DBMS PROJECT

```
CREATE TABLE CUSTOMERS
(
email varchar(15) primary key,
name varchar(20),
phone_number number(10) ,
password varchar(15),
rating number(1) check(rating in (1,2,3,4,5))
);

CREATE TABLE RIDER
(
email varchar(15) primary key,
no_of_seats number(2),
foreign key (email) references  CUSTOMERS(email));

CREATE TABLE DRIVER
(
email varchar(15) primary key,
licence number(15) not null,
foreign key (email) references  CUSTOMERS(email)
);

insert into CUSTOMERS values('will@gmail.com','WILL',9829371345,'123QWE',5);
insert into CUSTOMERS values('jace@gmail.com','JACE',8890164345,'987ASD',5);
insert into CUSTOMERS values('clary@gmail.com','CLARY',9928090345,'34J0',4);
insert into CUSTOMERS values('mag@gmail.com','MAGNUS',9829871230,'23WER',3);
insert into CUSTOMERS values('alec@gmail.com','ALEC',9876543210,'PPL9',1);
insert into CUSTOMERS values('izzy@gmail.com','CLARY',9786453211,'23ER',2);
insert into CUSTOMERS values('bruh@gmail.com','BRUH',9879879870,'45th',2);
insert into CUSTOMERS values('hehe@gmail.com','TESS',6547987654,'asdfl',3);
insert into CUSTOMERS values('jim@gmail.com','JIM',9876543212,'ppoik',4);
insert into CUSTOMERS values('zach@gmail.com','WILL',9879879870,'23erf',1);
insert into CUSTOMERS values('jonah@gmail.com','ALEC',6545431231,'98uhi',4);



insert into RIDER values('will@gmail.com',6);
insert into RIDER values('izzy@gmail.com',2);
insert into RIDER values('mag@gmail.com',2);
insert into RIDER values('clary@gmail.com',3);


insert into DRIVER values('alec@gmail.com',98765);
insert into DRIVER values('jim@gmail.com',6756);
insert into DRIVER values('zach@gmail.com',23456);
insert into DRIVER values('jonah@gmail.com',09867);
insert into DRIVER values('bruh@gmail.com',4356);

CREATE TABLE CAR(
car_number_plate varchar(15) primary key,
seats number(2)
);

CREATE TABLE TRIP(
trip_id number(8) primary key,
car_number_plate varchar(15),
source varchar(15),
destination varchar(15),
fare number(6),
```

```
email_driver varchar(15),
availability number(2),
startTime number,
endTime number,

foreign key (email_driver) references DRIVER (email),
foreign key (car_number_plate) references CAR (car_number_plate)
);


CREATE TABLE PAYMENT(
payment_id number(9) primary key,
trip_id number(8),
email_rider varchar(15),
foreign key (trip_id) references TRIP(trip_id),
foreign key (email_rider) references RIDER(email)
);

insert into CAR values('UK-05-5867',2);
insert into CAR values('RJ-02-2092',9);
insert into CAR values('CH-99-9986',4);
insert into CAR values('HR-11-3452',2);
insert into CAR values('UP-02-1098',5);
insert into CAR values('GJ-05-2346',4);

insert into TRIP values(1002,'UK-05-5867','Udaipur','Chandigarh',6500,'jim@gmail.com',3,1445,1545);
insert into TRIP values(1003,'RJ-02-2092','Delhi','Chandigarh',2000,'alec@gmail.com',5,0245,0445);
insert into TRIP values(1004,'RJ-02-2092','Udaipur','Shimla',6000,'jim@gmail.com',6,0845,0945);
insert into TRIP values(1005,'HR-11-3452','Delhi','Darjeeling',7000,'jonah@gmail.com',8,0145,0345);
insert into TRIP values(1006,'HR-11-3452','Mussorie','Chandigarh',3000,'zach@gmail.com',3,0945,1030);

insert into PAYMENT values(9000,1002,'will@gmail.com');
insert into PAYMENT values(9001,1002,'izzy@gmail.com');
insert into PAYMENT values(9002,1003,'mag@gmail.com');
insert into PAYMENT values(9003,1003,'clary@gmail.com');
insert into PAYMENT values(9004,1004,'mag@gmail.com');
insert into PAYMENT values(9005,1004,'izzy@gmail.com');


--Procedure #1
--Login in app

CREATE OR REPLACE PROCEDURE LOGIN_APP(user IN VARCHAR2, pass IN VARCHAR2)
IS
  passAux CUSTOMERS.password%TYPE;
  incorrect_password EXCEPTION;
BEGIN
  SELECT password INTO passAux
  FROM CUSTOMERS
  WHERE email LIKE user;

  IF passAux LIKE pass THEN
    DBMS_OUTPUT.PUT_LINE('User ' || user || ' loging succesfull');
  ELSE
    RAISE incorrect_password;
  END IF;

  EXCEPTION
  WHEN no_data_found OR incorrect_password THEN
       DBMS_OUTPUT.PUT_LINE('Incorrect username or password');
END;


--SET SERVEROUTPUT ON;
DECLARE
  user CUSTOMERS.email%TYPE;
  pass CUSTOMERS.password%TYPE;
BEGIN
    user := 'izzy@gmail.com';
    pass:='23ER';
--pass := &password;
--user := &email;
  LOGIN_APP(user,pass);
END;

--Procedure with cursor #2
--Calculating price using cursor

CREATE OR REPLACE PROCEDURE price_per_availability
  AS
  CURSOR S IS SELECT availability
     FROM TRIP;
     CURSOR B IS SELECT fare
     FROM TRIP;
     AVAI TRIP.availability%TYPE;
     PRICE TRIP.fare%TYPE;
     C NUMBER;
```

```
BEGIN
    OPEN S;
    OPEN B;
    LOOP
      FETCH S INTO AVAI;
        exit WHEN S%NOTFOUND;
        dbms_output.put_line(AVAI);

        FETCH B INTO PRICE;
        exit WHEN B%NOTFOUND;
        dbms_output.put_line(PRICE);
        C:=PRICE/AVAI;
        dbms_output.put_line(C);
        END LOOP;
    --CLOSE B;
END;

Declare

Begin
price_per_availability;

End;




--Procedure #3
--Adding a new customer to the database
CREATE OR REPLACE PROCEDURE AddNewCustomer(
c_email IN VARCHAR2,
c_name IN VARCHAR2 ,
c_phone IN NUMBER,
c_pass IN CUSTOMERS.password%type ,
c_rating in number
) AS
BEGIN
  INSERT INTO CUSTOMERS VALUES (c_email,c_name,c_phone,c_pass,c_rating);
END AddNewCustomer;
/

BEGIN
    AddNewCustomer ('geeta@gmail.com','geetakasanamaam',9829371345,'qwerty',5);
END;
select * from customers;


--Trigger#1
CREATE OR REPLACE TRIGGER TripInterval
BEFORE INSERT OR UPDATE ON Trip
FOR EACH ROW

 BEGIN
 IF :NEW.StartTime > :NEW.EndTime
 THEN
      RAISE_APPLICATION_ERROR(-20001, 'Invalid trip time');
  END IF;
END;

declare
begin
insert into TRIP values(9876,'hr-05-3456','Shimla','Delhi',5600,'bruh@gmail.com',8,0245,0145);
end;

--Trigger#2
CREATE  or replace TRIGGER foreign_key_trip3
  BEFORE INSERT
  ON Trip
  FOR EACH ROW
  DECLARE
var DRIVER.email%TYPE;
BEGIN
SELECT email INTO var FROM DRIVER WHERE email=:new.email_driver;
EXCEPTION
WHEN NO_DATA_FOUND THEN
RAISE_APPLICATION_ERROR(-20016,'NOT A registered user');
END;

--Procedure with cursor#2
--Search ride
DECLARE
Cursor SearchRide(src TRIP.source%type, dest TRIP.destination%type, avail TRIP.availability%type)
IS SELECT car_number_plate,source,destination,fare,email_driver,availability
From TRIP
where source=src AND destination=dest AND availability>=avail;
REC SearchRide%ROWTYPE;
BEGIN
```

```
OPEN SearchRide('Udaipur','Chandigarh',2);
LOOP
FETCH SearchRide INTO REC;
If SearchRide%NOTFOUND THEN
dbms_output.put_line('No data found');
END IF;
EXIT WHEN SearchRide%NOTFOUND;
dbms_output.put_line('Available Rides:');
dbms_output.put_line(REC.car_number_plate);
dbms_output.put_line(REC.source);
dbms_output.put_line(REC.destination);
dbms_output.put_line(REC.fare);
dbms_output.put_line(REC.email_driver);
dbms_output.put_line(REC.availability);
END LOOP;
CLOSE SearchRide;
END SearchRide;
/*
--Trigger#3
--Password update
create or replace trigger password_update
after update of password on CUSTOMERS for each row
begin
dbms_output.put_line('old password: '||:old.password);
dbms_output.put_line('new password: '||:new.password);
end;

declare
begin
password_update
*/

--Cursor#3
--display names of top 5 rating customers
declare
custname CUSTOMERS.name%type;
custrate CUSTOMERS.rating%type;
cursor temp1 is select name, rating from CUSTOMERS order by rating desc;
begin
open temp1;
dbms_output.put_line('Name'||'    ' || 'Rating');

loop
fetch temp1 into custname, custrate;
exit when temp1%rowcount>5 or temp1%notfound;
dbms_output.put_line(custname||'    ' || custrate);
end loop;
close temp1;

--cursor#4
--cursor to display details of all the customers
declare
custemail CUSTOMERS.email%type;
custname CUSTOMERS.name%type;
custphone CUSTOMERS.phone_number%type;
custpass CUSTOMERS.password%type;
custrate CUSTOMERS.rating%type;
cursor c is select email, name, phone_number, password, rating from CUSTOMERS;
begin
open c;
dbms_output.put_line('CustomerMail'||'    '||'CustomerName'||'     '||'CustomerName'||'     '||'CustomerPasswrd'||'      '||'CustomerR

loop
fetch c into custemail,custname,custphone,custpass,custrate;
exit when c%NOTFOUND;
dbms_output.put_line(custemail||'    '||custname||'     '||custphone||'     '||custpass||'      '||custrate);
end loop;
close c;
end;

--Procedure with exception#2
CREATE OR REPLACE PROCEDURE Book(
paymentid IN PAYMENT.payment_id%type,
tripid IN TRIP.trip_id%type,
email_cust IN RIDER.email%type
) IS
BEGIN
  INSERT INTO PAYMENT VALUES (paymentid,tripid,email_cust);
  UPDATE TRIP SET availability=availability-1 WHERE trip_id=tripid;
EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
dbms_output.put_line('PAYMENT ID NOT UNIQUE');
WHEN OTHERS THEN
dbms_output.put_line('INCORRECT EMAIL OR TRIP ID');
END Book;

begin
```

```
book(12345,234,'clary@gmail.com');
end;

--Procedure with exception#3

CREATE OR REPLACE PROCEDURE PublishTrip(
tripid  Trip.trip_id%TYPE,
cnp  Trip.car_number_plate%TYPE,
sl  Trip.source%TYPE ,
dl  Trip.destination%TYPE ,
f  Trip.fare%TYPE,
ed Trip.email_driver%TYPE ,
a   Trip.availability%TYPE,
st  Trip.StartTime%TYPE,
et  Trip.EndTime%TYPE
) AS
BEGIN
  INSERT INTO trip VALUES (tripid,cnp,sl,dl,f,ed,a,st,et);
EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
dbms_output.put_line('TRIP ID NOT UNIQUE');
WHEN OTHERS THEN
dbms_output.put_line('INCORRECT  DRIVER EMAIL OR CAR NUMBER PLATE');
END PublishTrip;

begin
PublishTrip(8989,'UK-05-5867','ghaziabad','gurgaon',350,'alec@gmail.com',4,900,1200);
end;

select * from customers;
select * from rider;
select * from driver;
select * from trip;
select * from payment;
select * from car;
```

```
--Sequence of report

CREATE TABLE CUSTOMERS
(
email varchar(15) primary key,
name varchar(20),
phone_number number(10) ,
password varchar(15),
rating number(1) check(rating in (1,2,3,4,5))
);

CREATE TABLE RIDER
(
email varchar(15) primary key,
no_of_seats number(2),
foreign key (email) references  CUSTOMERS(email));

CREATE TABLE DRIVER
(
email varchar(15) primary key,
licence number(15) not null,
foreign key (email) references  CUSTOMERS(email)
);

insert into CUSTOMERS values('will@gmail.com','WILL',9829371345,'123QWE',5);
insert into CUSTOMERS values('jace@gmail.com','JACE',8890164345,'987ASD',5);
insert into CUSTOMERS values('clary@gmail.com','CLARY',9928090345,'34J0',4);
insert into CUSTOMERS values('mag@gmail.com','MAGNUS',9829871230,'23WER',3);
insert into CUSTOMERS values('alec@gmail.com','ALEC',9876543210,'PPL9',1);
insert into CUSTOMERS values('izzy@gmail.com','CLARY',9786453211,'23ER',2);
insert into CUSTOMERS values('bruh@gmail.com','BRUH',9879879870,'45th',2);
insert into CUSTOMERS values('hehe@gmail.com','TESS',6547987654,'asdfl',3);
insert into CUSTOMERS values('jim@gmail.com','JIM',9876543212,'ppoik',4);
insert into CUSTOMERS values('zach@gmail.com','WILL',9879879870,'23erf',1);
insert into CUSTOMERS values('jonah@gmail.com','ALEC',6545431231,'98uhi',4);



insert into RIDER values('will@gmail.com',6);
insert into RIDER values('izzy@gmail.com',2);
insert into RIDER values('mag@gmail.com',2);
insert into RIDER values('clary@gmail.com',3);


insert into DRIVER values('alec@gmail.com',98765);
insert into DRIVER values('jim@gmail.com',6756);
insert into DRIVER values('zach@gmail.com',23456);
```

```
insert into DRIVER values('jonah@gmail.com',09867);
insert into DRIVER values('bruh@gmail.com',4356);

CREATE TABLE CAR(
car_number_plate varchar(15) primary key,
seats number(2)
);

CREATE TABLE TRIP(
trip_id number(8) primary key,
car_number_plate varchar(15),
source varchar(15),
destination varchar(15),
fare number(6),
email_driver varchar(15),
availability number(2),
startTime number,
endTime number,

foreign key (email_driver) references DRIVER (email),
foreign key (car_number_plate) references CAR (car_number_plate)
);


CREATE TABLE PAYMENT(
payment_id number(9) primary key,
trip_id number(8),
email_rider varchar(15),
foreign key (trip_id) references TRIP(trip_id),
foreign key (email_rider) references RIDER(email)
);

insert into CAR values('UK-05-5867',2);
insert into CAR values('RJ-02-2092',9);
insert into CAR values('CH-99-9986',4);
insert into CAR values('HR-11-3452',2);
insert into CAR values('UP-02-1098',5);
insert into CAR values('GJ-05-2346',4);

insert into TRIP values(1002,'UK-05-5867','Udaipur','Chandigarh',6500,'jim@gmail.com',3,1445,1545);
insert into TRIP values(1003,'RJ-02-2092','Delhi','Chandigarh',2000,'alec@gmail.com',5,0245,0445);
insert into TRIP values(1004,'RJ-02-2092','Udaipur','Shimla',6000,'jim@gmail.com',6,0845,0945);
insert into TRIP values(1005,'HR-11-3452','Delhi','Darjeeling',7000,'jonah@gmail.com',8,0145,0345);
insert into TRIP values(1006,'HR-11-3452','Mussorie','Chandigarh',3000,'zach@gmail.com',3,0945,1030);

insert into PAYMENT values(9000,1002,'will@gmail.com');
insert into PAYMENT values(9001,1002,'izzy@gmail.com');
insert into PAYMENT values(9002,1003,'mag@gmail.com');
insert into PAYMENT values(9003,1003,'clary@gmail.com');
insert into PAYMENT values(9004,1004,'mag@gmail.com');
insert into PAYMENT values(9005,1004,'izzy@gmail.com');



select * from customers;
select * from rider;
select * from driver;
select * from trip;
select * from payment;
select * from car;

--Procedure #1 with (exception#1)
--Login in app

CREATE OR REPLACE PROCEDURE LOGIN_APP(user IN VARCHAR2, pass IN VARCHAR2)
IS
  passAux CUSTOMERS.password%TYPE;
  incorrect_password EXCEPTION;
BEGIN
  SELECT password INTO passAux
  FROM CUSTOMERS
  WHERE email LIKE user;

  IF passAux LIKE pass THEN
    DBMS_OUTPUT.PUT_LINE('User ' || user || ' loging succesfull');
  ELSE
    RAISE incorrect_password;
  END IF;

  EXCEPTION
  WHEN no_data_found OR incorrect_password THEN
       DBMS_OUTPUT.PUT_LINE('Incorrect username or password');
END;


--SET SERVEROUTPUT ON;
DECLARE
```

```
  user CUSTOMERS.email%TYPE;
  pass CUSTOMERS.password%TYPE;
BEGIN
    user := 'izzy@gmail.com';
    pass:='23ER';
--pass := &password;
--user := &email;
  LOGIN_APP(user,pass);
END;

--Procedure #2 with (cursor #1)
--Calculating price using cursor
CREATE OR REPLACE PROCEDURE price_per_availability
  AS
  CURSOR S IS SELECT availability
     FROM TRIP;
     CURSOR B IS SELECT fare
     FROM TRIP;
     AVAI TRIP.availability%TYPE;
     PRICE TRIP.fare%TYPE;
     C NUMBER;
BEGIN
    OPEN S;
    OPEN B;
    LOOP
      FETCH S INTO AVAI;
        exit WHEN S%NOTFOUND;
        dbms_output.put_line(AVAI);

        FETCH B INTO PRICE;
        exit WHEN B%NOTFOUND;
        dbms_output.put_line(PRICE);
        C:=PRICE/AVAI;
        dbms_output.put_line(C);
        END LOOP;
    --CLOSE B;
END;

Declare

Begin
price_per_availability;

End;




--Procedure #3
--Adding a new customer to the database
CREATE OR REPLACE PROCEDURE AddNewCustomer(
c_email IN VARCHAR2,
c_name IN VARCHAR2 ,
c_phone IN NUMBER,
c_pass IN CUSTOMERS.password%type ,
c_rating in number
) AS
BEGIN
  INSERT INTO CUSTOMERS VALUES (c_email,c_name,c_phone,c_pass,c_rating);
END AddNewCustomer;
/

BEGIN
    AddNewCustomer ('geeta@gmail.com','geetakasanamaam',9829371345,'qwerty',5);
END;
select * from customers;


--Procedure #4 with (cursor#2)
--Search ride
DECLARE
Cursor SearchRide(src TRIP.source%type, dest TRIP.destination%type, avail TRIP.availability%type)
IS SELECT car_number_plate,source,destination,fare,email_driver,availability
From TRIP
where source=src AND destination=dest AND availability>=avail;
REC SearchRide%ROWTYPE;
BEGIN
OPEN SearchRide('Udaipur','Chandigarh',2);
LOOP
FETCH SearchRide INTO REC;
If SearchRide%NOTFOUND THEN
dbms_output.put_line('No data found');
END IF;
EXIT WHEN SearchRide%NOTFOUND;
dbms_output.put_line('Available Rides:');
dbms_output.put_line(REC.car_number_plate);
dbms_output.put_line(REC.source);
```

```
dbms_output.put_line(REC.destination);
dbms_output.put_line(REC.fare);
dbms_output.put_line(REC.email_driver);
dbms_output.put_line(REC.availability);
END LOOP;
CLOSE SearchRide;
END SearchRide;


--Procedure#5 with (exception#2)
CREATE OR REPLACE PROCEDURE Book(
paymentid IN PAYMENT.payment_id%type,
tripid IN TRIP.trip_id%type,
email_cust IN RIDER.email%type
) IS
BEGIN
  INSERT INTO PAYMENT VALUES (paymentid,tripid,email_cust);
  UPDATE TRIP SET availability=availability-1 WHERE trip_id=tripid;
EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
dbms_output.put_line('PAYMENT ID NOT UNIQUE');
WHEN OTHERS THEN
dbms_output.put_line('INCORRECT EMAIL OR TRIP IP');
END Book;

begin
book(12345,234,'clary@gmail.com');
end;

--Procedure#6 with (exception#3)

CREATE OR REPLACE PROCEDURE PublishTrip(
tripid  Trip.trip_id%TYPE,
cnp  Trip.car_number_plate%TYPE,
sl  Trip.source%TYPE ,
dl  Trip.destination%TYPE ,
f  Trip.fare%TYPE,
ed Trip.email_driver%TYPE ,
a   Trip.availability%TYPE,
st  Trip.StartTime%TYPE,
et  Trip.EndTime%TYPE
) AS
BEGIN
  INSERT INTO trip VALUES (tripid,cnp,sl,dl,f,ed,a,st,et);
EXCEPTION
WHEN DUP_VAL_ON_INDEX THEN
dbms_output.put_line('TRIP ID NOT UNIQUE');
WHEN OTHERS THEN
dbms_output.put_line('INCORRECT  DRIVER EMAIL OR CAR NUMBER PLATE');
END PublishTrip;

begin
PublishTrip(8989,'UK-05-5867','ghaziabad','gurgaon',350,'alec@gmail.com',4,900,1200);
end;



--Trigger#1
CREATE OR REPLACE TRIGGER TripInterval
BEFORE INSERT OR UPDATE ON Trip
FOR EACH ROW

 BEGIN
 IF :NEW.StartTime > :NEW.EndTime
 THEN
      RAISE_APPLICATION_ERROR(-20001, 'Invalid trip time');
  END IF;
END;

declare
begin
insert into TRIP values(9876,'hr-05-3456','Shimla','Delhi',5600,'bruh@gmail.com',8,0245,0145);
end;

--Trigger#2
CREATE  or replace TRIGGER foreign_key_trip3
  BEFORE INSERT
  ON Trip
  FOR EACH ROW
  DECLARE
var DRIVER.email%TYPE;
BEGIN
SELECT email INTO var FROM DRIVER WHERE email=:new.email_driver;
EXCEPTION
WHEN NO_DATA_FOUND THEN
RAISE_APPLICATION_ERROR(-20016,'NOT A registered user');
END;
```

```
/*
--Trigger#3
--Password update
create or replace trigger password_update
after update of password on CUSTOMERS for each row
begin
dbms_output.put_line('old password: '||:old.password);
dbms_output.put_line('new password: '||:new.password);
end;

declare
begin
password_update
*/


--Cursor#3
--display names of top 5 rating customers
declare
custname CUSTOMERS.name%type;
custrate CUSTOMERS.rating%type;
cursor temp1 is select name, rating from CUSTOMERS order by rating desc;
begin
open temp1;
dbms_output.put_line('Name'||'    ' || 'Rating');

loop
fetch temp1 into custname, custrate;
exit when temp1%rowcount>5 or temp1%notfound;
dbms_output.put_line(custname||'    ' || custrate);
end loop;
close temp1;

--cursor#4
--cursor to display details of all the customers
declare
custemail CUSTOMERS.email%type;
custname CUSTOMERS.name%type;
custphone CUSTOMERS.phone_number%type;
custpass CUSTOMERS.password%type;
custrate CUSTOMERS.rating%type;
cursor c is select email, name, phone_number, password, rating from CUSTOMERS;
begin
open c;
dbms_output.put_line('CustomerMail'||'    '||'CustomerName'||'     '||'CustomerName'||'     '||'CustomerPasswrd'||'      '||'CustomerR

loop
fetch c into custemail,custname,custphone,custpass,custrate;
exit when c%NOTFOUND;
dbms_output.put_line(custemail||'    '||custname||'     '||custphone||'     '||custpass||'      '||custrate);
end loop;
close c;
end;
```