**Queries:**

1. A user wants to go to X state for vacation then how many cities can he visit?

**SELECT cityid, cityname, state FROM City WHERE State=’X’;**

2. User is searching for hotels in Abu, he want 3 rooms in a hotel in delux category. What are the options available according to his need?

**SELECT \* FROM roomtype AS r NATURAL JOIN(SELECT h. FROM Hotels AS h NATURAL JOIN City AS c WHERE c.cityname=’Abu’) as T WHERE r.noofroomavailable > 2 AND r.typeofroom = ’duplex’;**

3. Retrieve all the popular hill station spots for summer. {Order by city}

**SELECT p.pname, c.cityname, c.state FROM Places AS p NATURAL JOIN City AS c where p.category='HS' AND p.besttimetovisit = 'summer' AND p.rating > 3.5 ORDER BY c.cityname;**

4. Avg cost of restaurant in a particular city.

**SELECT c.cityname, avg(r.avgcostperperson) FROM City AS c NATURAL JOIN**

**restaurants as r where c.cityname='Mumbai' GROUP BY c.cityname;**

5. Retrieve city wise total count of number of places of category “Temple” in all cities.

**SELECT c.cityname, count(p.pname) FROM Places AS p NATURAL JOIN City AS c where**

**p.category='Temple' GROUP BY c.cityname;**

6. List the number of distinct user views of this site who searched for city Jaipur.

**SELECT u.username FROM "User" AS u NATURAL JOIN Viewhistory AS v**

**NATURAL JOIN City AS c where c.cityname='Jaipur';**

7. List promocodes available to user Paxon Heasly.

**SELECT p.promoid, p.validity, p.discount FROM "User" AS u NATURAL JOIN**

**Promocodeavailabletouser AS pav NATURAL JOIN promocode AS p where**

**u.username='Paxon Heasly';**

8. List all users whose promocode is valid only for a week.

**SELECT u.loginid, u.username FROM "User" AS u NATURAL JOIN**

**Promocodeavailabletouser AS pav NATURAL JOIN promocode AS p where**

**(current\_date-p.validity) <8;**

9. User is from ajmer and he wants the list of all the cities where he can visit in the diatance less than 200kms.

**SELECT c.cityname, ic.cityid2,ic.distance FROM City AS c JOIN intercitytransportation AS ic ON ic.cityid2=C.cityid where**

**ic.cityid1=(SELECT c.cityid FROM City AS c where c.cityname='Ajmer' )and ic.distance< 200;**

10.User lives in city Ahmedabad and user have vacation for 7 days ,user want to visit Himachal Pradesh.Display option available to user with expenses on transportation and hotels.

**SELECT state\_expense(Himachal Pradesh,Ahmedabad,7);**

11. User has an overall budget of 25000 for transportation and hotels. What are the options available to the user for 4 days trip. The user starts his trip from Ahmedabad.

**SELECT Budgeting(Ahmedabad,25000,4);**

12. Total Transaction per date.

**SELECT BookingDate, sum((r.CostPerDay \* b.RoomsBooked )- ((r.CostPerDay \* b.RoomsBooked )\*p.Discount)/100) FROM promocode AS p NATURAL JOIN booking AS b NATURAL JOIN roomtype AS r GROUP BY BookingDate;**

13. List the discount provided to all the users per date.

**select LoginID, (b.cost \* b.noofroom\*discount/100) as discount from (booking natural join roomtype) as b natural join promocode as p group by LoginID, BookingDate**

**14.** Avg number of users that book from our site after searching for cities.

**SELECT (count(b.loginid)\*100)/(count(v.loginid)) AS avgbooking FROM Booking AS b NATURAL JOIN Viewhistory AS v;**

**15.** Get count of users booked today.

**SELECT b.loginid, u.usename FROM Booking AS b NATURAL JOIN User AS u WHERE b.bookindate=currentdate.**

16**.** Total transaction amount for all cities.

**SELECT Transition();**

17. How many rooms are available on a particular date for a particular hotel?

**SELECT BookingDate, HotelID, h.RoomsAvailable-count(b.RoomsBooked) FROM Booking as b NATURAL JOIN Hotel as h GROUP BY BookingDate, HotelID;**

**TRIGGER FUNCTIONS**

1. Trigger to delete promoID from PromoCodeAvailableToUser when that promoID is

used in booking.

**SET SEARCH\_PATH TO etourism;**

**CREATE TRIGGER booking\_trigger AFTER INSERT OR UPDATE OR DELETE**

**ON etourism.booking FOR EACH ROW**

**EXECUTE PROCEDURE process\_booking\_trigger();**

**CREATE FUNCTION process\_booking\_trigger()**

**RETURNS TRIGGER AS $booking\_trigger$**

**BEGIN**

**IF (TG\_OP = 'INSERT') THEN**

**IF (booking.promoid IS NOT NULL OR booking.promoid IS NOT Not\_available) THEN**

**DELETE FROM promocodeavailabletouser**

**WHERE booking.promoid = promocodeavailabletouser.promoid AND booking.loginid = promocodeavailabletouser.loginid;**

**END IF;**

**END;**

**$booking\_trigger$ LANGUAGE plpgsql;**

2. Trigger to update the view history of the user.

**CREATE TRIGGER trigger\_on\_hasViewed INSTEAD OF INSERT ON hasViewed**

**FOR EACH ROW EXECUTE PROCEDURE triggerfunc\_on\_hasViewed();**

**CREATE FUNCTION triggerfunc\_on\_hasViewed() RETURNS trigger AS $$**

**BEGIN**

**DECLARE boolean flag = false;**

**DECLARE record hasViewed%rowtype;**

**IF (TG\_OP = 'INSERT') THEN**

**newloop : LOOP**

**FOR record IN SELECT \* FROM hasViewed where loginID IS NOT NULL and loginID = new.loginID**

**IF(cityID = new.cityID) THEN**

**UPDATE hasViewed**

**SET hasViewed.count = hasViewed.count + 1;**

**flag = true;**

**END IF;**

**END LOOP newloop;**

**IF (flag = false) THEN**

**INSERT INTO hasViewed VALUES (new.loginID, new.cityID);**

**END IF;**

**END IF;**

**END;**

**$$ LANGUAGE plpgsql**

**STORED PROCEDURES**

1. Stored Procedure to show the best city, hotel and transportation available according to user’s budget.

**CREATE TYPE t\_type AS (cityid varchar(8), hotelname varchar(8), roomtype**

**varchar(10), transport varchar(8), amount integer);**

**CREATE FUNCTION Budgeting(user\_city varchar(15), budget integer, days integer) RETURNS SETOF t\_type AS $BODY$**

**DECLARE**

**sum\_late time;**

**days int4;**

**Cityname varchar(15);**

**intercity intercitytransportation%rowtype;**

**hotel hotels%rowtype;**

**room roomtype%rowtype;**

**buscost integer;**

**traincost integer;**

**flightcost integer;**

**ctid City.cityid%type;**

**counter integer;**

**rec t\_type;**

**total integer[];**

**BEGIN**

**select c.cityid as ctid from City AS c where c.cityname = user\_city;**

**for intercity in select \* from intercitytransportation AS i where i.cityid1 = ctid LOOP**

**buscost:=intercity.avgcostofbus;**

**traincost:=intercity.avgcostoftrain;**

**flightcost:=intercity.avgcostofflight;**

**for hotel in select \* from hotels AS h where h.cityid = intercity.cityid2 LOOP**

**for room in select \* from roomtype AS r where r.hotelid=hotel.hotelid LOOP**

**if(buscost!=0)then**

**total[0]:=buscost+days\*costperday;**

**Else**

**total[0]:=0;**

**End if;**

**if(traincost!=0)then**

**total[1]:=traincost+days\*costperday;**

**Else**

**total[1]:=0;**

**End if;**

**if(flightcost!=0)then**

**total[2]:=flightcost+days\*costperday;**

**Else**

**total[2]:=0;**

**End if;**

**LOOP**

**EXIT when counter>2;**

**IF (total[counter] < budget)**

**Then**

**rec.cityid = intercity.cityid2;**

**rec.hotelname = hotel.hotelname;**

**rec.roomtype = room.typename;**

**IF ((counter=0) and (total[0]>0)) THEN**

**rec.transport:=’Bus’;**

**rec.amount:=total[0];**

**ELSIF ((counter=1) and (total[1]>0))THEN**

**rec.transport:=’Train’;**

**rec.amount:=total[1];**

**ELSIF ((counter=3) and (total[2]>0)) THEN**

**rec.transport:=’Flight’;**

**rec.amount:=total[2];**

**END IF;**

**counter := counter+1;**

**END IF;**

**RETURN NEXT rec;**

**END LOOP;**

**END LOOP;**

**END LOOP;**

**END LOOP;**

**RETURN;**

**END; $BODY$ LANGUAGE'plpgsql';**

2. Give the expense of each city of a particular state from the user’s city.

**CREATE TYPE t\_type AS (cityid varachar(8) ,hotelname varchar(8),roomtype**

**varchar(10),transport varchar(8) amount integer);**

**CREATE FUNCTION state\_expense(statename varchar(20), user\_city varchar(20),days integer)**

**RETURNS SETOF t\_type AS**

**$BODY$**

**DECLARE**

**sum\_late time;**

**days integer;**

**My\_cityid integer;**

**Cityname varchar(15);**

**intercity intercitytransportation%rowtype;**

**hotel hotels%rowtype;**

**room roomtype%rowtype;**

**buscost integer;**

**traincost integer;**

**flightcost integer;**

**ctid City.cityid%type;**

**counter integer;**

**rec t\_type;**

**total integer[];**

**BEGIN**

**select c.cityid as my\_cityid from City AS c where c.cityname = user\_city;**

**for ctid in select c.cityid from City AS c WHERE c.state=statename LOOP**

**for intercity in select \* from intercitytransportation AS i**

**WHERE i.cityid2=ctid and i.cityid1=my\_cityid LOOP**

**buscost:=intercity.avgcostofbus;**

**traincost:=intercity.avgcostoftrain;**

**flightcost:=intercity.avgcostofflight;**

**for hotel in select \* from hotels AS h WHERE**

**h.cityid=intercity.cityid2 LOOP**

**for room in select\* from roomtype AS r WHERE**

**r.hotelid=hotel.hotellid LOOP**

**if(buscost!=0)then**

**total[0]:=buscost+days\*costperday;**

**Else**

**total[0]:=0;**

**End if;**

**if(traincost!=0)then**

**total[1]:=traincost+days\*costperday;**

**Else**

**total[1]:=0;**

**End if;**

**if(flightcost!=0)then**

**total[2]:flightcost+days\*costperday;**

**Else**

**total[2]:=0;**

**End if;**

**LOOP**

**EXIT when counter>2;**

**rec.cityid=intercity.cityid2;**

**rec.hotelname=hotel.hotelname;**

**rec.roomtype=room.typename;**

**IF ((counter=0 ) and (total[0]>0)) THEN**

**rec.transport:=’bus’;**

**rec.amount:=total[0];**

**ELSIF ((counter=1) and (total[1]>0)) THEN**

**rec.transport:=’train’;**

**rec.amount:=total[1];**

**ELSIF ((counter=2) and (total[2]>0)) then**

**rec.transport:=’flight’;**

**rec.amount:=total[2];**

**END IF;**

**counter := counter+1;**

**RETURN NEXT rec;**

**END LOOP;**

**END LOOP;**

**END LOOP;**

**END LOOP;**

**END LOOP;**

**RETURN:**

**END$BODY$ LANGUAGE=’plpgsql’;**

3. Stored procedure to find nearby hotel or restaurant from a given address.

**SET SEARCH\_PATH TO etourism;**

**CREATE FUNCTION nearby\_places(block varchar(15), street varchar(15), address varchar(15), city varchar(8) ) RETURNS SETOF places AS $BODY$**

**DECLARE**

**p places%rowtype;**

**BEGIN**

**for p in select \* from places WHERE cityID=city and block=p.block and street=p.street and address=p.address loop**

**RETURN NEXT p;**

**END LOOP;**

**for p in select \* from places WHERE cityID=city and block!=p.block and street=p.street and address=p.address loop**

**RETURN NEXT p;**

**END LOOP;**

**for p in select \* from places WHERE cityID=city and block!=p.block and street!=p.street and address=p.address loop**

**RETURN NEXT p;**

**END LOOP;**

**RETURN;**

**END;**

**$BODY$LANGUAGE 'plpgsql';**

4. Give the transaction details done from our site for a particular hotel.

**SET SEARCH\_PATH TO etourism;**

**CREATE TYPE l\_type AS (hotelid varchar(8),hotelname varchar(30),amount integer);**

**CREATE FUNCTION transactions\_hotels() RETURNS SETOF l\_type AS $BODY$**

**DECLARE**

**rec l\_type;**

**c integer;**

**bk booking%rowtype;**

**ht hotels%rowtype;**

**Rt roomtype%rowtype;**

**hotel\_amount integer[];**

**BEGIN**

**c:=0;**

**for ht in select\* from hotels LOOP**

**rec.hotelid:=ht.hotelid;**

**rec.hotelname:=ht.hotelname;**

**for bk in select \* from booking LOOP**

**if(bk.hotelid=ht.hotelid) then**

**for rt in select \* from roomtype LOOP**

**if (rt.hotelid=ht.hotelid and**

**bk.typename=rt.typename) then**

**hotel\_amount[bk.hotelid-2001]:=hotel\_amount[bk.hotelid-2001]+rt.costperday\*(checkoutdate-checkindate);**

**End if;**

**Return next rec;**

**End LOOP;**

**End if;**

**End LOOP;**

**rec.amount:=hotel\_amount[ht.hotelid-2001];**

**RETURN NEXT rec;**

**End LOOP;**

**END;**

**$BODY$ LANGUAGE 'plpgsql';**

5. Give the transaction details of all the hotels in a particular city.

**CREATE TYPE m\_type AS (cityid varchar(8),cityname varchar(20),sum integer);**

**CREATE FUNCTION transaction() RETURNS SETOF m\_type AS $BODY$**

**DECLARE**

**book booking%rowtype;**

**c city%rowtype;**

**room roomtype%rowtype;**

**dis promocode%rowtype;**

**rec m\_type;**

**sum integer;**

**BEGIN**

**for c in select \* from city loop**

**sum:=0;**

**rec.cityID=cityID;**

**rec.cityname=cityname;**

**for book in select \* from booking where cityID=c.cityID loop**

**select \* into room from roomtype where hotelID=book.hotelID and typename=book.typename;**

**select \* into dis from promocode where promoID=book.promoID;**

**sum := sum + room.costperday \* book.noofroomsbooked -room.costperday\*book.noofroomsbooked\* dis.discount/100;**

**END LOOP;**

**rec.sum=sum;**

**RETURN NEXT rec;**

**END LOOP;**

**RETURN;**

**END; $BODY$ LANGUAGE'plpgsql';**