**QUERIES:**

JDBC/Subsequent SQL queries

As our application is java based web application we have included the JDBC SQL and subsequent SQL. The main advantage of using JDBC is that our queries could be dynamic and live.

Simple queries

1)SELECT PASSWORD FROM USERS where email\_id= '"+username+"'";

SELECT PASSWORD FROM USERS where email\_id= 'abc@gmail.com';

2)select city from region order by city;

3)INSERT INTO USERS VALUES(USERS\_ID\_SEQ.nextval,?,?,?,?,?);

INSERT INTO USERS VALUES(USERS\_ID\_SEQ.nextval,'Sherlock','51bakerstreet','sherlock@gmail.com','9999999999','IAMSHERLOCKED');

4)UPDATE users SET password=? WHERE email\_id= ? ;

UPDATE users SET password='abc' WHERE email\_id= 'sherlock@gmail.com' ;

5)select user\_id from users where email\_id='"+username+"'";

select user\_id from users where email\_id='sherlock@gmail.com';

6)

insert into DEPENDENT\_DETAILS values (userId, name, relationship, sex, age);

7)select name from users where user\_id='"+user\_ID+"';

select name from users where user\_id='007';

Complex queries

1)select service\_name, price\_per\_mile from mode\_air where mode\_id in (select mode\_id from mode\_of\_transport where mode\_type = 'Air');

2)select hotel.hotel\_name,hotel.price\_per\_night,hotel.address\_line1,hotel.user\_ratings from hotel hotel inner join region region on hotel.region\_id=region.region\_id where region.city='"+itobj.getDEPARTURE\_CITY()+"'order by user\_ratings desc;

select hotel.hotel\_name,hotel.price\_per\_night,hotel.address\_line1,hotel.user\_ratings from hotel hotel inner join region region on hotel.region\_id=region.region\_id where region.city='Anchorage' order by user\_ratings desc;

3)select service\_name,price\_per\_mile from mode\_train where mode\_id in (select mode\_id from mode\_of\_transport where mode\_type LIKE 'Train');

4)select service\_name,price\_per\_mile from mode\_Road where mode\_id in (select mode\_id from mode\_of\_transport where mode\_type LIKE 'Road');

5) We have calculated the price of travel (air,road,rail) by using the PROCEDURE RATE\_CALCULATION which in turn uses the users departure and arrival destination i.e the latitude and longitude of the city and calculates the price

create or replace PROCEDURE RATE\_CALCULATION

(

  FROM\_CITY IN VARCHAR2

, TO\_CITY IN VARCHAR2

, MODE\_SELECT IN VARCHAR2

) AUTHID CURRENT\_USER AS

lat1 number;

lon1 number;

lat2 number;

lon2 number;

dist number;

var0 varchar2(4000);

var1 varchar2(4000);

BEGIN

--Set values of variables

select lattitude into lat1 from region where city=from\_city;

select longitude into lon1 from region where city=from\_city;

select lattitude into lat2 from region where city=to\_city;

select longitude into lon2 from region where city=to\_city;

--Calculate distance

dist:=DISTANCE(lat1,lon1,lat2,lon2);

--Calculate price

var0:='Drop table TempTable';

EXECUTE IMMEDIATE var0;

case

when MODE\_SELECT='Air' then var1:='Create table TempTable as select \* from Mode\_Air';

when MODE\_SELECT='Rail' then var1:='Create table TempTable as select \* from Mode\_Train';

when MODE\_SELECT='Bus' then var1:='Create table TempTable as select \* from Mode\_Road';

end case;

EXECUTE IMMEDIATE var1;

Update TempTable set Price\_per\_mile=Price\_per\_mile\*dist;

END RATE\_CALCULATION;

Functions:

We have created a function to calculate distance between any two cities selected by user. The distance is calculated by using the latitude and longitude of cities.

create or replace FUNCTION distance (Lat1 IN NUMBER,

Lon1 IN NUMBER,

Lat2 IN NUMBER,

Lon2 IN NUMBER,

Radius IN NUMBER DEFAULT 3963) RETURN NUMBER IS

-- Convert degrees to radians

DegToRad NUMBER := 57.29577951;

BEGIN

RETURN(NVL(Radius,0) \* ACOS((sin(NVL(Lat1,0) / DegToRad) \* SIN(NVL(Lat2,0) / DegToRad)) +

(COS(NVL(Lat1,0) / DegToRad) \* COS(NVL(Lat2,0) / DegToRad) \*

COS(NVL(Lon2,0) / DegToRad - NVL(Lon1,0)/ DegToRad))));

END;