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;

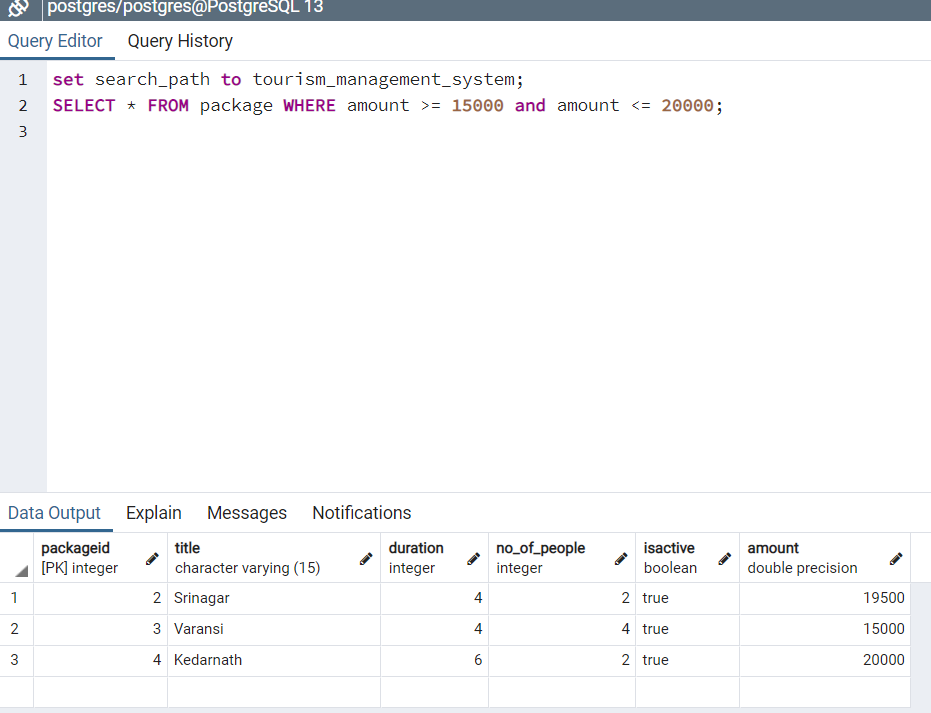
**--1) Retrieve the packages within a particular price range**

Relational Algebra:

(amount >=15000 AND amount <= 20000)(package)

SQL Query:

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



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

Relational Algebra:

r1 -> (bfp, packageidℱCOUNT(bid)(Booking\_for\_package))

r2 -> (p, package) ⋈<p.packageid = bfp.packageid>(r1)

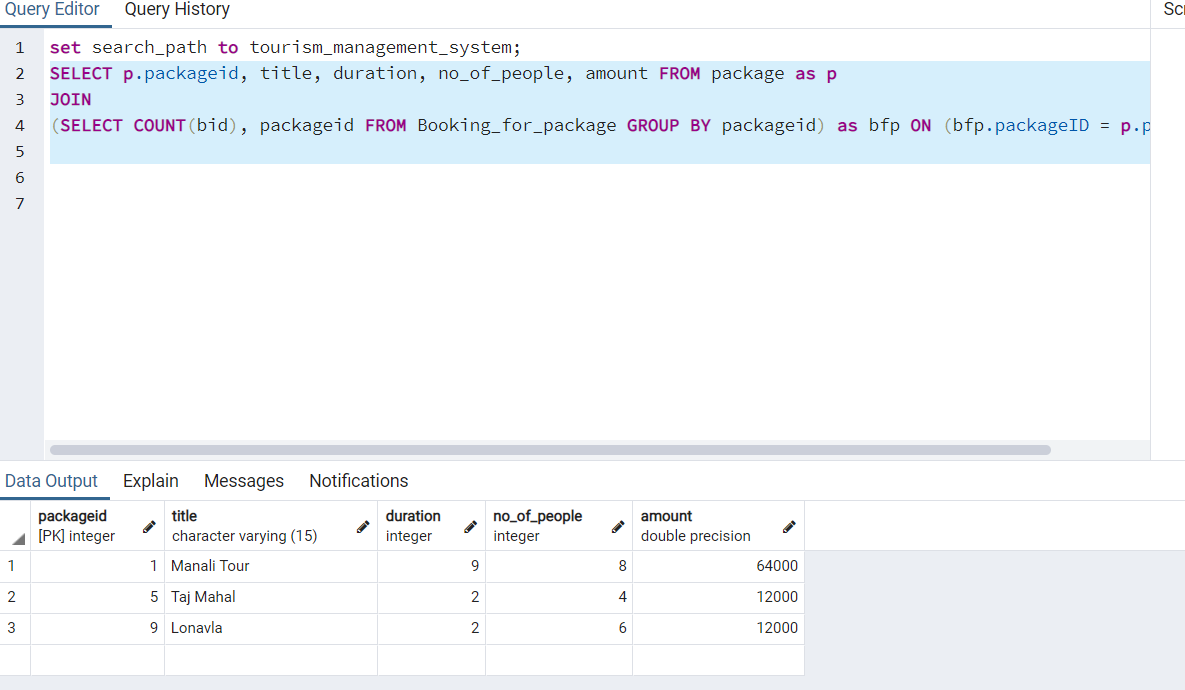
result -> p.packageid, title, duration, no\_of\_people, amount(r2)

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);



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

Relational Algebra:

r1 -> (ts, tourist\_spots) ⋈<ts.pincode = l.pincode>(l, location)

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

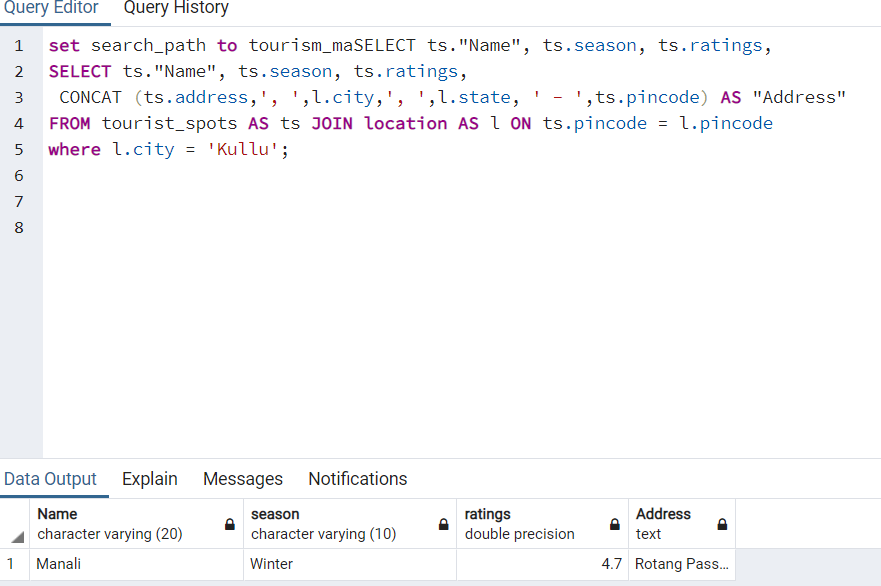
SQL Query:

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';



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

Relational Algebra:

r1 -> (l, location) ⋈<l.pincode = ts.pincode> (ts, tourist\_spots) ⋈<ts.spotid = pt.spotid> (pt, package\_includes\_spots) ⋈<pt.packageid = p.packageid> (p, package)

result -> (title, duration, no\_of\_people, amount, “Name”, rating, CONCAT (ts.address,', ',l.city,', ',l.state, ' - ',ts.pincode) -> Address) ( 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';



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

Relational Algebra:

r1 -> ℱMAX(ratings)->ratings ((ts2, tourist\_spots))

r2 -> r1 ⋈<ts2.ratings = ts1.ratings > (ts1, tourist\_spots) ⋈<ts1.pincode = l.pincode> (l, location))

result ->  “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

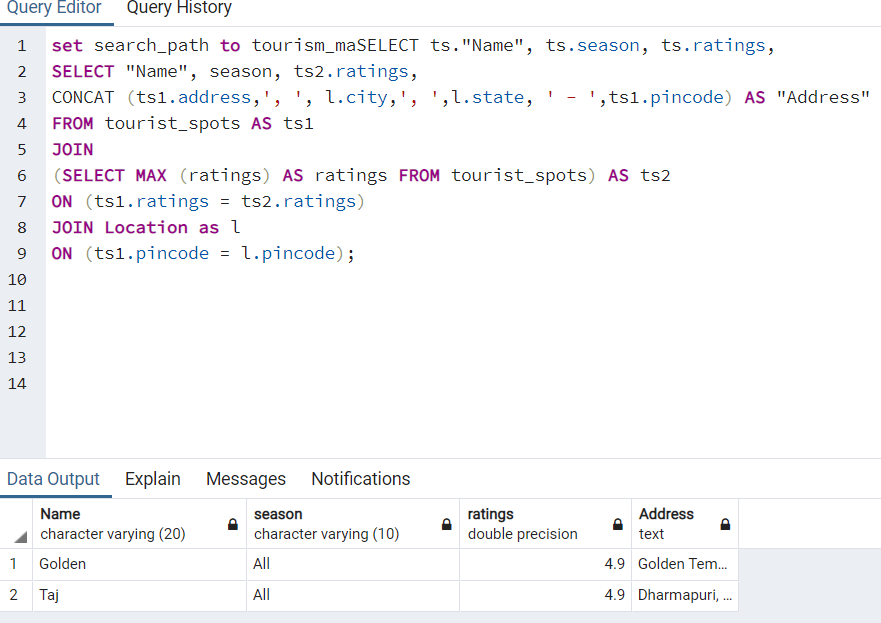
JOIN

(SELECT MAX (ratings) AS ratings FROM tourist\_spots) AS ts2

ON (ts1.ratings = ts2.ratings)

JOIN Location as l

ON (ts1.pincode = l.pincode);



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

Relational Algebra:

result -> ( “Name”, phone, foodtype, rating, CONCAT (r.address,', ', l.city, ', ', l.state, ' - ', r.pincode) -> Address) ( foodtype=”Veg”( (r, restaurant) ⋈<r.pincode = l.pincode> (l, location)))

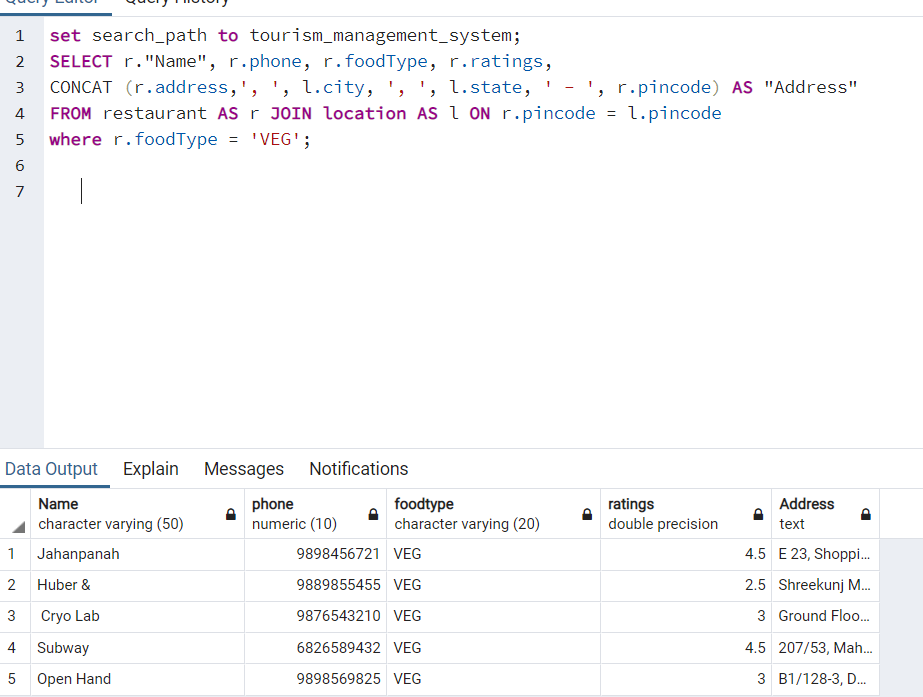
SQL Query:

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';



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

Relational Algebra:

result -> ( “Name”, phone, foodtype, rating, CONCAT (r.address,', ', l.city, ', ', l.state, ' - ', r.pincode) -> Address) ( city=”Ahmedabad”( (r, restaurant) ⋈<r.pincode = l.pincode> (l, location)))

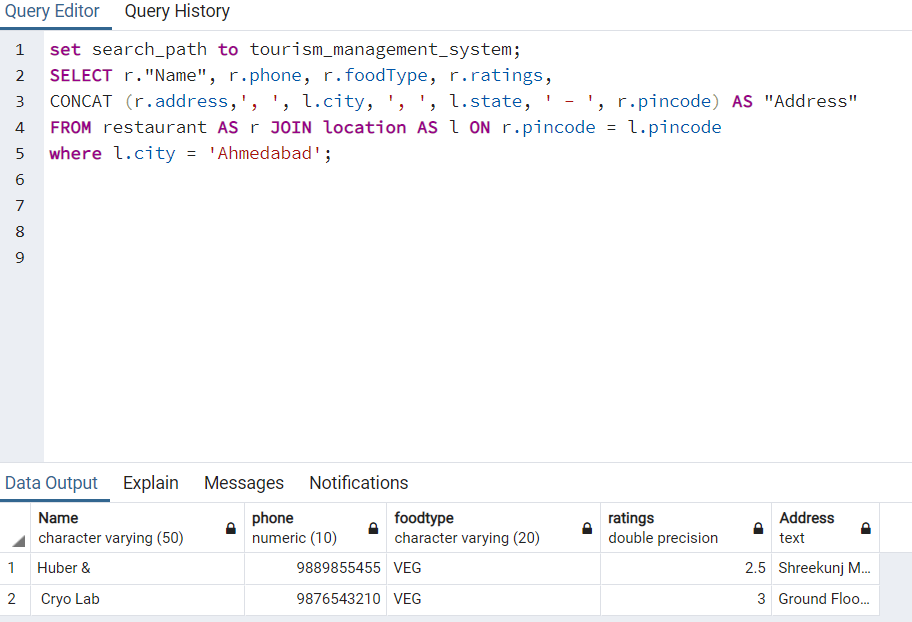
SQL Query:

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';



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

Relational Algebra:

r1 -> (l, location) ⋈<l.pincode = r.pincode> (r, restaurant) ⋈<r.rid = rc.rid> (rc, restaurant\_cuisines)

result -> ( “Name”, phone, foodtype, ratings, cuisines, CONCAT (r.address,', ', l.city, ', ', l.state, ' - ', r.pincode) -> Address) ( cuisines=”Chinese”(r1))

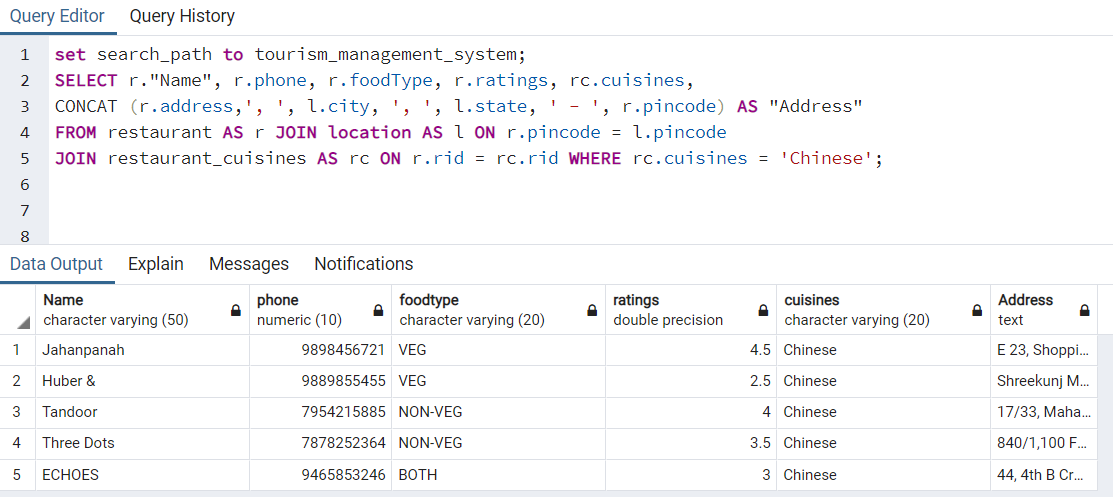
SQL Query:

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';



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

Relational Algebra:

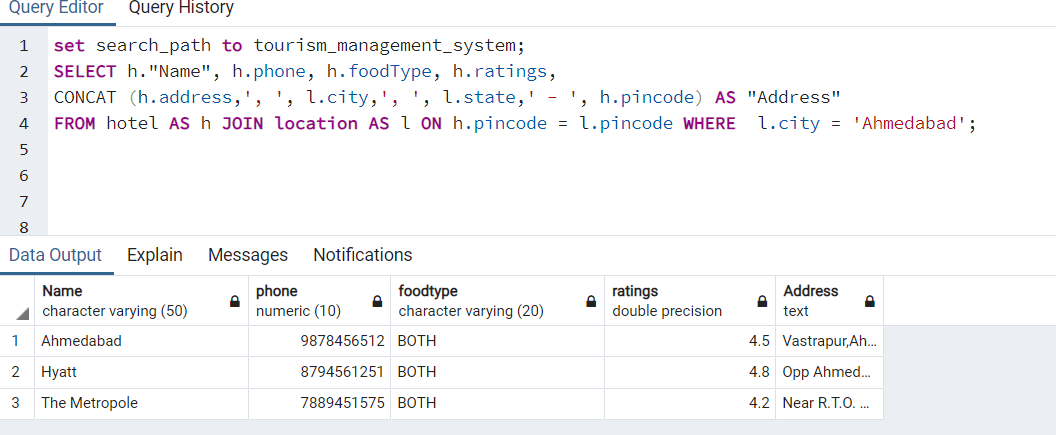
result -> ( “Name”, phone, foodtype, ratings, cuisines, CONCAT (h.address,', ', l.city,', ', l.state,' - ', h.pincode) -> Address) ( city=”Ahmedabad” ((h, hotel) ⋈<h.pincode = l.pincode> (l, location)))

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';



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

Relational Algebra:

r1 -> (l, location) ⋈<l.pincode = h.pincode> (h, hotel) ⋈<h.hotelid = hs.hotelid> (hs, hotel\_services)

result -> ( “Name”, phone, foodtype, ratings, services, CONCAT (h.address,', ', l.city,', ', l.state,' - ', h.pincode) -> Address) ( servies=”Gym”(r1))

SQL Query:

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

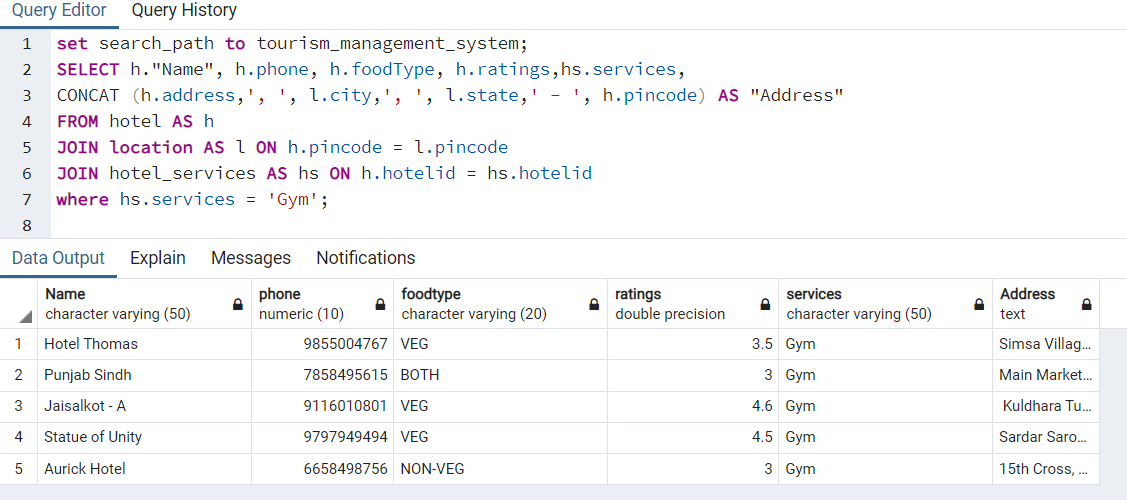
CONCAT (h.address,', ', l.city,', ', l.state,' - ', h.pincode) AS "Address"

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';



**--11) Retrieve the hotel with highest user ratings**

Relational Algebra:

r1 -> ℱMAX(ratings) (hotel)

r2 -> > (h, hotel) ⋈<h.pincode = l.pincode> (l, location)

result -> ( “Name”, phone, foodtype, ratings, services, CONCAT (h.address,', ', l.city,', ', l.state,' - ', h.pincode) -> Address) ( ratings IN (r1)=”Gym”(r2))

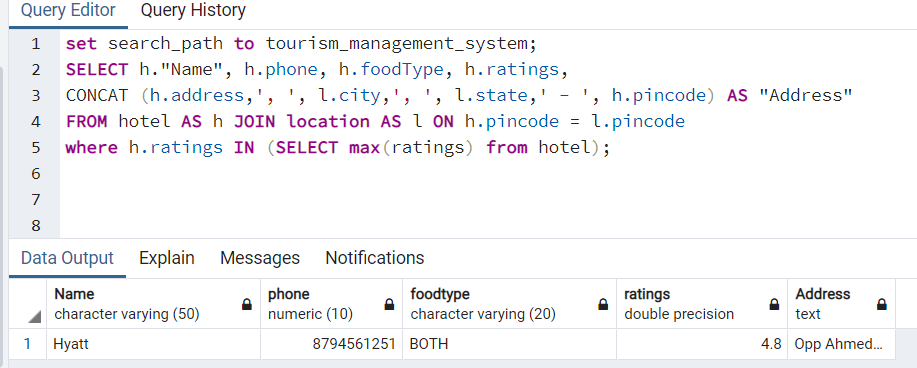
SQL Query:

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);



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

Relational Algebra:

r1-> (h, hotel) ⋈<h.pincode = l.pincode> (l, location)

result -> ( “Name”, phone, foodtype, ratings, services, CONCAT (h.address,', ', l.city,', ', l.state,' - ', h.pincode) -> Address) (ORDER BY h.ratings (r1))

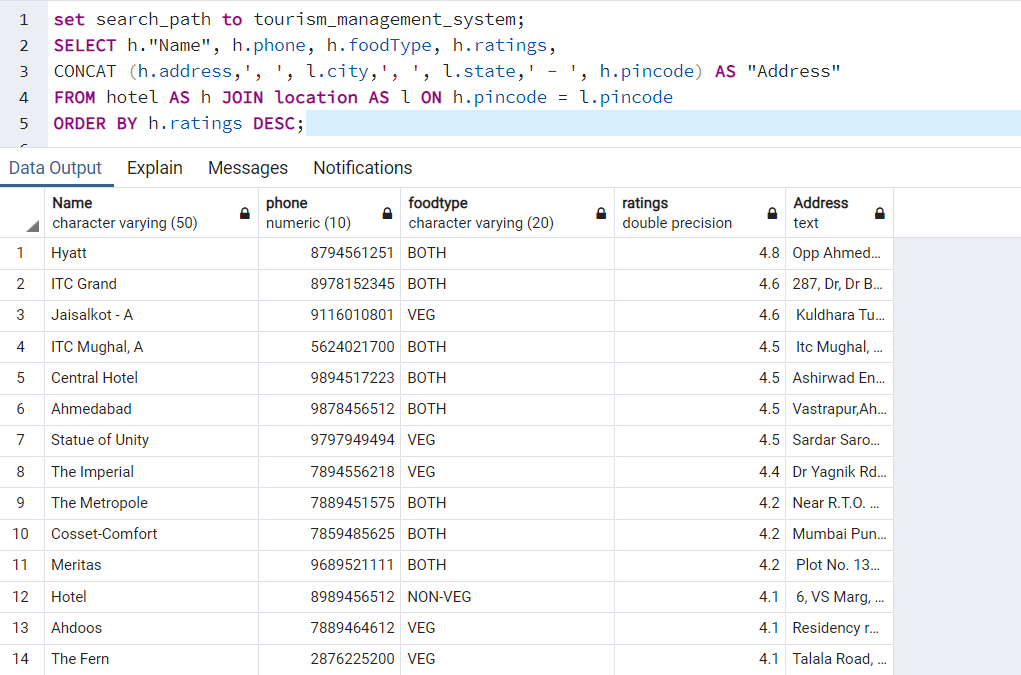
SQL Query:

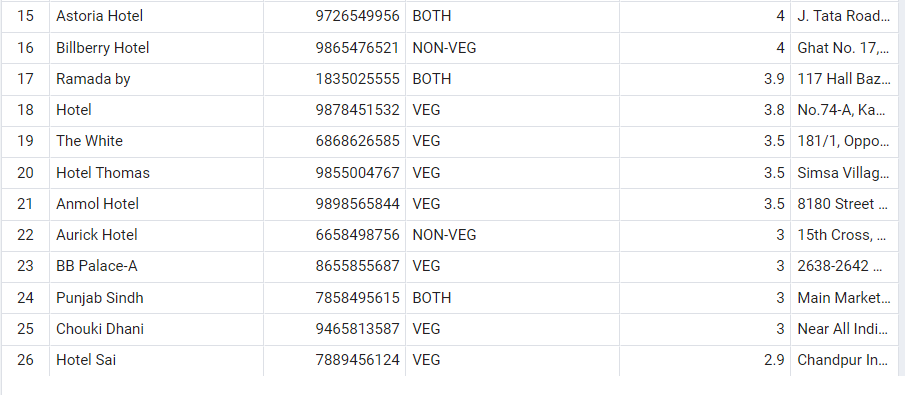
SELECT h."Name", h.phone, h.foodType, h.ratings,

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

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

ORDER BY h.ratings DESC;





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

Relational Algebra:

r1 -> (h, hotel) ⋈<h.hotelid = r.hotelid> (r, room)

result -> (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) ( ORDER BYr.rate,h.”name”,r.room\_no (r1))

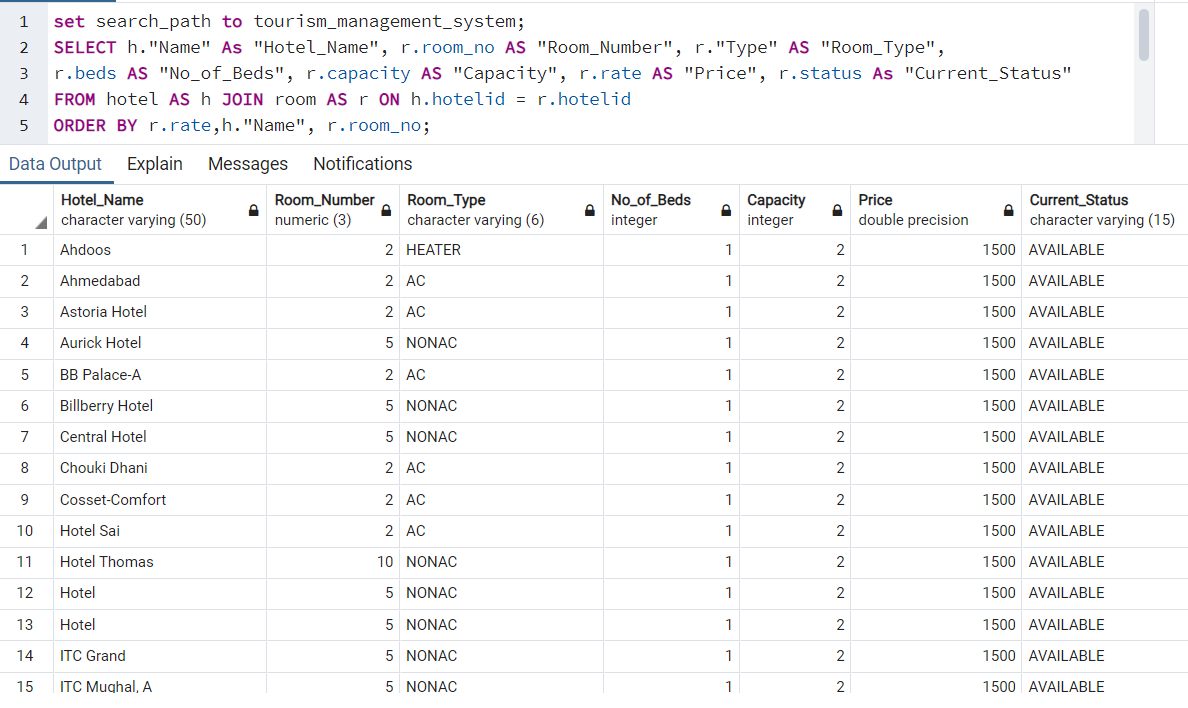
SQL Query:

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;



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

Relational Algebra:

r1 -> (h, hotel) ⋈<h.pincode=l.pincode> (l, location) ⋈<h.hotelid= r.hotelid> (r, room)

⋈< h.hotelid=rf.hotelid and r.room\_no=rf.roomno > (rf, room\_facilities)

Result -> (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,l.city) ((rf.facility='Cable TV' and l.city = 'Amritsar')

ORDER BYr.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

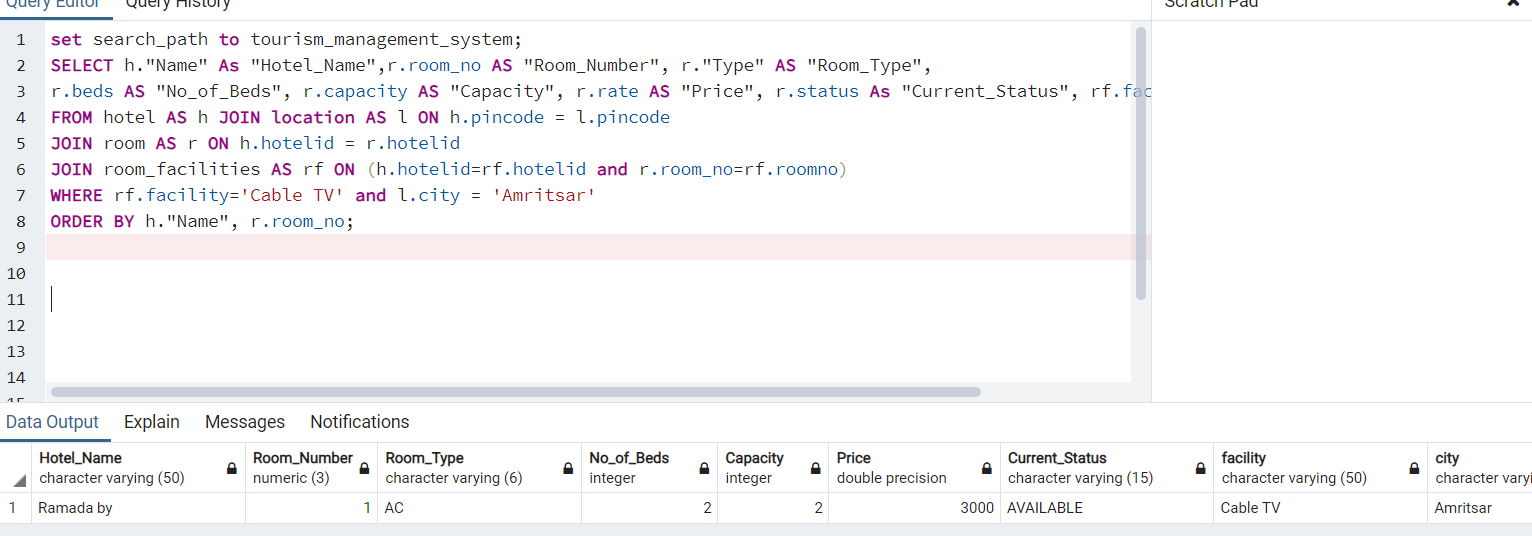
FROM hotel AS h JOIN location AS l ON h.pincode = l.pincode

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;



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

Relational Algebra:

r1 -> (g, guide) ⋈<g.guideid = pg.guideid> (pg, guideid) ⋈<pg.packageid = p.packageid> (p, package)

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

SQL Query:

SELECT CONCAT (g.fname,' ', g.lname) AS "Guide\_Name",

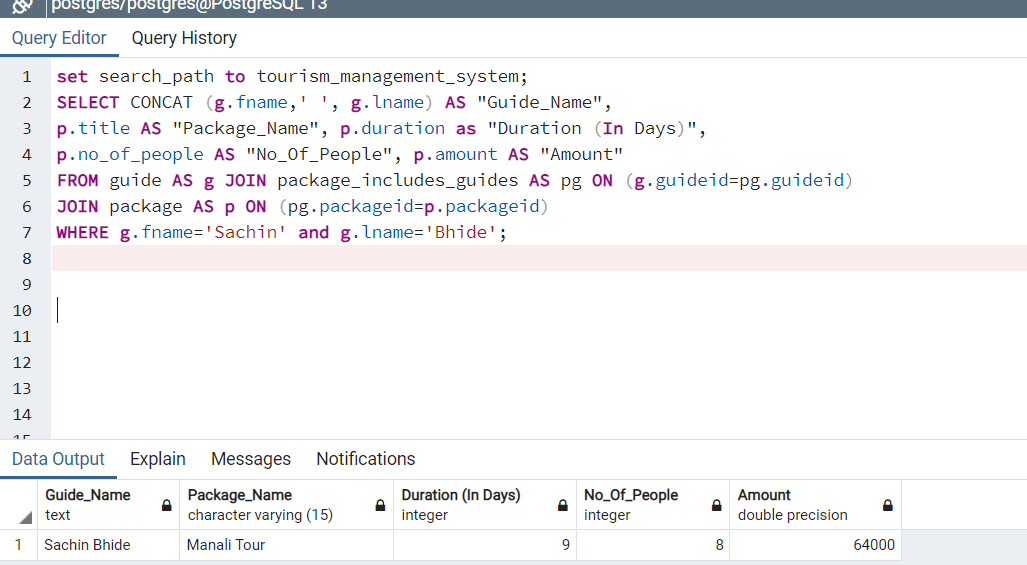
p.title AS "Package\_Name", p.duration as "Duration (In Days)",

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';



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

Relational Algebra:

r1 -> (h, hotel) ⋈<h.hotelid = ph.hotelid> (ph, package\_includes\_hotels) ⋈<pg.packageid = p.packageid> (p, package)

result -> (h.”Name” -> Hotel\_Name, p.title -> package\_name, p.duration -> Duration(in days), p.no\_of\_people -> No\_of\_people, p.amount -> Amount)( h."Name"='Hotel Thomas Villa' (r1))

SQL Query:

SELECT h."Name" AS "Hotel\_Name",

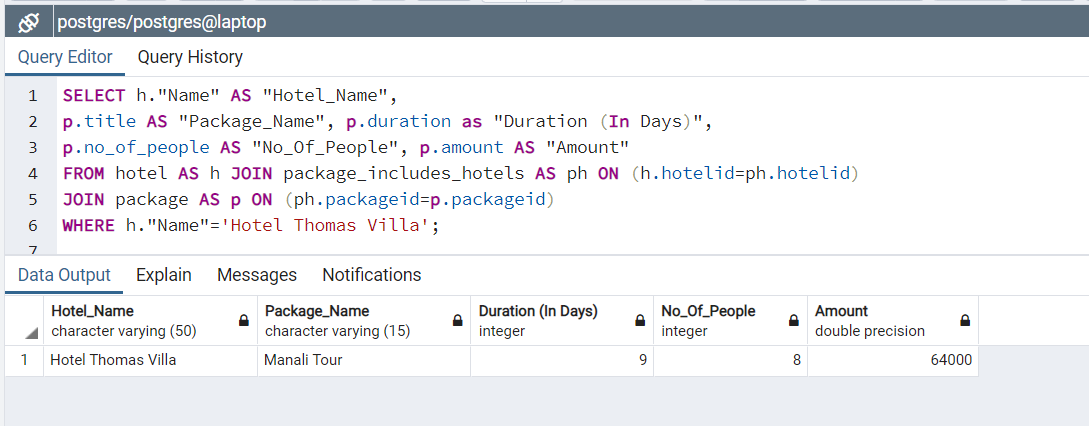
p.title AS "Package\_Name", p.duration as "Duration (In Days)",

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';



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

Relational Algebra:

r1 -> (t, tourist\_spots) ⋈<t.spotid = ps.spotid> (ps, package\_includes\_spots) ⋈<ps.packageid = p.packageid> (p, package)

result -> (t.”Name” -> Spot\_Name, p.title -> package\_name, p.duration -> Duration(in days), p.no\_of\_people -> No\_of\_people, p.amount -> Amount)( t."Name"='Manali' (r1))

SQL Query:

SELECT t."Name" AS "Spot\_Name",

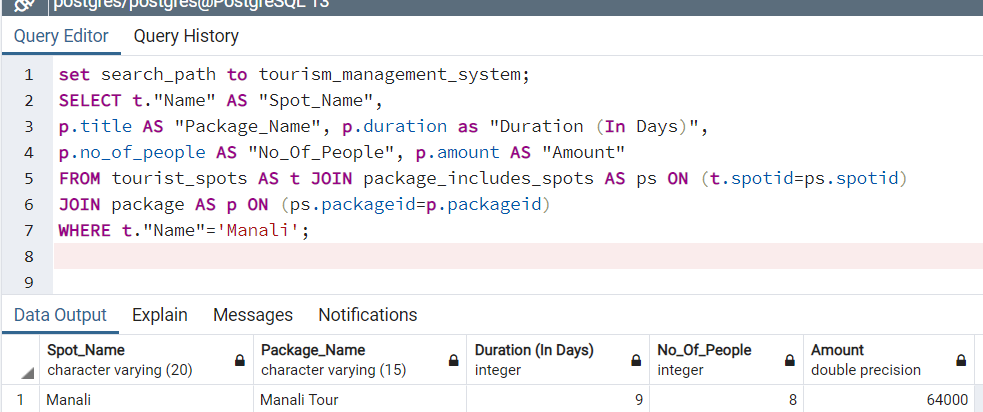
p.title AS "Package\_Name", p.duration as "Duration (In Days)",

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';



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

Relational Algebra:

r1 -> (ts, tourist\_spots) ⋈<ts.pincode = l.pincode> (l, location)

result -> (t.”Name” -> Spot\_Name, ts.season -> Season, ts.ratings -> Ratings,

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

SQL Query:

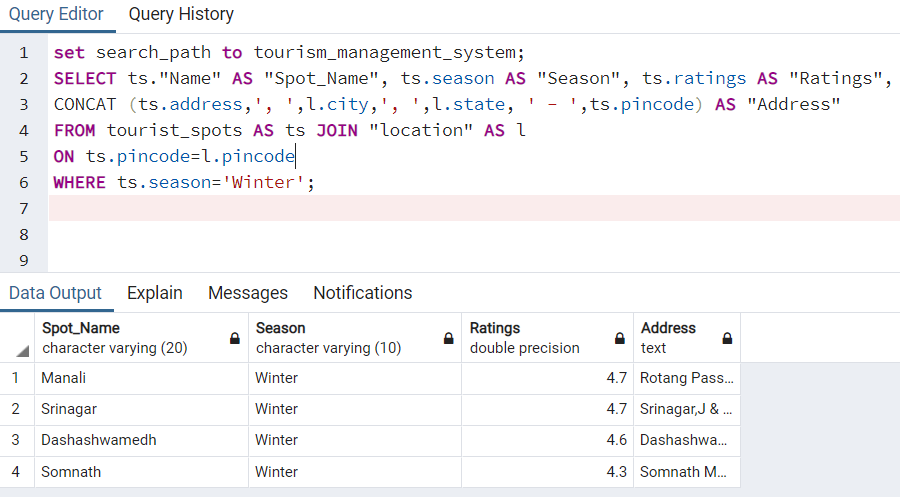
SELECT ts."Name" AS "Spot\_Name", ts.season AS "Season", ts.ratings AS "Ratings",

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';



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

Relational Algebra:

r1 -> (h, hotel) ⋈<h.pincode=l.pincode> (l, location) ⋈<h.hotelid= r.hotelid> (r, room)

LEFT ⋈< h.hotelid=rf.hotelid and r.room\_no=rf.roomno > (rf, room\_facilities)

Result -> (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)

((r.rate BETWEEN 1500 and 2000) ORDER BYr.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"

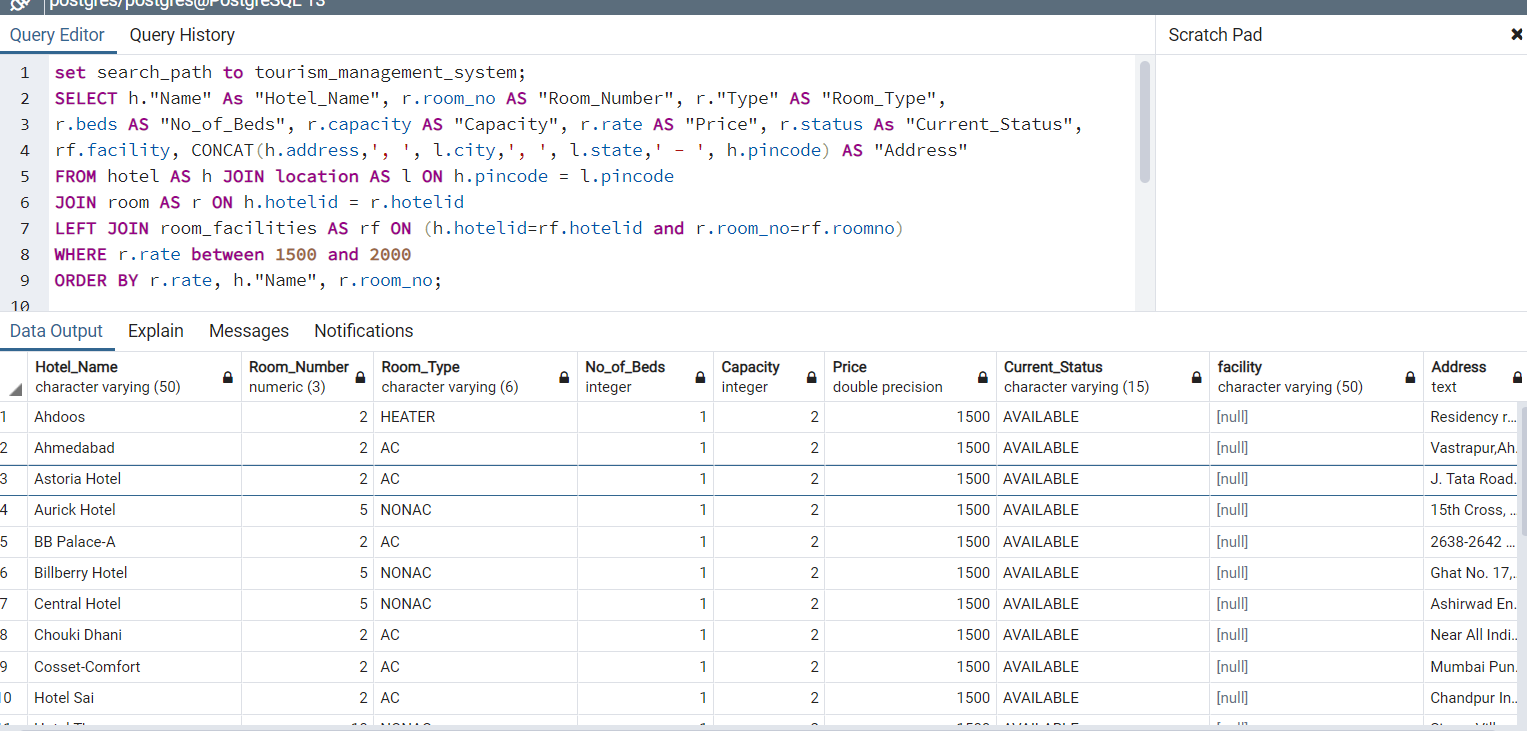
FROM hotel AS h JOIN location AS l ON h.pincode = l.pincode

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;



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

Relational Algebra:

r1-> (pg.guideid) ( (p.isActive=’TRUE’) ((pg, package\_include\_guide) ⋈<pg.package.id=p.packageid> (p,package)))

r2 -> (g,guide) ⋈<g.pincode=l.pincode> (l, location)

result-> (CONCAT(g.fname,’ ‘,’g.lname)->Guide\_Name,g.email,g.phone,g.age,g.gender,(g.address,’,’,l.city,’,’,l.state,’-‘,g.pincode)->Address) ( (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,

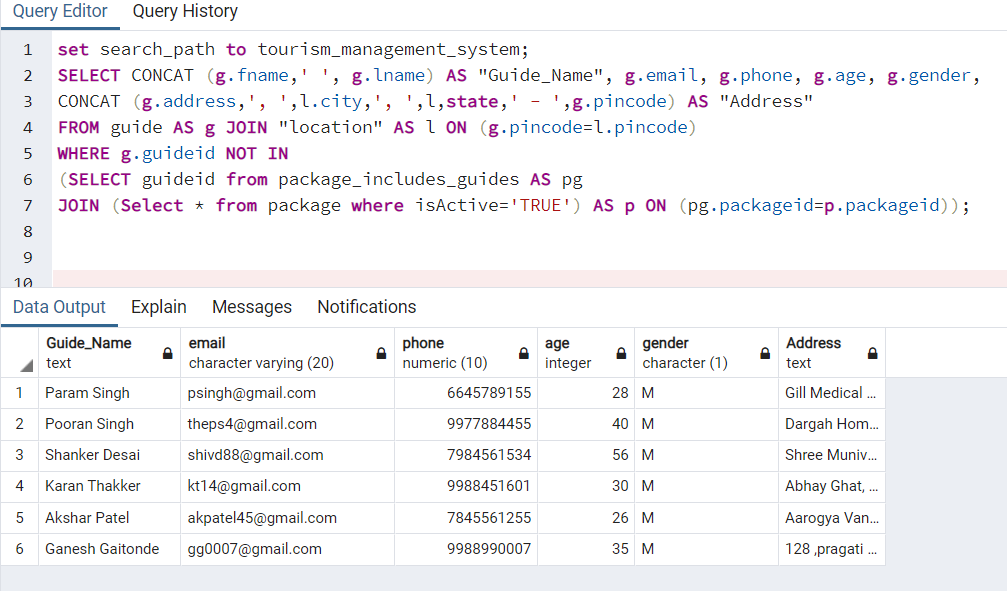
CONCAT (g.address,', ',l.city,', ',l,state,' - ',g.pincode) AS "Address"

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));



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

Relational Algebra:

r1 -> (hl, hotel) ⋈<hl.hotelid = r.hotelid> (r, hotelid ℱCOUNT(room\_no) -> rooms (room))

result -> (hl.”Name”,rooms, hl.ratings, hl.address)( rooms > 0 AND hl.isActive=true (r1))

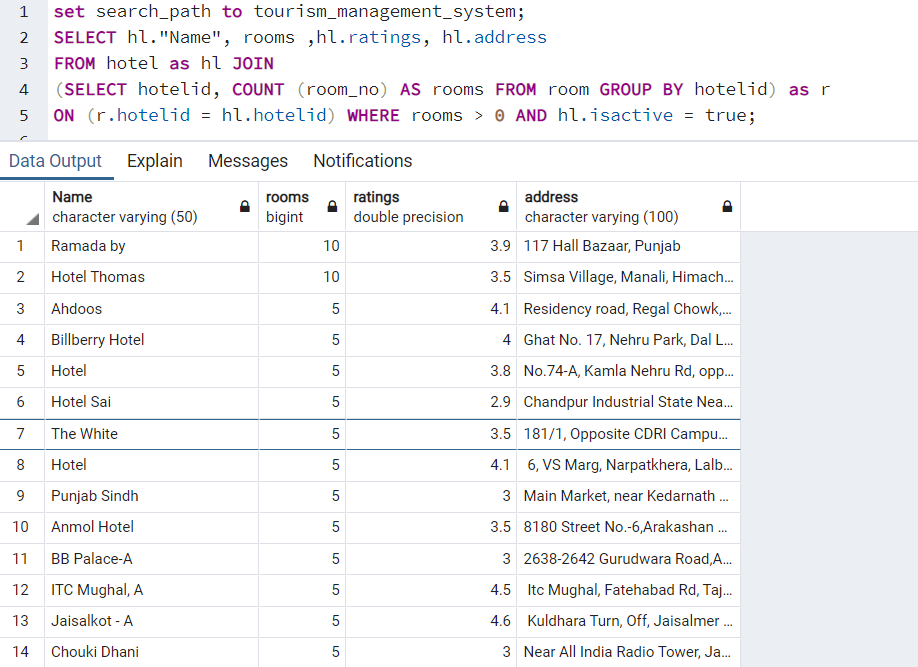
SQL Query:

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;



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

Relational Algebra:

r1 -> (h, hotel) ⋈<h.hotelid = r.hotelid> (r,ℱCOUNT(\*) (room))

result ->  h.”Name” = ‘Ahdoos Hotel’ AND r.status = ‘AVAILABLE’ (r1)

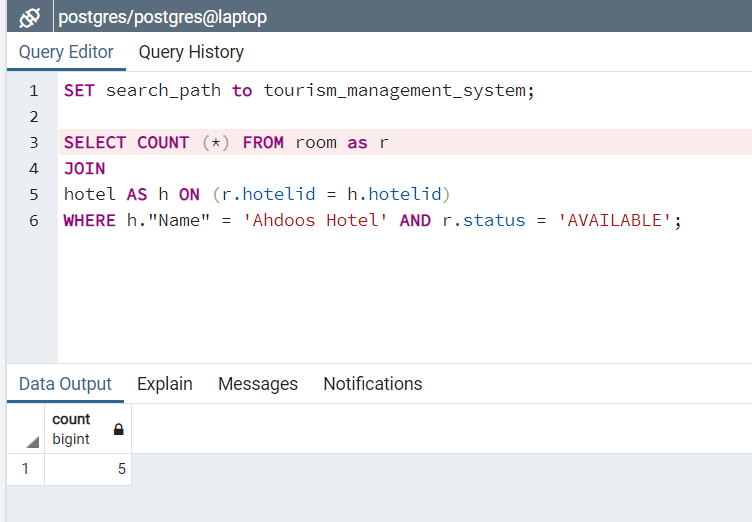
SQL Query:

SELECT COUNT (\*) FROM room as r

JOIN

hotel AS h ON (r.hotelid = h.hotelid)

WHERE h."Name" = 'Ahdoos Hotel' AND r.status = 'AVAILABLE';



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

Relational Algebra:

r1 -> (b, booking) LEFT ⋈<b.bid = bfc.bid> (bfc, bid ℱCOUNT(fname) -> no\_of\_co\_passengers (booking\_copassenger)) FULL OUTER ⋈<b.bid = bfp.bid> (bfp, booking\_for\_package) FULL OUTER ⋈<bfp.packageid = p.packageid> (p, package) FULL OUTER ⋈<b.bid = bfh.bid> (bfh, booking\_for\_hotel) FULL OUTER ⋈<bfh.hotelid = h.hotelid> (h, hotel) FULL OUTER ⋈<h.hotelid = pih.hotelid>  (pih, package\_includes\_hotels) FULL OUTER ⋈<pih.hotelid = hp.hotelid> (hp, hotel) ⋈<b.userid = u.userid> (u,”User”)

result -> (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) ( 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)

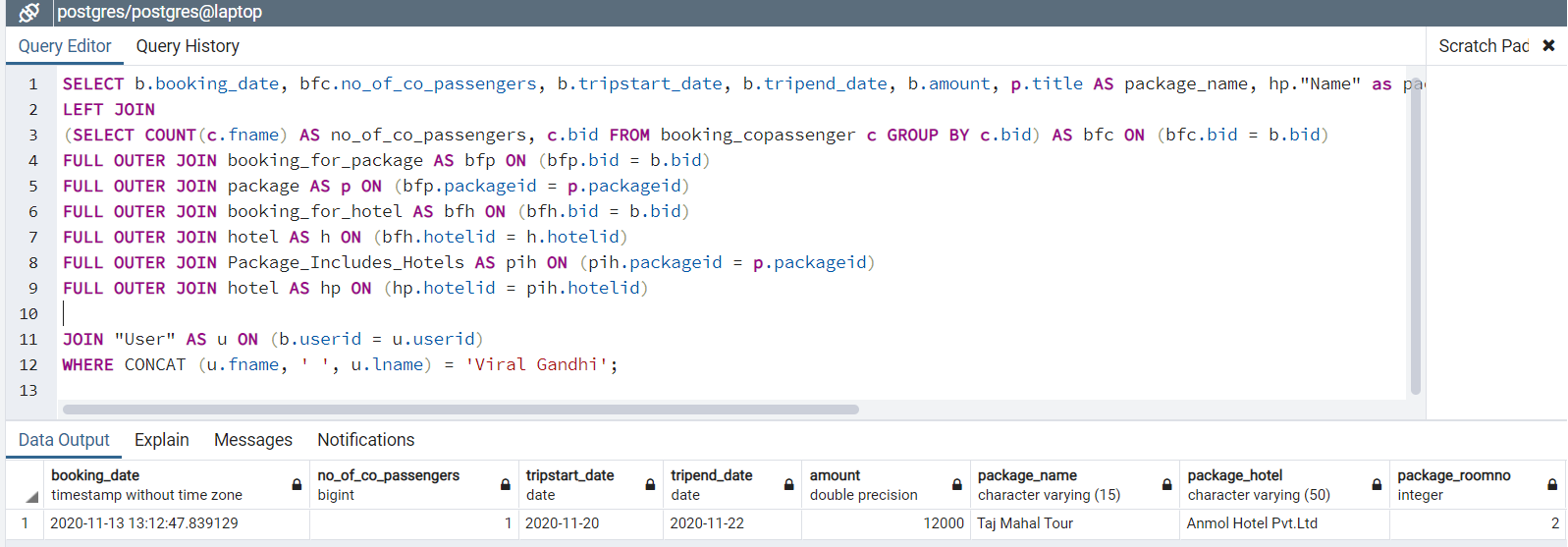
FULL OUTER JOIN booking\_for\_hotel AS bfh ON (bfh.bid = b.bid)

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'; 

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

Relational Algebra:

r1 -> (b, booking) LEFT ⋈<b.bid = bfc.bid> (bfc, bid ℱCOUNT(fname) -> no\_of\_co\_passengers (booking\_copassenger)) FULL OUTER ⋈<b.bid = bfp.bid> (bfp, booking\_for\_package) FULL OUTER ⋈<bfp.packageid = p.packageid> (p, package) FULL OUTER ⋈<b.bid = bfh.bid> (bfh, booking\_for\_hotel) FULL OUTER ⋈<bfh.hotelid = h.hotelid> (h, hotel) FULL OUTER ⋈<h.hotelid = pih.hotelid>  (pih, package\_includes\_hotels) FULL OUTER ⋈<pih.hotelid = hp.hotelid> (hp, hotel) ⋈<b.userid = u.userid> (u,”User”)

result -> (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) ( CONCAT (u.fname,’ ‘,u.lname) = ‘Viral Gandhi’ and b.booking\_date < '2020-11-14' AND b.booking\_date >= '2020-11-09' (r1))

SQL Query:

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)

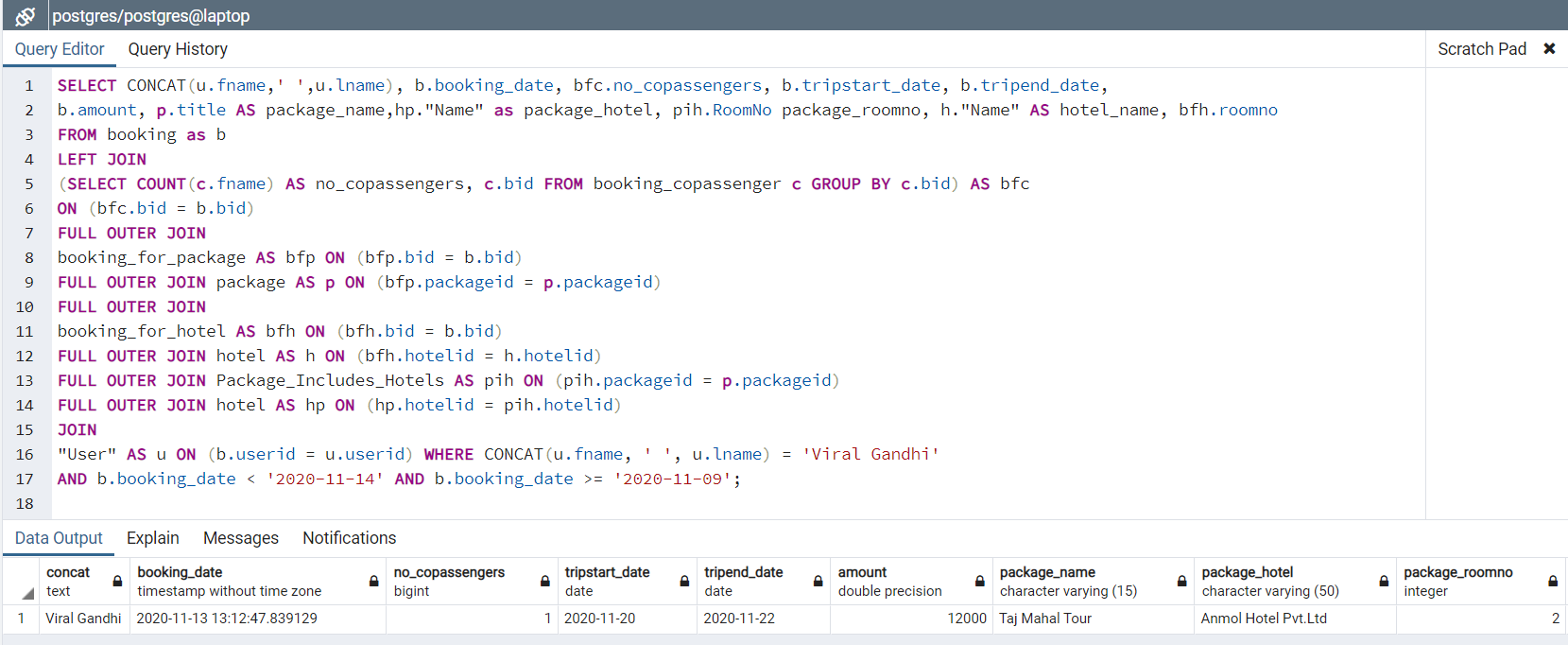
FULL OUTER JOIN Package\_Includes\_Hotels AS pih ON (pih.packageid = p.packageid)

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';



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

Relational Algebra:

r1 -> (b, booking) LEFT ⋈<b.bid = bfc.bid> (bfc, bid ℱCOUNT(fname) -> no\_of\_co\_passengers (booking\_copassenger)) FULL OUTER ⋈<b.bid = bfp.bid> (bfp, booking\_for\_package) FULL OUTER ⋈<bfp.packageid = p.packageid> (p, package) FULL OUTER ⋈<b.bid = bfh.bid> (bfh, booking\_for\_hotel) FULL OUTER ⋈<bfh.hotelid = h.hotelid> (h, hotel) FULL OUTER ⋈<h.hotelid = pih.hotelid>  (pih, package\_includes\_hotels) FULL OUTER ⋈<pih.hotelid = hp.hotelid> (hp, hotel) ⋈<b.userid = u.userid> (u,”User”)

result -> (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) (b.booking\_date < '2020-11-14' AND b.booking\_date >= '2020-11-09' (r1))

SQL Query:

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)

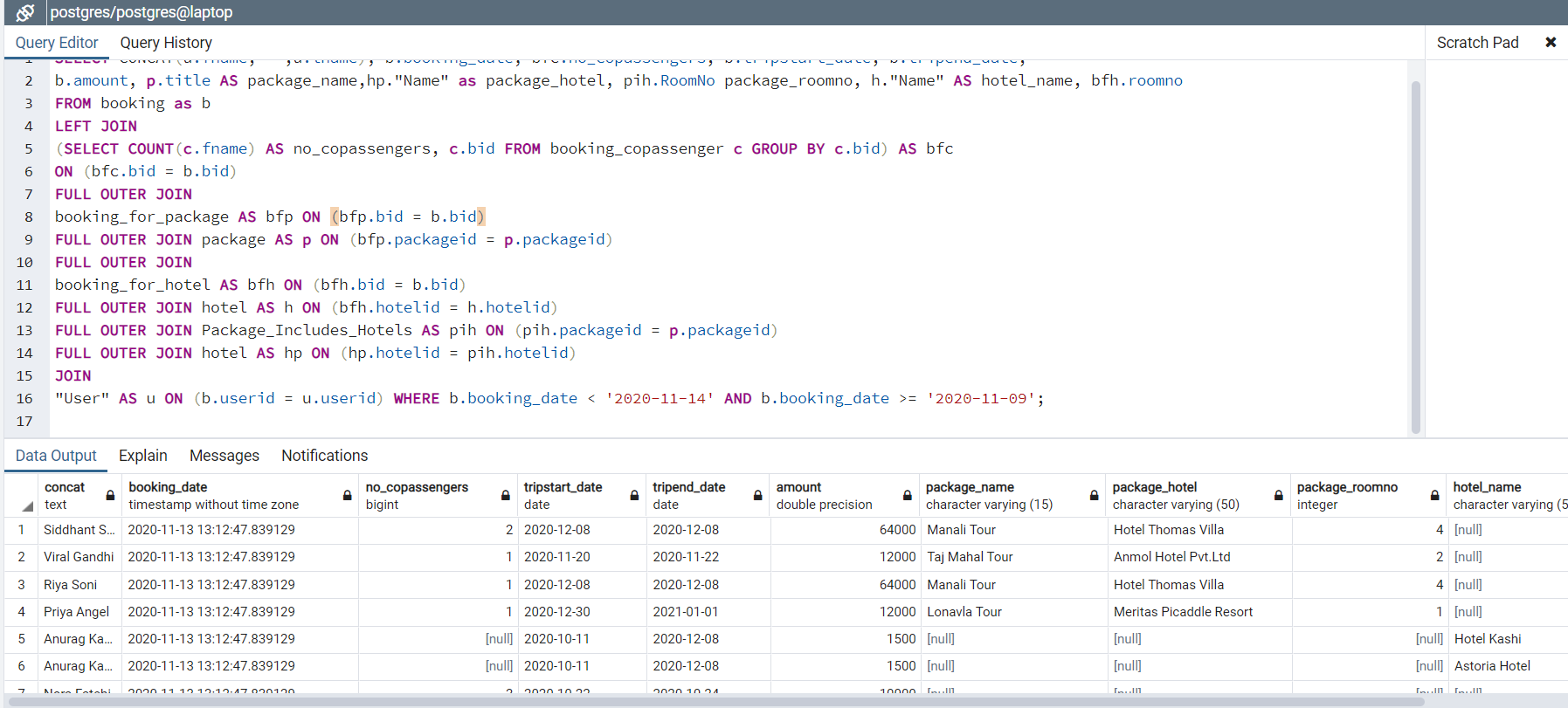
FULL OUTER JOIN hotel AS h ON (bfh.hotelid = h.hotelid)

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';



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

**Relational Algebra**

r1 -> (r,room) ⋈< r.hotelid = h.hotelid >(h,hotel)

r2-> (h."Name" = 'Hotel Thomas' and r.room\_no = 5 (r1))

r3-> ( bfh,booking\_for\_hotel)⋈<r2.hotelid = bfh.hotelid > (r2)

r4-> (b,booking)⋈<r3.bid = b.bid > (r3)

r5-> (u,User)⋈< u.userid = r4.userid > (r4)

result-> (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

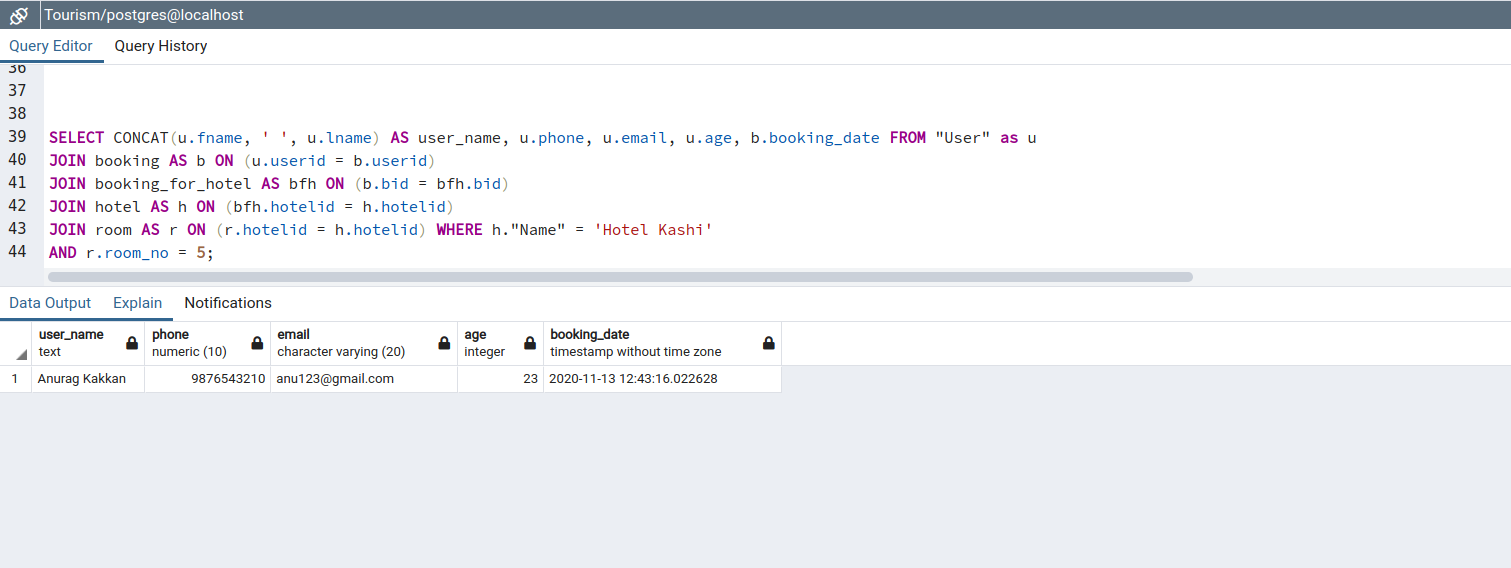
JOIN booking AS b ON (u.userid = b.userid)

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-> (bfh, booking\_for\_hotel) ⋈< bfh.hotelid = h.hotelid >(h,hotel)

r2->  h."Name" = 'Hotel Kashi'(r1)

r3-> (b,booking) ⋈<b.bid=r2.bid>(r2)

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

(CONCAT(u.fname, ' ', u.lname)->user\_name, b.booking\_date, bfh.roomno)(r4)

**SQL**

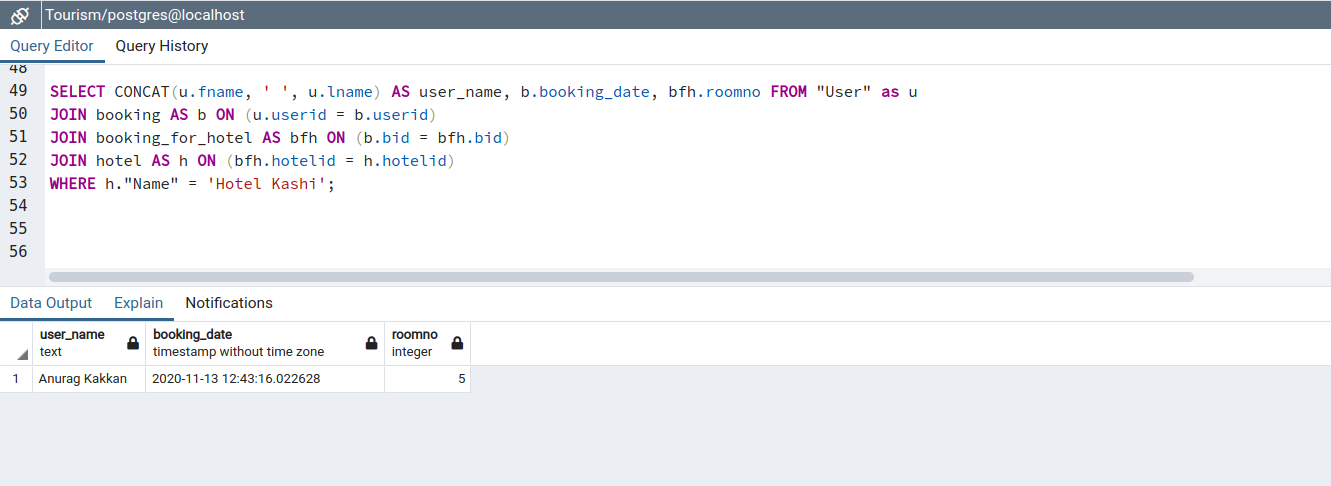
SELECT CONCAT(u.fname, ' ', u.lname) AS user\_name, b.booking\_date, bfh.roomno FROM "User" as u

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->(bc, booking\_copassenger) ⋈< bc.userid = c.uid AND bc.fname = c.fname AND

bc.lname = c.lname>(c,copassanger)

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

r3-> (b,booking) ⋈<b.bid=r2.bid>(r2)

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

( 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)[Messages](http://127.0.0.1:41835/datagrid/panel/4561795?is_query_tool=true&sgid=1&sid=1&server_type=pg&did=16389)

**SQL**

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';

