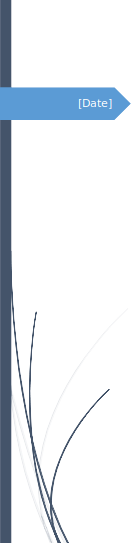


Lab 8 QUERIES

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**

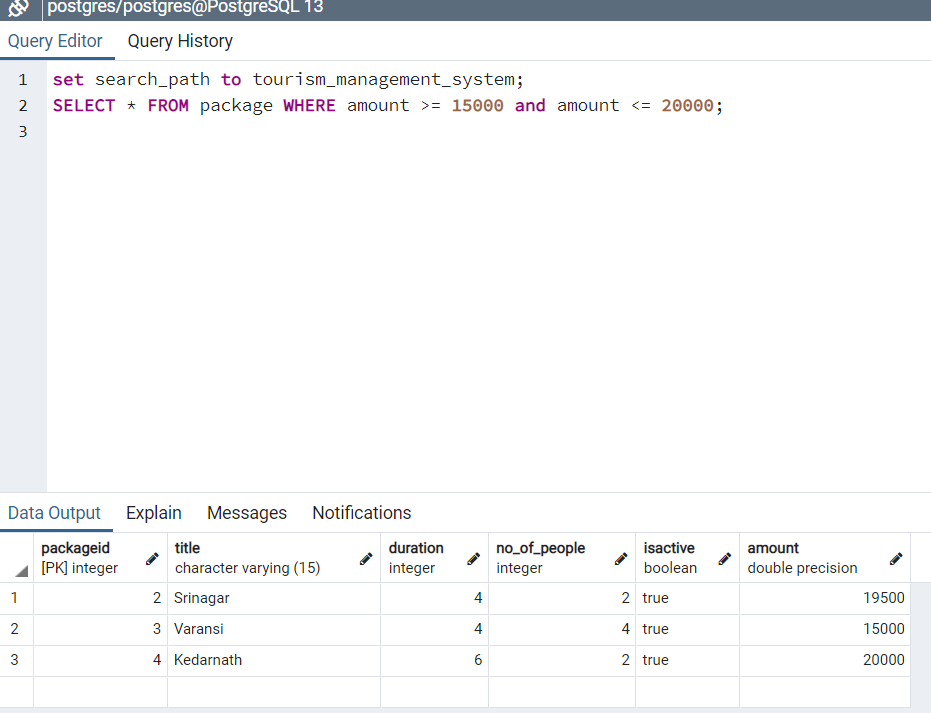
--Relational Algebra:

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

--SQL Query:

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

--Output:



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

--Relational Algebra:

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

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

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

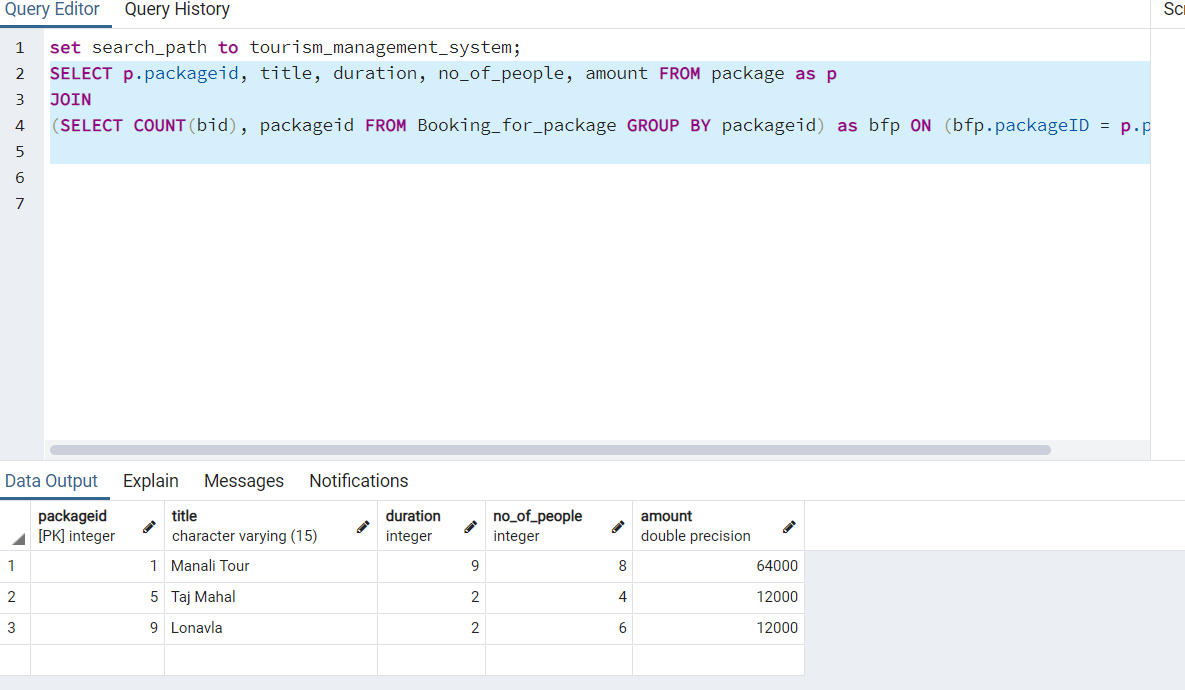
--SQL Query:

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);

--Output:



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

--Relational Algebra:

r1 -> ρ(ts, tourist\_spots) ⋈<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:

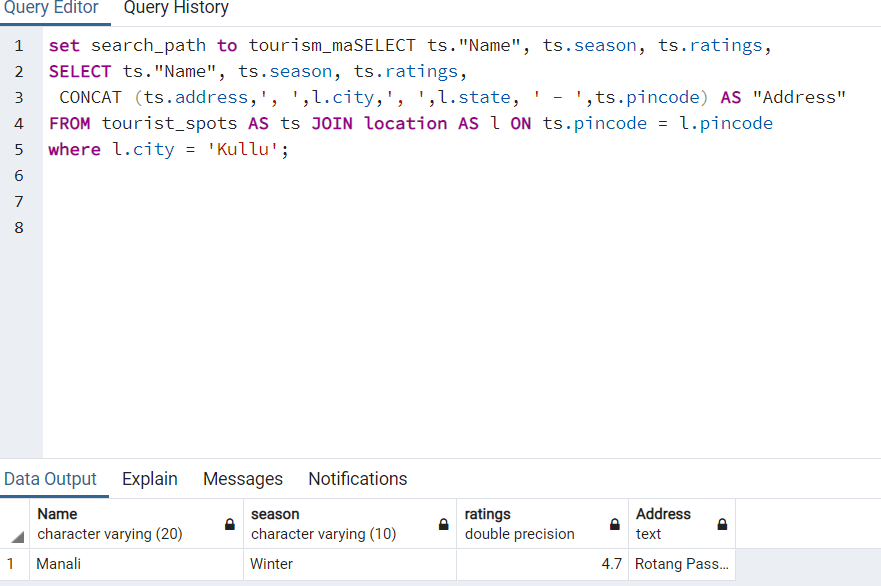
SELECT ts."Name", ts.season, ts.ratings,

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';

--Output:



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

--Relational Algebra:

r1 -> ρ(l, location) ⋈<l.pincode = ts.pincode> ρ(ts, tourist\_spots) ⋈<ts.spotid = pt.spotid> ρ(pt, package\_includes\_spots) ⋈<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) (σ 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';

--Output:



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

--Relational Algebra:

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

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

result -> Π “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

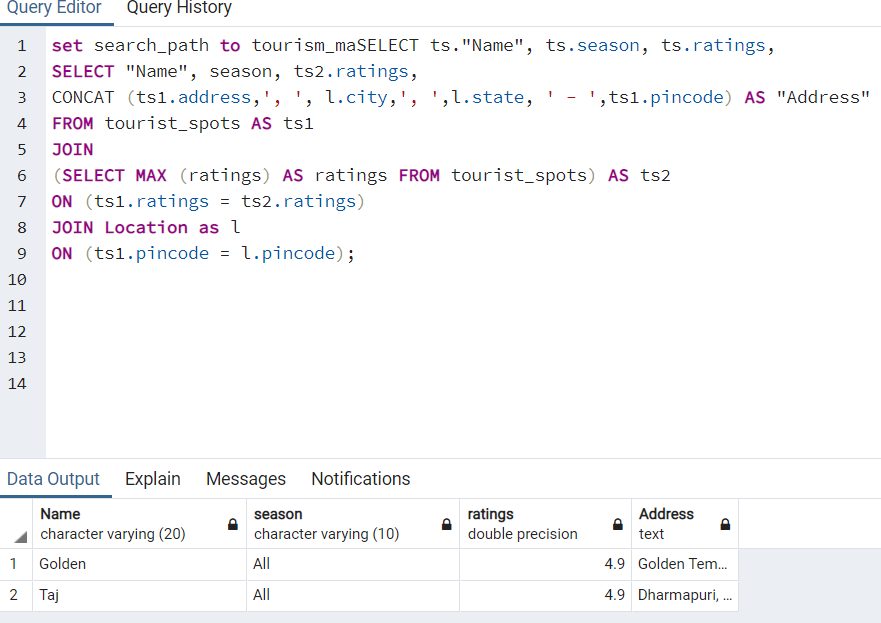
(SELECT MAX (ratings) AS ratings FROM tourist\_spots) AS ts2

ON (ts1.ratings = ts2.ratings)

JOIN Location as l

ON (ts1.pincode = l.pincode);

--Output:



**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) ⋈<r.pincode = l.pincode> ρ(l, location)))

--SQL Query:

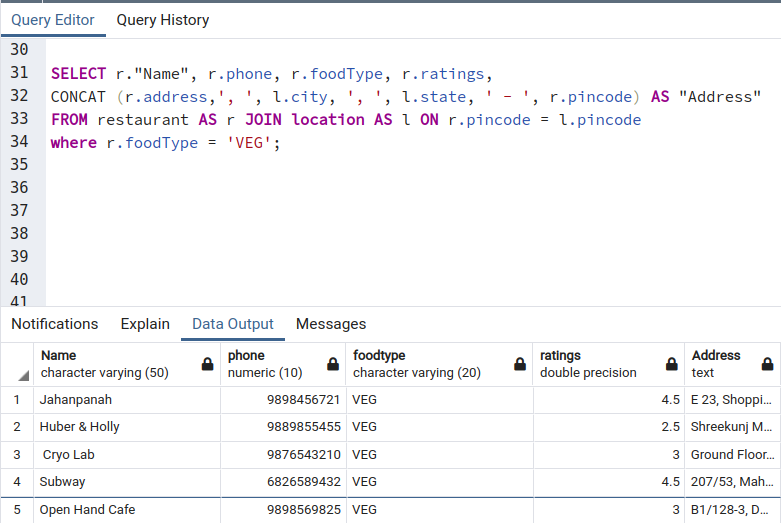
SELECT r."Name", r.phone, r.foodType, r.ratings,

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';

--Output:



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

--Relational Algebra:

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

--SQL Query:

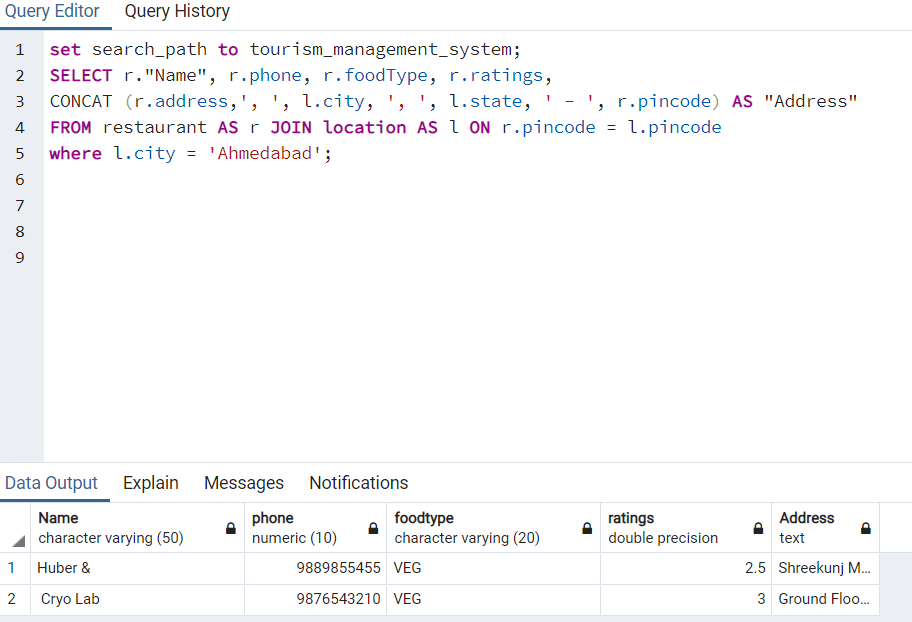
SELECT r."Name", r.phone, r.foodType, r.ratings,

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';

--Output:



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

--Relational Algebra:

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

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

--SQL Query:

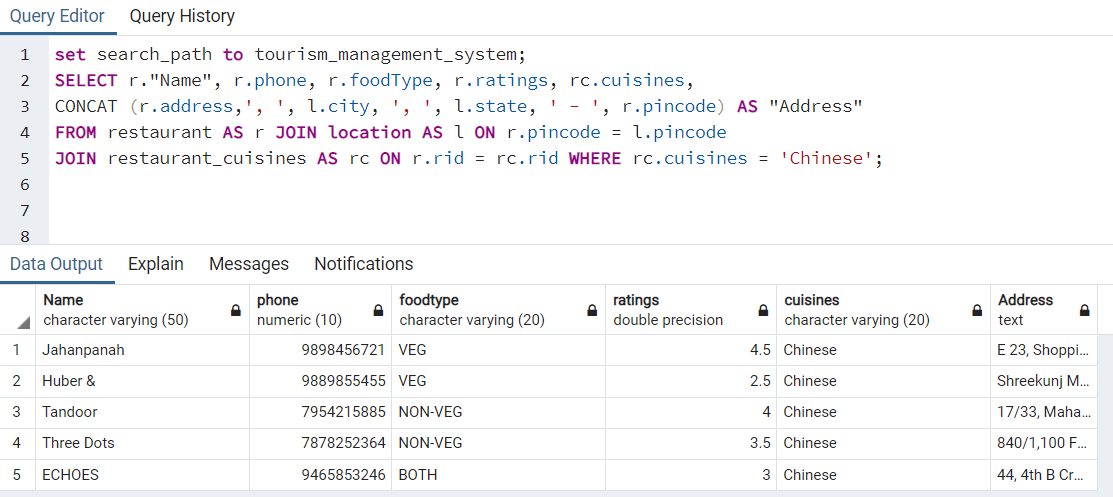
SELECT r."Name", r.phone, r.foodType, r.ratings, rc.cuisines,

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';

--Output:



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

--Relational Algebra:

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

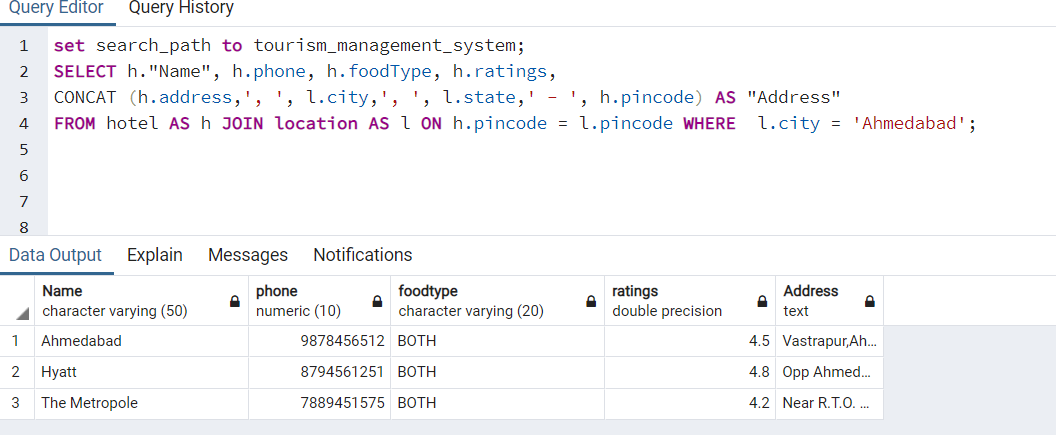
--SQL Query:

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';

--Output:



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

--Relational Algebra:

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

result -> Π( “Name”, phone, foodtype, ratings, services, CONCAT (h.address,', ', l.city,', ', l.state,' - ', h.pincode) -> Address) (σ 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"

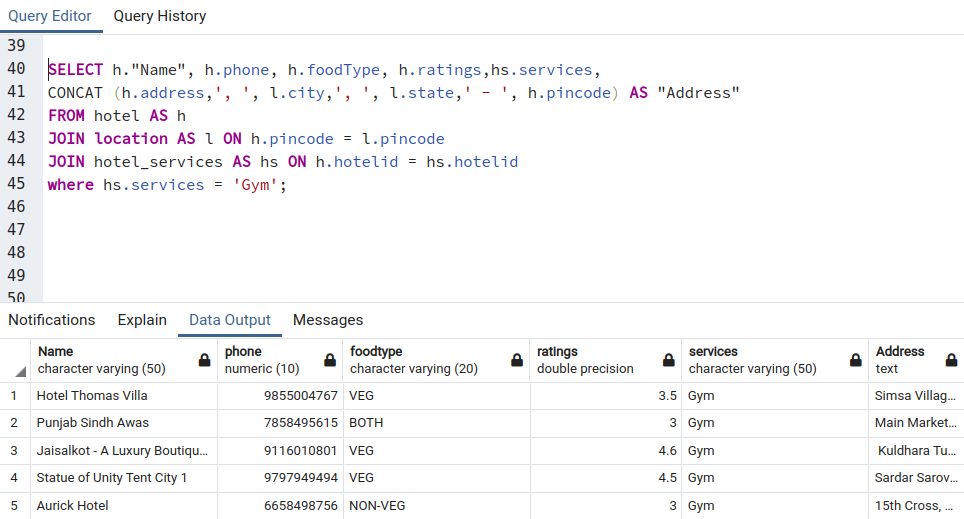
FROM hotel AS h

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

JOIN hotel\_services AS hs ON h.hotelid = hs.hotelid

where hs.services = 'Gym';

--Output:



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

--Relational Algebra:

r1 -> ℱMAX(ratings) (hotel)

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

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

--SQL Query:

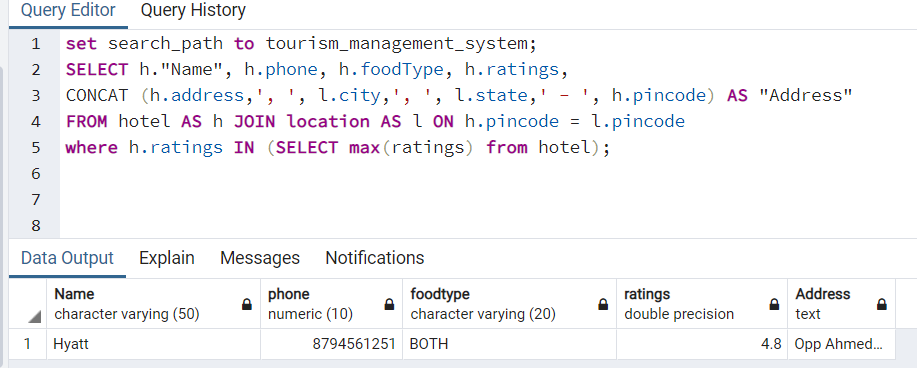
SELECT h."Name", h.phone, h.foodType, h.ratings,

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);

--Output:



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

--Relational Algebra:

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

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

--SQL Query:

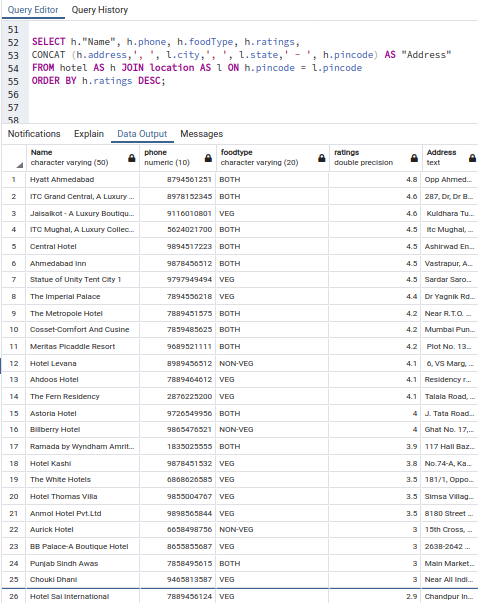
SELECT h."Name", h.phone, h.foodType, h.ratings,

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;

--Output:



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

--Relational Algebra:

r1 -> ρ(h, hotel) ⋈<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 BYr.rate,h.”name”,r.room\_no (r1))

--SQL Query:

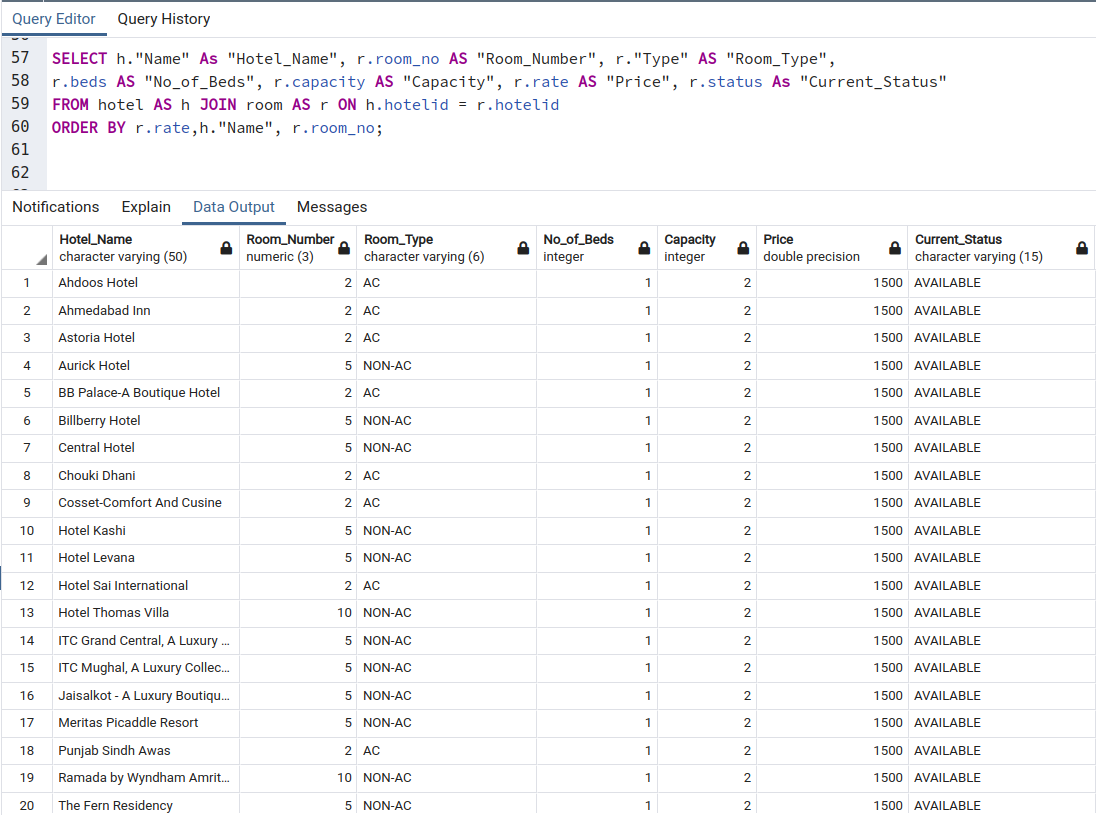
SELECT h."Name" As "Hotel\_Name", r.room\_no AS "Room\_Number", r."Type" AS "Room\_Type",

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;

--Output:



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

--Relational Algebra:

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

⋈< h.hotelid=rf.hotelid and r.room\_no=rf.room\_no > 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,l.city) (σ(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.room\_no)

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

ORDER BY h."Name", r.room\_no;

--Output:



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

--Relational Algebra:

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

result -> Π(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)(σ 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)",

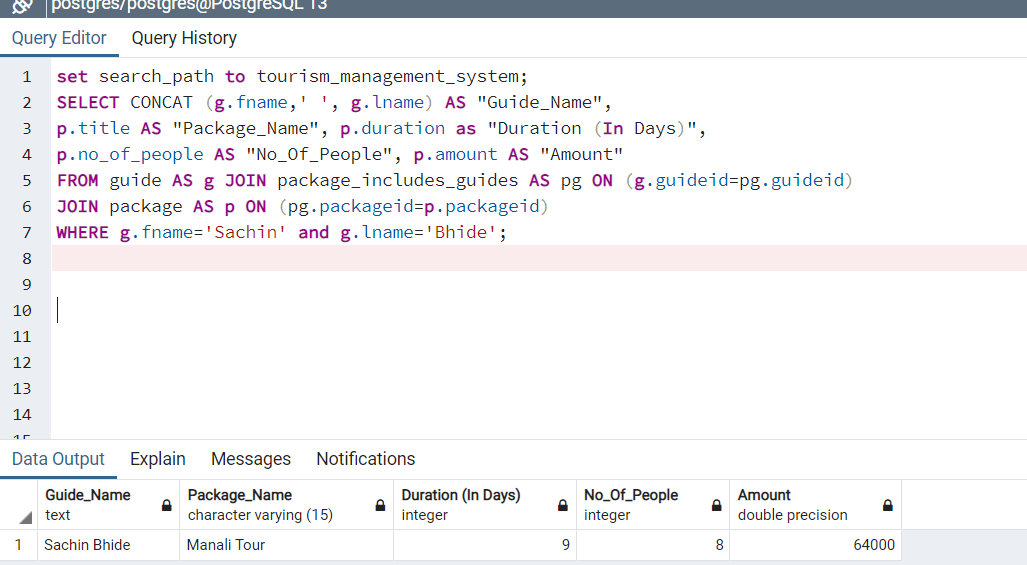
p.no\_of\_people AS "No\_Of\_People", p.amount AS "Amount"

FROM guide AS g JOIN package\_includes\_guides AS pg ON (g.GuideAadharNo=pg.GuideAadharNo)

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

WHERE g.fname='Sachin' and g.lname='Bhide';

--Output:



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

--Relational Algebra:

r1 -> ρ(h, hotel) ⋈<h.hotelid = ph.hotelid> ρ(ph, package\_includes\_hotels) ⋈<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)",

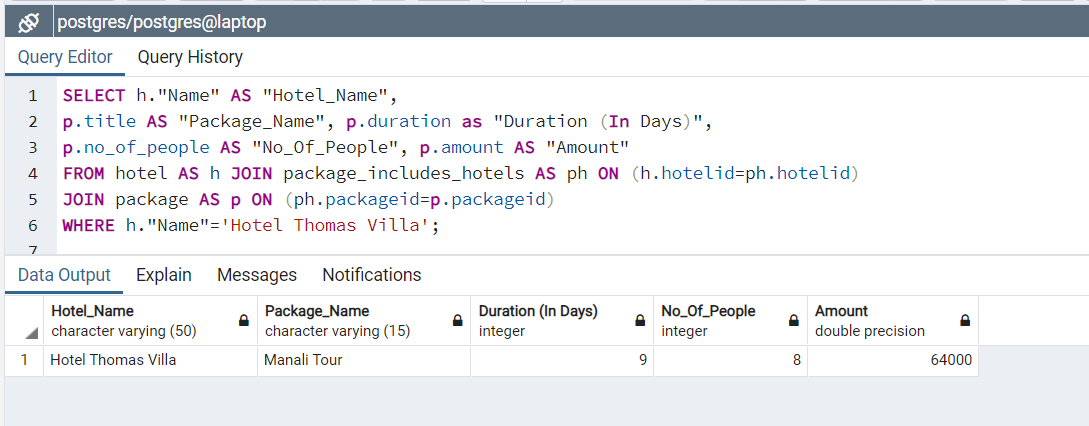
p.no\_of\_people AS "No\_Of\_People", p.amount AS "Amount"

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';

--Output:



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

--Relational Algebra:

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

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

--SQL Query:

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

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

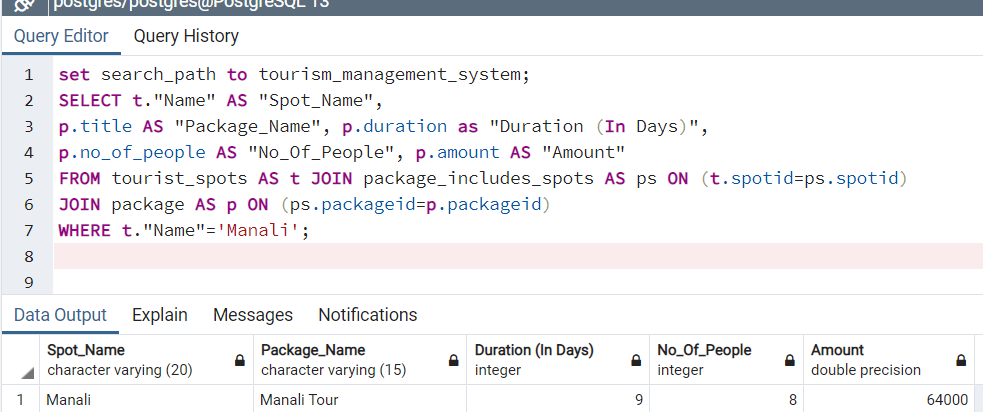
p.no\_of\_people AS "No\_Of\_People", p.amount AS "Amount"

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';

--Output:



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

--Relational Algebra:

r1 -> ρ(ts, tourist\_spots) ⋈<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",

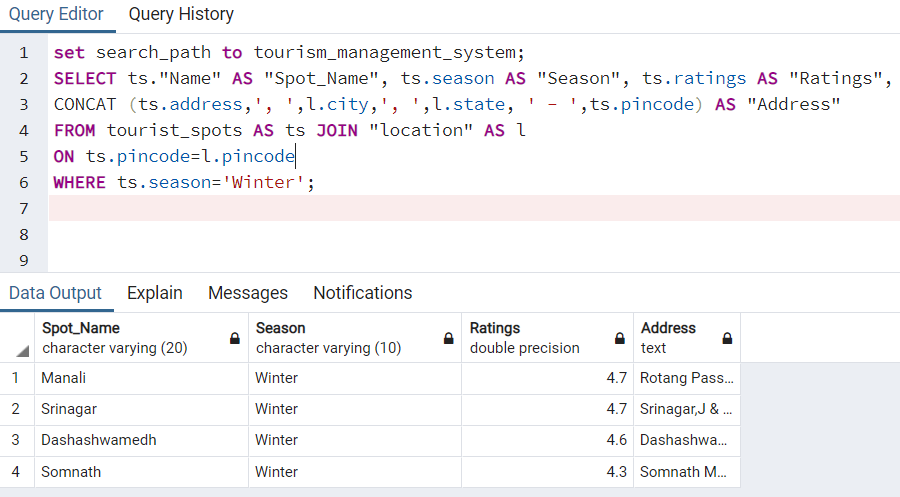
CONCAT (ts.address,', ',l.city,', ',l.state, ' - ',ts.pincode) AS "Address"

FROM tourist\_spots AS ts JOIN "location" AS l

ON ts.pincode=l.pincode

WHERE ts.season='Winter';

--Output:



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

--Relational Algebra:

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

LEFT ⋈< h.hotelid=rf.hotelid and r.room\_no=rf.room\_no > 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)

(σ(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

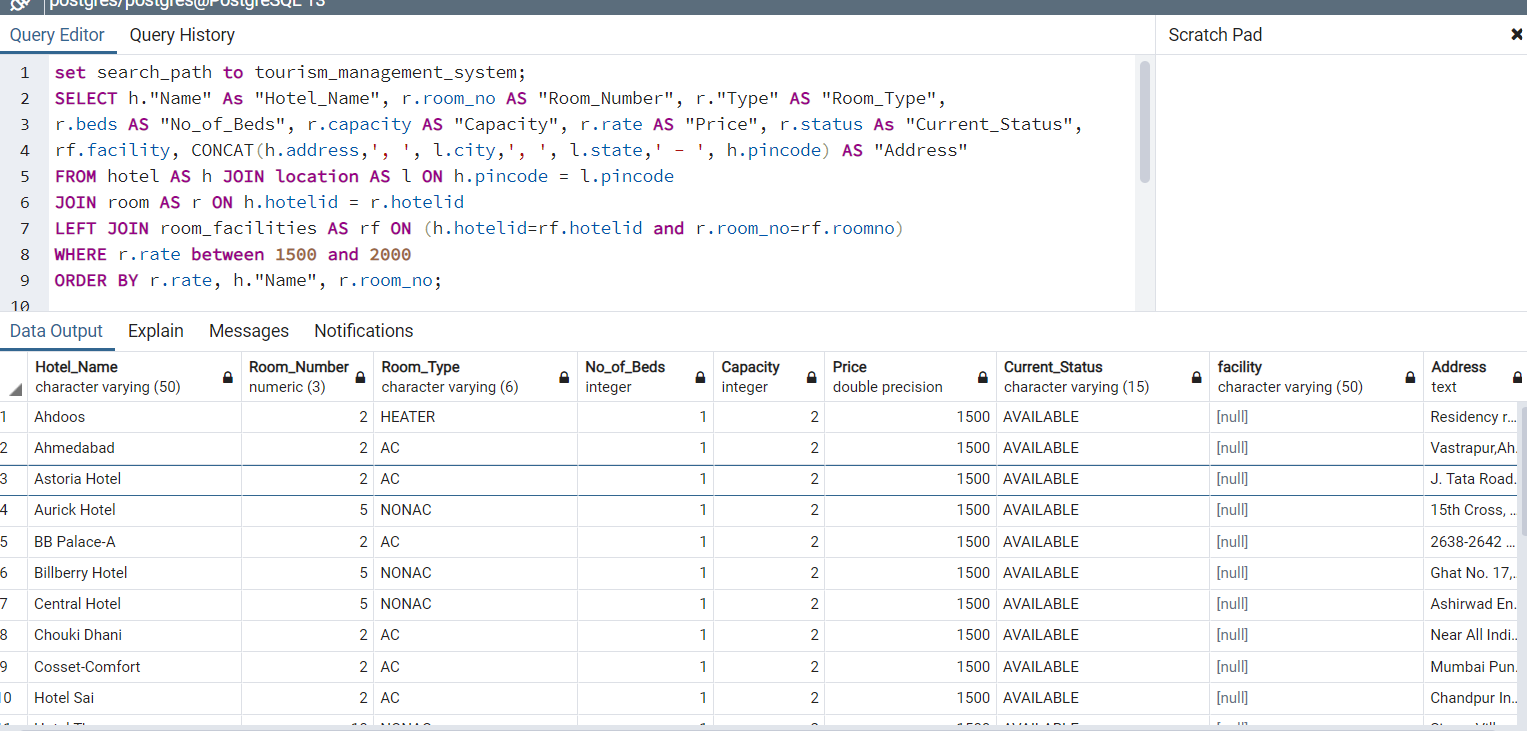
JOIN room AS r ON h.hotelid = r.hotelid

LEFT JOIN room\_facilities AS rf ON (h.hotelid=rf.hotelid and r.room\_no=rf.room\_no)

WHERE r.rate between 1500 and 2000

ORDER BY r.rate, h."Name", r.room\_no;

--Output:



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

--Relational Algebra:

r1-> Π(pg.GuideAadharNo) (σ (p.isActive=’TRUE’) (ρ(pg, package\_include\_guide) ⋈<pg.package.id=p.packageid> ρ(p,package)))

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

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

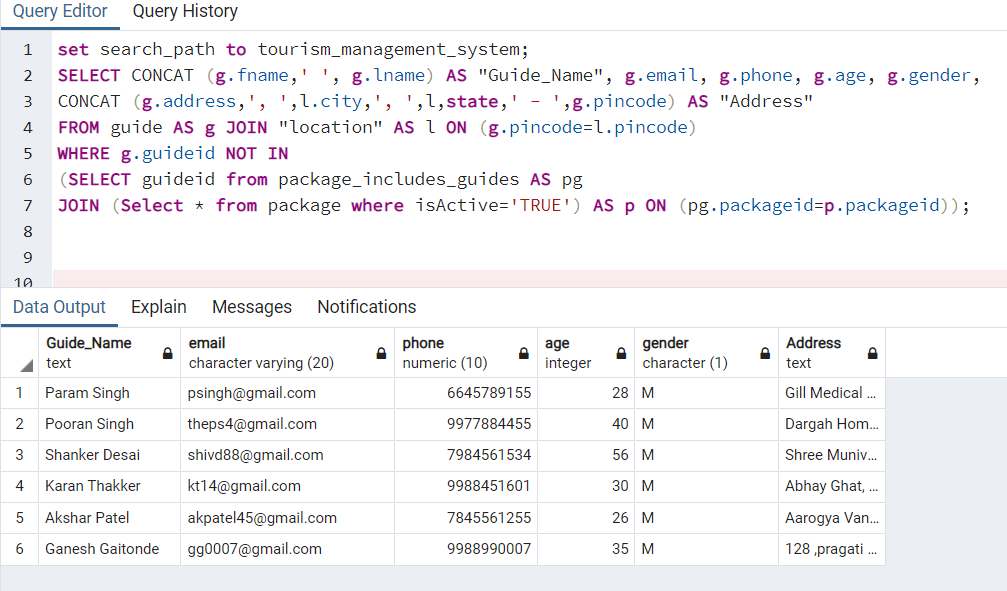
FROM guide AS g JOIN "location" AS l ON (g.pincode=l.pincode)

WHERE g.GuideAadharNo NOT IN

(SELECT GuideAadharNo from package\_includes\_guides AS pg

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

--Output:



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

--Relational Algebra:

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

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

--SQL Query:

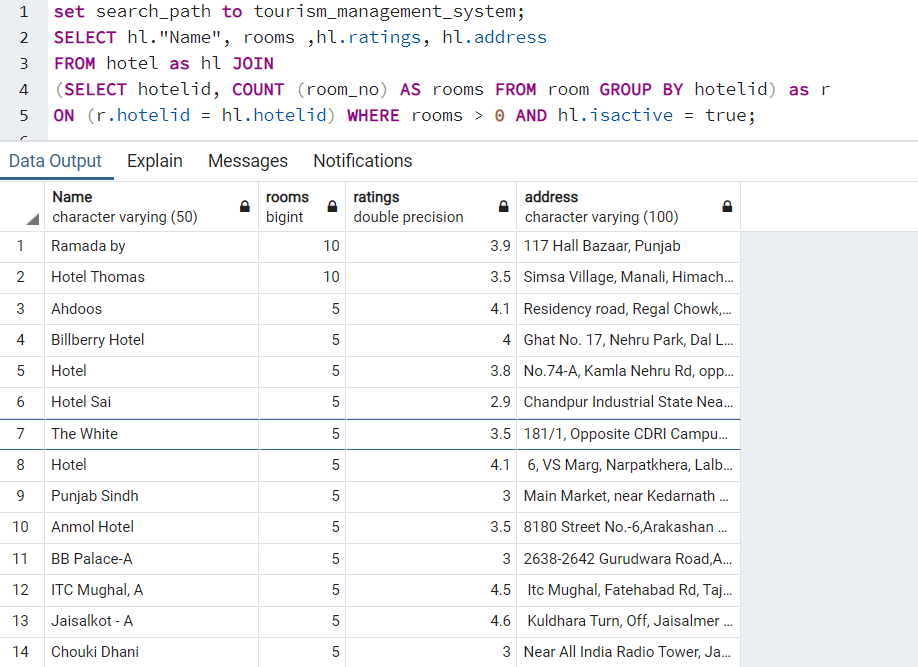
SELECT hl."Name", rooms ,hl.ratings, hl.address

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;

--Output:



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

--Relational Algebra:

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

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

--SQL Query:

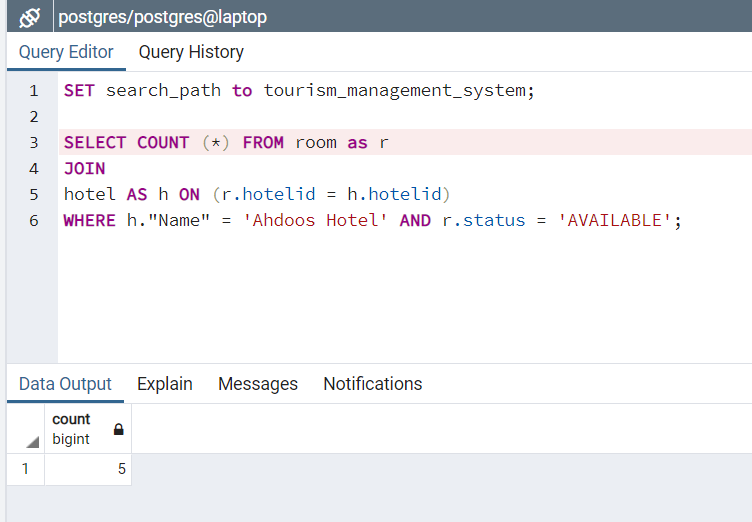
SELECT COUNT (\*) FROM room as r

JOIN

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

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

--Output:



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

--Relational Algebra:

r1 -> ρ(b, booking) LEFT ⋈<b.bid = bfc.bid> ρ(bfc, bid ℱCOUNT(CoPassID) -> 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.UserAadharNo = u.UserAadharNo> 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.room\_no -> package\_roomno, h.”Name” -> hotel\_name, bfh.room\_no) (σ 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.room\_no package\_roomno,h."Name" AS hotel\_name, bfh.room\_no FROM booking as b

LEFT JOIN

(SELECT COUNT(c.CoPassID) 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)

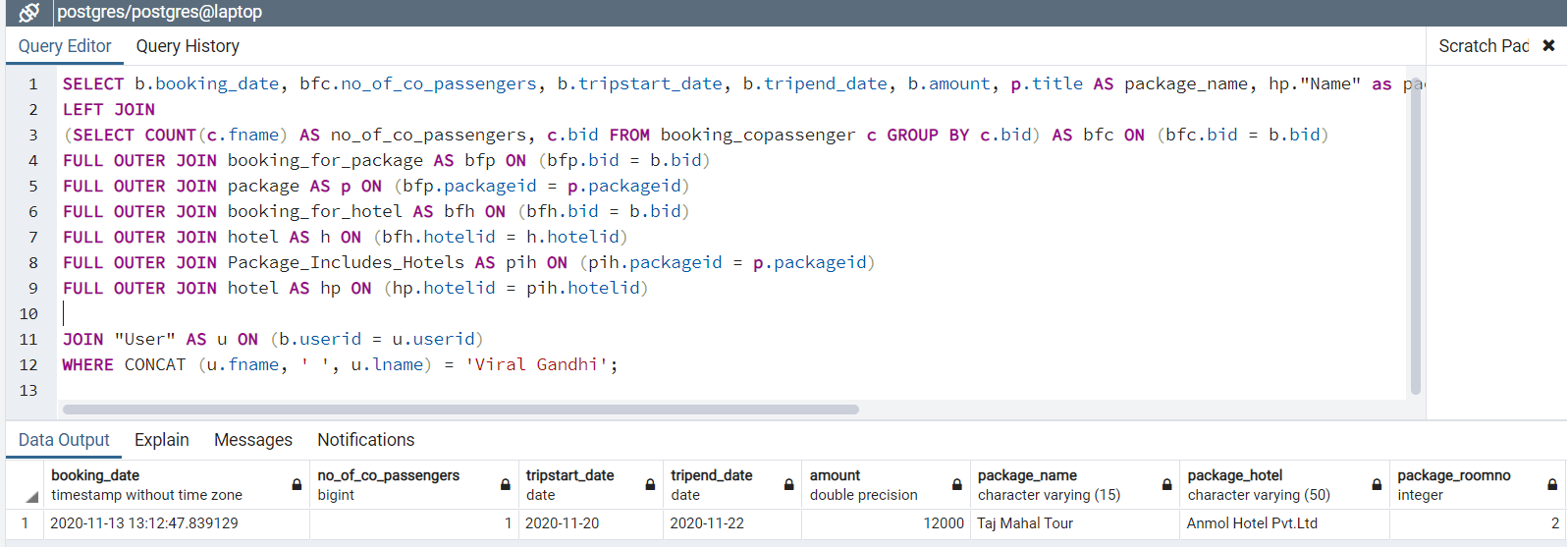
FULL OUTER JOIN Package\_Includes\_Hotels AS pih ON (pih.packageid = p.packageid)

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

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

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

--Output:



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

--Relational Algebra:

r1 -> ρ(b, booking) LEFT ⋈<b.bid = bfc.bid> ρ(bfc, bid ℱCOUNT(CoPassID) -> 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.UserAadharNo = u.UserAadharNo> 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 < '2020-11-30' AND b.booking\_date >= '2020-11-20' (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.room\_no package\_roomno, h."Name" AS hotel\_name, bfh.room\_no

FROM booking as b

LEFT JOIN

(SELECT COUNT(c.CoPassID) 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.UserAadharNo = u.UserAadharNo) WHERE CONCAT(u.fname, ' ', u.lname) = 'Viral Gandhi'

AND b.booking\_date < '2020-11-30' AND b.booking\_date >= '2020-11-20';

--Output:



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

--Relational Algebra:

r1 -> ρ(b, booking) LEFT ⋈<b.bid = bfc.bid> ρ(bfc, bid ℱCOUNT(CoPassID) -> 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.UserAadharNo = u.UserAadharNo> 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.room\_no -> package\_roomno, h.”Name” -> hotel\_name, bfh.room\_no) (σb.booking\_date < '2020-11-28' AND b.booking\_date >= '2020-11-26' (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.room\_no package\_roomno, h."Name" AS hotel\_name, bfh.room\_no

FROM booking as b

LEFT JOIN

(SELECT COUNT(c.CoPassID) 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)

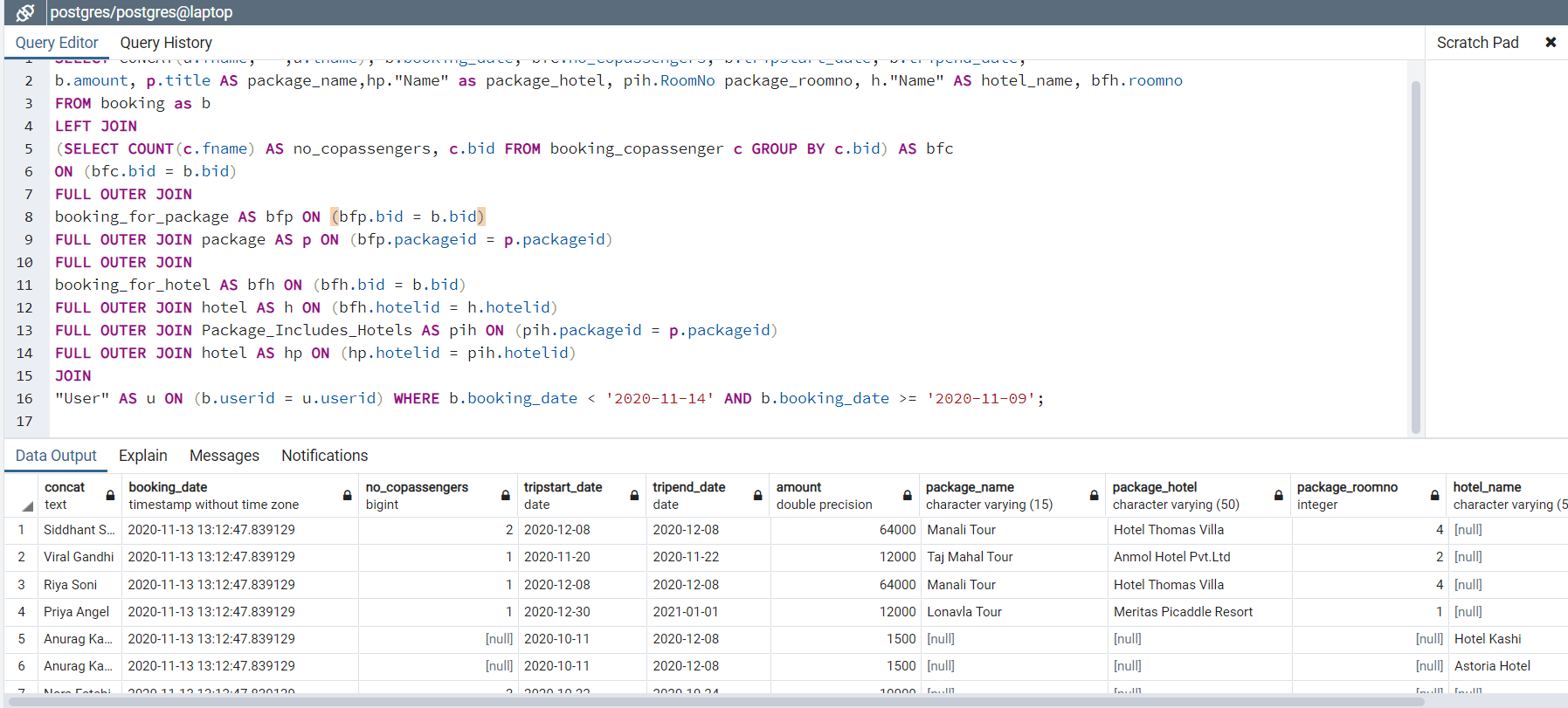
FULL OUTER JOIN Package\_Includes\_Hotels AS pih ON (pih.packageid = p.packageid)

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

JOIN

"User" AS u ON (b.UserAadharNo = u.UserAadharNo) WHERE b.booking\_date < '2020-11-28' AND b.booking\_date >= '2020-11-26';

--Output:



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

--Relational Algebra:

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

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

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

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

r5-> ρ(u,User)⋈< u.UserAadharNo = r4.UserAadharNo > (r4)

result-> Π(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.UserAadharNo = b.UserAadharNo)

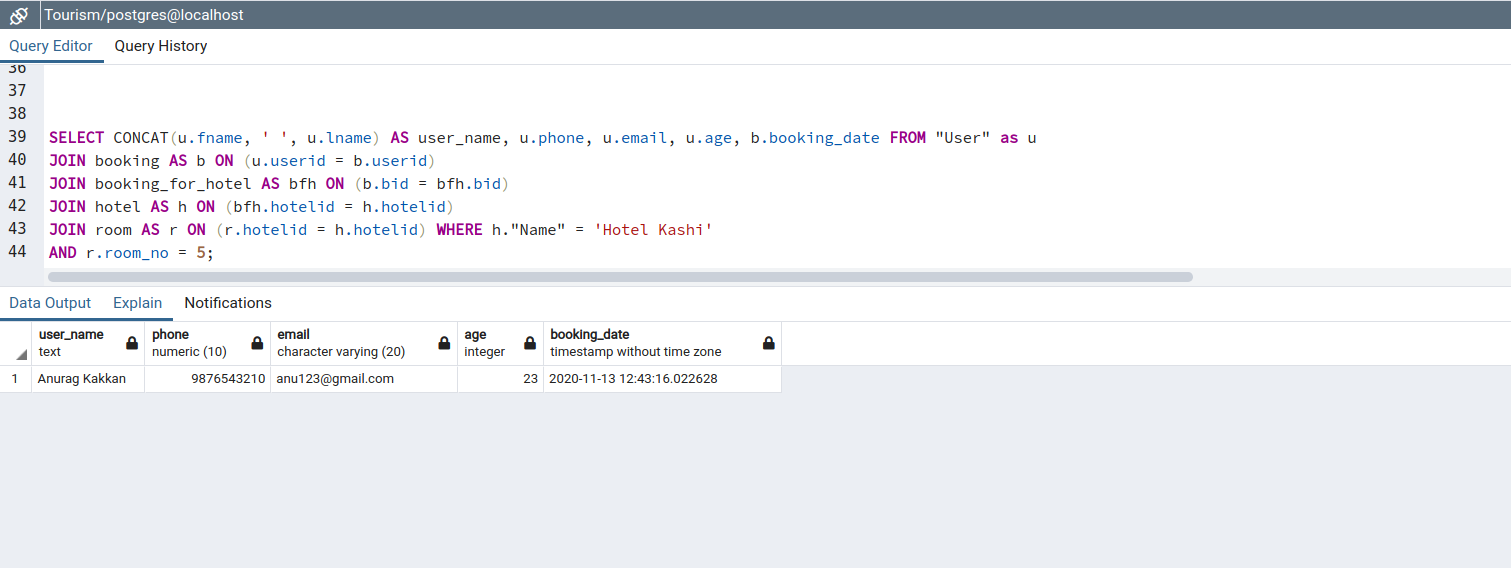
JOIN booking\_for\_hotel AS bfh ON (b.bid = bfh.bid)

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;

--Output:



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

--Relational Algebra:

r1-> ρ(bfh, booking\_for\_hotel) ⋈< bfh.hotelid = h.hotelid >ρ(h,hotel)

r2-> σ h."Name" = 'Hotel Kashi'(r1)

r3-> ρ(b,booking) ⋈<b.bid=r2.bid>(r2)

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

Π(CONCAT(u.fname, ' ', u.lname)->user\_name, b.booking\_date, bfh.room\_no)(r4)

--SQL Query:

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

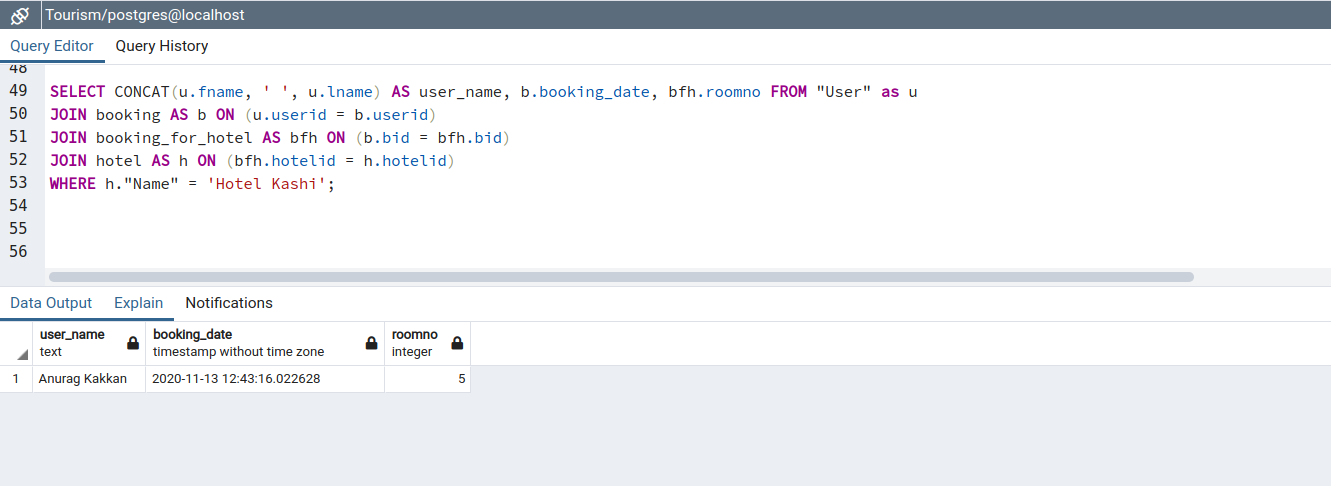
JOIN booking AS b ON (u.UserAadharNo = b.UserAadharNo)

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';

--Output:



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

--Relational Algebra:

r1->ρ(bc, booking\_copassenger) ⋈< bc.UserAadharNo = c.UserAadharNo AND bc.CoPassID = c.CoPassID> 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.UserAadharNo=r3.UserAadharNo>(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.UserAadharNo = b.UserAadharNo)

JOIN booking\_copassenger AS bc ON (bc.bid = b.bid)

JOIN copassenger AS c ON (bc.UserAadharNo = c.UserAadharNo AND bc.CoPassID = c.CoPassID)

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

--Output:

