	10	3.5	Simsa Village, Manali, Himach...
3	Ahdoos	5	4.1	Residency road, Regal Chowk,...
4	Billberry Hotel	5	4	Ghat No. 17, Nehru Park, Dal L...
5	Hotel	5	3.8	No.74-A, Kamla Nehru Rd, opp...
6	Hotel Sai	5	2.9	Chandpur Industrial State Nea...
7	The White	5	3.5	181/1, Opposite CDRI Campu...
8	Hotel	5	4.1	6, VS Marg, Narpatkhera, Lalb...
9	Punjab Sindh	5	3	Main Market, near Kedarnath ...
10	Anmol Hotel	5	3.5	8180 Street No.-6,Arakashan ...
11	BB Palace-A	5	3	2638-2642 Gurudwara Road,A...
12	ITC Mughal, A	5	4.5	Itc Mughal, Fatehabad Rd, Taj...
13	Jaisalkot - A	5	4.6	Kuldhara Turn, Off, Jaisalmer ...
14	Chouki Dhani	5	3	Near All India Radio Tower, Ja...

--22) Number of rooms available at a particular hotel right now.

Relational Algebra:

r1 -> $\rho(h, hotel) \bowtie_{<h.hotelid = r.hotelid>} \rho(r, \mathcal{F}_{COUNT(*)} (room))$

result -> $\sigma_{h.\text{"Name"} = \text{'Ahdoos Hotel' AND } r.status = \text{'AVAILABLE'}} (r1)$

SQL Query:

```
SELECT COUNT (*) FROM room as r
JOIN
hotel AS h ON (r.hotelid = h.hotelid)
WHERE h."Name" = 'Ahdoos Hotel' AND r.status = 'AVAILABLE';
```

postgres/postgres@laptop

Query Editor Query History

```
SET search_path to tourism_management_system;

SELECT COUNT (*) FROM room as r
JOIN
hotel AS h ON (r.hotelid = h.hotelid)
WHERE h."Name" = 'Ahdoos Hotel' AND r.status = 'AVAILABLE';
```

Data Output Explain Messages Notifications

	count bigint
1	5

--23) Retrieve all the previous bookings of user. (both)

Relational Algebra:

r1 -> ρ(b, booking) LEFT $\bowtie_{<b.bid = bfc.bid>}$ ρ(bfc, $_{bid}\mathcal{F}_{COUNT(fname) \rightarrow no\_of\_co\_passengers}$ (booking_copassenger)) FULL OUTER $\bowtie_{<b.bid = bfp.bid>}$ ρ(bfp, booking_for_package) FULL OUTER $\bowtie_{<bfp.packageid = p.packageid>}$ ρ(p, package) FULL OUTER $\bowtie_{<b.bid = bfh.bid>}$ ρ(bfh, booking_for_hotel) FULL OUTER $\bowtie_{<bfh.hotelid = h.hotelid>}$ ρ(h, hotel) FULL OUTER $\bowtie_{<h.hotelid = pih.hotelid>}$ ρ(pih, package_includes_hotels) FULL OUTER $\bowtie_{<pih.hotelid = hp.hotelid>}$ ρ(hp, hotel) $\bowtie_{<b.userid = u.userid>}$ ρ(u,"User")

result -> $\Pi_{(b.booking\_date, bfc.no\_of\_co\_passengers, b.tripstart\_date, b.tripend\_date, b.amount, p.title \rightarrow package\_name, hp."Name" \rightarrow package\_hotel, pih.RoomNo \rightarrow package\_roomno, h."Name" \rightarrow hotel\_name, bfh.roomno)}$ ($\sigma_{CONCAT\ (u.fname,'\ ',u.lname) = 'Viral\ Gandhi'}$ (r1))

SQL Query:

```
SELECT b.booking_date, bfc.no_of_co_passengers, b.tripstart_date, b.tripend_date,
b.amount, p.title AS package_name, hp."Name" as package_hotel, pih.RoomNo
package_roomno,h."Name" AS hotel_name, bfh.roomno  FROM booking as b

LEFT JOIN

(SELECT COUNT(c.fname) AS no_of_co_passengers, c.bid FROM booking_copassenger c
GROUP BY c.bid) AS bfc ON (bfc.bid = b.bid)

FULL OUTER JOIN booking_for_package AS bfp ON (bfp.bid = b.bid)

FULL OUTER JOIN package AS p ON (bfp.packageid = p.packageid)

FULL OUTER JOIN booking_for_hotel AS bfh ON (bfh.bid = b.bid)

FULL OUTER JOIN hotel AS h ON (bfh.hotelid = h.hotelid)

FULL OUTER JOIN Package_Includes_Hotels AS pih ON (pih.packageid = p.packageid)

FULL OUTER JOIN hotel AS hp ON (hp.hotelid = pih.hotelid)

JOIN "User" AS u ON (b.userid = u.userid)
```

WHERE CONCAT (u.fname, ' ', u.lname) = 'Viral Gandhi';

postgres/postgres@laptop

Query Editor | Query History | Scratch Pad

```
SELECT b.booking_date, bfc.no_of_co_passengers, b.tripstart_date, b.tripend_date, b.amount, p.title AS package_name, hp."Name" as pa
LEFT JOIN
(SELECT COUNT(c.fname) AS no_of_co_passengers, c.bid FROM booking_copassenger c GROUP BY c.bid) AS bfc ON (bfc.bid = b.bid)
FULL OUTER JOIN booking_for_package AS bfp ON (bfp.bid = b.bid)
FULL OUTER JOIN package AS p ON (bfp.packageid = p.packageid)
FULL OUTER JOIN booking_for_hotel AS bfh ON (bfh.bid = b.bid)
FULL OUTER JOIN hotel AS h ON (bfh.hotelid = h.hotelid)
FULL OUTER JOIN Package_Includes_Hotels AS pih ON (pih.packageid = p.packageid)
FULL OUTER JOIN hotel AS hp ON (hp.hotelid = pih.hotelid)

JOIN "User" AS u ON (b.userid = u.userid)
WHERE CONCAT (u.fname, ' ', u.lname) = 'Viral Gandhi';
```

Data Output | Explain | Messages | Notifications

	booking_date timestamp without time zone	no_of_co_passengers bigint	tripstart_date date	tripend_date date	amount double precision	package_name character varying (15)	package_hotel character varying (50)	package_roomno integer
1	2020-11-13 13:12:47.839129	1	2020-11-20	2020-11-22	12000	Taj Mahal Tour	Anmol Hotel Pvt.Ltd	2

--24) Retrieve the bookings between particular date of “xyz” user. (admin)

<u>Relational Algebra:</u>

r1 -> ρ(b, booking) LEFT ⋈<b.bid = bfc.bid> ρ(bfc, bid $\mathcal{F}$COUNT(fname) -> no_of_co_passengers (booking_copassenger)) FULL OUTER ⋈<b.bid = bfp.bid> ρ(bfp, booking_for_package) FULL OUTER ⋈<bfp.packageid = p.packageid> ρ(p, package) FULL OUTER ⋈<b.bid = bfh.bid> ρ(bfh, booking_for_hotel) FULL OUTER ⋈<bfh.hotelid = h.hotelid> ρ(h, hotel) FULL OUTER ⋈<h.hotelid = pih.hotelid> ρ(pih, package_includes_hotels) FULL OUTER ⋈<pih.hotelid = hp.hotelid> ρ(hp, hotel) ⋈<b.userid = u.userid> ρ(u,”User”)

result -> Π(b.booking_date, bfc.no_of_co_passengers, b.tripstart_date, b.tripend_date, b.amount, p.title -> package_name, hp.”Name” -> package_hotel, pih.RoomNo -> package_roomno, h.”Name” -> hotel_name, bfh.roomno) (σ CONCAT (u.fname,’ ‘,u.lname) = ‘Viral Gandhi’ and b.booking_date < '2020-11-14' AND b.booking_date >= '2020-11-09' (r1))

<u>SQL Query:</u>

SELECT CONCAT(u.fname,' ',u.lname), b.booking_date, bfc.no_copassengers, b.tripstart_date, b.tripend_date,

b.amount, p.title AS package_name,hp."Name" as package_hotel, pih.RoomNo package_roomno, h."Name" AS hotel_name, bfh.roomno

FROM booking as b

LEFT JOIN

(SELECT COUNT(c.fname) AS no_copassengers, c.bid FROM booking_copassenger c GROUP BY c.bid) AS bfc ON (bfc.bid = b.bid)

FULL OUTER JOIN booking_for_package AS bfp ON (bfp.bid = b.bid)

FULL OUTER JOIN package AS p ON (bfp.packageid = p.packageid)

FULL OUTER JOIN booking_for_hotel AS bfh ON (bfh.bid = b.bid)

FULL OUTER JOIN hotel AS h ON (bfh.hotelid = h.hotelid)

FULL OUTER JOIN Package_Includes_Hotels AS pih ON (pih.packageid = p.packageid)

FULL OUTER JOIN hotel AS hp ON (hp.hotelid = pih.hotelid)

JOIN "User" AS u ON (b.userid = u.userid)

WHERE CONCAT(u.fname, ' ', u.lname) = 'Viral Gandhi'

AND b.booking_date < '2020-11-14' AND b.booking_date >= '2020-11-09';

postgres/postgres@laptop

Query Editor | Query History | Scratch Pad

```
SELECT CONCAT(u.fname,' ',u.lname), b.booking_date, bfc.no_copassengers, b.tripstart_date, b.tripend_date,
b.amount, p.title AS package_name,hp."Name" as package_hotel, pih.RoomNo package_roomno, h."Name" AS hotel_name, bfh.roomno
FROM booking as b
LEFT JOIN
(SELECT COUNT(c.fname) AS no_copassengers, c.bid FROM booking_copassenger c GROUP BY c.bid) AS bfc
ON (bfc.bid = b.bid)
FULL OUTER JOIN
booking_for_package AS bfp ON (bfp.bid = b.bid)
FULL OUTER JOIN package AS p ON (bfp.packageid = p.packageid)
FULL OUTER JOIN
booking_for_hotel AS bfh ON (bfh.bid = b.bid)
FULL OUTER JOIN hotel AS h ON (bfh.hotelid = h.hotelid)
FULL OUTER JOIN Package_Includes_Hotels AS pih ON (pih.packageid = p.packageid)
FULL OUTER JOIN hotel AS hp ON (hp.hotelid = pih.hotelid)
JOIN
"User" AS u ON (b.userid = u.userid) WHERE CONCAT(u.fname, ' ', u.lname) = 'Viral Gandhi'
AND b.booking_date < '2020-11-14' AND b.booking_date >= '2020-11-09';
```

Data Output | Explain | Messages | Notifications

	concat text	booking_date timestamp without time zone	no_copassengers bigint	tripstart_date date	tripend_date date	amount double precision	package_name character varying (15)	package_hotel character varying (50)	package_roomno integer
1	Viral Gandhi	2020-11-13 13:12:47.839129	1	2020-11-20	2020-11-22	12000	Taj Mahal Tour	Anmol Hotel Pvt.Ltd	2

--25) Retrieve all the bookings between particular date. (admin)

<u>Relational Algebra:</u>

r1 -> ρ(b, booking) LEFT ⋈<b.bid = bfc.bid> ρ(bfc, bid $\mathcal{F}$COUNT(fname) -> no_of_co_passengers (booking_copassenger)) FULL OUTER ⋈<b.bid = bfp.bid> ρ(bfp, booking_for_package) FULL OUTER ⋈<bfp.packageid = p.packageid> ρ(p, package) FULL OUTER ⋈<b.bid = bfh.bid> ρ(bfh, booking_for_hotel) FULL OUTER ⋈<bfh.hotelid = h.hotelid> ρ(h, hotel) FULL OUTER ⋈<h.hotelid = pih.hotelid> ρ(pih, package_includes_hotels) FULL OUTER ⋈<pih.hotelid = hp.hotelid> ρ(hp, hotel) ⋈<b.userid = u.userid> ρ(u,"User")

result -> Π(b.booking_date, bfc.no_of_co_passengers, b.tripstart_date, b.tripend_date, b.amount, p.title -> package_name, hp.”Name” -> package_hotel, pih.RoomNo -> package_roomno, h.”Name” -> hotel_name, bfh.roomno) (σb.booking_date < '2020-11-14' AND b.booking_date >= '2020-11-09' (r1))

<u>SQL Query:</u>

SELECT CONCAT(u.fname,' ',u.lname), b.booking_date, bfc.no_copassengers, b.tripstart_date, b.tripend_date,

b.amount, p.title AS package_name,hp."Name" as package_hotel, pih.RoomNo package_roomno, h."Name" AS hotel_name, bfh.roomno

FROM booking as b

LEFT JOIN

(SELECT COUNT(c.fname) AS no_copassengers, c.bid FROM booking_copassenger c GROUP BY c.bid) AS bfc ON (bfc.bid = b.bid)

FULL OUTER JOIN booking_for_package AS bfp ON (bfp.bid = b.bid)

FULL OUTER JOIN package AS p ON (bfp.packageid = p.packageid)

FULL OUTER JOIN booking_for_hotel AS bfh ON (bfh.bid = b.bid)

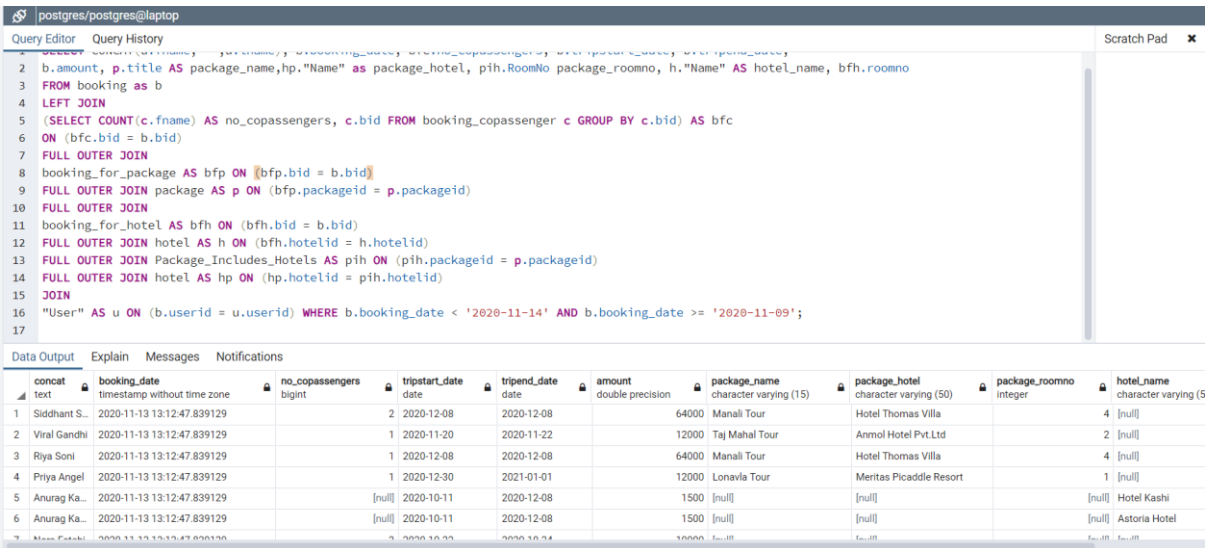
FULL OUTER JOIN hotel AS h ON (bfh.hotelid = h.hotelid)

FULL OUTER JOIN Package_Includes_Hotels AS pih ON (pih.packageid = p.packageid)

FULL OUTER JOIN hotel AS hp ON (hp.hotelid = pih.hotelid)

JOIN "User" AS u ON (b.userid = u.userid)

WHERE b.booking_date < '2020-11-14' AND b.booking_date >= '2020-11-09';



	concat text	booking_date timestamp without time zone	no_copassengers bigint	tripstart_date date	tripend_date date	amount double precision	package_name character varying (15)	package_hotel character varying (50)	package_roomno integer	hotel_name character varying (5
1	Siddhant S...	2020-11-13 13:12:47.839129	2	2020-12-08	2020-12-08	64000	Manali Tour	Hotel Thomas Villa	4	[null]
2	Viral Gandhi	2020-11-13 13:12:47.839129	1	2020-11-20	2020-11-22	12000	Taj Mahal Tour	Anmol Hotel Pvt.Ltd	2	[null]
3	Riya Soni	2020-11-13 13:12:47.839129	1	2020-12-08	2020-12-08	64000	Manali Tour	Hotel Thomas Villa	4	[null]
4	Priya Angel	2020-11-13 13:12:47.839129	1	2020-12-30	2021-01-01	12000	Lonavla Tour	Meritas Picaddle Resort	1	[null]
5	Anurag Ka...	2020-11-13 13:12:47.839129	[null]	2020-10-11	2020-12-08	1500	[null]	[null]	[null]	Hotel Kashi
6	Anurag Ka...	2020-11-13 13:12:47.839129	[null]	2020-10-11	2020-12-08	1500	[null]	[null]	[null]	Astoria Hotel

26) Retrieve all the details of user of “xyz” hotel room. (admin)

Relational Algebra:

r1 -> ρ(r,room) $\bowtie_{< r.hotelid = h.hotelid >}$ ρ(h,hotel)

r2-> ($\sigma_{h."Name" = 'Hotel Thomas' \text{ and } r.room\_no = 5}$ (r1))

r3-> ρ(bfh,booking_for_hotel)$\bowtie_{<r2.hotelid = bfh.hotelid >}$ (r2)

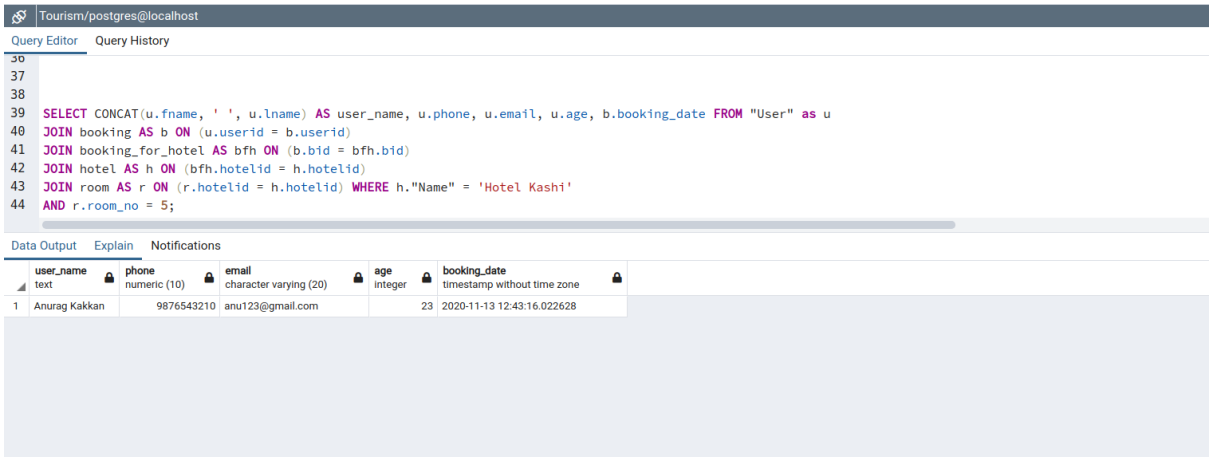
r4-> ρ(b,booking)$\bowtie_{<r3.bid = b.bid >}$ (r3)

r5-> ρ(u,User)$\bowtie_{< u.userid = r4.userid >}$ (r4)

result-> $\Pi_{(CONCAT(u.fname, ' ', u.lname)->user\_name,\ u.phone,\ u.email,\ u.age,\ b.booking\_date)}$ (r5)

SQL Query:

```
SELECT CONCAT(u.fname, ' ', u.lname) AS user_name, u.phone, u.email, u.age,
b.booking_date FROM "User" as u
JOIN booking AS b ON (u.userid = b.userid)
JOIN booking_for_hotel AS bfh ON (b.bid = bfh.bid)
JOIN hotel AS h ON (bfh.hotelid = h.hotelid)
JOIN room AS r ON (r.hotelid = h.hotelid) WHERE h."Name" = 'Hotel Kashi'
AND r.room_no = 5;
```



27) Retrieve all the user booking details of “xyz” hotel. (admin)

Relational Algebra:

r1-> ρ(bfh, booking_for_hotel) $\bowtie_{<\text{bfh.hotelid = h.hotelid}>}\rho$(h,hotel)

r2-> $\sigma_{\text{h."Name" = 'Hotel Kashi'}}$(r1)

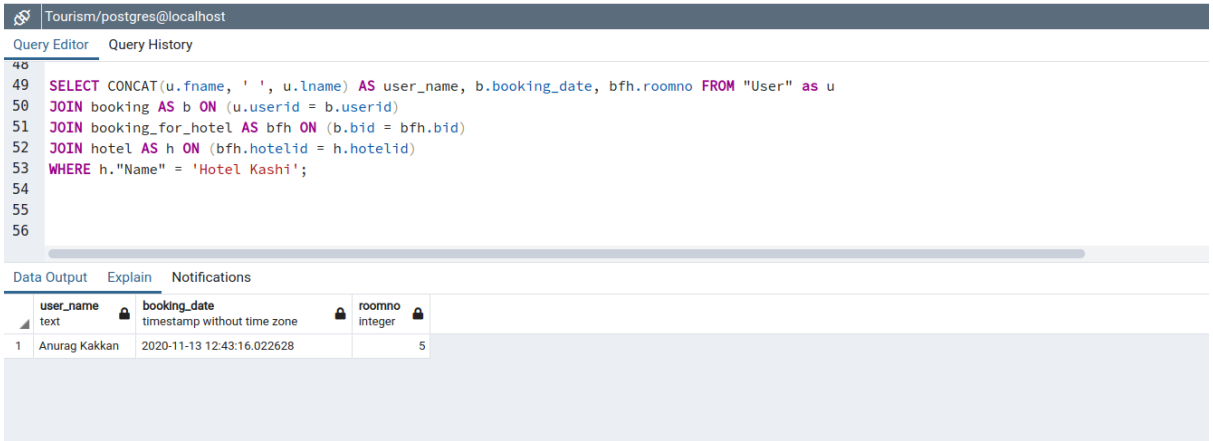
r3-> ρ(b,booking) $\bowtie_{<\text{b.bid=r2.bid}>}$(r2)

r4->ρ(u,user) $\bowtie_{<\text{u.userid=r3.userid}>}$(r3)

$\Pi_{\text{(CONCAT(u.fname, ' ', u.lname)->user_name, b.booking_date, bfh.roomno)}}$(r4)

SQL Query:

```
SELECT CONCAT(u.fname, ' ', u.lname) AS user_name, b.booking_date, bfh.roomno FROM "User" as u
JOIN booking AS b ON (u.userid = b.userid)
JOIN booking_for_hotel AS bfh ON (b.bid = bfh.bid)
JOIN hotel AS h ON (bfh.hotelid = h.hotelid)
WHERE h."Name" = 'Hotel Kashi';
```



28) Give details of co-passenger with “xyz” user with dates. (admin)

Relational Algebra:

r1->ρ(bc, booking_copassenger) ⋈< bc.userid = c.uid AND bc.fname = c.fname AND

bc.lname = c.lname>ρ(c,copassanger)

r2-> σ CONCAT(u.fname, ' ', u.lname) = 'Viral Gandhi'(r1)

r3-> ρ(b,booking) ⋈<b.bid=r2.bid>(r2)

r4->ρ(u,user) ⋈<u.userid=r3.userid>(r3)

Π(CONCAT(u.fname, ' ', u.lname) AS user_name, b.booking_date,

CONCAT(c.fname, ' ', c.lname) as copassenger, c.phone, c.gender, c.age)(r4)

SQL Query:

SELECT CONCAT(u.fname, ' ', u.lname) AS user_name, b.booking_date,

CONCAT(c.fname, ' ', c.lname) as copassenger, c.phone, c.gender, c.age

FROM "User" as u

JOIN booking AS b ON (u.userid = b.userid)

JOIN booking_copassenger AS bc ON (bc.bid = b.bid)

JOIN copassenger AS c ON (bc.userid = c.uid AND bc.fname = c.fname AND bc.lname = c.lname)

WHERE CONCAT(u.fname, ' ', u.lname) = 'Viral Gandhi';

postgres/postgres@PostgreSQL 13

Query Editor Query History

```
set search_path to tourism_management_system;
SELECT CONCAT(u.fname, ' ', u.lname) AS user_name, b.booking_date,
CONCAT(c.fname, ' ', c.lname) as copassenger, c.phone, c.gender, c.age
FROM "User" as u
JOIN booking AS b ON (u.userid = b.userid)
JOIN booking_copassenger AS bc ON (bc.bid = b.bid)
JOIN copassenger AS c ON (bc.userid = c.uid AND bc.fname = c.fname AND bc.lname = c.lname)
WHERE CONCAT(u.fname, ' ', u.lname) = 'Viral Gandhi';
```

Data Output Explain Messages Notifications

	user_name text	booking_date timestamp without time zone	copassenger text	phone numeric (12)	gender character (1)	age integer
1	Viral Gandhi	2020-10-16 10:52:52.192482	Raju Japani	6765462659	M	26