CAB BOOKING

PROCEDURES

APRIL 16, 2019

DBMS PL/SQL

1. Procedure for Insertion into driver table

```
SQL> create or replace procedure reg_driver(fname driver.first_name%type,lname
driver.last_name%type,bday driver.birth_date%type,lic_no driver.driving_licence_number%type,exp
driver.expiry_date%type,status driver.working%type)
   as
   begin
   insert into driver(first_name,last_name,birth_date,driving_licence_number,expiry_date,working)
values(fname,lname,bday,lic_no,exp,status);
   end;
   /
```

2. Procedure for Insertion into cab table

```
SQL> create or replace procedure reg_cab(plate cab.licence_plate%type,year
cab.manufacture_year%type,owner cab.owner_id%type,mname car_model.model_name%type) as
   begin
   insert into cab(licence_plate,manufacture_year,owner_id,model_name)
values(plate,year,owner,mname);
   end;
   /
```

3. Procedure for Insertion into owner table

```
SQL> create or replace procedure reg_owner(name owner.name%type)
   as
   begin
   insert into owner(name) values(name);
   end;
   /
```

4. Procedure for Insertion into car_model table

```
SQL>  create or replace procedure reg_car_model(name car_model.model_name%type,descr
car_model.model_description%type) as
       begin
        insert into car_model values(name,descr);
        end;
        /
```

5. Procedure for Insertion into driver_cab table

```
SQL> create or replace procedure reg_driver_cab(d_id driver_cab.driver_id%type,c_id
driver_cab.cab_id%type)
    as
    begin
    insert into driver_cab values(d_id,c_id);
    end;
    /
```

6. Procedure for Insertion into payment_type table

```
SQL> create or replace procedure reg_payment_type(pid payment_type.payment_id%type,name
payment_type.type_name%type)
    as
    begin
    insert into payment_type values(pid,name);
    end;
    /
```

7. Procedure for Insertion into cab_ride table

```
SQL> create or replace procedure reg_cab_ride(customer_id customer.cust_id%type,password
customer.pass%type,jf cab_ride.jfrom%type,jt cab_ride.jto%type)
  as
  p customer.pass%type;
  cursor c is select pass from customer where cust_id=customer_id;
  begin
  open c;
  fetch c into p;
  close c;
  if(p=password) then
          if(cab_booking(customer_id,jf,jt)=0) then
                    dbms_output.put_line('No cab is available right now.. wait for sometimes...');
          end if;
  else
          dbms_output.put_line('Wrong user credentials');
  end if;
/
```

8. Procedure for Insertion into distance_map table

```
SQL> create or replace procedure reg_distance_map(p1 distance_map.jfrom%type,p2
distance_map.jto%type,d distance_map.dis%type)
   as
   begin
   insert into distance_map values(p1,p2,d);
   insert into distance_map values(p2,p1,d);
   end;
   /
```

9. Procedure for Insertion into customer table

```
SQL> create or replace procedure reg_customer
   (
   password customer.pass%type,
   cust_name customer.cust_name%type,
   email customer.email%type,
   contact customer.contact%type,
   vpa customer.vpa%type
   )
   as
  begin
  insert into customer(pass,cust_name,email,contact,vpa)
values(cust_name,password,email,contact,vpa);
  end;
  /
```

10. Procedure for Insertion into cab_ride_history table

```
SQL> create or replace procedure reg_cab_ride_history(cbid cab_ride.cab_id%type,
  cust_id cab_ride.cust_id%type,
  st cab_ride.ride_start_time%type,
  et cab_ride.ride_end_time%type,
  jfrom cab_ride.jfrom%type,
  jto cab_ride.jto%type,
  cancelled cab_ride.cancelled%type,
  ptype cab_ride.payment_id%type,
  charges cab_ride.charges%type)
  as
  begin
  insert into cab_ride_history values(cbid,cust_id,st,et,jfrom,jto,cancelled,ptype,charges);
  end;
  /
```