--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;

--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);

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

Relational Algebra:

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

result -> $\Pi_{(\text{“Name”, season, ratings, CONCAT (ts.address,', ',l.city,', ',l.state, ' - ',ts.pincode) -> Address})}$ ($\sigma_{\text{l.city=”kullu”}}$(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 -> 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 -> $\Pi_{(\text{title, duration, no_of_people, amount, “Name”, rating, CONCAT (ts.address,', ',l.city,', ',l.state, ' - ',ts.pincode) -> Address})}$ ($\sigma_{\text{title=”Manali Tour”}}$(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 -> $\mathcal{F}_{MAX(ratings)->ratings}(\rho(ts2, tourist\_spots))$

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

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);
```

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

Relational Algebra:

result ->

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

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 ->

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

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 -> $\rho(l, \text{location}) \bowtie_{<l.pincode = r.pincode>} \rho(r, \text{restaurant}) \bowtie_{<r.rid = rc.rid>} \rho(rc, \text{restaurant\_cuisines})$

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';

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

Relational Algebra:

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

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

Relational Algebra:

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

--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)
```

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

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.

```
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.

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

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

--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'
```

--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 -> 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'
```

--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';
```

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

```
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.

```
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))
```