**Trigger**

1. CREATE or REPLACE TRIGGER office\_time

BEFORE INSERT OR UPDATE OR DELETE ON TOLL

FOR EACH ROW

BEGIN

IF(TO\_CHAR(SYSDATE,'DAY')in('FRI','SAT') or TO\_CHAR(SYSDATE,'HH24') NOT BETWEEN '08' and '22') then

raise\_application\_error(-20501,'INVALID TIMING');

end if;

END;

2. CREATE or REPLACE TRIGGER toll\_price\_limit

BEFORE INSERT OR UPDATE ON TOLL

FOR EACH ROW

BEGIN

IF(:new.PRICE NOT BETWEEN 100 and 1000) then

raise\_application\_error(-20500,'INVALID AMOUNT');

end if;

END;

3. CREATE or REPLACE TRIGGER pay\_price\_limit

BEFORE INSERT ON TRANSACTION

FOR EACH ROW

BEGIN

IF(:new.AMOUNT < 0) then

raise\_application\_error(-20504,'INVALID AMOUNT');

end if;

END;

4. CREATE or REPLACE TRIGGER office\_time\_users

BEFORE DELETE ON users

FOR EACH ROW

BEGIN

IF(TO\_CHAR(SYSDATE,'DAY')in('FRI','SAT') or TO\_CHAR(SYSDATE,'HH24') NOT BETWEEN '08' and '17') then

raise\_application\_error(-20507,'INVALID TIMING');

end if;

END;

**VIEW**

GRANT ALL PRIVILEGES TO scott;

1. CREATE VIEW date\_wise\_transaction

AS

SELECT USERID, PLACENAME, AMOUNT, TIME, VEHICLENUMBER

FROM TRANSACTION;

**PROCEDURE**

create or replace procedure p\_getuserID(un IN USERS.USERNAME%type, ui OUT USERS.USERID%type)

AS

BEGIN

ui := getuserID(un);

END;

2. create or replace procedure p\_isadmin(un IN USERS.USERNAME%type, a OUT USERS.USERID%type)

AS

BEGIN

a := isadmin(un);

END;

3. create or replace procedure p\_getvehicleid(vn IN VEHICLE.VEHICLENUMBER%type, v OUT VEHICLE.VEHICLEID %type)

AS

BEGIN

v := getvehicleid(vn);

END;

4. create or replace procedure p\_add\_vehicle(bn IN VEHICLE.BRANDNAME%type, vm IN VEHICLE.VEHICLEMODEL%type, vn IN VEHICLE.VEHICLENUMBER%type, vc IN VEHICLE.VEHICLECLASS%type,

vt IN VEHICLE.VEHICLETYPE%type, ui IN VEHICLE.USERID%type, a OUT VEHICLE.USERID%type)

AS

BEGIN

a := add\_vehicle(bn, vm, vn, vc, vt, ui);

END;

5. create or replace procedure p\_delete\_vehicle(vn IN VEHICLE.vehiclenumber%type, a OUT VEHICLE.USERID%type)

AS

BEGIN

a := delete\_vehicle(vn);

END;

6. create or replace procedure p\_update\_password(np IN USERS.PASSWORD%type, ui IN USERS.USERNAME%type, cp IN USERS.PASSWORD%type, a OUT USERS.USERID%type)

AS

BEGIN

a := update\_password(np, ui, cp);

END;

**Function**

1. create or replace function getuserID(un USERS.USERNAME%type) return NUMBER

is

id NUMBER(11);

BEGIN

select userid into id from users where username = un;

return id;

END;

2. create or replace function isadmin(un USERS.USERNAME%type) return NUMBER

is

a NUMBER(11);

BEGIN

select admin into a from users where username = un;

return a;

END;

3. create or replace function getvehicleid(vn VEHICLE.VEHICLENUMBER%type) return NUMBER

is

v NUMBER(11);

BEGIN

select VEHICLEID into v from VEHICLE where VEHICLENUMBER = vn;

return v;

END;

4. create or replace function add\_vehicle(bn VEHICLE.BRANDNAME%type, vm VEHICLE.VEHICLEMODEL%type, vn VEHICLE.VEHICLENUMBER%type, vc VEHICLE.VEHICLECLASS%type,

vt VEHICLE.VEHICLETYPE%type, ui VEHICLE.USERID%type) return number

is

BEGIN

INSERT INTO vehicle (vehicleid, brandname, vehiclemodel, vehiclenumber, vehicleclass, vehicletype, userid) VALUES (seq\_vehicle.nextval, bn, vm, vn, vc, vt, ui);

return ui;

END;

5. create or replace function delete\_vehicle(vn VEHICLE.vehiclenumber%type) return number

is

BEGIN

DELETE FROM vehicle WHERE vehiclenumber = vn;

return 1;

END;

6. create or replace function update\_password(np USERS.PASSWORD%type, ui USERS.USERNAME%type, cp USERS.PASSWORD%type) return number

is

BEGIN

UPDATE users SET password = np WHERE username = ui AND password = cp;

return 1;

END;