**--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;

**--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);

**--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, address, ts.pincode) (σ l.city=”kullu”(r1))

SQL Query:

SELECT ts."Name", ts.season, ts.ratings, ts.address, ts.pincode, l.city, l.state 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) ⋈<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, address, ts.pincode, city, state) (σ title=”Manali Tour”(r1))

SQL Query:

SELECT p.title, p.duration, p.no\_of\_people, p.amount, ts."Name", ts.season, ts.ratings, ts.address, ts.pincode, l.city,l.state 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 (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, address, ts1.pincode, city, state

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 ->

Π( “Name”, phone, foodtype, rating, address, r.pincode, city, state) (σ foodtype=”Veg”( ρ(r, restaurant) ⋈<r.pincode = l.pincode> ρ(l, location)))

SQL Query:

SELECT r."Name", r.phone, r.foodType, r.ratings, r.address, r.pincode, l.city,l.state

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 ->

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

SQL Query:

SELECT r."Name", r.phone, r.foodType, r.ratings, r.address, r.pincode, l.city,l.state

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 -> 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, address, r.pincode, city, state) (σ cuisines=”Chinese”(r1))

SQL Query:

SELECT r."Name", r.phone, r.foodType, r.ratings, rc.cuisines, r.address, r.pincode, l.city,l.state

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

SQL Query:

SELECT h."Name", h.phone, h.foodType, h.ratings, h.address, h.pincode, l.city,l.state

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

result -> Π( “Name”, phone, foodtype, ratings, services, address, h.pincode, city, state) (σ servies=”Gym”(r1))

SQL Query:

SELECT h."Name", h.phone, h.foodType, h.ratings,hs.services, h.address, h.pincode, l.city,l.state

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';