**Data-Base Creation Relationship**

CREATE TABLE us\_er

(

user\_id INT,

user\_name CHAR(50),

user\_mail CHAR(50),

user\_age INT,

user\_gender CHAR(50),

user\_phone\_no INT,

registration\_id INT,

PRIMARY KEY (user\_id)

);

insert into us\_er values (101,'user1','user1@gmail.com', 20, 'male', 1111111111, 201);

insert into us\_er values (102,'user2','user2@gmail.com', 21, 'female', 1111111112, 202);

insert into us\_er values (103,'user3','user3@gmail.com', 22, 'male', 1111111113, 203);

insert into us\_er values (104,'user4','user4@gmail.com', 23, 'female', 1111111114, 204);

insert into us\_er values (105,'user5','user5@gmail.com', 24, 'male', 1111111115, 205);

select \* from us\_er;

CREATE TABLE waste

(

waste\_type CHAR(50),

waste\_weight INT,

waste\_unit CHAR(50),

waste\_id INT,

PRIMARY KEY (waste\_id)

);

insert into waste values ('type1', 1,'kg', 301);

insert into waste values ('type2', 2,'kg', 302);

insert into waste values ('type3', 3,'kg', 303);

insert into waste values ('type4', 4,'kg', 304);

insert into waste values ('type5', 5,'kg', 305);

select \* from waste;

CREATE TABLE complain

(

complain\_id INT,

image INT,

complain\_status CHAR(50),

description CHAR(50),

pin\_code INT,

waste\_unit INT,

user\_id INT,

waste\_id INT,

PRIMARY KEY (complain\_id),

FOREIGN KEY (user\_id) REFERENCES us\_er(user\_id),

FOREIGN KEY (waste\_id) REFERENCES waste(waste\_id)

);

insert into complain values (401, 10, 'complete', 'description1', 100000, 1, 101, 301);

insert into complain values (402, 11, 'incomplete', 'description2', 200000, 2, 102, 302);

insert into complain values (403, 12, 'complete', 'description3', 300000, 3, 103, 303);

insert into complain values (404, 13, 'incomplete', 'description4', 400000, 4, 104, 304);

insert into complain values (405, 14, 'complete', 'description5', 500000, 5, 105, 305);

select \* from complain;

CREATE TABLE customer\_address

(

pin\_code INT,

street CHAR(50),

area CHAR(50),

city CHAR(50),

state CHAR(50),

country CHAR(50),

user\_id INT,

PRIMARY KEY (pin\_code, user\_id),

FOREIGN KEY (user\_id) REFERENCES us\_er(user\_id)

);

insert into customer\_address values (100000, 'street1', 'area1', 'city1', 'state1', 'country1', 101);

insert into customer\_address values (200000, 'street2', 'area2', 'city2', 'state2', 'country2', 102);

insert into customer\_address values (300000, 'street3', 'area3', 'city3', 'state3', 'country3', 103);

insert into customer\_address values (400000, 'street4', 'area4', 'city4', 'state4', 'country4', 104);

insert into customer\_address values (500000, 'street5', 'area5', 'city5', 'state5', 'country5', 105);

select \* from customer\_address;

CREATE TABLE info

(

manager\_id INT,

manager\_name CHAR(50),

manager\_phone\_no INT,

manager\_email CHAR(50),

location CHAR(60),

complain\_id INT,

PRIMARY KEY (manager\_id),

FOREIGN KEY (complain\_id) REFERENCES complain(complain\_id)

);

insert into info values (501, 'mname1', 1111111121, 'mname1@gamil.com', 'location1', 401);

insert into info values (502, 'mname2', 1111111122, 'mname2@gamil.com', 'location2', 402);

insert into info values (503, 'mname3', 1111111123, 'mname3@gamil.com', 'location3', 403);

insert into info values (504, 'mname4', 1111111124, 'mname4@gamil.com', 'location4', 404);

insert into info values (505, 'mname5', 1111111125, 'mname5@gamil.com', 'location5', 405);

select \* from info;

CREATE TABLE government

(

gov\_id INT,

gov\_no INT,

gov\_area CHAR(50),

gov\_name CHAR(50),

manager\_id INT,

PRIMARY KEY (gov\_id),

FOREIGN KEY (manager\_id) REFERENCES info(manager\_id)

);

insert into government values (601, 21, 'garea1','gname1', 501);

insert into government values (602, 22, 'garea2','gname2', 502);

insert into government values (603, 23, 'garea3','gname3', 503);

insert into government values (604, 24, 'garea4','gname4', 504);

insert into government values (605, 25, 'garea5','gname5', 505);

select \* from government;

CREATE TABLE private

(

private\_id INT,

private\_name CHAR(50),

private\_area CHAR(50),

price INT,

manager\_id INT,

PRIMARY KEY (private\_id),

FOREIGN KEY (manager\_id) REFERENCES info(manager\_id)

);

insert into private values (701, 'pname1', 'parea1', 100, 501);

insert into private values (702, 'pname2', 'parea2', 110, 502);

insert into private values (703, 'pname3', 'parea3', 120, 503);

insert into private values (704, 'pname4', 'parea4', 130, 504);

insert into private values (705, 'pname5', 'parea5', 140, 505);

select \* from private;

CREATE TABLE driver

(

driver\_id INT,

driver\_name CHAR(50),

phone\_no INT,

provider CHAR(50),

PRIMARY KEY (driver\_id)

);

insert into driver values (801, 'dname1', 1111111131, 'provider1');

insert into driver values (802, 'dname2', 1111111132, 'provider2');

insert into driver values (803, 'dname3', 1111111133, 'provider3');

insert into driver values (804, 'dname4', 1111111134, 'provider4');

insert into driver values (805, 'dname5', 1111111135, 'provider5');

select \* from driver;

CREATE TABLE dump\_details

(

dump\_name CHAR(50),

dump\_area CHAR(50),

phone\_no INT,

PRIMARY KEY (phone\_no)

);

insert into dump\_details values ('dumpname1', 'dumparea1', 1111111131);

insert into dump\_details values ('dumpname2', 'dumparea2', 1111111132);

insert into dump\_details values ('dumpname3', 'dumparea3', 1111111133);

insert into dump\_details values ('dumpname4', 'dumparea4', 1111111134);

insert into dump\_details values ('dumpname5', 'dumparea5', 1111111135);

select \* from dump\_details;

CREATE TABLE History

(

driver\_id INT,

gov\_id INT,

private\_id INT,

waste\_id INT,

user\_id INT,

FOREIGN KEY (driver\_id) REFERENCES driver(driver\_id),

FOREIGN KEY (gov\_id) REFERENCES government(gov\_id),

FOREIGN KEY (private\_id) REFERENCES private(private\_id),

FOREIGN KEY (waste\_id) REFERENCES waste(waste\_id),

FOREIGN KEY (user\_id) REFERENCES us\_er(user\_id)

);

insert into History values (801, 601, 701, 301, 101);

insert into History values (802, 602, 702, 302, 102);

insert into History values (803, 603, 703, 303, 103);

insert into History values (804, 604, 704, 304, 104);

insert into History values (805, 605, 705, 305, 105);

select \* from History;

CREATE TABLE waste\_user

(

user\_id INT,

waste\_id INT,

FOREIGN KEY (user\_id) REFERENCES us\_er(user\_id),

FOREIGN KEY (waste\_id) REFERENCES waste(waste\_id)

);

insert into waste\_user values (101, 301);

insert into waste\_user values (102, 302);

insert into waste\_user values (103, 303);

insert into waste\_user values (104, 304);

insert into waste\_user values (105, 305);

select \* from waste\_user;

CREATE TABLE vehicle

(

vehicle\_no CHAR(50),

driver\_id INT,

private\_id INT,

gov\_id INT,

PRIMARY KEY (vehicle\_no, driver\_id),

FOREIGN KEY (driver\_id) REFERENCES driver(driver\_id),

FOREIGN KEY (private\_id) REFERENCES private(private\_id),

FOREIGN KEY (gov\_id) REFERENCES government(gov\_id)

);

insert into vehicle values ('a901', 801, 701, 601);

insert into vehicle values ('a902', 802, 702, 602);

insert into vehicle values ('a903', 803, 703, 603);

insert into vehicle values ('a904', 804, 704, 604);

insert into vehicle values ('a905', 805, 705, 605);

select \* from vehicle;

CREATE TABLE supply

(

supply\_date DATE,

supply\_time DATE,

supply\_status CHAR(50),

supply\_id INT,

private\_id INT,

gov\_id INT,

phone\_no INT,

PRIMARY KEY (supply\_id),

FOREIGN KEY (private\_id) REFERENCES private(private\_id),

FOREIGN KEY (gov\_id) REFERENCES government(gov\_id),

FOREIGN KEY (phone\_no) REFERENCES dump\_details(phone\_no)

);

insert into supply values ('01-april-2022', '01-april-2022', 'complete', 1001, 701, 601, 1111111131);

insert into supply values ('02-april-2022', '02-april-2022', 'incomplete', 1002, 702, 602, 1111111132);

insert into supply values ('03-april-2022', '03-april-2022', 'complete', 1003, 703, 603, 1111111133);

insert into supply values ('04-april-2022', '04-april-2022', 'incomplete', 1004, 704, 604, 1111111134);

insert into supply values ('05-april-2022', '05-april-2022', 'complete', 1005, 705, 605, 1111111135);

select \* from supply;

CREATE TABLE temp

(

complain\_id INT,

image INT,

complain\_status CHAR(50),

description CHAR(50),

pin\_code INT, waste\_unit INT,

user\_id INT,

waste\_id INT,

waste\_type CHAR(50),

waste\_weight INT

);

CREATE TABLE temp2

(

pin\_code INT,

street CHAR(50),

area CHAR(50),

city CHAR(50),

state CHAR(50),

country CHAR(50),

user\_id INT,

PRIMARY KEY (pin\_code, user\_id),

FOREIGN KEY (user\_id) REFERENCES us\_er(user\_id)

);

**Function, Procedure and Trigger**

1. Function (User Wants to Update Details)

create or replace function update\_data(id int, name char, mail char, age int, gender char, phone int) return int as

cursor c\_p is select \* from us\_er;

r\_p c\_p%rowtype;

begin

for r\_p in c\_p loop

update us\_er set user\_name = name, user\_mail = mail, user\_age = age, user\_gender = gender, user\_phone\_no = phone where r\_p.user\_id = id;

end loop;

dbms\_output.put\_line('successfully updated!');

return null;

end;

/

2. Function (Government and Private Details)

create or replace function disp\_details(p\_id int, g\_id int, pg\_unable int) return int as

cursor c\_p1 is select \* from government;

r\_p1 c\_p1%rowtype;

cursor c\_p2 is select \* from private;

r\_p2 c\_p2%rowtype;

begin

if(pg\_unable = 0) then

for r\_p1 in c\_p1 loop

if(r\_p1.gov\_id = g\_id) then

dbms\_output.put\_line(r\_p1.gov\_no || r\_p1.gov\_area || r\_p1.manager\_id);

end if;

end loop;

else

for r\_p2 in c\_p2 loop

if(r\_p2.private\_id = p\_id) then

dbms\_output.put\_line(r\_p2.private\_name || r\_p2.private\_area || r\_p2.price || r\_p2.manager\_id);

end if;

end loop;

end if;

return null;

end;

/

3. Procedure (Managers Information)

create or replace procedure us\_info as

cursor c\_p is select \* from info;

r\_p c\_p%rowtype;

begin

for r\_p in c\_p loop

dbms\_output.put\_line(r\_p.manager\_name || r\_p.manager\_phone\_no || r\_p.manager\_email || r\_p.location);

end loop;

end;

/

4. Trigger (Waste id and User id Insert to User\_Waste)

create or replace trigger waste\_usr after insert on complain

for each row

begin

insert into waste\_user values(:new.user\_id,:new.waste\_id);

end;

/

5.Trigger (Government Private Trigger)

create or replace trigger comp\_add after insert on complain

for each row

declare

cursor c\_p1 is select area from customer\_address;

r\_p1 c\_p1%rowtype;

cursor c\_p2 is select gov\_area, gov\_name from government;

r\_p2 c\_p2%rowtype;

cursor c\_p3 is select private\_area, private\_name from private;

r\_p3 c\_p3%rowtype;

chk1 int;

chk2 int;

begin

chk1 := 0;

chk2 := 0;

for r\_p1 in c\_p1 loop

for r\_p2 in c\_p2 loop

if(r\_p1.area = r\_p2.gov\_area) then

dbms\_output.put\_line(r\_p2.gov\_name);

chk1 := 1;

end if;

end loop;

for r\_p3 in c\_p3 loop

if(r\_p1.area = r\_p3.private\_area) then

dbms\_output.put\_line(r\_p3.private\_name); chk2 := 1;

end if;

end loop;

end loop;

if(chk1 = 0) then

dbms\_output.put\_line(‘Service not found in your area’);

end if;

if(chk2 = 0) then

dbms\_output.put\_line(‘Service not found in your area’);

end if;

end;

/

6.Function (Supply Details)

create or replace function sho\_supp(sup\_id int, i\_d int) return int as

cursor c\_p is select \* from supply;

r\_p c\_p%rowtype;

begin

for r\_p in c\_p loop

if(sup\_id = r\_p.supply\_id and i\_d = 0) then

dbms\_output.put\_line(r\_p.supply\_id || r\_p.supply\_date || r\_p.supply\_time || r\_p.supply\_status || r\_p.gov\_id);

else

dbms\_output.put\_line(r\_p.supply\_id || r\_p.supply\_date || r\_p.supply\_time || r\_p.supply\_status || r\_p.private\_id);

end if;

end loop;

return null;

end;

/

7. Function (displays all complain given by user for waste)

create or replace function usr\_comp(usr\_id int, com\_id int) return int as

cursor c\_p1 is select \* from waste;

r\_p1 c\_p1%rowtype;

cursor c\_p2 is select waste\_id, user\_id, complain\_id, description from complain;

r\_p2 c\_p2%rowtype;

begin

for r\_p2 in c\_p2 loop

for r\_p1 in c\_p1 loop

if(usr\_id = r\_p2.user\_id and com\_id = r\_p2.complain\_id and r\_p1.waste\_id = r\_p2.waste\_id) then

dbms\_output.put\_line(r\_p2.complain\_id || r\_p2.waste\_id || r\_p1.waste\_type || r\_p1.waste\_eight || r\_p1.waste\_unit || r\_p2.description);

end if;

end loop;

end loop;

return null;

end;

/

8. Procedure (Driver Details)

create or replace procedure driv\_no as

cursor c\_p is select \* from driver;

r\_p c\_p%rowtype;

begin

for r\_p in c\_p loop

dbms\_output.put\_line(r\_p.driver\_id || r\_p.driver\_name || r\_p.phone\_no);

end loop;

end;

/

9. Function (User Details)

create or replace function usr\_dt(usr\_id int) return int as

cursor c\_p is select \* from us\_er;

r\_p c\_p%rowtype;

begin

for r\_p in c\_p loop

if(r\_p.user\_id = usr\_id) then

dbms\_output.put\_line(r\_p.user\_name || r\_p.user\_id || r\_p.user\_mail || r\_p.user\_age || r\_p.user\_gender || r\_p.user\_phone\_no);

end if;

end loop;

return null;

end;

/

10. Function (display complain history of a user)

create or replace function view\_history(usr\_id int) return int as

cursor c\_p is select \* from history;

r\_p c\_p%rowtype;

begin

for r\_p in c\_p loop

if(r\_p.user\_id = usr\_id) then

dbms\_output.put\_line(r\_p.user\_id || r\_p.driver\_id || r\_p.gov\_id || r\_p.private\_id || r\_p.waste\_id);

end if;

end loop;

return null;

end;

/

11. Function (Use if user wants to delete the complain)

create or replace function del\_com(com\_id int, usr\_id int) return int as

cursor c\_p is select user\_id, complain\_id from complain;

r\_p c\_p%rowtype;

begin

for r\_p in c\_p loop

if (com\_id = r\_p.complain\_id and usr\_id = r\_p.user\_id) then

delete from complain where complain\_id = com\_id;

end if;

end loop;

return null;

end;

/

12. Function (if user wants to update the complain details in complain table)

create or replace function upd\_com(com\_id int, usr\_id int, wst\_id int, wst\_ut int, add int, de\_sc char) return int as

cursor c\_p is select complain\_id, user\_id from complain;

r\_p c\_p%rowtype;

begin

for r\_p in c\_p loop

if(com\_id = r\_p.complain\_id and usr\_id = r\_p.user\_id) then

update complain set waste\_id = wst\_id, waste\_unit = wst\_ut, pin\_code = add, description = de\_sc where r\_p.complain\_id = com\_id;

end if;

end loop;

return null;

end;

/

13. Function (if the company want to update the manager information)

create or replace function upd\_info(mng\_id int, mng\_name char, mag\_no int, mng\_mid char, mng\_location char) return int as

cursor c\_p is select manager\_id from info;

r\_p c\_p%rowtype;

begin

for r\_p in c\_p loop

if(mng\_id = r\_p.manager\_id) then

update info set manager\_name = mng\_name, manager\_phone\_no = mag\_no, manager\_email = mng\_mid, location = mng\_location;

end if;

end loop;

return null;

end;

/

14.Trigger (separating the data inserted by user while complaining and separated into complain and waste table)

create or replace trigger sep\_tem after insert on temp

For each row

begin

insert into complain values(:new.complain\_id, :new.image, :new.complain\_status, :new.description, :new.pin\_code, :new.waste\_unit, :new.user\_id, :new.waste\_id);

insert into waste values(:new.waste\_id, :new.waste\_type, :new.waste\_weight, :new.waste\_unit);

end;

/

15. Function (User Log-in Details)

create or replace function log\_in(usr\_email char) return int as

cursor c\_p is select user\_mail from us\_er;

r\_p c\_p%rowtype;

begin

for r\_p in c\_p loop

if(r\_p.user\_mail = usr\_email) then

dbms\_output.put\_line('Access Granted for the account');

else

dbms\_output.put\_line('email not found, create your account');

end if;

end loop;

return null;

end;

/

16. Trigger (if there is no service in user area display the error)

create or replace trigger chk\_ser after insert on temp2

for each row

declare

cursor c\_p1 is select area from temp2;

r\_p1 c\_p1%rowtype;

cursor c\_p2 is select gov\_area from government;

r\_p2 c\_p2%rowtype;

cursor c\_p3 is select private\_area from private;

r\_p3 c\_p3%rowtype;

begin

for r\_p1 in c\_p1 loop

for r\_p2 in c\_p2 loop

for r\_p3 in c\_p3 loop

if(r\_p1.area = r\_p2.gov\_area or r\_p1.area = r\_p3.private\_area) then

dbms\_output.put\_line('Address Added');

insert into customer\_address values(:new.pin\_code, :new.user\_id, :new.street, :new.area, :new.city, :new.state, :new.country);

else

dbms\_output.put\_line('No service found in your area!');

end if;

end loop;

end loop;

end loop;

end;

/

17. Trigger (Discount on the price of private sector)

create or replace trigger disk\_prc after insert on private

for each row

declare

cursor c\_p1 is select \* from private;

r\_p1 c\_p1%rowtype;

cursor c\_p2 is select \* from waste;

r\_p2 c\_p2%rowtype;

cursor c\_p3 is select \* from history;

r\_p3 c\_p3%rowtype;

discount int;

begin

for r\_p1 in c\_p1 loop

for r\_p2 in c\_p2 loop

for r\_p3 in c\_p3 loop

if(r\_p1.private\_id = r\_p3.private\_id and r\_p2.waste\_id = r\_p3.waste\_id) then

if(r\_p2.waste\_weight > 10 and r\_p2.waste\_weight <= 20) then

discount := (r\_p1.price\*5)/100;

dbms\_output.put\_line('Discount = '||discount);

elsif(r\_p2.waste\_weight >20 and r\_p2.waste\_weight <= 30) then

discount := (r\_p1.price\*10)/100;

dbms\_output.put\_line('Discount = '||discount);

elsif(r\_p2.waste\_weight > 30 and r\_p2.waste\_weight <= 40) then

discount := (r\_p1.price\*15)/100;

dbms\_output.put\_line('Discount = '||discount);

elsif(r\_p2.waste\_weight > 40) then

discount := (r\_p1.price\*20)/100;

dbms\_output.put\_line('Discount = '||discount);

else

dbms\_output.put\_line('Discount = 0');

end if;

end if;

end loop;

end loop;

end loop;

end;

/