**12.A employee\_info**

**Query:**

create or replace view employee\_info as select a.first\_name||' '||a.middle\_name||' '||a.last\_name as Name\_of\_employee,a.street\_address as address,

a.HOME\_PHONE\_NUMBER,a.CELL\_PHONE\_NUMBER,a.ZIP\_CODE,a.SSN,a.SALARY,a.BIRTH\_DATE, TRUNC(MONTHS\_BETWEEN(SYSDATE, a.birth\_date))/12 as age,

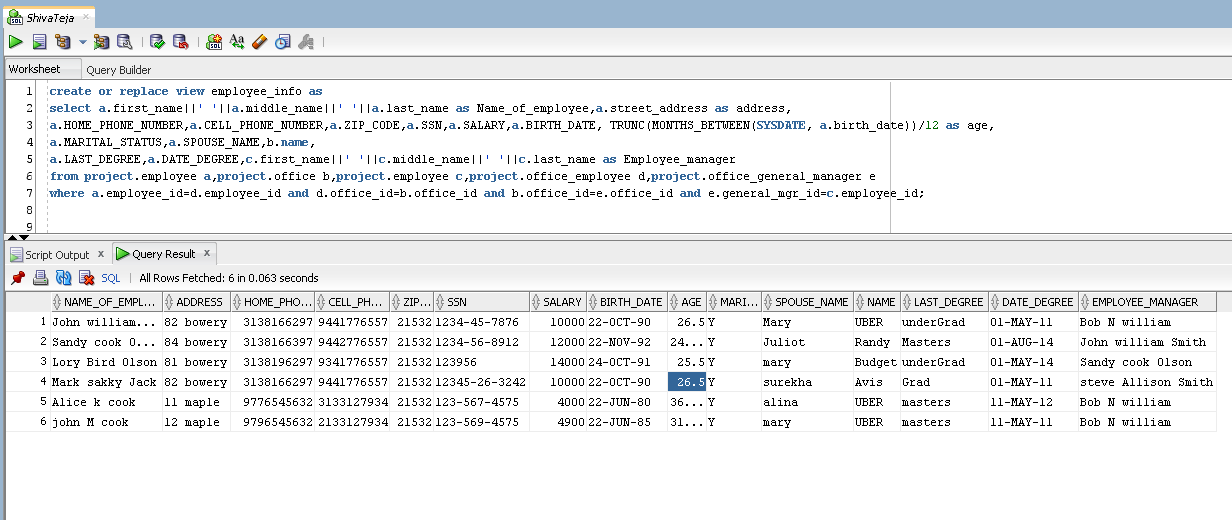
a.MARITAL\_STATUS,a.SPOUSE\_NAME,b.name,

a.LAST\_DEGREE,a.DATE\_DEGREE,c.first\_name||' '||c.middle\_name||' '||c.last\_name as Employee\_manager

from project.employee a,project.office b,project.employee c,project.office\_employee d,project.office\_general\_manager e

where a.employee\_id=d.employee\_id and d.office\_id=b.office\_id and b.office\_id=e.office\_id and e.general\_mgr\_id=c.employee\_id;

**Output:**



**12.b** **employee\_salary Query:**

create or replace view employee\_salary as

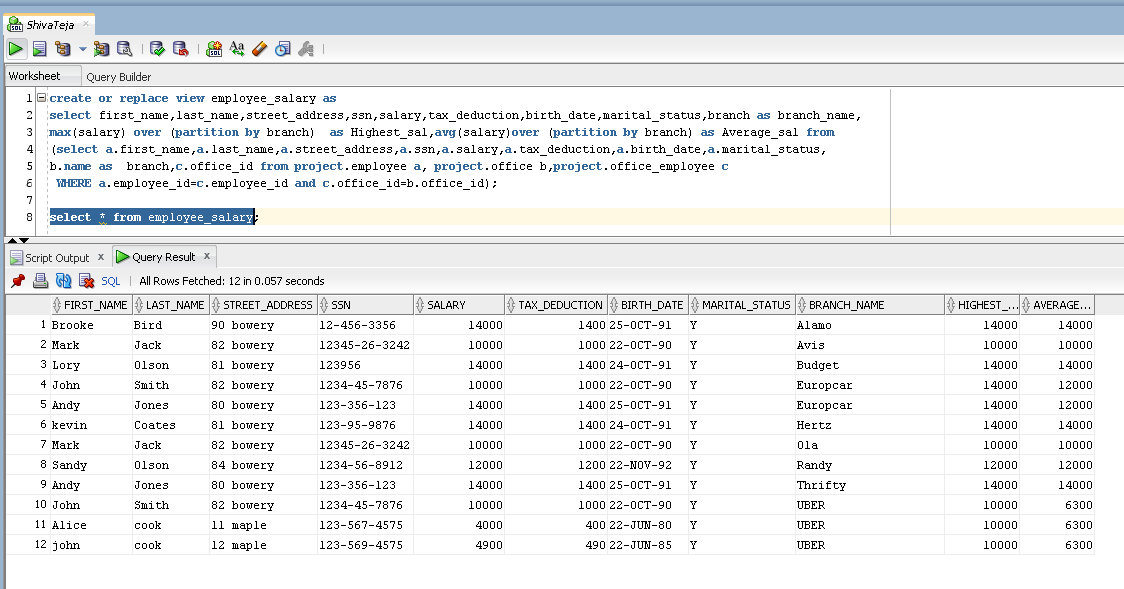
select first\_name,last\_name,street\_address,ssn,salary,tax\_deduction,birth\_date,marital\_status,branch as branch\_name,

max(salary) over (partition by branch) as Highest\_sal,avg(salary)over (partition by branch) as Average\_sal from

(select a.first\_name,a.last\_name,a.street\_address,a.ssn,a.salary,a.tax\_deduction,a.birth\_date,a.marital\_status,

b.name as branch,c.office\_id from project.employee a, project.office b,project.office\_employee c

WHERE a.employee\_id=c.employee\_id and c.office\_id=b.office\_id);



**12.c branch\_info**

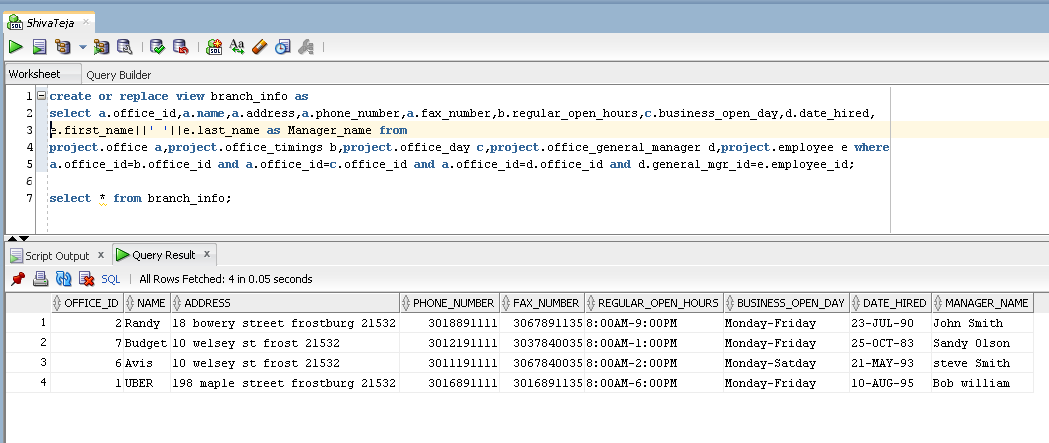
**Query:**

create or replace view branch\_info as

select a.office\_id,a.name,a.address,a.phone\_number,a.fax\_number,b.regular\_open\_hours,c.business\_open\_day,d.date\_hired,e.first\_name||' '||e.last\_name as Manager\_name from

project.office a,project.office\_timings b,project.office\_day c,project.office\_general\_manager d,project.employee e where

a.office\_id=b.office\_id and a.office\_id=c.office\_id and a.office\_id=d.office\_id and d.general\_mgr\_id=e.employee\_id;



**12.d valued\_customers**

**Query:**

create or replace view valued\_customers as

select full\_name,age,cell\_number,address,zipCode,email,License\_number,license\_issued\_state,number\_of\_times\_rented from

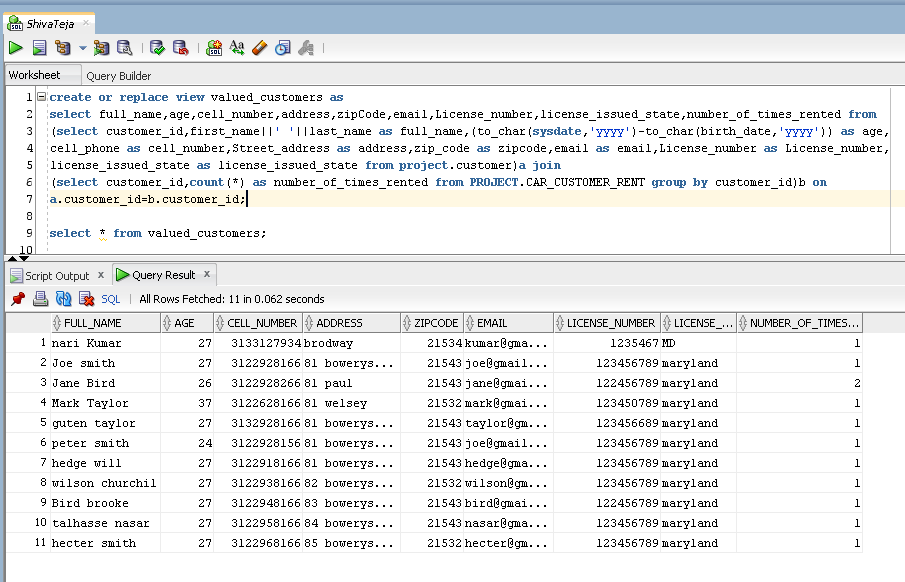
(select customer\_id,first\_name||' '||last\_name as full\_name,(to\_char(sysdate,'yyyy')-to\_char(birth\_date,'yyyy')) as age,

cell\_phone as cell\_number,Street\_address as address,zip\_code as zipcode,email as email,License\_number as License\_number,

license\_issued\_state as license\_issued\_state from project.customer)a join

(select customer\_id,count(\*) as number\_of\_times\_rented from PROJECT.CAR\_CUSTOMER\_RENT group by customer\_id)b on

a.customer\_id=b.customer\_id;



**12.e Query:**

create or replace view list\_of\_cars as

select VECH.car\_id,VECH.car\_plate\_no,VECH.car\_state\_registeration,VECH.year,VECH.first\_name||' '||VECH.last\_name as Owner\_name,

VECH.class,VECH.cost\_damage,VECH.make,VECH.color,CCR.count\_ccr as No\_times\_car\_rented,ACD.count\_ACD as No\_times\_car\_accidents from

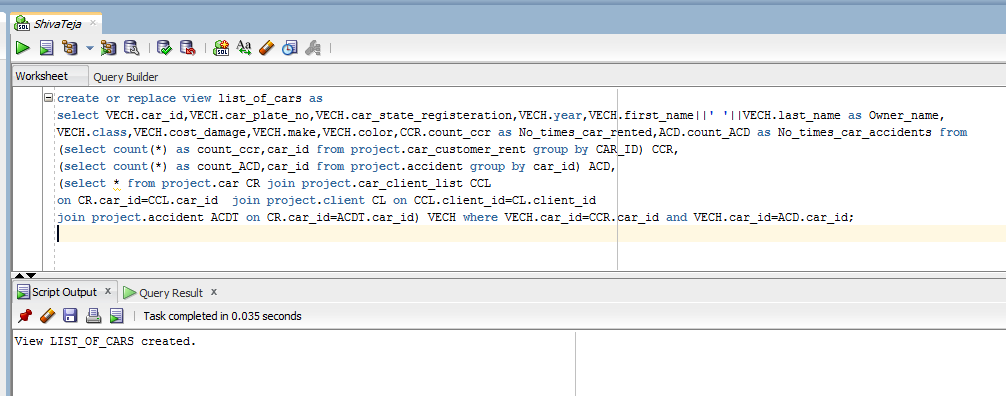
(select count(\*) as count\_ccr,car\_id from project.car\_customer\_rent group by CAR\_ID) CCR,

(select count(\*) as count\_ACD,car\_id from project.accident group by car\_id) ACD,

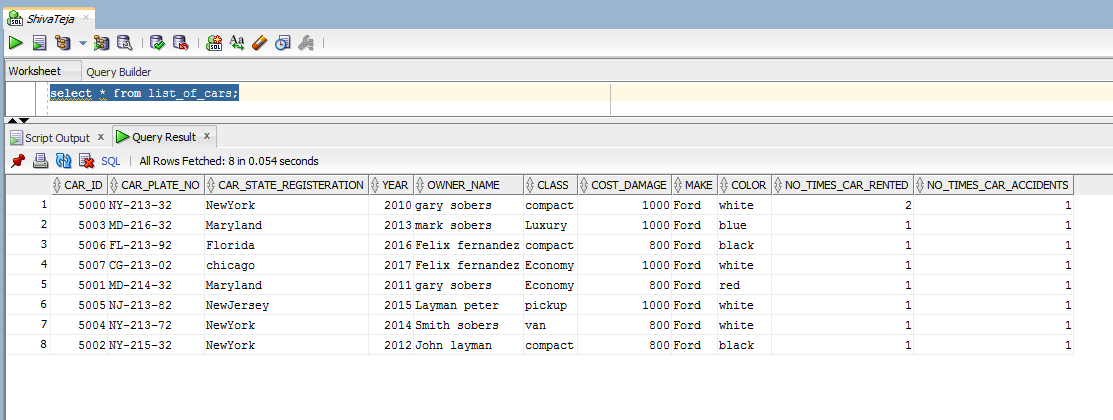
(select \* from project.car CR join project.car\_client\_list CCL

on CR.car\_id=CCL.car\_id join project.client CL on CCL.client\_id=CL.client\_id

join project.accident ACDT on CR.car\_id=ACDT.car\_id) VECH where VECH.car\_id=CCR.car\_id and VECH.car\_id=ACD.car\_id;



Select \* from list\_of\_cars;



**12.f list\_of\_transaction**

**Query**

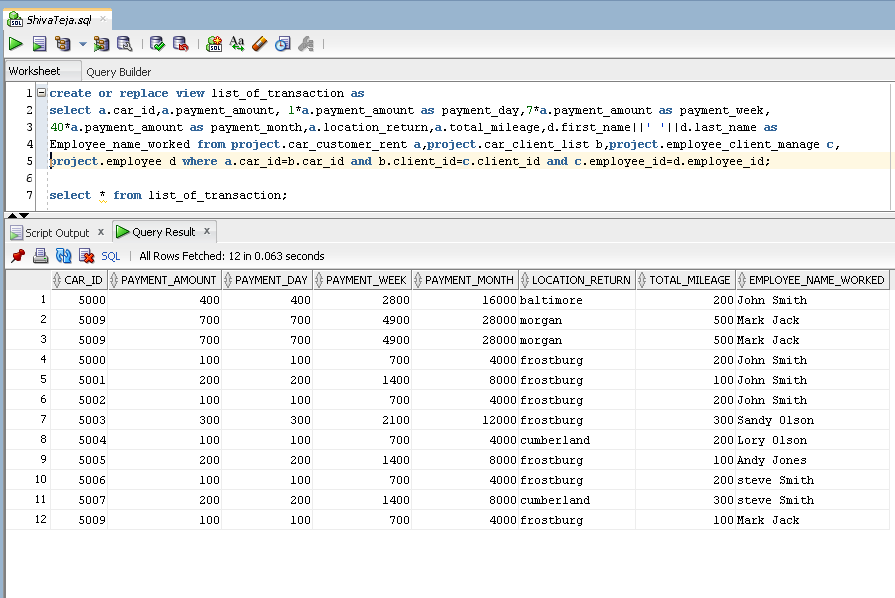
create or replace view list\_of\_transaction as

select a.car\_id,a.payment\_amount, 1\*a.payment\_amount as payment\_day,7\*a.payment\_amount as payment\_week,

40\*a.payment\_amount as payment\_month,a.location\_return,a.total\_mileage,d.first\_name||' '||d.last\_name as

Employee\_name\_worked from project.car\_customer\_rent a,project.car\_client\_list b,project.employee\_client\_manage c,

project.employee d where a.car\_id=b.car\_id and b.client\_id=c.client\_id and c.employee\_id=d.employee\_id;



**12.g Query**

create or replace view Statistics\_by\_Branch as

select office\_id,owner\_name,customer\_name,No\_times\_car\_listed,year,total\_amount\_received

from

(select ccl.client\_id,count(ccr.car\_id) as no\_times\_car\_listed,to\_char(ccr.date\_of\_rent,'yyyy') as year,sum(ccr.payment\_amount) as total\_amount\_received

from project.car\_customer\_rent CCR,project.car\_client\_list CCL where CCR.car\_id=CCL.car\_id group by

ccl.client\_id,to\_char(ccr.date\_of\_rent,'yyyy'))CCL\_CCR

join

(select distinct OFC.office\_id as office\_id,CCL.client\_id,CLT.first\_name|| ' ' ||CLT.last\_name as owner\_name,CST.first\_name|| ' ' ||CST.last\_name as customer\_name

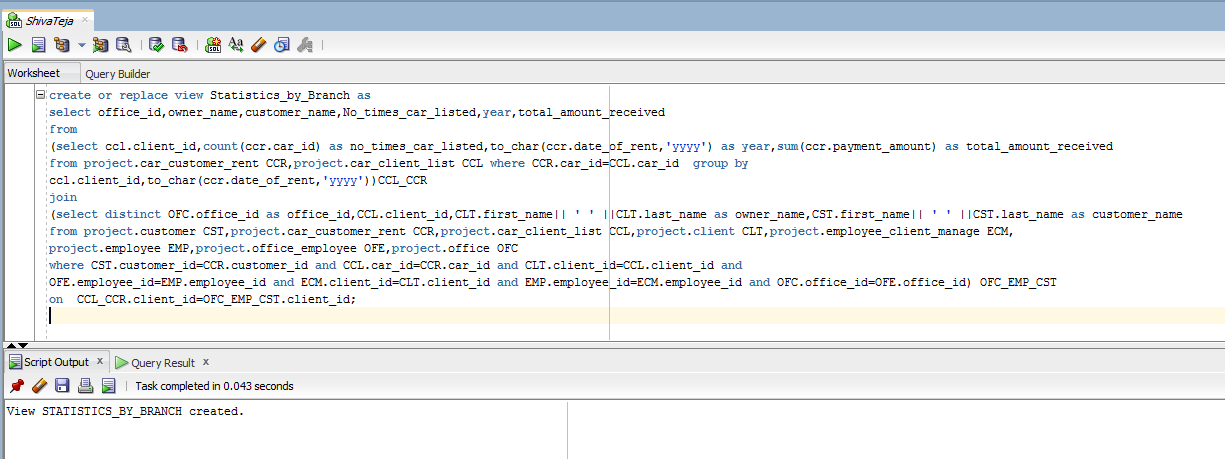
from project.customer CST,project.car\_customer\_rent CCR,project.car\_client\_list CCL,project.client CLT,project.employee\_client\_manage ECM,

project.employee EMP,project.office\_employee OFE,project.office OFC

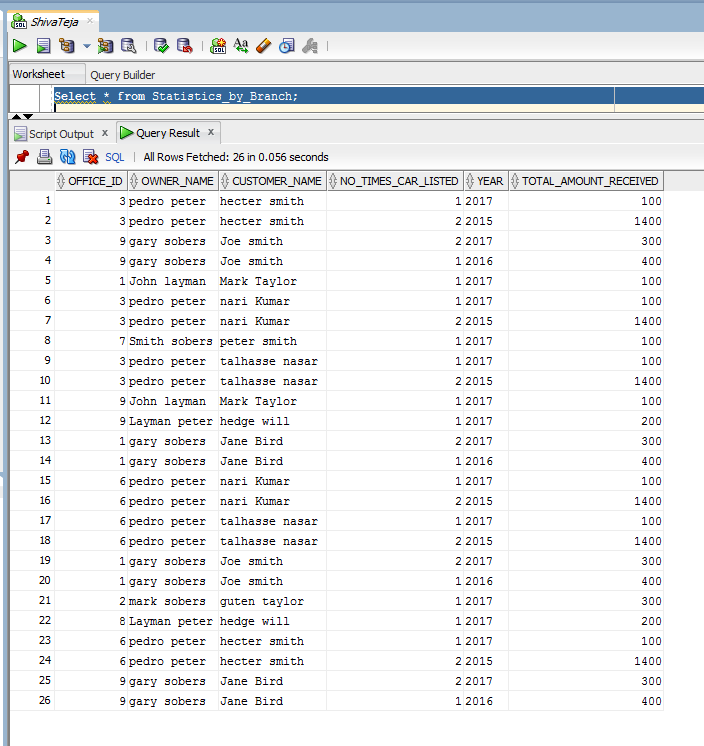
where CST.customer\_id=CCR.customer\_id and CCL.car\_id=CCR.car\_id and CLT.client\_id=CCL.client\_id and

OFE.employee\_id=EMP.employee\_id and ECM.client\_id=CLT.client\_id and EMP.employee\_id=ECM.employee\_id and OFC.office\_id=OFE.office\_id) OFC\_EMP\_CST

on CCL\_CCR.client\_id=OFC\_EMP\_CST.client\_id;



Select \* from Statistics\_by\_Branch;



**12.h last\_employee\_branch**

**Query**

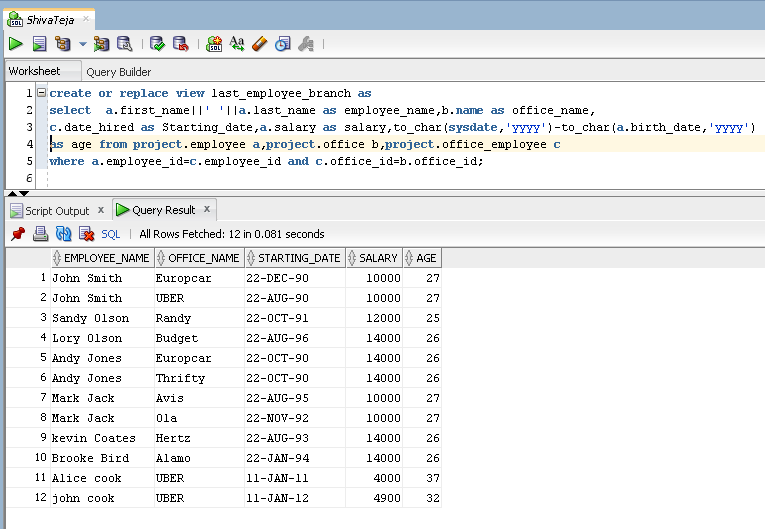
create or replace view last\_employee\_branch as

select a.first\_name||' '||a.last\_name as employee\_name,b.name as office\_name,

c.date\_hired as Starting\_date,a.salary as salary,to\_char(sysdate,'yyyy')-to\_char(a.birth\_date,'yyyy')

as age from project.employee a,project.office b,project.office\_employee c

where a.employee\_id=c.employee\_id and c.office\_id=b.office\_id;



**12.i managers**

**Query:**

create or replace view managers as

select total\_employees,starting\_date,manager\_name from

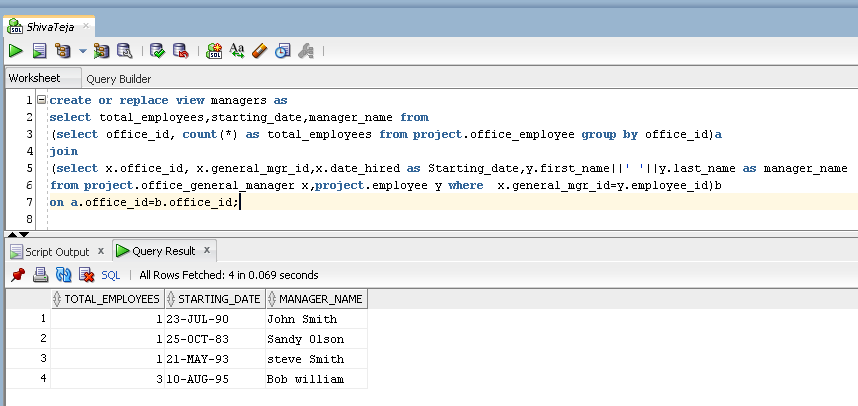
(select office\_id, count(\*) as total\_employees from project.office\_employee group by office\_id)a

join

(select x.office\_id, x.general\_mgr\_id,x.date\_hired as Starting\_date,y.first\_name||' '||y.last\_name as manager\_name

from project.office\_general\_manager x,project.employee y where x.general\_mgr\_id=y.employee\_id)b

on a.office\_id=b.office\_id;



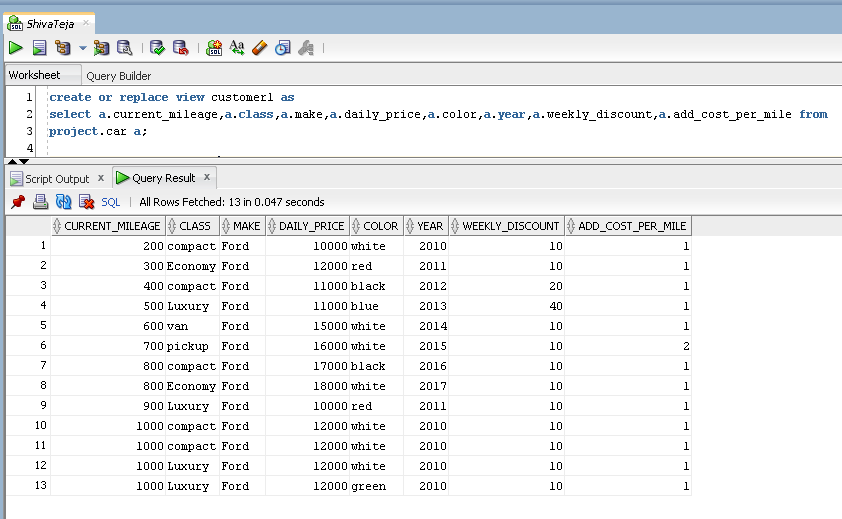
**12.j Customers1**

**Query**

create or replace view customer1 as

select a.current\_mileage,a.class,a.make,a.daily\_price,a.color,a.year,a.weekly\_discount,a.add\_cost\_per\_mile from

project.car a;



**12.k. customer2**

**Query**

create or replace view car\_feedback as

select carid,current\_mileage,class,features,make,year,daily\_price,car\_desc,rating,feedback\_given,comments from

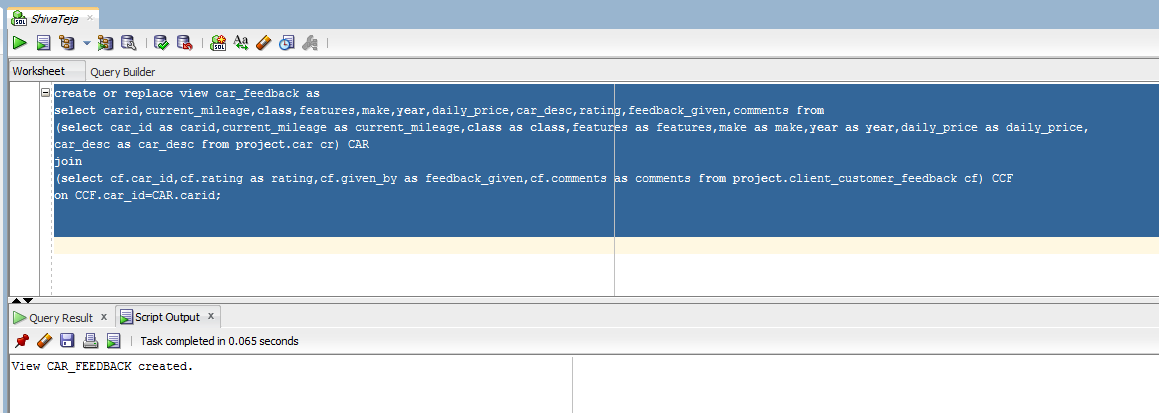
(select car\_id as carid,current\_mileage as current\_mileage,class as class,features as features,make as make,year as year,daily\_price as daily\_price,

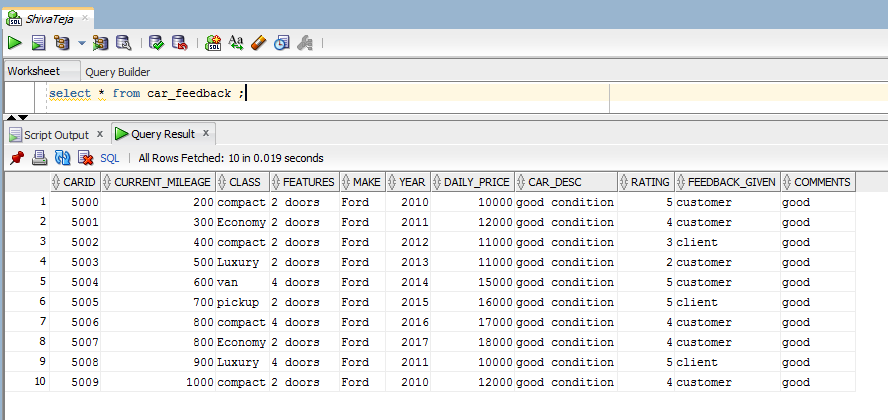
car\_desc as car\_desc from project.car cr) CAR

join

(select cf.car\_id,cf.rating as rating,cf.given\_by as feedback\_given,cf.comments as comments from project.client\_customer\_feedback cf) CCF

on CCF.car\_id=CAR.carid;



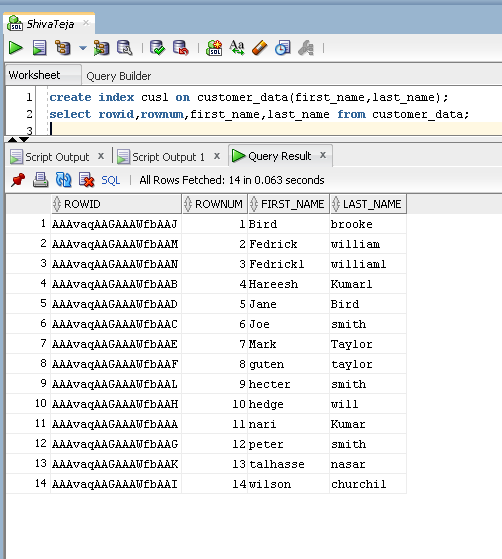


**13.a Index**

**Query:**

create index cus1 on customer\_data(first\_name,last\_name);

select rowid,rownum,first\_name,last\_name from customer\_data;

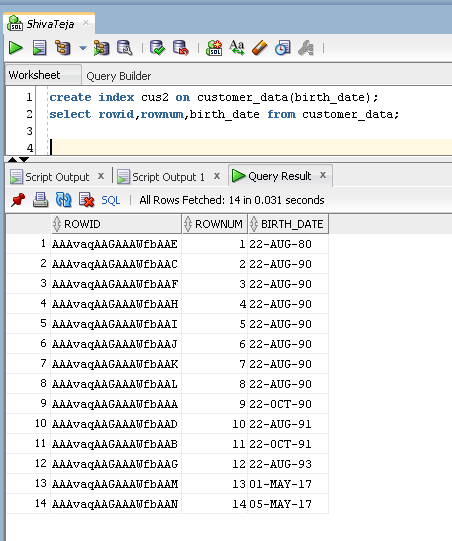


**13.b Index**

**Query**

create index cus2 on customer\_data(birth\_date);

select rowid,rownum,birth\_date from customer\_data;



**14.a ID\_generator**

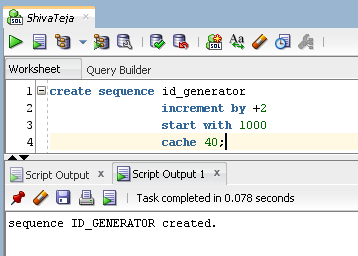
**Query**

create sequence id\_generator

increment by +2

start with 1000

cache 40;



**14.b Transaction id**

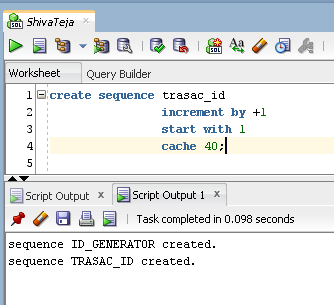
**Query**

create sequence trasac\_id

increment by +1

start with 1

cache 40;



**14.c Tracknumber**

**Query**

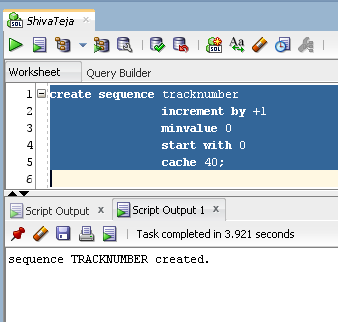
create sequence tracknumber

increment by +1

minvalue 0

start with 0

cache 40;



**14.d**

create table inquiries

(Inquiry\_Date date default sysdate,

type\_of\_request varchar2(20),

Full\_name varchar2(20),

email varchar2(20),

tracknumber number primary key);

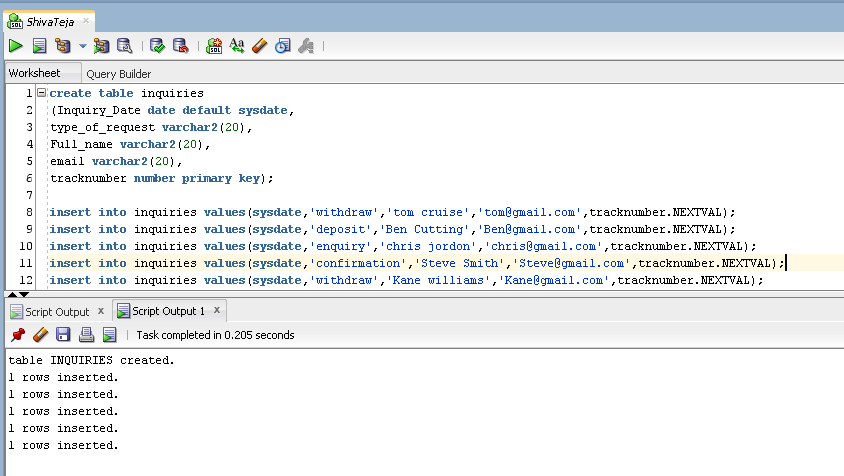
insert into inquiries values(sysdate,'withdraw','tom cruise','tom@gmail.com',tracknumber.NEXTVAL);

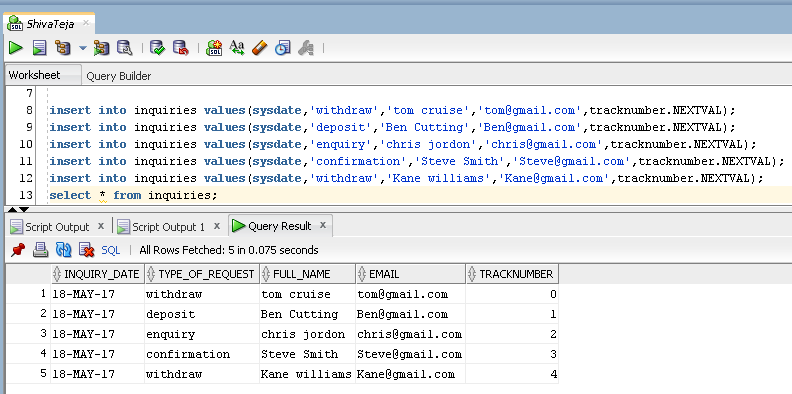
insert into inquiries values(sysdate,'deposit','Ben Cutting','Ben@gmail.com',tracknumber.NEXTVAL);

insert into inquiries values(sysdate,'enquiry','chris jordon','chris@gmail.com',tracknumber.NEXTVAL);

insert into inquiries values(sysdate,'confirmation','Steve Smith','Steve@gmail.com',tracknumber.NEXTVAL);

insert into inquiries values(sysdate,'withdraw','Kane williams','Kane@gmail.com',tracknumber.NEXTVAL);





**15.a EmpInsert\_pkg**

**Query:**

create or replace package EmpInsert\_pkg as

PROCEDURE insert\_new\_Employee(employee\_id IN project.employee.employee\_id%TYPE,first\_name IN project.employee.first\_name%TYPE,

last\_name in project.employee.last\_name%type,middle\_name IN project.employee.middle\_name%TYPE,

street\_address IN project.employee.street\_address%TYPE,city IN project.employee.city%TYPE,state IN project.employee.state%TYPE,

zip\_code IN project.employee.zip\_code%TYPE,ssn IN project.employee.ssn%TYPE,salary IN project.employee.salary%TYPE,

tax\_deduction IN project.employee.tax\_deduction%TYPE,birthdate IN project.employee.birth\_date%TYPE,

gender IN project.employee.gender%TYPE, marital\_status IN project.employee.marital\_status%TYPE,

spouse\_name IN project.employee.spouse\_name%TYPE,last\_degree IN project.employee.last\_degree%TYPE,

date\_degree IN project.employee.date\_degree%TYPE,number\_of\_dependents IN project.employee.NO\_OF\_DEPENDENTS%TYPE,

home\_phone\_number IN project.employee.home\_phone\_number%TYPE,cell\_phone\_number IN project.employee.cell\_phone\_number%TYPE,

date\_cert\_received IN project.employee\_certificate\_dates.date\_cert\_received%TYPE,list\_of\_certificates IN project.employee\_certificates.list\_of\_certificates%TYPE);

PROCEDURE update\_marital\_statue\_employee(employee\_id IN project.employee.employee\_ID%TYPE,

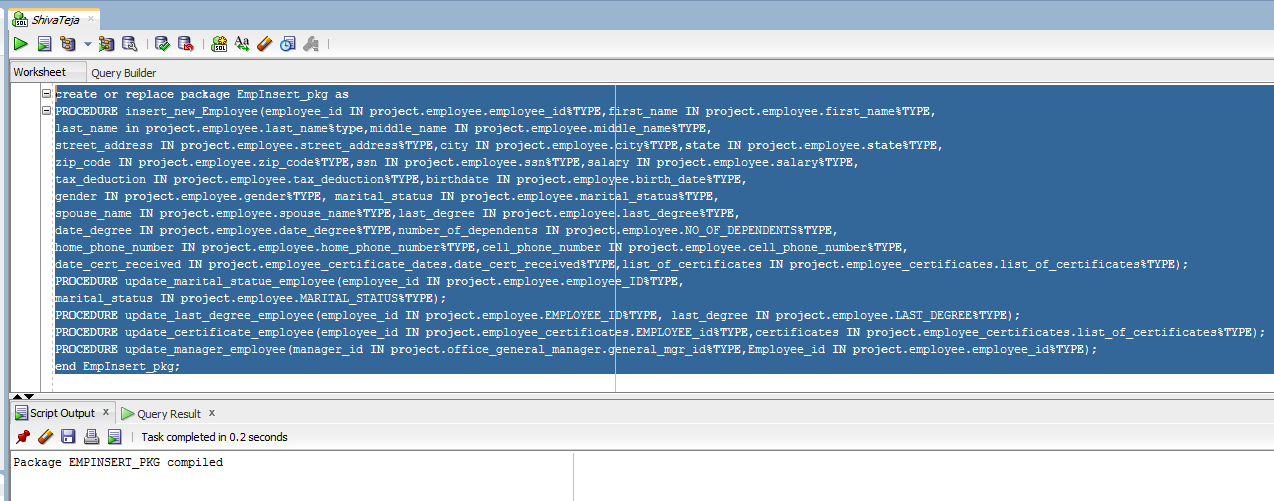
marital\_status IN project.employee.MARITAL\_STATUS%TYPE);

PROCEDURE update\_last\_degree\_employee(employee\_id IN project.employee.EMPLOYEE\_ID%TYPE, last\_degree IN project.employee.LAST\_DEGREE%TYPE);

PROCEDURE update\_certificate\_employee(employee\_id IN project.employee\_certificates.EMPLOYEE\_id%TYPE,certificates IN project.employee\_certificates.list\_of\_certificates%TYPE);

PROCEDURE update\_manager\_employee(manager\_id IN project.office\_general\_manager.general\_mgr\_id%TYPE,Employee\_id IN project.employee.employee\_id%TYPE);

end EmpInsert\_pkg;



create or replace package body EmpInsert\_pkg as

PROCEDURE insert\_new\_Employee(employee\_id IN project.employee.employee\_id%TYPE,first\_name IN project.employee.first\_name%TYPE,

last\_name in project.employee.last\_name%type,middle\_name IN project.employee.middle\_name%TYPE,

street\_address IN project.employee.street\_address%TYPE,city IN project.employee.city%TYPE,state IN project.employee.state%TYPE,

zip\_code IN project.employee.zip\_code%TYPE,ssn IN project.employee.ssn%TYPE,salary IN project.employee.salary%TYPE,

tax\_deduction IN project.employee.tax\_deduction%TYPE,birthdate IN project.employee.birth\_date%TYPE,

gender IN project.employee.gender%TYPE, marital\_status IN project.employee.marital\_status%TYPE,

spouse\_name IN project.employee.spouse\_name%TYPE,last\_degree IN project.employee.last\_degree%TYPE,

date\_degree IN project.employee.date\_degree%TYPE,number\_of\_dependents IN project.employee.NO\_OF\_DEPENDENTS%TYPE,

home\_phone\_number IN project.employee.home\_phone\_number%TYPE,cell\_phone\_number IN project.employee.cell\_phone\_number%TYPE,

date\_cert\_received IN project.employee\_certificate\_dates.date\_cert\_received%TYPE,list\_of\_certificates IN project.employee\_certificates.list\_of\_certificates%TYPE) IS

BEGIN

INSERT INTO employee\_temp(employee\_id,first\_name,last\_name, middle\_name, street\_address, city,state, zip\_code, ssn,

salary, tax\_deduction, birth\_date, gender, marital\_status, spouse\_name, last\_degree, date\_degree, NO\_OF\_DEPENDENTS, home\_phone\_number, cell\_phone\_number)

VALUES (employee\_id, first\_name ,last\_name, middle\_name ,street\_address ,city ,state,zip\_code ,ssn ,salary ,

tax\_deduction ,birthdate ,gender ,marital\_status ,spouse\_name ,last\_degree ,date\_degree ,number\_of\_dependents ,home\_phone\_number ,cell\_phone\_number);

INSERT INTO employee\_crtf\_dates\_temp(employee\_id,date\_cert\_received) values(employee\_id,date\_cert\_received);

INSERT INTO employee\_certificates\_temp(employee\_id,list\_of\_certificates) values(employee\_id,date\_cert\_received);

exception

when others then

dbms\_output.put\_line('error: entered invalid values');

end;

PROCEDURE update\_marital\_statue\_employee(employee\_id IN project.employee.employee\_ID%TYPE,

marital\_status IN project.employee.MARITAL\_STATUS%TYPE)IS

BEGIN

UPDATE employee\_temp SET MARITAL\_STATUS = marital\_status WHERE employee\_ID = employee\_id;

exception

when no\_data\_found then

dbms\_output.put\_line('error: occured due to entered value employee\_id is in-correct');

when others then

dbms\_output.put\_line('error: invalid data entered');

END;

PROCEDURE update\_last\_degree\_employee(employee\_id IN project.employee.EMPLOYEE\_ID%TYPE, last\_degree IN project.employee.LAST\_DEGREE%TYPE)IS

BEGIN

UPDATE employee\_temp SET LAST\_DEGREE= last\_degree WHERE employee\_id = employee\_id;

exception

when no\_data\_found then

dbms\_output.put\_line('error: occured due to entered value employee\_id is in correct');

when others then

dbms\_output.put\_line('error: invalid data entered');

END;

PROCEDURE update\_certificate\_employee(employee\_id IN project.employee\_certificates.EMPLOYEE\_id%TYPE,

certificates IN project.employee\_certificates.list\_of\_certificates%TYPE) IS

BEGIN

UPDATE employee\_certificates\_temp SET LIST\_OF\_CERTIFICATES = certificates

WHERE employee\_id = employee\_id;

exception

when no\_data\_found then

dbms\_output.put\_line('error: occured due to entered value employee\_id is in correct');

when others then

dbms\_output.put\_line('error: invalid data entered');

END;

PROCEDURE update\_manager\_employee(manager\_id IN project.office\_general\_manager.general\_mgr\_id%TYPE,Employee\_id IN project.employee.employee\_id%TYPE) IS

BEGIN

UPDATE manager\_temp SET general\_mgr\_id = Employee\_id WHERE general\_mgr\_id = manager\_id;

exception

when no\_data\_found then

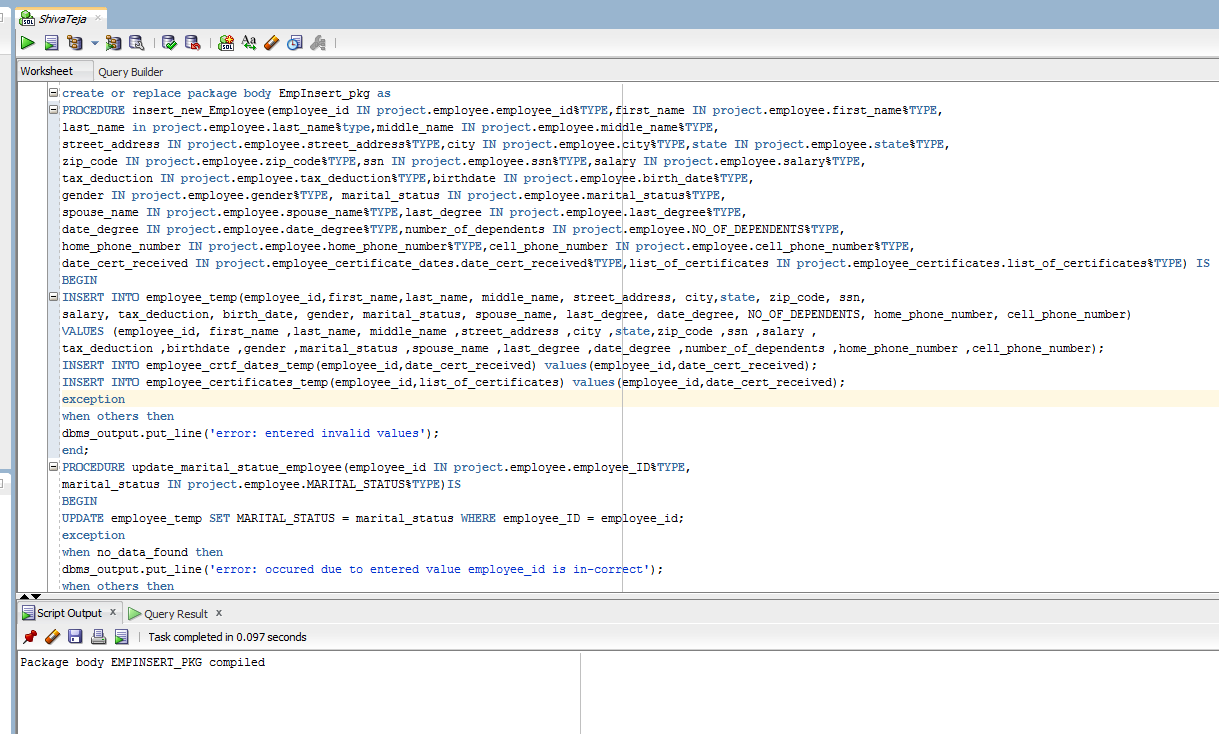
dbms\_output.put\_line('error: occured due to entered value employee\_id is in correct');

when others then

dbms\_output.put\_line('error: invalid data entered');

END;

end empInsert\_pkg;



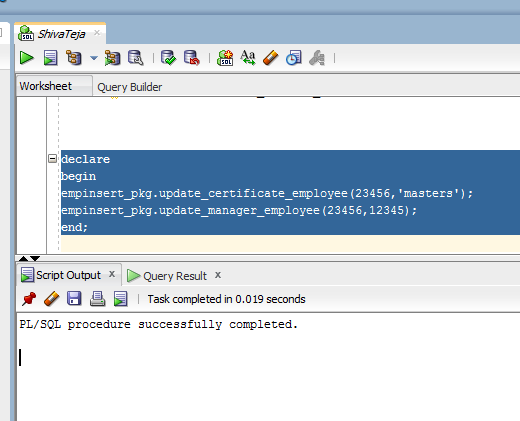
declare

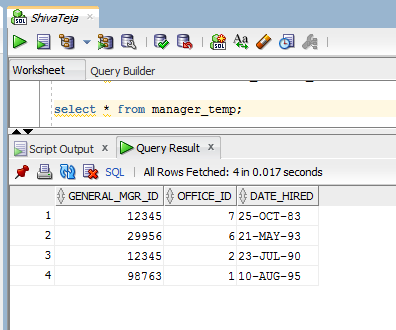
begin

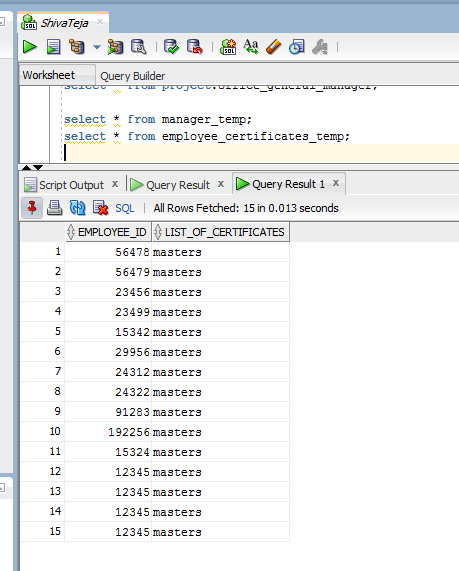
empinsert\_pkg.update\_certificate\_employee(23456,'masters');

empinsert\_pkg.update\_manager\_employee(23456,12345);

end;





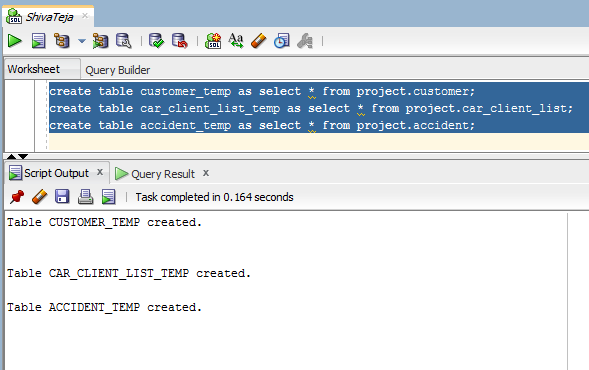


**15.b**

create table customer\_temp as select \* from project.customer;

create table car\_client\_list\_temp as select \* from project.car\_client\_list;

create table accident\_temp as select \* from project.accident;



create or replace package insert\_pkg as

procedure new\_customer\_insert(customer\_id IN project.customer.CUSTOMER\_ID%TYPE,first\_name IN project.customer.FIRST\_NAME%TYPE,

last\_name IN project.customer.LAST\_NAME%TYPE,middle\_initial IN project.customer.MIDDLE\_INITIAL%TYPE,birth\_date IN project.customer.BIRTH\_DATE%TYPE,

cell\_phone IN project.customer.CELL\_PHONE%TYPE,street\_address IN project.customer.STREET\_ADDRESS%TYPE,city IN project.customer.CITY%TYPE,

state1 IN project.customer.STATE%TYPE,zip\_code IN project.customer.ZIP\_CODE%TYPE,email IN project.customer.EMAIL%TYPE,

license\_issued\_state IN project.customer.LICENSE\_ISSUED\_STATE%TYPE,license\_number IN project.customer.LICENSE\_NUMBER%TYPE,

credit\_card\_number IN project.customer.CREDIT\_CARD\_NUMBER%TYPE,expiration\_date IN project.customer.EXPIRATION\_DATE%TYPE,

ssn1 in project.customer.ssn%type);

procedure insert\_new\_car\_owner(car\_id project.car\_client\_list.car\_id%type,client\_id project.car\_client\_list.client\_id%type,

available\_hrs project.car\_client\_list.available\_hrs%type,start\_date project.car\_client\_list.start\_date%type,

end\_date project.car\_client\_list.end\_date%type);

procedure info\_update\_customer(cust\_id project.customer.customer\_id%type,first\_name IN project.customer.FIRST\_NAME%TYPE,

last\_name IN project.customer.LAST\_NAME%TYPE,middle\_initial IN project.customer.MIDDLE\_INITIAL%TYPE,birth\_date IN project.customer.BIRTH\_DATE%TYPE,

cell\_phone IN project.customer.CELL\_PHONE%TYPE,street\_address IN project.customer.STREET\_ADDRESS%TYPE,city IN project.customer.CITY%TYPE,

state1 IN project.customer.STATE%TYPE,zip\_code IN project.customer.ZIP\_CODE%TYPE,email IN project.customer.EMAIL%TYPE,

license\_issued\_state IN project.customer.LICENSE\_ISSUED\_STATE%TYPE,license\_number IN project.customer.LICENSE\_NUMBER%TYPE,

credit\_card\_number IN project.customer.CREDIT\_CARD\_NUMBER%TYPE,expiration\_date IN project.customer.EXPIRATION\_DATE%TYPE,

ssn1 in project.customer.ssn%type);

procedure info\_update\_carowner(existing\_car\_id project.car\_client\_list.car\_id%type,client\_id project.car\_client\_list.client\_id%type,

available\_hrs project.car\_client\_list.available\_hrs%type,start\_date project.car\_client\_list.start\_date%type,

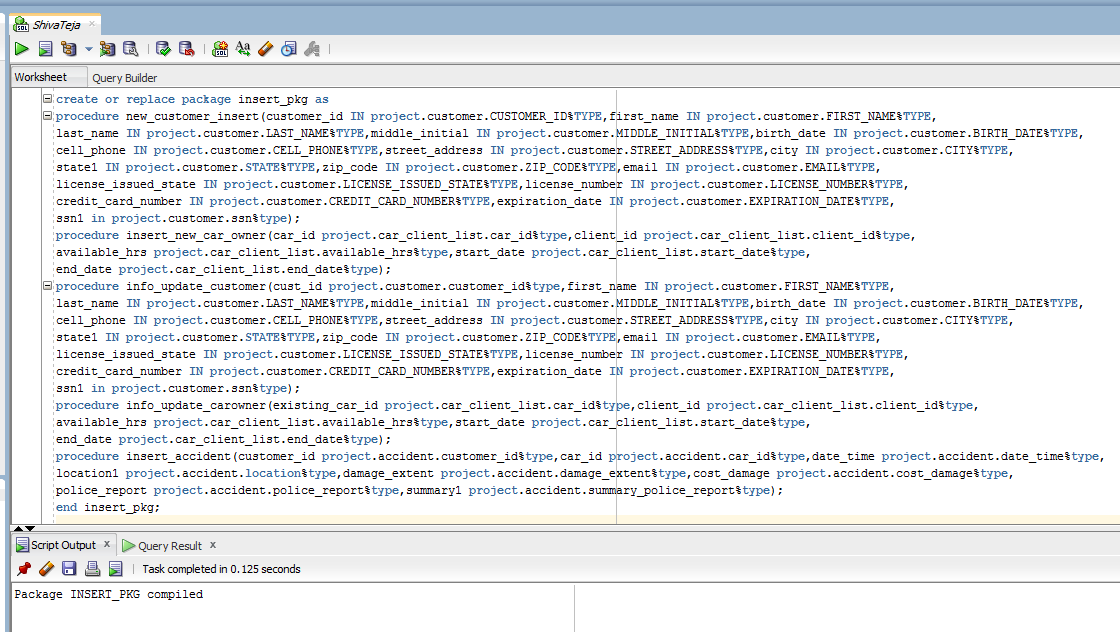
end\_date project.car\_client\_list.end\_date%type);

procedure insert\_accident(customer\_id project.accident.customer\_id%type,car\_id project.accident.car\_id%type,date\_time project.accident.date\_time%type,

location1 project.accident.location%type,damage\_extent project.accident.damage\_extent%type,cost\_damage project.accident.cost\_damage%type,

police\_report project.accident.police\_report%type,summary1 project.accident.summary\_police\_report%type);

end insert\_pkg;



create or replace package body insert\_pkg as

procedure new\_customer\_insert(customer\_id IN project.customer.CUSTOMER\_ID%TYPE,first\_name IN project.customer.FIRST\_NAME%TYPE,

last\_name IN project.customer.LAST\_NAME%TYPE,middle\_initial IN project.customer.MIDDLE\_INITIAL%TYPE,birth\_date IN project.customer.BIRTH\_DATE%TYPE,

cell\_phone IN project.customer.CELL\_PHONE%TYPE,street\_address IN project.customer.STREET\_ADDRESS%TYPE,city IN project.customer.CITY%TYPE,

state1 IN project.customer.STATE%TYPE,zip\_code IN project.customer.ZIP\_CODE%TYPE,email IN project.customer.EMAIL%TYPE,

license\_issued\_state IN project.customer.LICENSE\_ISSUED\_STATE%TYPE,license\_number IN project.customer.LICENSE\_NUMBER%TYPE,

credit\_card\_number IN project.customer.CREDIT\_CARD\_NUMBER%TYPE,expiration\_date IN project.customer.EXPIRATION\_DATE%TYPE,

ssn1 in project.customer.ssn%type) as

begin

insert into customer\_temp(customer\_id,first\_name,last\_name,middle\_initial,birth\_date,cell\_phone,street\_address,city,state,

zip\_code,email,license\_issued\_state,license\_Number,credit\_card\_number,Expiration\_date,ssn)

values(customer\_id,first\_name,last\_name,middle\_initial,birth\_date,cell\_phone,street\_address,city,state1,zip\_code,email,

license\_issued\_state,license\_number,credit\_card\_number,expiration\_date,ssn1);

exception

when others then

dbms\_output.put\_line('your entered data cannot be inserted into table');

END;

procedure insert\_new\_car\_owner(car\_id project.car\_client\_list.car\_id%type,client\_id project.car\_client\_list.client\_id%type,

available\_hrs project.car\_client\_list.available\_hrs%type,start\_date project.car\_client\_list.start\_date%type,

end\_date project.car\_client\_list.end\_date%type) as

begin

insert into car\_client\_list\_temp(car\_id,client\_id,available\_hrs,start\_date,end\_date)

values(car\_id,client\_id,available\_hrs,start\_date,end\_date);

exception

when others then

dbms\_output.put\_line('error occured because enter data is invalid');

end;

procedure info\_update\_customer(cust\_id project.customer.customer\_id%type,first\_name IN project.customer.FIRST\_NAME%TYPE,

last\_name IN project.customer.LAST\_NAME%TYPE,middle\_initial IN project.customer.MIDDLE\_INITIAL%TYPE,birth\_date IN project.customer.BIRTH\_DATE%TYPE,

cell\_phone IN project.customer.CELL\_PHONE%TYPE,street\_address IN project.customer.STREET\_ADDRESS%TYPE,city IN project.customer.CITY%TYPE,

state1 IN project.customer.STATE%TYPE,zip\_code IN project.customer.ZIP\_CODE%TYPE,email IN project.customer.EMAIL%TYPE,

license\_issued\_state IN project.customer.LICENSE\_ISSUED\_STATE%TYPE,license\_number IN project.customer.LICENSE\_NUMBER%TYPE,

credit\_card\_number IN project.customer.CREDIT\_CARD\_NUMBER%TYPE,expiration\_date IN project.customer.EXPIRATION\_DATE%TYPE,

ssn1 in project.customer.ssn%type) as

begin

update customer\_temp set license\_issued\_state = license\_issued\_state where customer\_id = cust\_id;

exception

when no\_data\_found then

dbms\_output.put\_line('error occured due to entered value cust\_id is in correct');

when others then

dbms\_output.put\_line('incorrect data as entered');

end;

procedure info\_update\_carowner(existing\_car\_id project.car\_client\_list.car\_id%type,client\_id project.car\_client\_list.client\_id%type,

available\_hrs project.car\_client\_list.available\_hrs%type,start\_date project.car\_client\_list.start\_date%type,

end\_date project.car\_client\_list.end\_date%type)

as

begin

update car\_client\_list\_temp set start\_date = start\_date where car\_id = existing\_car\_id;

exception

when no\_data\_found then

dbms\_output.put\_line('error occured due to entered value cust\_id is in correct');

when others then

dbms\_output.put\_line('incorrect data is entered');

end;

procedure insert\_accident(customer\_id project.accident.customer\_id%type,car\_id project.accident.car\_id%type,date\_time project.accident.date\_time%type,

location1 project.accident.location%type,damage\_extent project.accident.damage\_extent%type,cost\_damage project.accident.cost\_damage%type,

police\_report project.accident.police\_report%type,summary1 project.accident.summary\_police\_report%type)

as

begin

insert into accident\_temp(customer\_id,car\_id,date\_time,location,damage\_extent,cost\_damage,police\_report,summary\_police\_report)

values (customer\_id,car\_id,date\_time,location1,damage\_extent,cost\_damage,police\_report,summary1);

exception

when no\_data\_found then

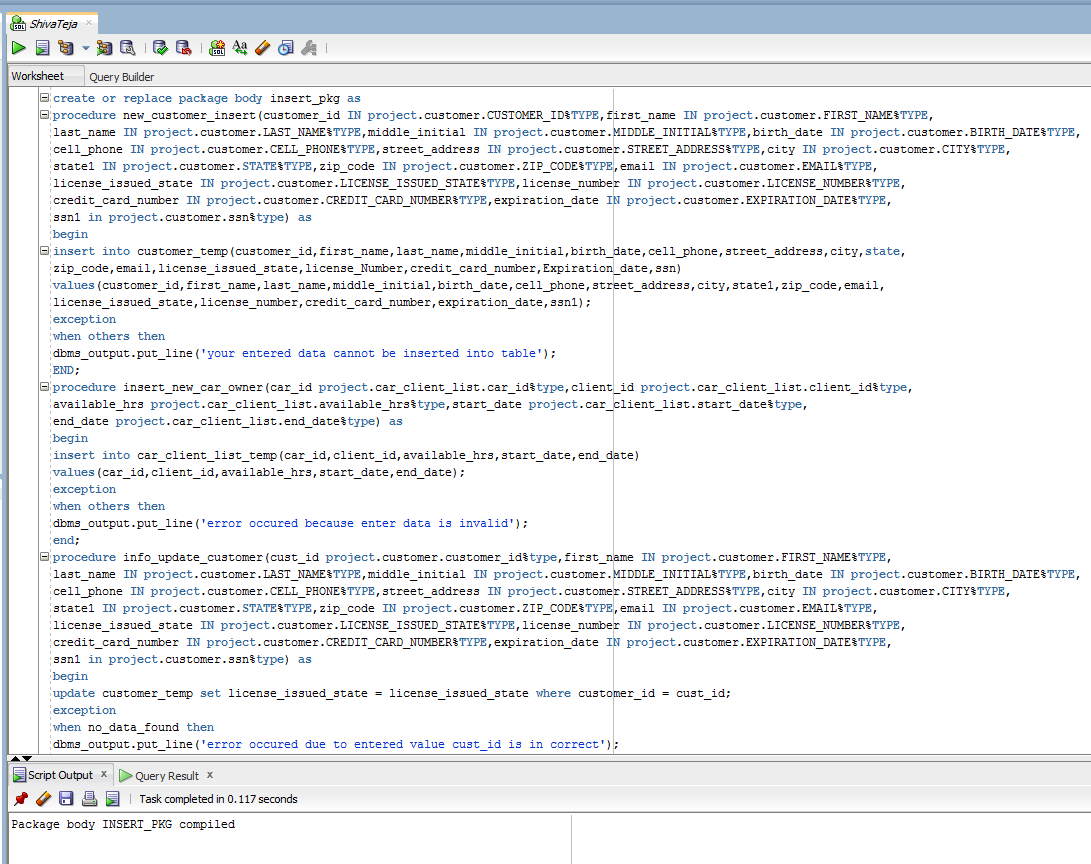
dbms\_output.put\_line('error occured due to entered value cust\_id is in correct');

when others then

dbms\_output.put\_line('error occured because enter data is invalid');

end;

end insert\_pkg;

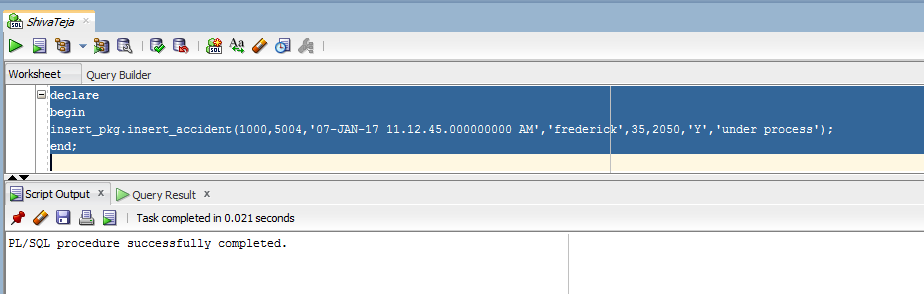


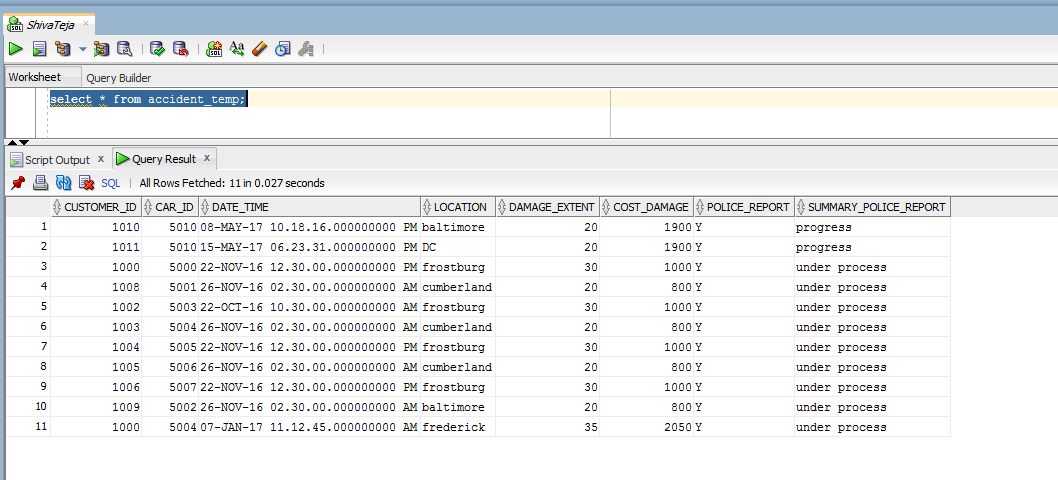
declare

begin

insert\_pkg.insert\_accident(1000,5004,'07-JAN-17 11.12.45.000000000 AM','frederick',35,2050,'Y','under process');

end;





**15.c Vehicle\_pkg**

create or replace package Vehicle\_pkg as

procedure car\_info(vehicle\_id in project.car.car\_id%type,Plate\_no out project.car.car\_plate\_no%type,state out project.car.car\_state\_registeration%type,

year out project.car.year%type,current\_mileage out project.car.current\_mileage%type,class out project.car.class%type, features out project.car.features%type,

make out project.car.make%type, color out project.car.color%type);

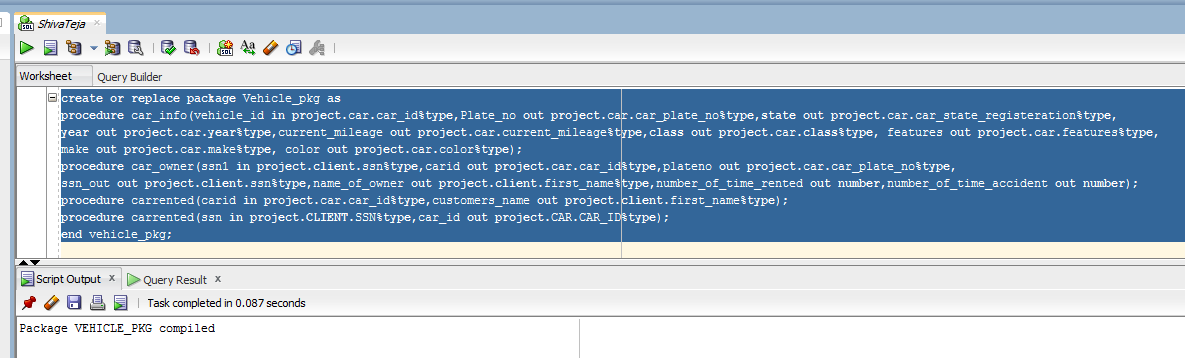
procedure car\_owner(ssn1 in project.client.ssn%type,carid out project.car.car\_id%type,plateno out project.car.car\_plate\_no%type,

ssn\_out out project.client.ssn%type,name\_of\_owner out project.client.first\_name%type,number\_of\_time\_rented out number,number\_of\_time\_accident out number);

procedure carrented(carid in project.car.car\_id%type,customers\_name out project.client.first\_name%type);

procedure carrented(ssn in project.CLIENT.SSN%type,car\_id out project.CAR.CAR\_ID%type);

end vehicle\_pkg;



create or replace package body Vehicle\_pkg as

procedure car\_info(vehicle\_id in project.car.car\_id%type,Plate\_no out project.car.car\_plate\_no%type,state out project.car.car\_state\_registeration%type,

year out project.car.year%type,current\_mileage out project.car.current\_mileage%type,class out project.car.class%type, features out project.car.features%type,

make out project.car.make%type, color out project.car.color%type) as

begin

select car\_plate\_no,car\_state\_registeration,year,current\_mileage,class,features,make,color into

plate\_no,state,year,current\_mileage,class,features,make,color from project.car where car\_id=vehicle\_ID;

exception

when no\_data\_found then

dbms\_output.put\_line('error: entered value car\_id is incorrect');

when others then

dbms\_output.put\_line('error: invalid data entered');

end;

procedure car\_owner(ssn1 in project.client.ssn%type,carid out project.car.car\_id%type,plateno out project.car.car\_plate\_no%type,

ssn\_out out project.client.ssn%type,name\_of\_owner out project.client.first\_name%type,number\_of\_time\_rented out number,number\_of\_time\_accident out number) as

begin

select ccl.car\_id,ca.car\_plate\_no,cl.ssn,cl.first\_name|| ' ' || cl.last\_name as full\_name into carid,plateno,ssn\_out,name\_of\_owner from project.client cl

join project.car\_client\_list ccl on cl.client\_id=ccl.client\_id join project.car ca on ca.car\_id=ccl.car\_id where cl.ssn=ssn1;

select count(\*) into number\_of\_time\_rented from project.car\_customer\_rent ccr join project.car ca on ccr.car\_id=ca.car\_id group by ca.car\_id having ca.car\_id=carid;

select count(\*) into number\_of\_time\_accident from project.car ca join project.accident pa on pa.car\_id=ca.car\_id group by ca.car\_id having ca.car\_id=carid;

exception

when no\_data\_found then

dbms\_output.put\_line('error: entered value owner\_ssn is incorrect');

when others then

dbms\_output.put\_line('error: invalid data entered');

end;

procedure carrented(carid in project.car.car\_id%type,customers\_name out project.client.first\_name%type) as

begin

select CST.first\_name|| ' ' || CST.last\_name as full\_name into customers\_name from project.car CAR,

project.car\_customer\_rent CCR, project.customer CST where CAR.car\_id=CCR.car\_id and CCR.customer\_id=CST.customer\_id and CAR.car\_id=carid;

exception

when no\_data\_found then

dbms\_output.put\_line('error: entered invalid car\_id ');

when others then

dbms\_output.put\_line('error: invalid data entered');

end;

procedure carrented(ssn in project.CLIENT.SSN%type,car\_id out project.CAR.CAR\_ID%type) as

begin

select CCL.car\_id into car\_id from project.client CLT,project.car\_client\_list CCL where CLT.client\_id=CCL.client\_id and CLT.ssn=ssn;

exception

when no\_data\_found then

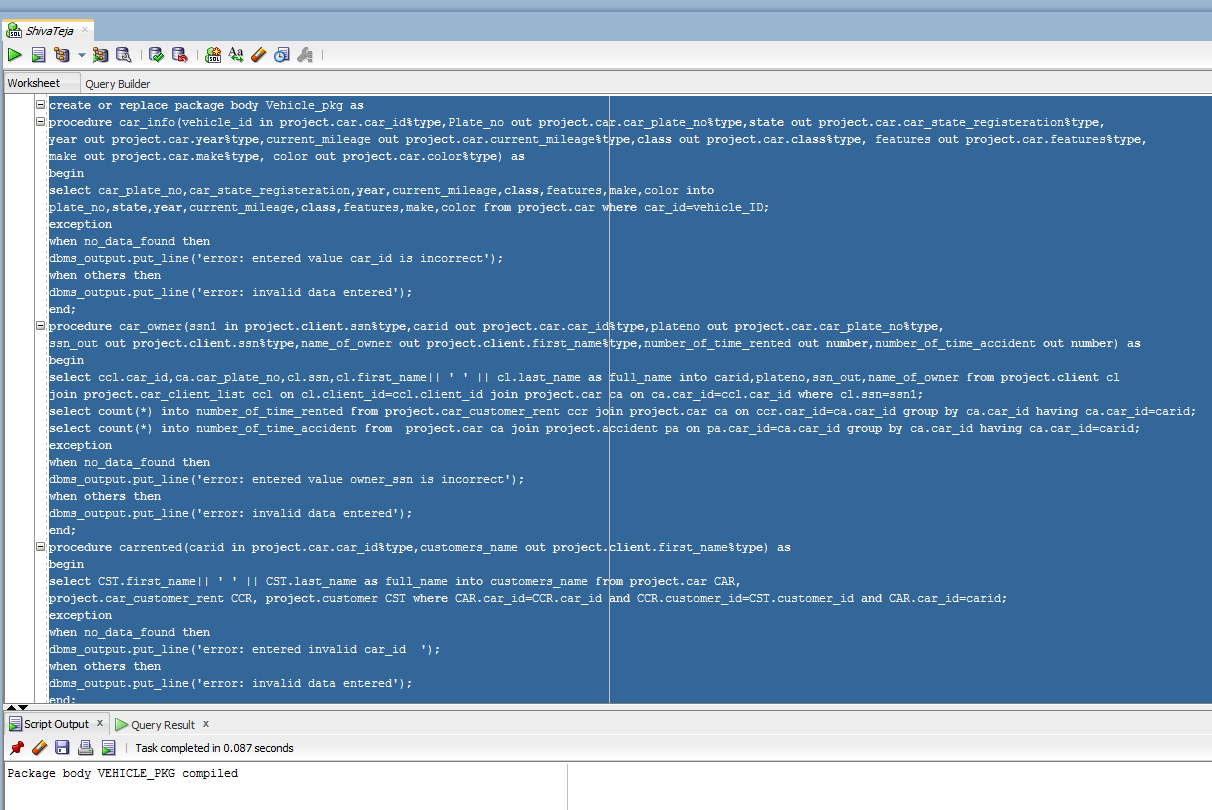
dbms\_output.put\_line('error: entered invalid SSN');

when others then

dbms\_output.put\_line('error: invalid data entered');

end;

end Vehicle\_pkg;



declare

carid project.car.car\_id%type;

plateno project.car.car\_state\_registeration%type;

ssn\_out project.client.ssn%type;

name\_of\_owner project.client.first\_name%type;

number\_of\_time\_rented number;

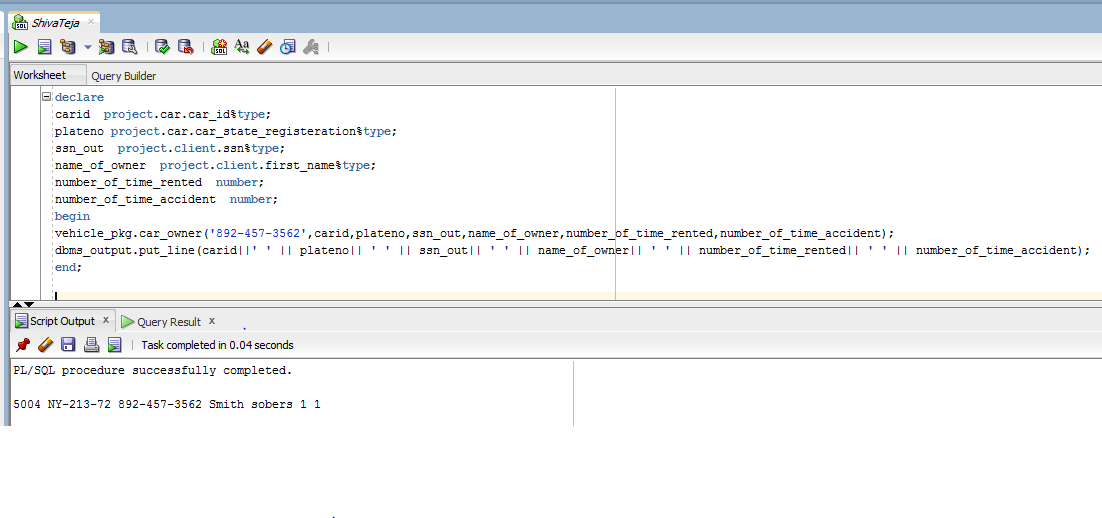
number\_of\_time\_accident number;

begin

vehicle\_pkg.car\_owner('892-457-3562',carid,plateno,ssn\_out,name\_of\_owner,number\_of\_time\_rented,number\_of\_time\_accident);

dbms\_output.put\_line(carid||' ' || plateno|| ' ' || ssn\_out|| ' ' || name\_of\_owner|| ' ' || number\_of\_time\_rented|| ' ' || number\_of\_time\_accident);

end;



**15.d accident\_pkg**

create or replace package accident\_pkg as

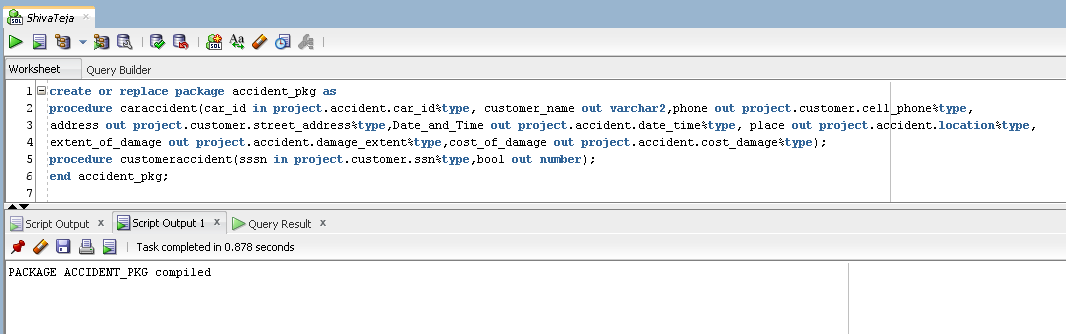
procedure caraccident(car\_id in project.accident.car\_id%type, customer\_name out varchar2,phone out project.customer.cell\_phone%type,

address out project.customer.street\_address%type,Date\_and\_Time out project.accident.date\_time%type, place out project.accident.location%type,

extent\_of\_damage out project.accident.damage\_extent%type,cost\_of\_damage out project.accident.cost\_damage%type);

procedure customeraccident(sssn in project.customer.ssn%type,bool out number);

end accident\_pkg;



create or replace package body accident\_pkg as

procedure caraccident(car\_ide in project.accident.car\_id%type, customer\_name out varchar2,phone out project.customer.cell\_phone%type,

address out project.customer.street\_address%type,Date\_and\_Time out project.accident.date\_time%type, place out project.accident.location%type,

extent\_of\_damage out project.accident.damage\_extent%type,cost\_of\_damage out project.accident.cost\_damage%type) is

cust\_id project.customer.customer\_id%type;

begin

select customer\_id,date\_time,location,damage\_extent,cost\_damage into cust\_id,date\_and\_time,place,extent\_of\_damage,cost\_of\_damage from project.accident where car\_id=car\_ide;

select first\_name||' '||last\_name,cell\_phone,street\_address into customer\_name,phone,address from project.customer where customer\_id=cust\_id;

end;

procedure customeraccident(sssn in project.customer.ssn%type,bool out number) is

cust\_id project.customer.customer\_id%type;

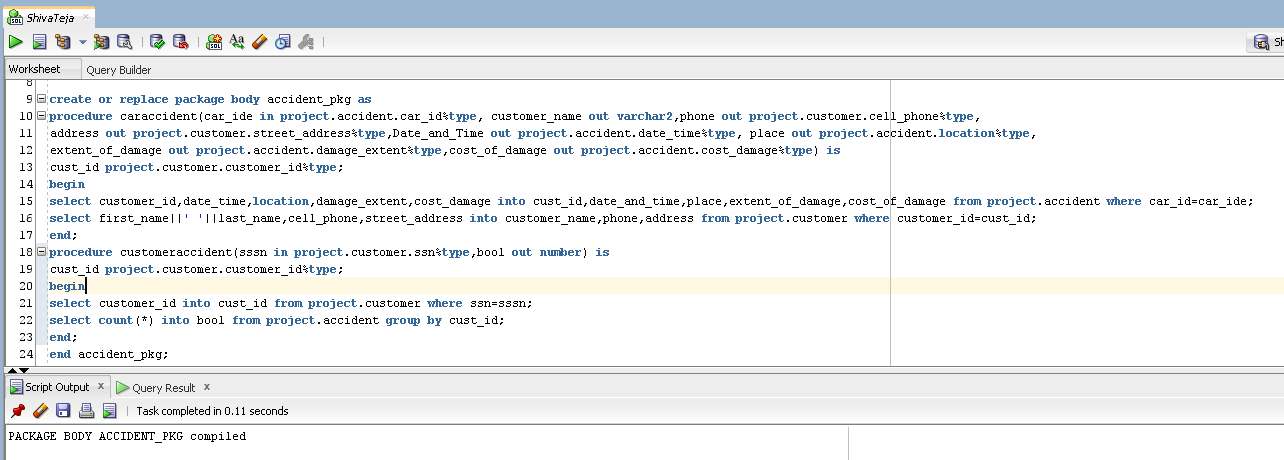
begin

select customer\_id into cust\_id from project.customer where ssn=sssn;

select count(\*) into bool from project.accident group by cust\_id;

end;

end accident\_pkg;



declare

customer\_name varchar2(100);phone project.customer.cell\_phone%type;

address project.customer.street\_address%type;Date\_and\_Time project.accident.date\_time%type; place project.accident.location%type;

extent\_of\_damage project.accident.damage\_extent%type;cost\_of\_damage project.accident.cost\_damage%type;bool number default 0;

begin

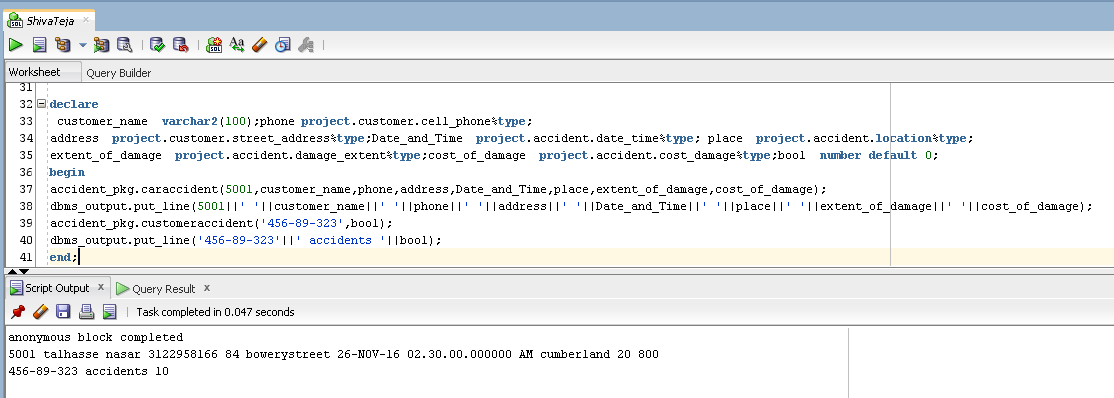
accident\_pkg.caraccident(5001,customer\_name,phone,address,Date\_and\_Time,place,extent\_of\_damage,cost\_of\_damage);

dbms\_output.put\_line(5001||' '||customer\_name||' '||phone||' '||address||' '||Date\_and\_Time||' '||place||' '||extent\_of\_damage||' '||cost\_of\_damage);

accident\_pkg.customeraccident('456-89-323',bool);

dbms\_output.put\_line('456-89-323'||' accidents '||bool);

end;



**15.e BirthdaySub:**

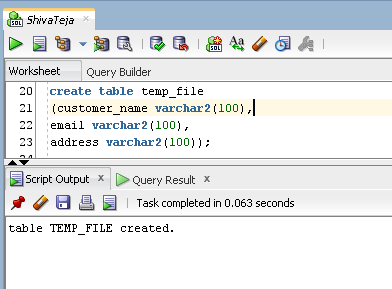
**Query:**

create table temp\_file

(customer\_name varchar2(100),

email varchar2(100),

address varchar2(100));



create or replace procedure birthdaysub(today in date,customer\_name out varchar2,email out project.customer.email%type,

address out project.customer.street\_address%type ) is

birthday project.customer.birth\_date%type;

cursor c\_temp is

select birth\_date,first\_name||' '||last\_name,email,street\_address from project.customer;

begin

open c\_temp;

loop fetch c\_temp into birthday,customer\_name,email,address;

if extract(month from birthday)=extract(month from today) then

if extract(day from birthday)-extract(day from today)<=15 then

execute immediate 'truncate table temp\_file';

insert into temp\_file values(customer\_name,email,address);

end if;

if extract(day from today)-extract(month from birthday)<=15 then

execute immediate 'truncate table temp\_file';

insert into temp\_file values(customer\_name,email,address);

end if;

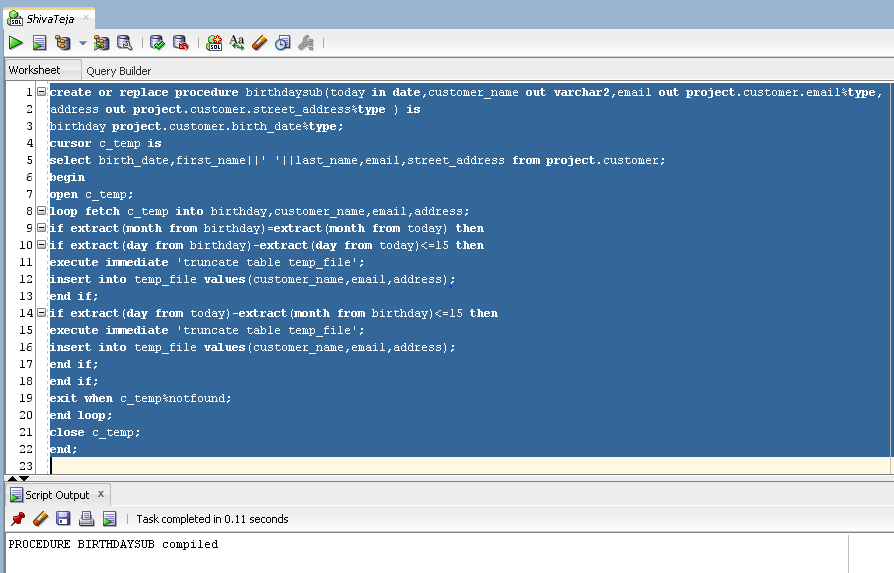
end if;

exit when c\_temp%notfound;

end loop;

close c\_temp;

end;



declare

today date default sysdate;

customer\_name varchar2(100);

email project.customer.email%type;

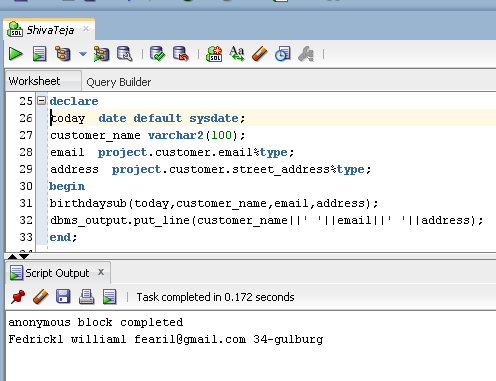
address project.customer.street\_address%type;

begin

birthdaysub(today,customer\_name,email,address);

dbms\_output.put\_line(customer\_name||' '||email||' '||address);

end;



**15.f**

**Query:**

create or replace procedure office\_info is

branch\_id project.office.office\_id%type;

branch\_name project.office.name%type;

manager\_name varchar2(100);

hired\_date project.office\_general\_manager.date\_hired%type;

emp\_id project.employee.employee\_id%type;

branch\_id1 project.office.office\_id%type;

cursor c\_temp is

select office\_id,name from project.office;

cursor c\_temp2 is

select general\_mgr\_id,office\_id,date\_hired from project.office\_general\_manager;

begin

open c\_temp;

loop fetch c\_temp into branch\_id,branch\_name;

open c\_temp2 ;

loop fetch c\_temp2 into emp\_id,branch\_id1,hired\_date;

if branch\_id1=branch\_id then

select first\_name||' '||last\_name into manager\_name from project.employee where employee\_id=emp\_id;

dbms\_output.put\_line(branch\_id||' '||branch\_name||' '||' '||manager\_name||' '||hired\_date);

end if;

exit when c\_temp2%notfound;

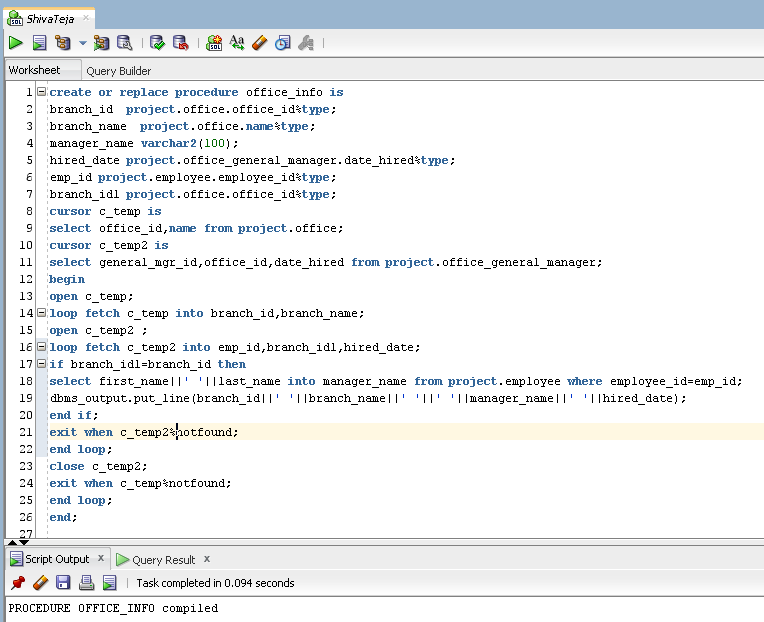
end loop;

close c\_temp2;

exit when c\_temp%notfound;

end loop;

end;

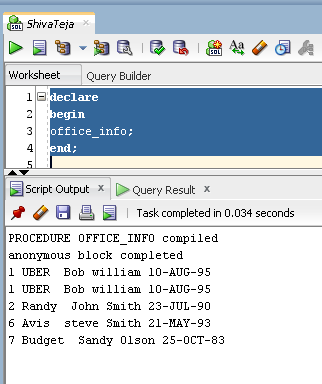


declare

begin

office\_info;

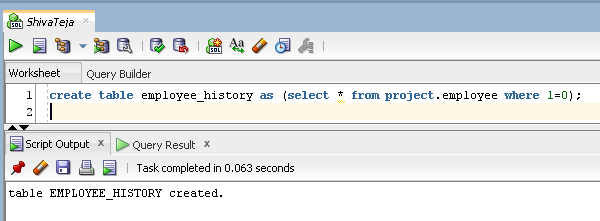
end;

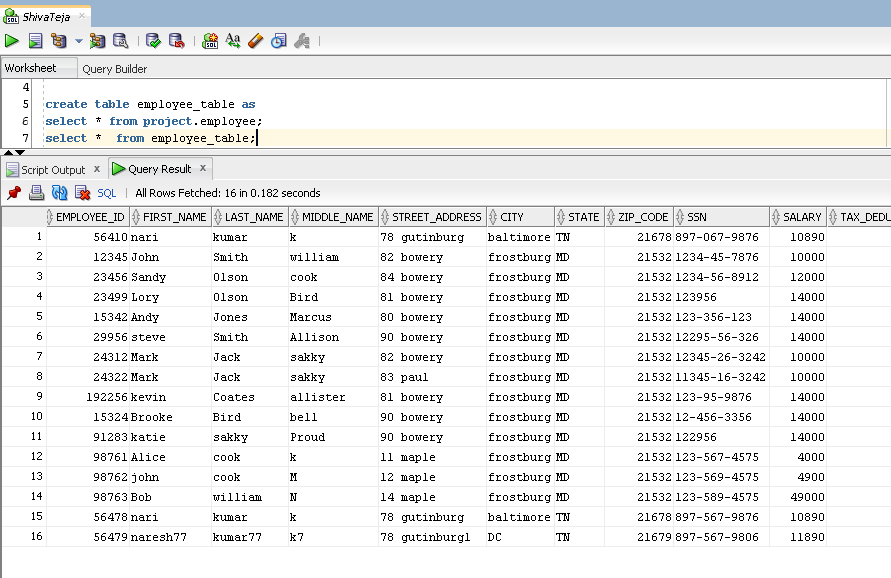


**16.g Employee History**

**Query:**

create table employee\_history as (select \* from project.employee where 1=0);





create or replace trigger before\_deletion

before delete on employee\_table for each row

declare

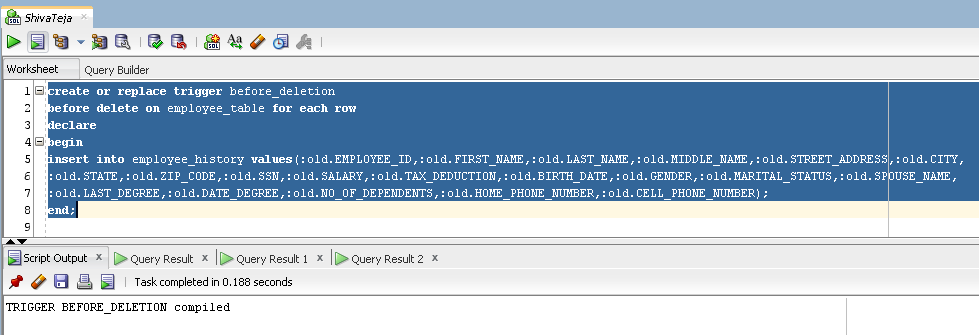
begin

insert into employee\_history values(:old.EMPLOYEE\_ID,:old.FIRST\_NAME,:old.LAST\_NAME,:old.MIDDLE\_NAME,:old.STREET\_ADDRESS,:old.CITY,

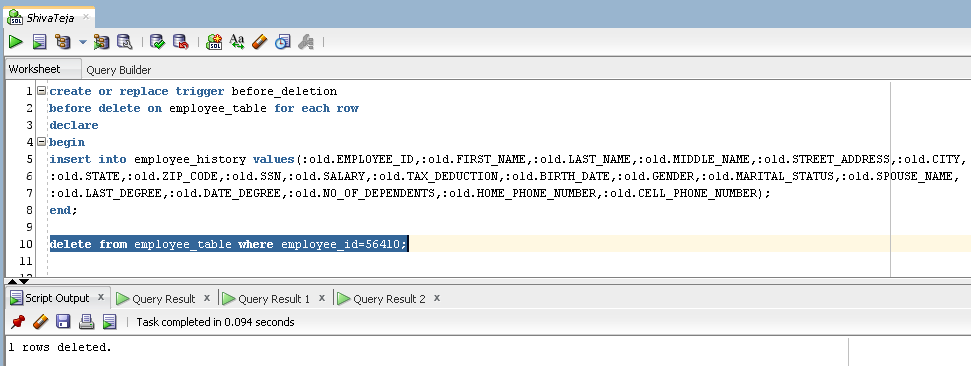
:old.STATE,:old.ZIP\_CODE,:old.SSN,:old.SALARY,:old.TAX\_DEDUCTION,:old.BIRTH\_DATE,:old.GENDER,:old.MARITAL\_STATUS,:old.SPOUSE\_NAME,

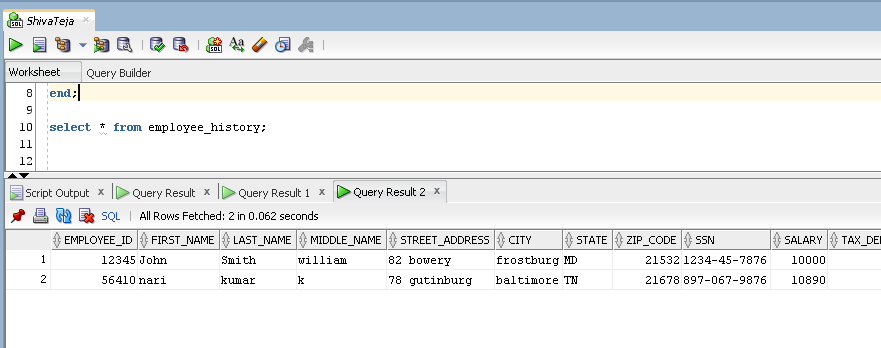
:old.LAST\_DEGREE,:old.DATE\_DEGREE,:old.NO\_OF\_DEPENDENTS,:old.HOME\_PHONE\_NUMBER,:old.CELL\_PHONE\_NUMBER);

end;



delete from employee\_table where employee\_id=56410;





**16.h modifications**

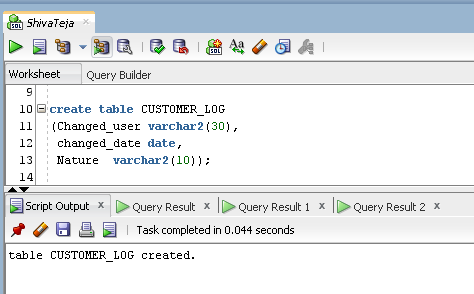
**Query:**

create table CUSTOMER\_LOG

(Changed\_user varchar2(30),

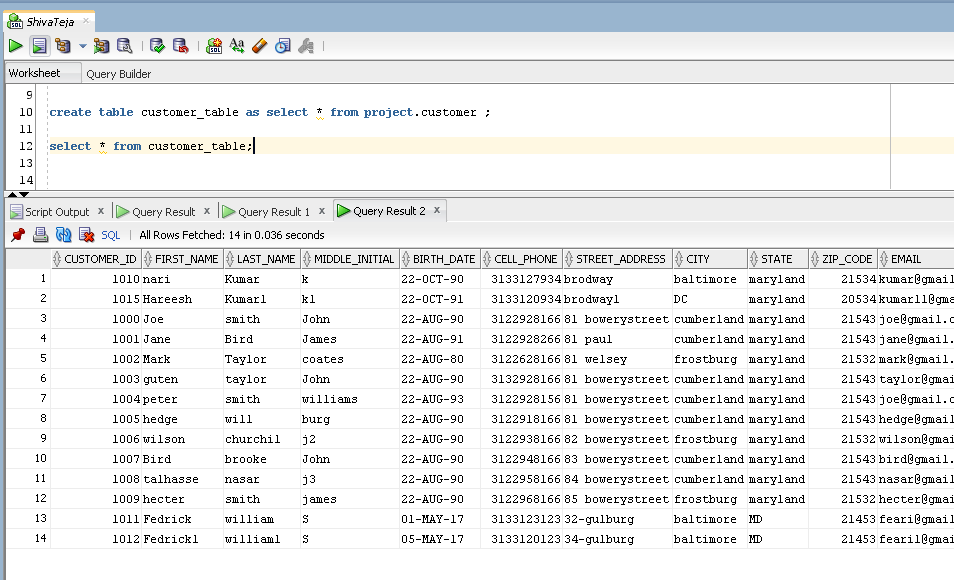
changed\_date date,

Nature varchar2(10));



create table customer\_table as select \* from project.customer ;

select \* from customer\_table;



create or replace trigger modification

before update or delete or insert on customer\_table for each row

begin

if inserting then

insert into customer\_log values(user,sysdate,'I');

elsif deleting then

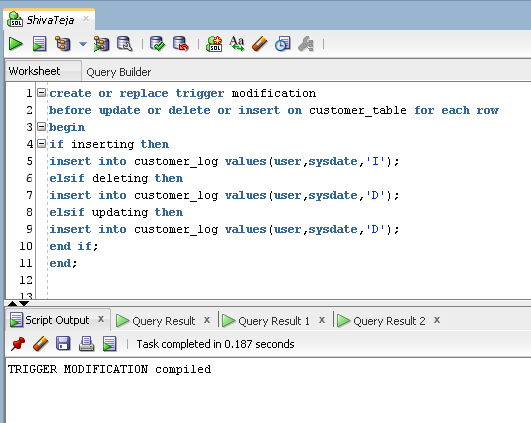
insert into customer\_log values(user,sysdate,'D');

elsif updating then

insert into customer\_log values(user,sysdate,'D');

end if;

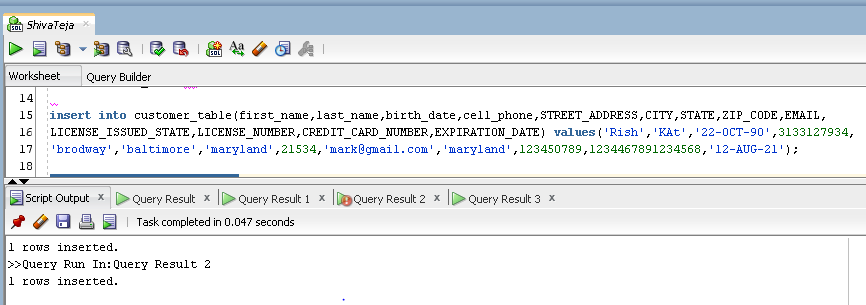
end;



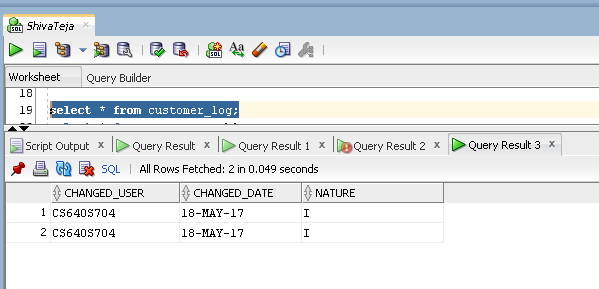
insert into customer\_table(first\_name,last\_name,birth\_date,cell\_phone,STREET\_ADDRESS,CITY,STATE,ZIP\_CODE,EMAIL,

LICENSE\_ISSUED\_STATE,LICENSE\_NUMBER,CREDIT\_CARD\_NUMBER,EXPIRATION\_DATE) values('Rish','KAt','22-OCT-90',3133127934,

'brodway','baltimore','maryland',21534,'mark@gmail.com','maryland',123450789,1234467891234568,'12-AUG-21');

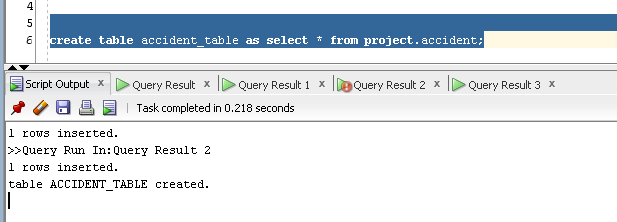


select \* from customer\_log;



**16.i**

create table accident\_table as select \* from project.accident;

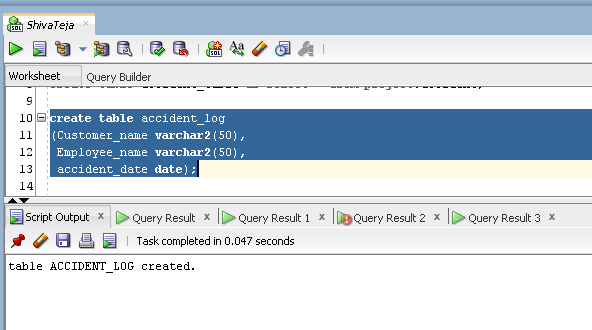


create table accident\_log

(Customer\_name varchar2(50),

Employee\_name varchar2(50),

accident\_date date);



create or replace trigger accident\_trigger

after insert on accident\_table for each row

declare

customer\_name varchar2(50);

Employee\_name varchar2(50);

emp\_id project.employee.employee\_id%type;

begin

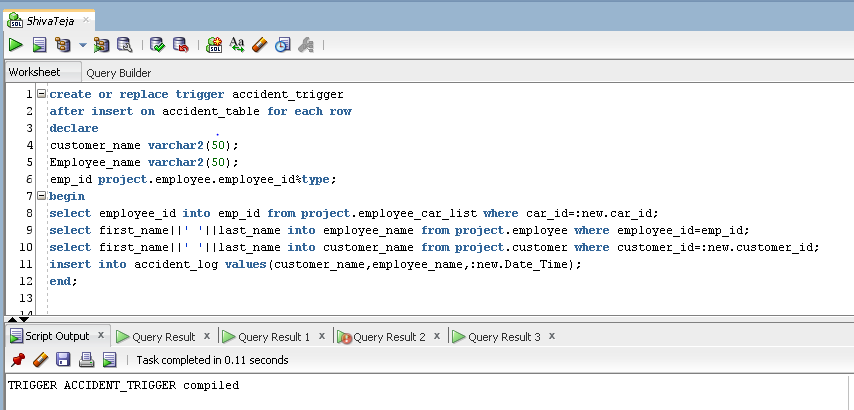
select employee\_id into emp\_id from project.employee\_car\_list where car\_id=:new.car\_id;

select first\_name||' '||last\_name into employee\_name from project.employee where employee\_id=emp\_id;

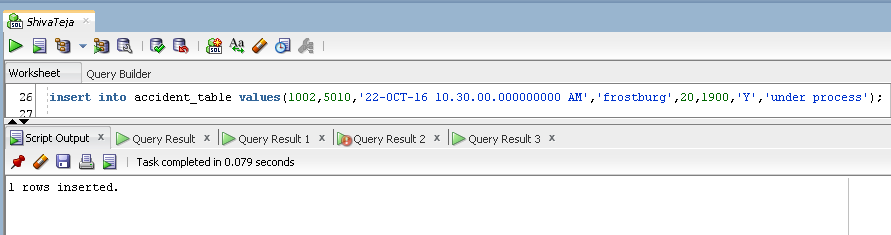
select first\_name||' '||last\_name into customer\_name from project.customer where customer\_id=:new.customer\_id;

insert into accident\_log values(customer\_name,employee\_name,:new.Date\_Time);

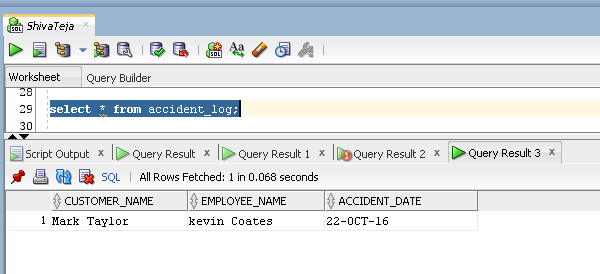
end;



insert into accident\_table values(1002,5010,'22-OCT-16 10.30.00.000000000 AM','frostburg',20,1900,'Y','under process');



select \* from accident\_log;



**17**

create table error\_log

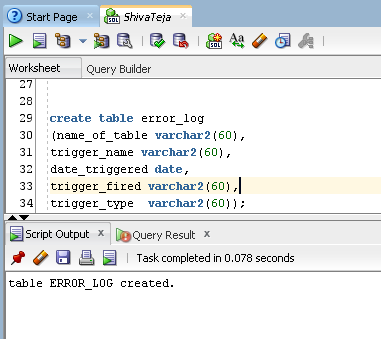
(name\_of\_table varchar2(60),

trigger\_name varchar2(60),

date\_triggered date,

trigger\_fired varchar2(60),

trigger\_type varchar2(60));



create or replace trigger before\_deletion

before delete on employee\_table for each row

declare

begin

insert into employee\_history values(:old.EMPLOYEE\_ID,:old.FIRST\_NAME,:new.LAST\_NAME,:old.MIDDLE\_NAME,:old.STREET\_ADDRESS,:old.CITY,

:old.STATE,:old.ZIP\_CODE,:old.SSN,:old.SALARY,:old.TAX\_DEDUCTION,:old.BIRTH\_DATE,:old.GENDER,:old.MARITAL\_STATUS,:old.SPOUSE\_NAME,

:old.LAST\_DEGREE,:old.DATE\_DEGREE,:old.NO\_OF\_DEPENDENTS,:old.HOME\_PHONE\_NUMBER,:old.CELL\_PHONE\_NUMBER);

exception

when others then

declare

tri\_status user\_triggers.status%type;

tri\_type user\_triggers.trigger\_type%type;

begin

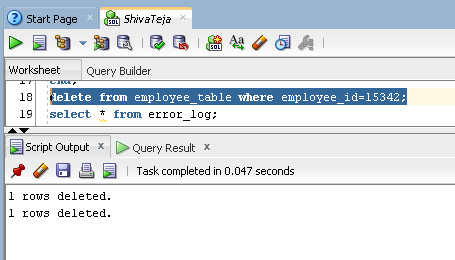
select trigger\_type,status into tri\_type,tri\_status from user\_triggers where trigger\_name like '%before\_deletion%';

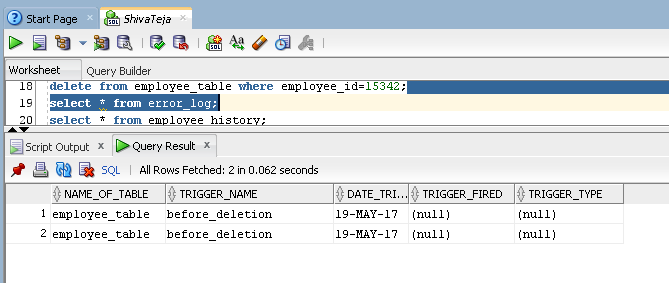
insert into error\_log values('employee\_table','before\_deletion',sysdate,tri\_status,tri\_type);

end;

end;







create or replace trigger modification

before update or delete or insert on customer\_table for each row

declare

err exception;

tri\_status varchar2(100);

tri\_type varchar2(100);

begin

select trigger\_type,status into tri\_type,tri\_status from user\_triggers where trigger\_name like '%MODIFICATION%';

if inserting then

insert into customer\_log values(user,sysdate,'jdbkjsdblk');

raise err;

elsif deleting then

insert into customer\_log values(user,sysdate,'D');

elsif updating then

insert into customer\_log values(user,sysdate,'U');

end if;

exception

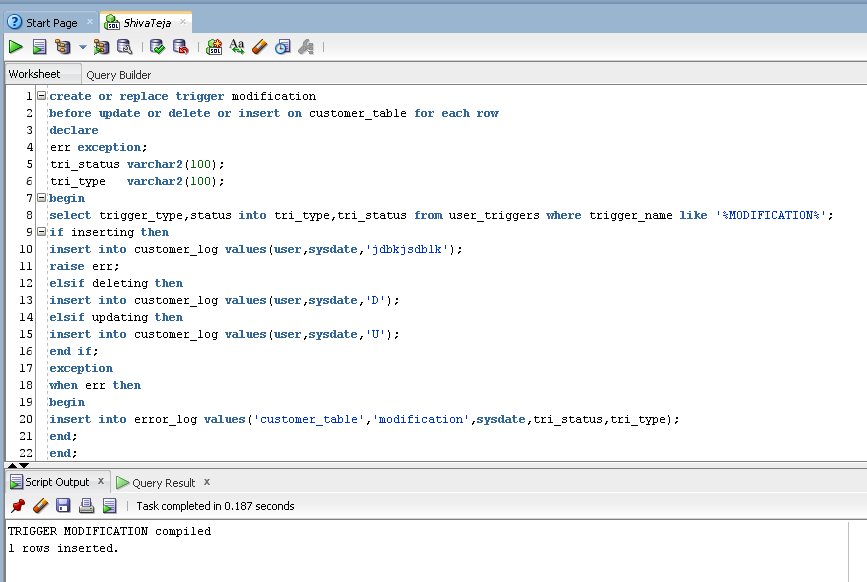
when err then

begin

insert into error\_log values('customer\_table','modification',sysdate,tri\_status,tri\_type);

end;

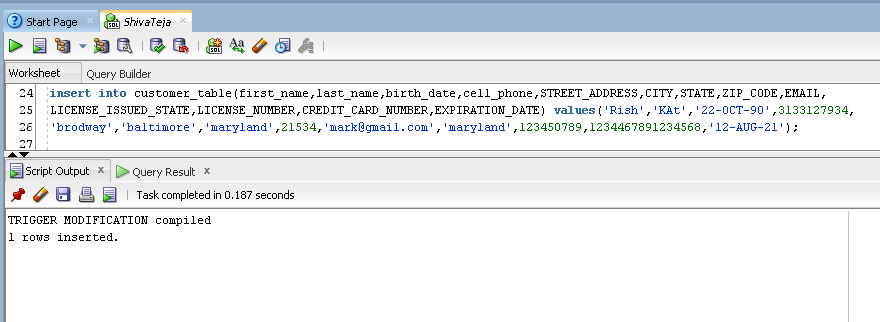
end;



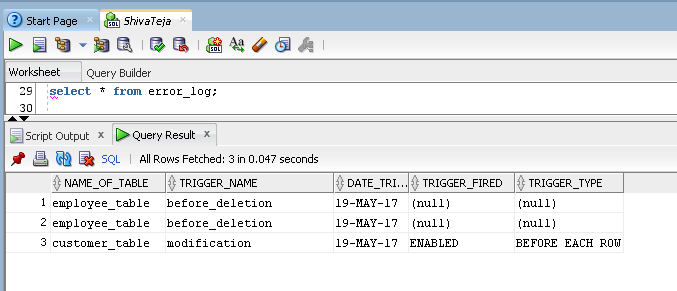
insert into customer\_table(first\_name,last\_name,birth\_date,cell\_phone,STREET\_ADDRESS,CITY,STATE,ZIP\_CODE,EMAIL,

LICENSE\_ISSUED\_STATE,LICENSE\_NUMBER,CREDIT\_CARD\_NUMBER,EXPIRATION\_DATE) values('Rish','KAt','22-OCT-90',3133127934,

'brodway','baltimore','maryland',21534,'mark@gmail.com','maryland',123450789,1234467891234568,'12-AUG-21');



select \* from error\_log;



create or replace trigger accident\_trigger

after insert on accident\_table for each row

declare

customer\_name varchar2(50);

Employee\_name varchar2(50);

tri\_status varchar2(100);

tri\_type varchar2(100);

emp\_id project.employee.employee\_id%type;

begin

select trigger\_type,status into tri\_type,tri\_status from user\_triggers where trigger\_name like '%ACCIDENT\_TRIGGER%';

select employee\_id into emp\_id from project.employee\_car\_list where car\_id=:old.car\_id;

select first\_name||' '||last\_name into employee\_name from project.employee where employee\_id=emp\_id;

select first\_name||' '||last\_name into customer\_name from project.customer where customer\_id=:new.customer\_id;

insert into accident\_log values(customer\_name,employee\_name,:new.Date\_Time);

exception

when others then

begin

insert into error\_log values('accident\_table','ACCIDENT\_TRIGGER',sysdate,tri\_status,tri\_type);

end;

end;

