**Project Report on designing a Database for Car Rental Company**

(This report is continuation of project report submitted in Dec 2016)

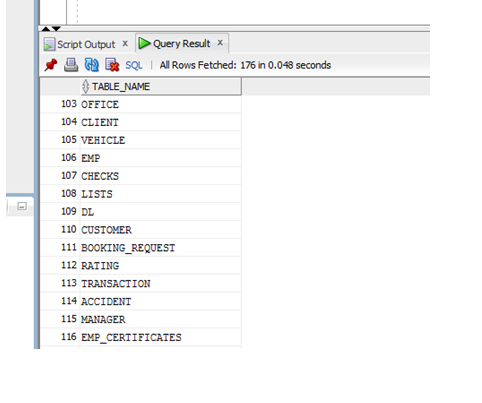
**Submitted By :**

**Srikanth Mandula**

**FSU ID: 3080786**

**Frostburg State University**

Below are the Tables considered for Project development. These tables are developed as part of DBMS-1 project.



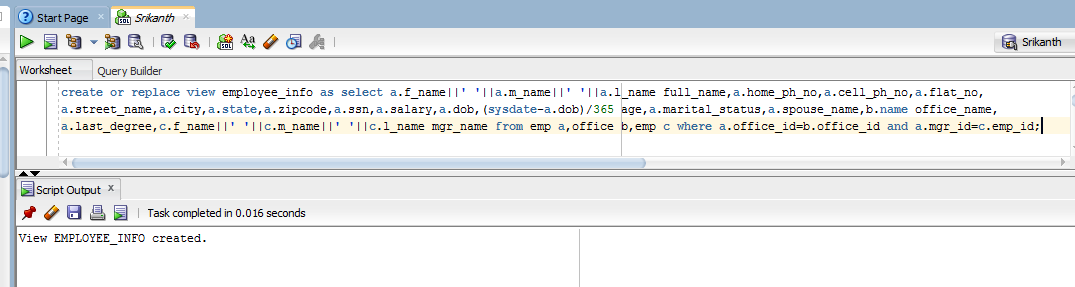
**12 a) Creating *Employee\_Info* view**

**Query:**

create or replace view employee\_info as select a.f\_name||' '||a.m\_name||' '||a.l\_name full\_name,a.home\_ph\_no,a.cell\_ph\_no,a.flat\_no,a.street\_name,a.city,a.state,a.zipcode,a.ssn,a.salary,a.dob,(sysdate-a.dob)/365 age,a.marital\_status,a.spouse\_name,b.name office\_name,

a.last\_degree,c.f\_name||' '||c.m\_name||' '||c.l\_name mgr\_name from emp a,office b,emp c where a.office\_id=b.office\_id and a.mgr\_id=c.emp\_id;

**Output:**



**12 b) Creating *Employee\_Salary* view**

**Query:**

create or replace view employee\_salary as select a.f\_name,a.l\_name ,a.flat\_no,

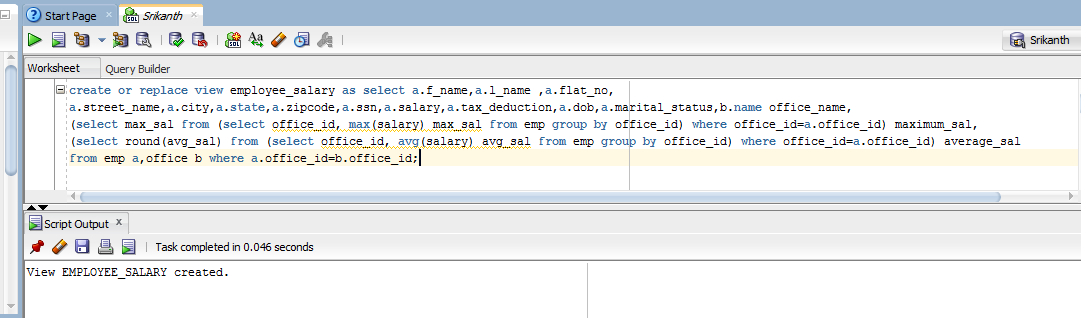
a.street\_name,a.city,a.state,a.zipcode,a.ssn,a.salary,a.tax\_deduction,a.dob,a.marital\_status,b.name office\_name,

(select max\_sal from (select office\_id, max(salary) max\_sal from emp group by office\_id) where office\_id=a.office\_id) maximum\_sal,

(select round(avg\_sal) from (select office\_id, avg(salary) avg\_sal from emp group by office\_id) where office\_id=a.office\_id) average\_sal

from emp a,office b where a.office\_id=b.office\_id;

**Output:**



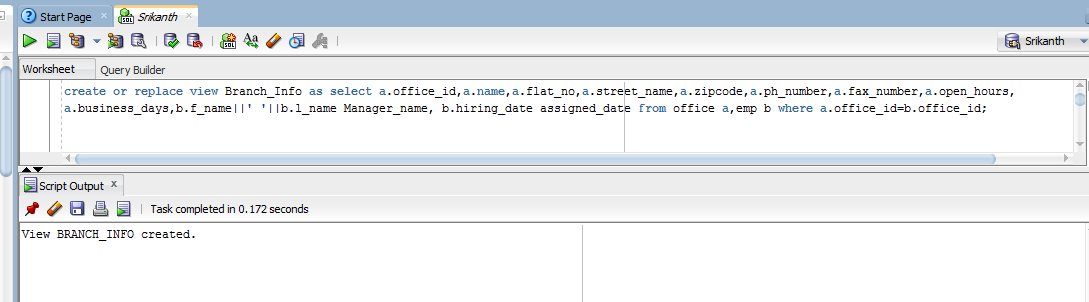
**12 c) Creating *Branch\_Info* view**

**Query:**

create or replace view Branch\_Info as select a.office\_id,a.name,a.flat\_no,a.street\_name,a.zipcode,a.ph\_number,a.fax\_number,a.open\_hours,

a.business\_days,b.f\_name||' '||b.l\_name Manager\_name, b.hiring\_date assigned\_date from office a,emp b where a.office\_id=b.office\_id;

**Output:**



**12 d ) Creating *Valued\_Customers* view**

**Query:**

create or replace view Valued\_Customers as select a.f\_name||' '||a.m\_name||' '||a.l\_name Name,round((sysdate-a.birth\_date)/365) age,

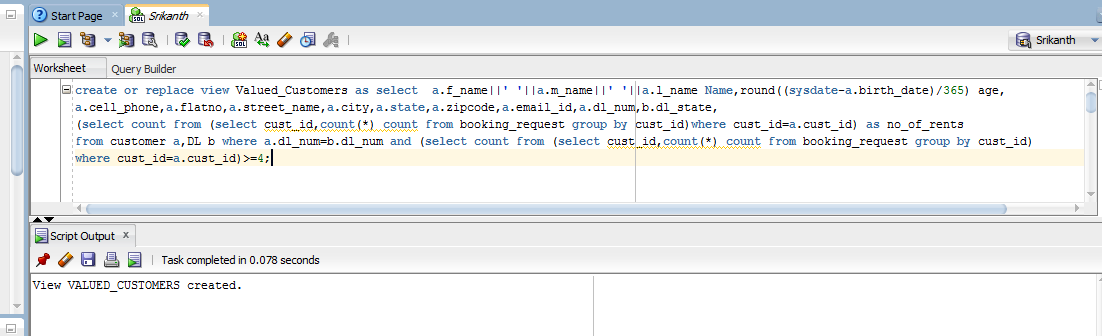
a.cell\_phone,a.flatno,a.street\_name,a.city,a.state,a.zipcode,a.email\_id,a.dl\_num,b.dl\_state,

(select count from (select cust\_id,count(\*) count from booking\_request group by cust\_id)where cust\_id=a.cust\_id) as no\_of\_rents

from customer a,DL b where a.dl\_num=b.dl\_num and (select count from (select cust\_id,count(\*) count from booking\_request group by cust\_id)

where cust\_id=a.cust\_id)>=4;

**Output:**



**12 e ) Creating *List\_Of\_Cars* view**

**Query:**

create or replace view List\_Of\_Cars as

select a.car\_id,a.year,b.f\_name||' '||b.l\_name owner\_name,a.class,

(select count from (select car\_id,count(\*) count from booking\_request group by car\_id) where car\_id=a.car\_id) no\_of\_times\_car\_rented,

a.make,a.color,

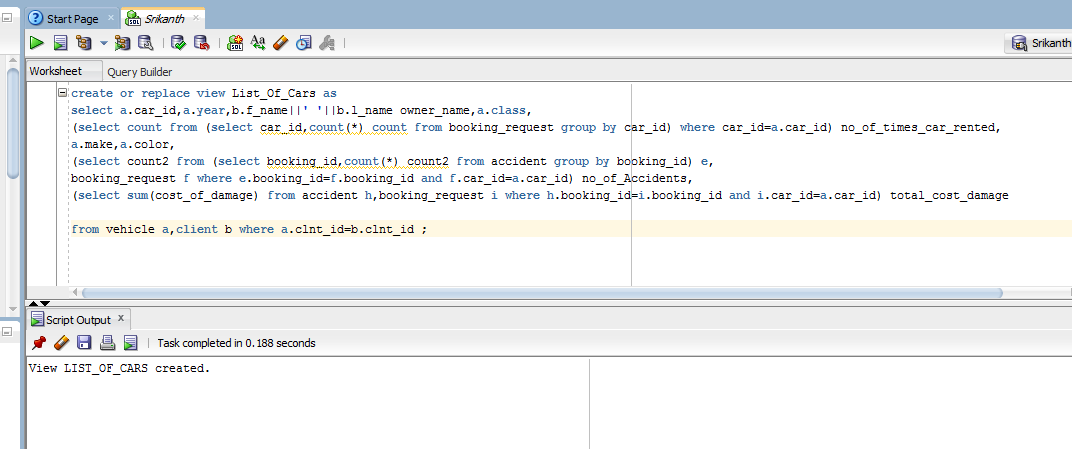
(select count2 from (select booking\_id,count(\*) count2 from accident group by booking\_id) e,

booking\_request f where e.booking\_id=f.booking\_id and f.car\_id=a.car\_id) no\_of\_Accidents,

(select sum(cost\_of\_damage) from accident h,booking\_request i where h.booking\_id=i.booking\_id and i.car\_id=a.car\_id) total\_cost\_damage

from vehicle a,client b where a.clnt\_id=b.clnt\_id ;

**Output:**



**12 f ) Creating *List\_Of\_Transactions* view**

**Query:**

create or replace view List\_Of\_Transaction as

select a.tran\_id,(b.vehicle\_req\_upto -b.req\_date) no\_of\_days,a.amount total\_cost\_of\_transaction,

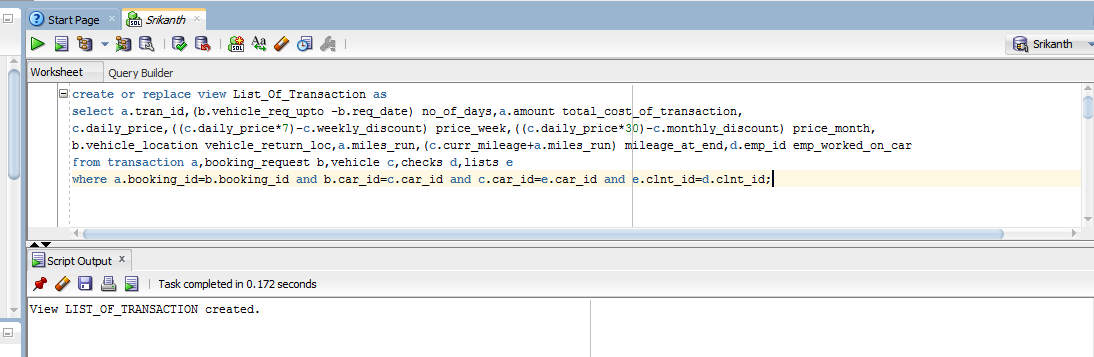
c.daily\_price,((c.daily\_price\*7)-c.weekly\_discount) price\_week,((c.daily\_price\*30)-c.monthly\_discount) price\_month,

b.vehicle\_location vehicle\_return\_loc,a.miles\_run,(c.curr\_mileage+a.miles\_run) mileage\_at\_end,d.emp\_id emp\_worked\_on\_car

from transaction a,booking\_request b,vehicle c,checks d,lists e

where a.booking\_id=b.booking\_id and b.car\_id=c.car\_id and c.car\_id=e.car\_id and e.clnt\_id=d.clnt\_id;

**Output:**



**12 g ) Creating *Statistics\_by\_Branch\_Year* view**

**Query:** create or replace view Statistics\_by\_Branch\_year as

select distinct a.office\_id,b.f\_name||' '||b.l\_name client\_name,to\_char(c.registered\_date,'yyyy') year,

(select count1 from (select clnt\_id,to\_char(registered\_date,'yyyy') year\_temp,count(\*) count1 from lists

group by clnt\_id,to\_char(registered\_date,'yyyy')) where clnt\_id=b.clnt\_id and year\_temp=to\_char(c.registered\_date,'yyyy')) count\_of\_cars,

(select count2 from (select distinct d.clnt\_id,to\_char(e.req\_date,'yyyy') req\_date,count(\*) count2 from vehicle d,booking\_request e

where e.car\_id=d.car\_id group by d.clnt\_id,to\_char(e.req\_date,'yyyy')) where clnt\_id=b.clnt\_id and req\_date=to\_char(c.registered\_date,'yyyy')) no\_of\_trans,

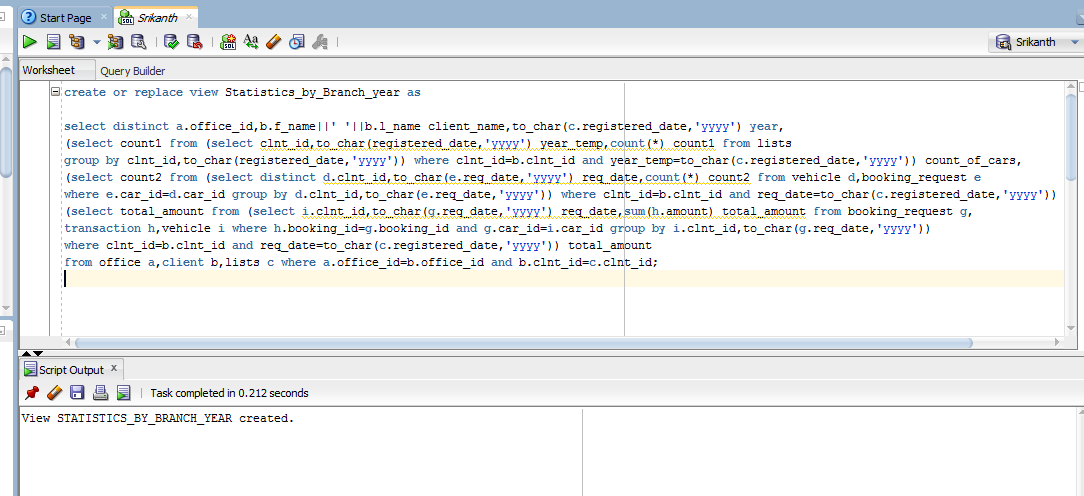
(select total\_amount from (select i.clnt\_id,to\_char(g.req\_date,'yyyy') req\_date,sum(h.amount) total\_amount from booking\_request g,

transaction h,vehicle i where h.booking\_id=g.booking\_id and g.car\_id=i.car\_id group by i.clnt\_id,to\_char(g.req\_date,'yyyy'))

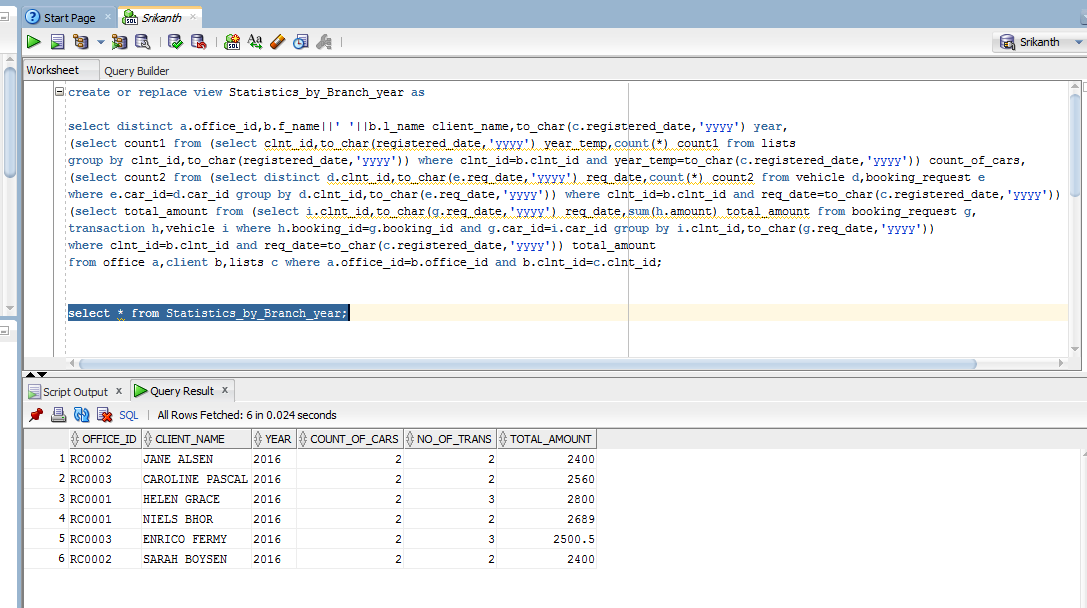
where clnt\_id=b.clnt\_id and req\_date=to\_char(c.registered\_date,'yyyy')) total\_amount

from office a,client b,lists c where a.office\_id=b.office\_id and b.clnt\_id=c.clnt\_id;

**Output :**



**Output :**



**12 h ) Creating *Last\_employee\_Branch* view**

**Query:**

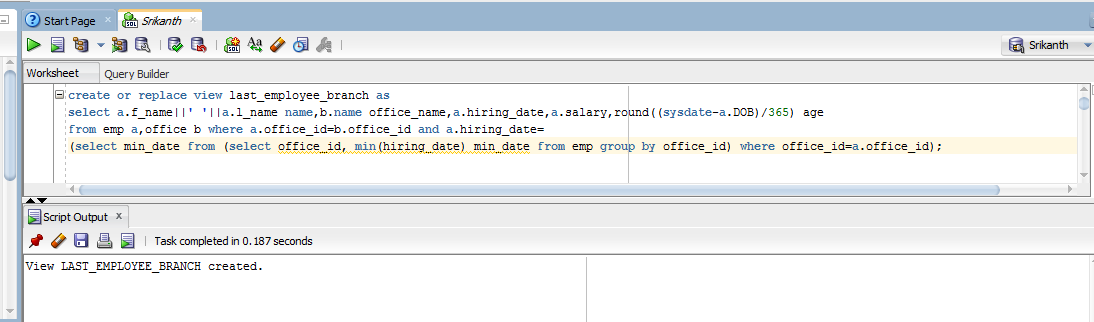
create or replace view last\_employee\_branch as

select a.f\_name||' '||a.l\_name name,b.name office\_name,a.hiring\_date,a.salary,round((sysdate-a.DOB)/365) age

from emp a,office b where a.office\_id=b.office\_id and a.hiring\_date=

(select min\_date from (select office\_id, min(hiring\_date) min\_date from emp group by office\_id) where office\_id=a.office\_id);

**Output:**



**12 i ) Creating *Managers* view**

**Query:**

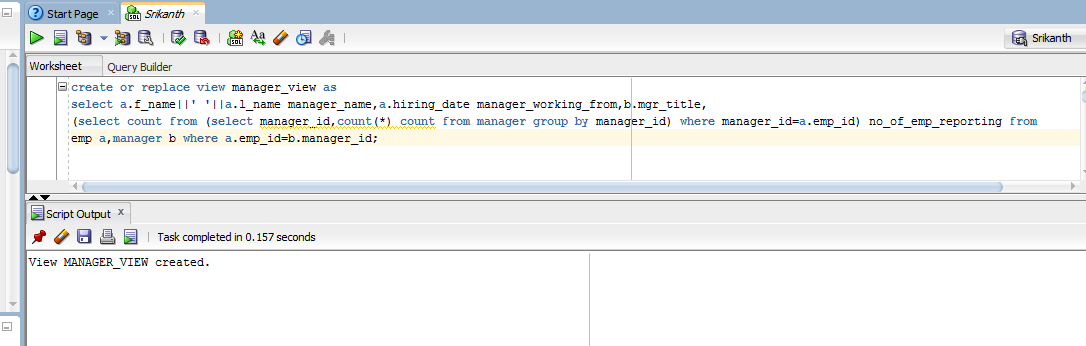
create or replace view manager\_view as

select a.f\_name||' '||a.l\_name manager\_name,a.hiring\_date manager\_working\_from,b.mgr\_title,

(select count from (select manager\_id,count(\*) count from manager group by manager\_id) where manager\_id=a.emp\_id) no\_of\_emp\_reporting from

emp a,manager b where a.emp\_id=b.manager\_id;

**Output:**



**12 j)**

Creating a view to list the number of bookings a customer has done at his registered office.

**Query:**

create or replace view cust\_bookings\_at\_home\_office as

select a.cust\_id,a.f\_name||' '||a.l\_name cust\_name,a.office\_id cust\_registered\_office,

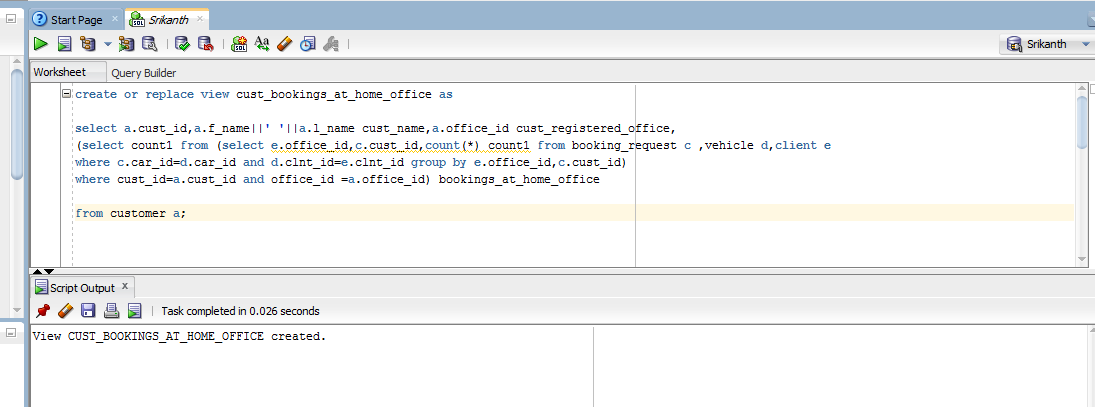
(select count1 from (select e.office\_id,c.cust\_id,count(\*) count1 from booking\_request c ,vehicle d,client e

where c.car\_id=d.car\_id and d.clnt\_id=e.clnt\_id group by e.office\_id,c.cust\_id)

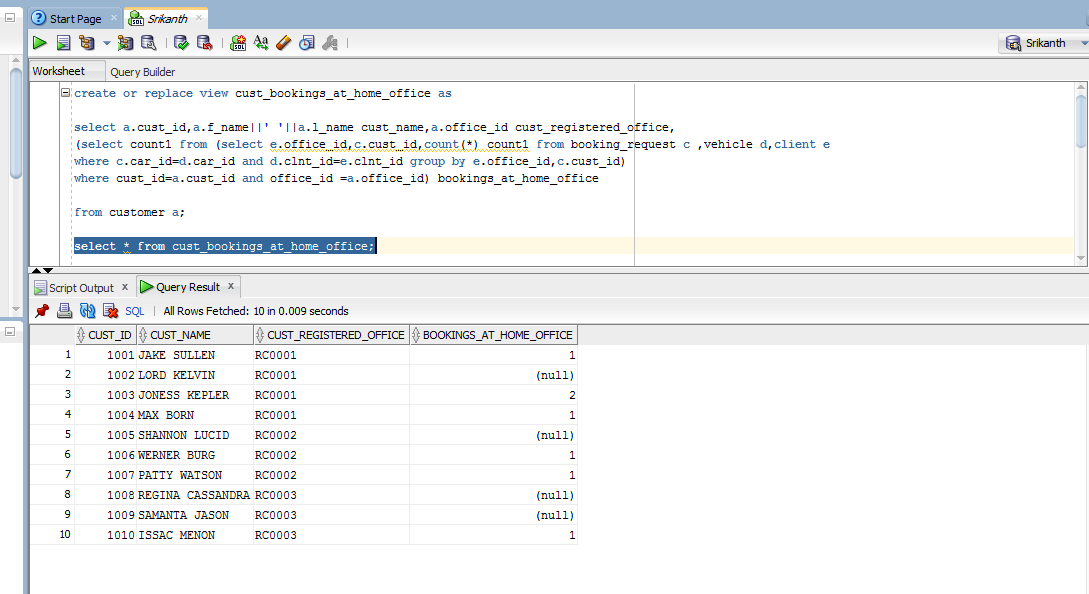
where cust\_id=a.cust\_id and office\_id =a.office\_id) bookings\_at\_home\_office

from customer a;

**Output:**



**Output:**



**12 k)**

creating a view to list top 3 frequent visiting customers at any office.

**Query:**

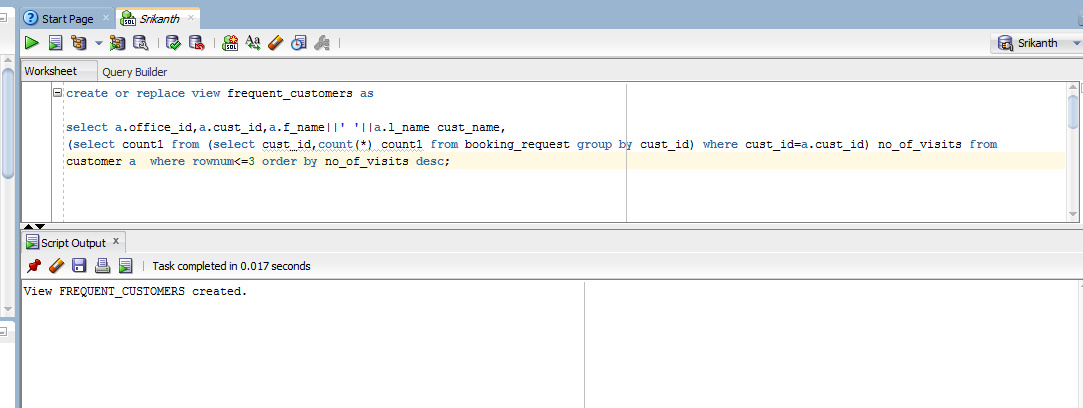
create or replace view frequent\_customers as

select a.office\_id,a.cust\_id,a.f\_name||' '||a.l\_name cust\_name,

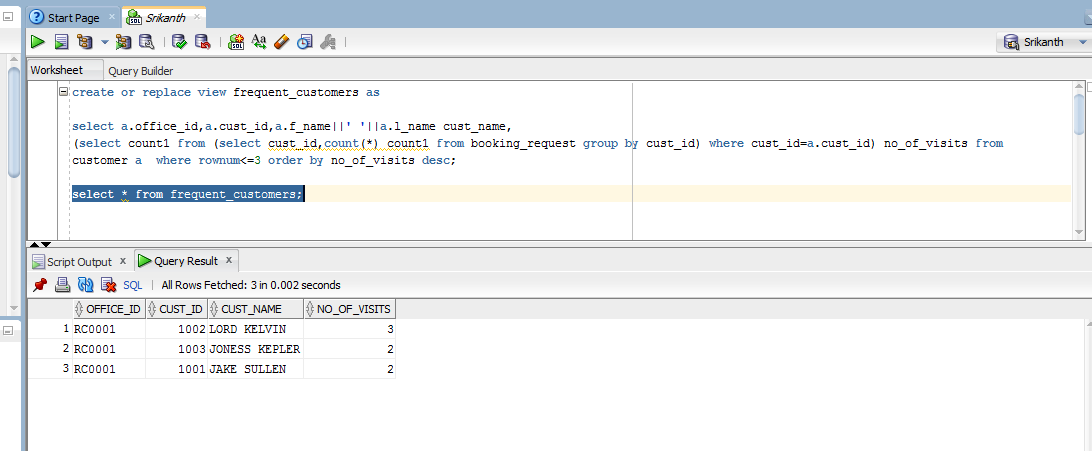
(select count1 from (select cust\_id,count(\*) count1 from booking\_request group by cust\_id) where cust\_id=a.cust\_id) no\_of\_visits from

customer a where rownum<=3 order by no\_of\_visits desc;

**Output:**



**Output:**

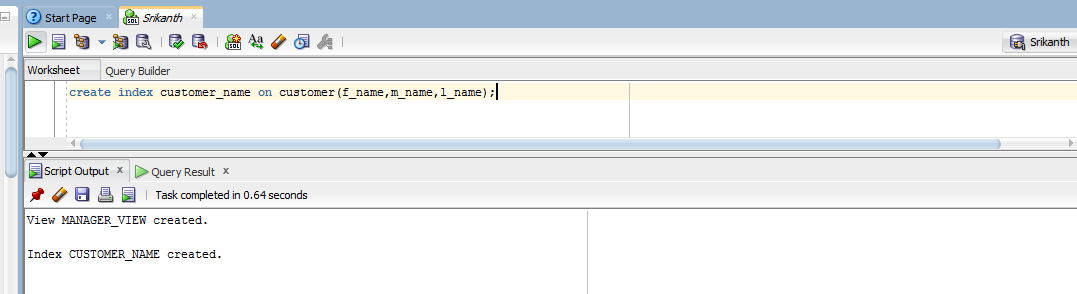


**13 a)**  Index on Customer First & Last name

**Query:**

create index customer\_name on customer(f\_name,m\_name,l\_name);

**Output:**

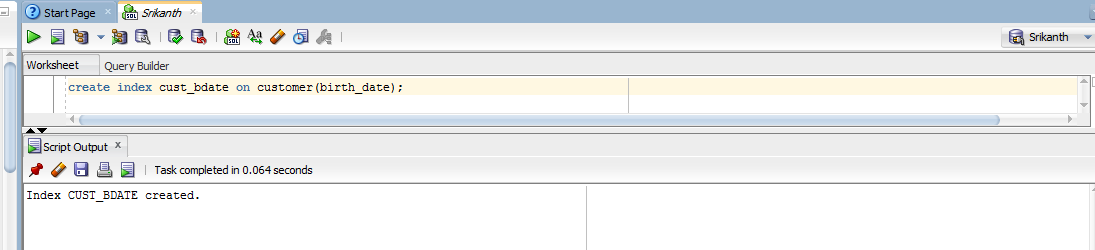


**13 b)** Creating Index on Birthdate of Customer

**Query:**

create index cust\_bdate on customer(birth\_date);

**Output:**

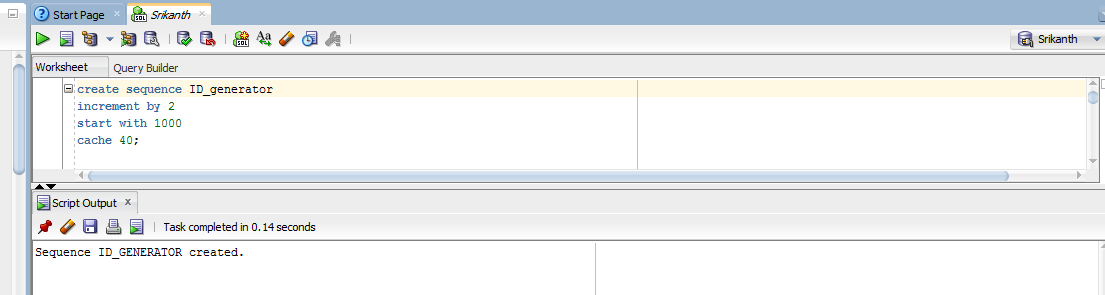


**14 a)** Creating sequence ‘ ID\_generator’

**Query:**

create sequence ID\_generator increment by 2 start with 1000 cache 40;

**Output:**

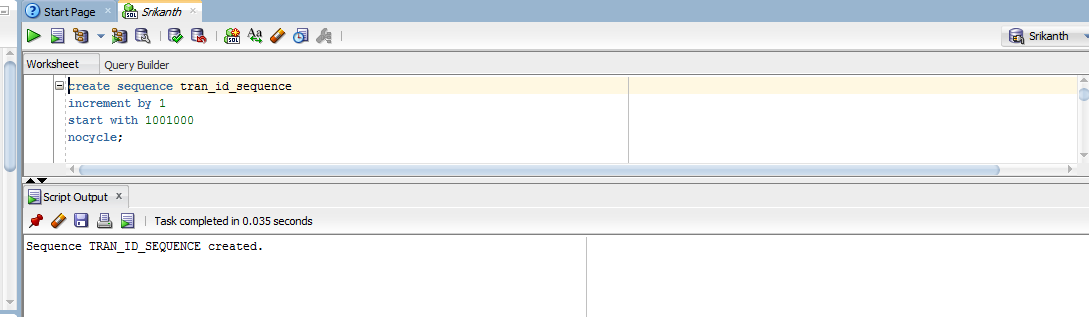


**14 b)** Creating sequence for the Transaction ID.

**Query:**

create sequence tran\_id\_sequence increment by 1 start with 1001000 nocycle;

**Output:**



**14 c)** creating sequence called track number,start with zero and increment by one.

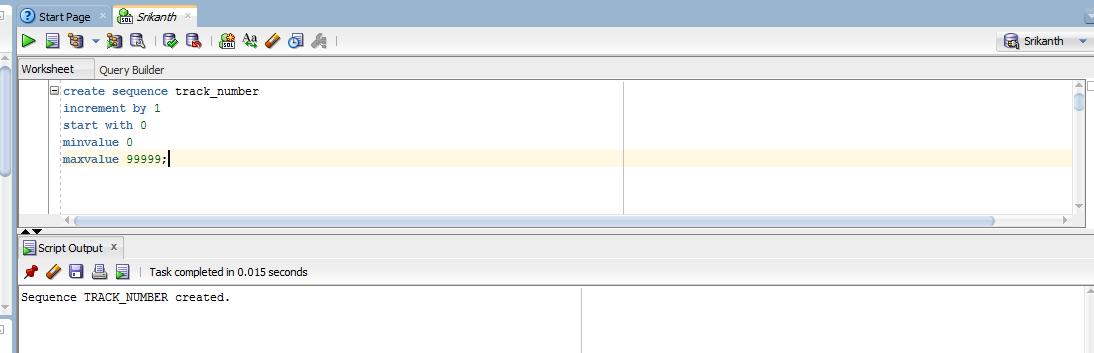
**Query:**

create sequence track\_number increment by 1 start with 0

minvalue 0

maxvalue 99999;

**Output:**



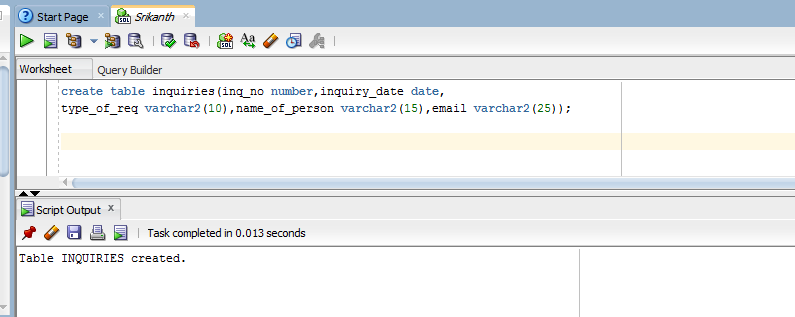
**14 d)**

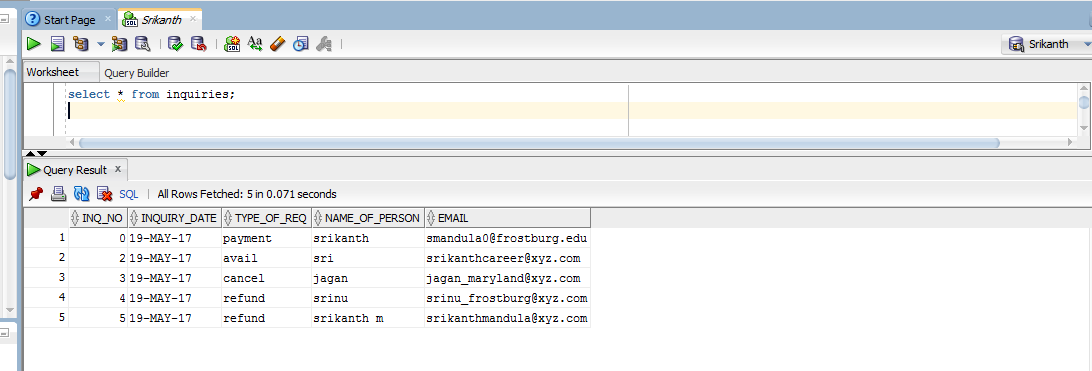
**Query:**

create table inquiries(inq\_no number,inquiry\_date date,

type\_of\_req varchar2(10),name\_of\_person varchar2(15),email varchar2(25));

**Output:**





**SubPrograms & Packages with exception handling :**

**15 a)** Creating package EmpInser\_pkg

**Program :**

create or replace package EmpInsert\_pkg as

procedure insert\_new\_emp(v\_emp\_id emp.emp\_id%type,v\_office\_id emp.office\_id%type,v\_f\_name emp.f\_name%type,v\_m\_name emp.m\_name%type,

v\_l\_name emp.l\_name%type,v\_gender emp.gender%type,v\_flat\_no emp.flat\_no%type,v\_street\_name emp.street\_name%type,

v\_city emp.city%type,v\_state emp.state%type,v\_zipcode emp.zipcode%type,v\_ssn emp.ssn%type,v\_salary emp.salary%type,

v\_tax\_deduction emp.tax\_deduction%type,v\_dob emp.dob%type,v\_marital\_status emp.marital\_status%type,v\_spouse\_name emp.spouse\_name%type,

v\_hiring\_date emp.hiring\_date%type,v\_divorce\_status emp.divorce\_status%type,v\_previous\_name emp.previous\_name%type,

v\_no\_of\_years\_at\_office emp.no\_of\_years\_at\_office%type,v\_no\_of\_years\_at\_company emp.no\_of\_years\_at\_company%type,

v\_last\_degree emp.last\_degree%type,v\_no\_of\_dependents emp.no\_of\_dependents%type,v\_home\_ph\_no emp.home\_ph\_no%type,

v\_cell\_ph\_no emp.cell\_ph\_no%type,v\_emp\_mgr\_id emp.emp\_id%type,v\_mgr\_title manager.mgr\_title%type);

procedure emp\_marital\_status\_update(v\_emp emp.emp\_id%type,v\_marital\_status emp.marital\_status%type);

procedure emp\_last\_degree\_update(v\_emp emp.emp\_id%type,v\_last\_degree emp.last\_degree%type);

procedure emp\_certificate\_update(v\_emp emp\_certificates.emp\_id%type,v\_certificate emp\_certificates.certificate%type);

procedure emp\_manager\_update(v\_emp emp.emp\_id%type,v\_mgr\_id emp.mgr\_id%type);

end EmpInsert\_pkg;

create or replace package body EmpInsert\_pkg as

procedure insert\_new\_emp(v\_emp\_id emp.emp\_id%type,v\_office\_id emp.office\_id%type,v\_f\_name emp.f\_name%type,v\_m\_name emp.m\_name%type,

v\_l\_name emp.l\_name%type,v\_gender emp.gender%type,v\_flat\_no emp.flat\_no%type,v\_street\_name emp.street\_name%type,

v\_city emp.city%type,v\_state emp.state%type,v\_zipcode emp.zipcode%type,v\_ssn emp.ssn%type,v\_salary emp.salary%type,

v\_tax\_deduction emp.tax\_deduction%type,v\_dob emp.dob%type,v\_marital\_status emp.marital\_status%type,v\_spouse\_name emp.spouse\_name%type,

v\_hiring\_date emp.hiring\_date%type,v\_divorce\_status emp.divorce\_status%type,v\_previous\_name emp.previous\_name%type,

v\_no\_of\_years\_at\_office emp.no\_of\_years\_at\_office%type,v\_no\_of\_years\_at\_company emp.no\_of\_years\_at\_company%type,

v\_last\_degree emp.last\_degree%type,v\_no\_of\_dependents emp.no\_of\_dependents%type,v\_home\_ph\_no emp.home\_ph\_no%type,

v\_cell\_ph\_no emp.cell\_ph\_no%type,v\_emp\_mgr\_id emp.emp\_id%type,v\_mgr\_title manager.mgr\_title%type) as

begin

insert into MANAGER values(v\_emp\_mgr\_id,v\_emp\_id,v\_mgr\_title);

insert into emp values( v\_emp\_id,v\_office\_id,v\_f\_name,v\_m\_name,v\_l\_name,v\_gender,v\_flat\_no,v\_street\_name,v\_city,v\_state,

v\_zipcode,v\_ssn,v\_salary,v\_tax\_deduction,v\_dob,v\_marital\_status,v\_spouse\_name,v\_hiring\_date,v\_divorce\_status,v\_previous\_name,

v\_no\_of\_years\_at\_office,v\_no\_of\_years\_at\_company,v\_last\_degree,v\_no\_of\_dependents,v\_home\_ph\_no,v\_cell\_ph\_no,v\_emp\_mgr\_id);

dbms\_output.put\_line('new employee data inserted successfully into emp table and Manger table is also updated');

exception

when others then

dbms\_output.put\_line('Error occured in updating the employee or manager table');

end;

procedure emp\_marital\_status\_update(v\_emp emp.emp\_id%type,v\_marital\_status emp.marital\_status%type) as

begin

update emp set marital\_status = v\_marital\_status where emp\_id=v\_emp;

dbms\_output.put\_line('Employee marital status updated successfully');

exception

when no\_data\_found then

dbms\_output.put\_line('There is no employee available with the given data');

when too\_many\_rows then

dbms\_output.put\_line('there are multiple employees available with the given data');

when others then

dbms\_output.put\_line('there is some error occured .Please check');

end;

procedure emp\_last\_degree\_update(v\_emp emp.emp\_id%type,v\_last\_degree emp.last\_degree%type) as

begin

update emp set last\_degree = v\_last\_degree where emp\_id=v\_emp;

dbms\_output.put\_line('Employee last degree updated successfully');

exception

when no\_data\_found then

dbms\_output.put\_line('There is no employee available with the given data');

when too\_many\_rows then

dbms\_output.put\_line('there are multiple employees available with the given data');

when others then

dbms\_output.put\_line('there is some error occured .Please check');

end;

procedure emp\_certificate\_update(v\_emp emp\_certificates.emp\_id%type,v\_certificate emp\_certificates.certificate%type) as

begin

update emp\_certificates set certificate = v\_certificate where emp\_id=v\_emp;

dbms\_output.put\_line('Employee certificate status updated successfully');

exception

when no\_data\_found then

dbms\_output.put\_line('There is no employee available with the given data');

when too\_many\_rows then

dbms\_output.put\_line('there are multiple employees available with the given data');

when others then

dbms\_output.put\_line('there is some error occured .Please check');

end;

procedure emp\_manager\_update(v\_emp emp.emp\_id%type,v\_mgr\_id emp.mgr\_id%type) as

begin

update emp set mgr\_id = v\_mgr\_id where emp\_id=v\_emp;

update manager set manager\_id = v\_mgr\_id where rep\_emp\_id=v\_emp;

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

exception

when no\_data\_found then

dbms\_output.put\_line('There is no employee available with the given data');

when too\_many\_rows then

dbms\_output.put\_line('there are multiple employees available with the given data');

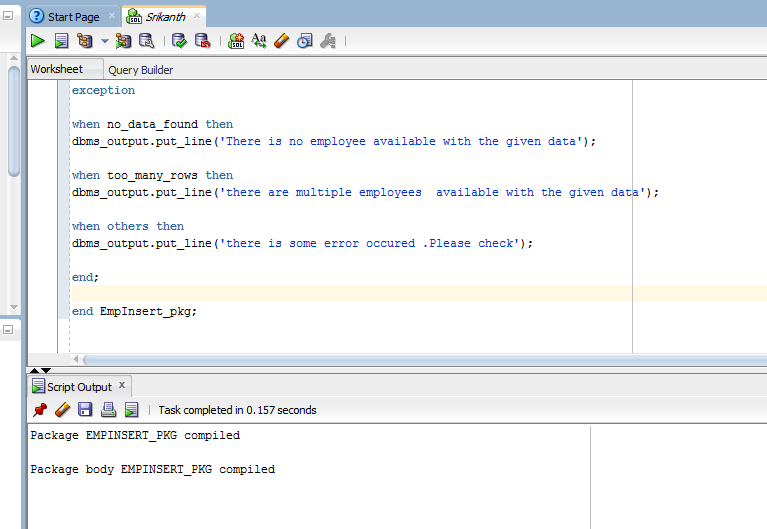
when others then

dbms\_output.put\_line('there is some error occured .Please check');

end;

end EmpInsert\_pkg;

**Output:**



**15 b) Creating *Insert\_pkg***

**Program :**

create or replace package Insert\_pkg as

procedure insert\_new\_customer(v\_cust\_id customer.cust\_id%type,v\_office\_id customer.office\_id%type,v\_f\_name customer.f\_name%type,v\_m\_name customer.m\_name%type,

v\_l\_name customer.l\_name%type,v\_birth\_date customer.birth\_date%type,v\_cell\_phone customer.cell\_phone%type,

v\_flatno customer.flatno%type,v\_street\_name customer.street\_name%type,v\_city customer.city%type,v\_state customer.state%type,

v\_zipcode customer.zipcode%type,v\_email\_id customer.email\_id%type,v\_dl\_num customer.dl\_num%type,

v\_credit\_card\_no customer.credit\_card\_no%type,v\_card\_exp\_date customer.card\_exp\_date%type,v\_dl\_state dl.dl\_state%type,

v\_dl\_issued\_date dl.dl\_issued\_date%type,v\_dl\_valid\_upto dl.dl\_valid\_upto%type,v\_remarks dl.remarks%type);

procedure insert\_new\_client(v\_clnt\_id client.clnt\_id%type,v\_office\_id client.office\_id%type,v\_ssn client.ssn%type,

v\_f\_name client.f\_name%type,v\_m\_name client.m\_name%type,v\_l\_name client.l\_name%type,v\_dob client.dob%type,

v\_flat\_no client.flat\_no%type,v\_street\_name client.street\_name%type,v\_city client.city%type,v\_state client.state%type,

v\_zipcode client.zipcode%type,v\_home\_ph client.home\_ph%type,v\_cell\_ph client.cell\_ph%type,

v\_email client.email%type,v\_cc\_no client.cc\_no%type,v\_card\_exp\_date client.card\_exp\_date%type,v\_emp\_id emp.emp\_id%type,v\_is\_employee checks.is\_employee%type,

v\_clnt\_emp\_id checks.clnt\_emp\_id%type,v\_credit\_score checks.credit\_score%type,v\_scored\_date checks.scored\_date%type,

v\_remarks checks.remarks%type);

procedure customer\_info\_update(v\_cust\_id customer.cust\_id%type,v\_cell\_phone customer.cell\_phone%type);

procedure client\_info\_update(v\_clnt\_id client.clnt\_id%type,v\_email client.email%type);

procedure insert\_new\_tran(v\_tran\_id transaction.tran\_id%type,v\_booking\_id transaction.booking\_id%type,

v\_tran\_date transaction.tran\_date%type,v\_amount transaction.amount%type,v\_miles\_run transaction.miles\_run%type);

procedure insert\_accident\_record(v\_booking\_id accident.booking\_id%type,v\_accident\_date accident.accident\_date%type,

v\_accident\_time accident.accident\_time%type,v\_place accident.place%type,v\_ext\_of\_damage accident.ext\_of\_damage%type,

v\_cost\_of\_damage accident.cost\_of\_damage%type,v\_police\_report accident.police\_report%type,v\_summary\_report accident.summary\_report%type,

v\_remarks accident.remarks%type);

end Insert\_pkg;

create or replace package body Insert\_pkg as

procedure insert\_new\_customer(v\_cust\_id customer.cust\_id%type,v\_office\_id customer.office\_id%type,v\_f\_name customer.f\_name%type,v\_m\_name customer.m\_name%type,

v\_l\_name customer.l\_name%type,v\_birth\_date customer.birth\_date%type,v\_cell\_phone customer.cell\_phone%type,

v\_flatno customer.flatno%type,v\_street\_name customer.street\_name%type,v\_city customer.city%type,v\_state customer.state%type,

v\_zipcode customer.zipcode%type,v\_email\_id customer.email\_id%type,v\_dl\_num customer.dl\_num%type,

v\_credit\_card\_no customer.credit\_card\_no%type,v\_card\_exp\_date customer.card\_exp\_date%type,v\_dl\_state dl.dl\_state%type,

v\_dl\_issued\_date dl.dl\_issued\_date%type,v\_dl\_valid\_upto dl.dl\_valid\_upto%type,v\_remarks dl.remarks%type) as

begin

insert into DL values(v\_dl\_num,v\_dl\_state,v\_dl\_issued\_date,v\_dl\_valid\_upto,v\_remarks);

insert into customer values(v\_cust\_id,v\_office\_id,v\_f\_name,v\_m\_name,v\_l\_name,v\_birth\_date,v\_cell\_phone,

v\_flatno,v\_street\_name,v\_city,v\_state,v\_zipcode,v\_email\_id,v\_dl\_num,v\_credit\_card\_no,v\_card\_exp\_date);

dbms\_output.put\_line('new customer data inserted successfully into customer table and DL table is also updated');

exception

when others then

dbms\_output.put\_line('Error occured in updating customer table');

end;

procedure insert\_new\_client(v\_clnt\_id client.clnt\_id%type,v\_office\_id client.office\_id%type,v\_ssn client.ssn%type,

v\_f\_name client.f\_name%type,v\_m\_name client.m\_name%type,v\_l\_name client.l\_name%type,v\_dob client.dob%type,

v\_flat\_no client.flat\_no%type,v\_street\_name client.street\_name%type,v\_city client.city%type,v\_state client.state%type,

v\_zipcode client.zipcode%type,v\_home\_ph client.home\_ph%type,v\_cell\_ph client.cell\_ph%type,

v\_email client.email%type,v\_cc\_no client.cc\_no%type,v\_card\_exp\_date client.card\_exp\_date%type,v\_emp\_id emp.emp\_id%type,v\_is\_employee checks.is\_employee%type,

v\_clnt\_emp\_id checks.clnt\_emp\_id%type,v\_credit\_score checks.credit\_score%type,v\_scored\_date checks.scored\_date%type,

v\_remarks checks.remarks%type) as

begin

insert into client values(v\_clnt\_id,v\_office\_id,v\_ssn,v\_f\_name,v\_m\_name,v\_l\_name,v\_dob,v\_flat\_no,v\_street\_name,v\_city,

v\_state,v\_zipcode,v\_home\_ph,v\_cell\_ph,v\_email,v\_cc\_no,v\_card\_exp\_date);

insert into checks values(v\_emp\_id,v\_clnt\_id,v\_is\_employee,v\_clnt\_emp\_id,v\_credit\_score,v\_scored\_date,v\_remarks);

exception

when others then

dbms\_output.put\_line('there is some error occured .Please check');

end;

procedure customer\_info\_update(v\_cust\_id customer.cust\_id%type,v\_cell\_phone customer.cell\_phone%type) as

begin

update customer set cell\_phone = v\_cell\_phone where cust\_id=v\_cust\_id;

dbms\_output.put\_line('Customer cell phone updated successfully');

exception

when no\_data\_found then

dbms\_output.put\_line('There is no customer available with the given data');

when too\_many\_rows then

dbms\_output.put\_line('there are multiple customers available with the given data');

when others then

dbms\_output.put\_line('there is some error occured .Please check');

end;

procedure client\_info\_update(v\_clnt\_id client.clnt\_id%type,v\_email client.email%type) as

begin

update client set email = v\_email where clnt\_id=v\_clnt\_id;

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

exception

when no\_data\_found then

dbms\_output.put\_line('There is no client available with the given data');

when too\_many\_rows then

dbms\_output.put\_line('there are multiple clients available with the given data');

when others then

dbms\_output.put\_line('there is some error occured .Please check');

end;

procedure insert\_new\_tran(v\_tran\_id transaction.tran\_id%type,v\_booking\_id transaction.booking\_id%type,

v\_tran\_date transaction.tran\_date%type,v\_amount transaction.amount%type,v\_miles\_run transaction.miles\_run%type) as

begin

insert into transaction values(v\_tran\_id,v\_booking\_id,v\_tran\_date,v\_amount,v\_miles\_run);

dbms\_output.put\_line('transaction record added successfully');

exception

when others then

dbms\_output.put\_line('there is some error occured .Please check');

end;

procedure insert\_accident\_record(v\_booking\_id accident.booking\_id%type,v\_accident\_date accident.accident\_date%type,

v\_accident\_time accident.accident\_time%type,v\_place accident.place%type,v\_ext\_of\_damage accident.ext\_of\_damage%type,

v\_cost\_of\_damage accident.cost\_of\_damage%type,v\_police\_report accident.police\_report%type,v\_summary\_report accident.summary\_report%type,

v\_remarks accident.remarks%type) as

begin

insert into accident values(v\_booking\_id,v\_accident\_date,v\_accident\_time,v\_place,v\_ext\_of\_damage,v\_cost\_of\_damage,

v\_police\_report,v\_summary\_report,v\_remarks);

dbms\_output.put\_line('accident record added successfully');

exception

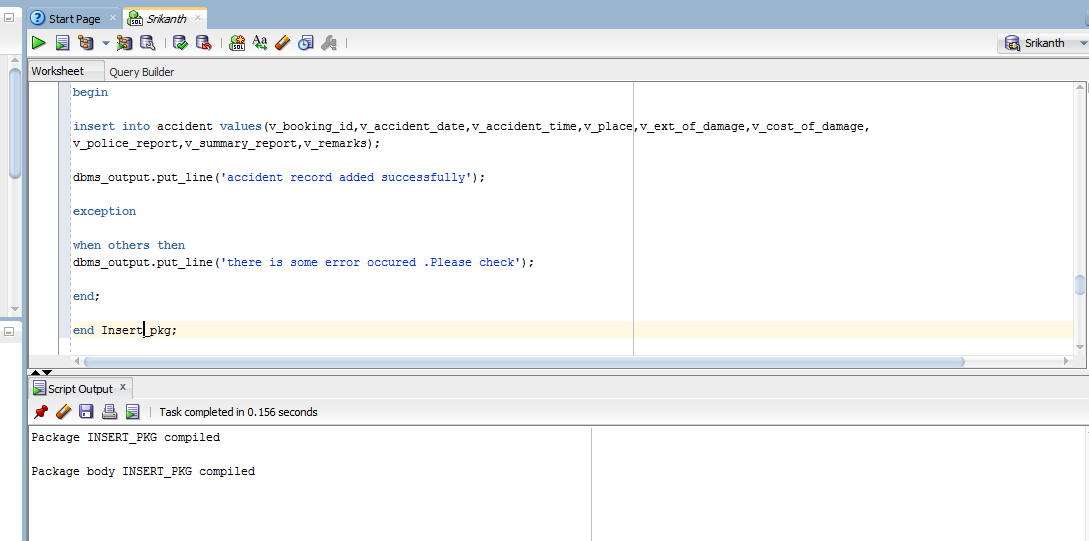
when others then

dbms\_output.put\_line('there is some error occured .Please check');

end;

end Insert\_pkg;

**Output:**



**15 c) Creating *Vehicle\_pkg***

**Program :**

create or replace package Vehicle\_pkg as

function Car\_info(v\_car\_id vehicle.car\_id%type) return varchar2;

procedure CarOwner(SSN1 client.ssn%type);

procedure CarRented(v\_car\_id vehicle.car\_id%type);

procedure CarRented(v\_cust\_id customer.cust\_id%type);

end Vehicle\_pkg;

create or replace package body Vehicle\_pkg as

function Car\_info(v\_car\_id vehicle.car\_id%type) return varchar2 as

v\_plate\_no vehicle.car\_info%type;

v\_make vehicle.make%type;

v\_year vehicle.year%type;

v\_curr\_mileage vehicle.curr\_mileage%type;

v\_class vehicle.class%type;

v\_color vehicle.color%type;

v\_feature vehicle.ffeatures%type;

v\_car\_description vehicle.car\_description%type;

result varchar2(500);

begin

select car\_info,make,year,curr\_mileage,class,color,ffeatures,car\_description into v\_plate\_no,v\_make,v\_year,v\_curr\_mileage,v\_class,v\_color,v\_feature,v\_car\_description from vehicle where car\_id=v\_car\_id;

result := 'plate no :'||v\_plate\_no||' ,make : '||v\_make||' ,year : '||v\_year||' ,current mileage : '||v\_curr\_mileage||' ,class : '||v\_class||' ,color : '||v\_color||' ,features : '||v\_feature||' ,car description : '||v\_car\_description;

return result;

exception

when no\_data\_found then

dbms\_output.put\_line('There is no vehicle available with the given data');

when too\_many\_rows then

dbms\_output.put\_line('there are multiple vehicles available with the given data');

when others then

dbms\_output.put\_line('Error occured in updating customer table');

end;

procedure CarOwner(SSN1 client.ssn%type) as

v\_name varchar2(100);

v\_car\_id vehicle.car\_id%type;

v\_car\_info vehicle.car\_info%type;

v\_no\_of\_rents number default 0;

v\_no\_of\_accidents number default 0;

v\_booking\_id booking\_request.booking\_id%type;

v\_temp\_car\_id vehicle.car\_id%type;

cursor c1 is

select car\_id,car\_info from vehicle where clnt\_id=(select clnt\_id from client where ssn=ssn1);

cursor c2 is

select booking\_id from accident;

begin

select f\_name||' '||m\_name||' '||l\_name into v\_name from client where ssn=ssn1;

dbms\_output.put\_line('Car Owner SSN is : '||SSN1);

dbms\_output.put\_line('Car Owner name is : '||v\_name);

dbms\_output.put\_line('details of car/cars owned by this owner :');

open c1;

loop

fetch c1 into v\_car\_id,v\_car\_info;

exit when c1%notfound;

dbms\_output.put\_line('car\_id : '||v\_car\_id||' , plate No : '||v\_car\_info);

select count(\*) into v\_no\_of\_rents from booking\_request where car\_id=v\_car\_id;

dbms\_output.put\_line('Number of times this car rented is : '||v\_no\_of\_rents);

open c2;

loop

fetch c2 into v\_booking\_id;

exit when c2%notfound;

select car\_id into v\_temp\_car\_id from booking\_request where booking\_id=v\_booking\_id;

if(v\_temp\_car\_id=v\_car\_id) then

v\_no\_of\_accidents :=v\_no\_of\_accidents+1;

end if;

end loop;

dbms\_output.put\_line('Number of times this car involved in accidents is : '||v\_no\_of\_accidents);

close c2;

end loop;

close c1;

exception

when others then

dbms\_output.put\_line('there is some error occured .Please check');

end;

procedure CarRented(v\_car\_id vehicle.car\_id%type) as

v\_cust\_id booking\_request.cust\_id%type;

v\_customer\_name varchar2(100);

cursor cusor\_cust\_id is

select distinct cust\_id from booking\_request where car\_id=v\_car\_id;

begin

dbms\_output.put\_line('Customers who rented the car : '||v\_car\_id||' are : ');

open cusor\_cust\_id;

loop

fetch cusor\_cust\_id into v\_cust\_id;

exit when cusor\_cust\_id%notfound;

select f\_name||' '||m\_name||' '||l\_name into v\_customer\_name from customer where cust\_id=v\_cust\_id;

dbms\_output.put\_line('Name : '||v\_customer\_name);

end loop;

exception

when no\_data\_found then

dbms\_output.put\_line('There is no car available with the given data');

when too\_many\_rows then

dbms\_output.put\_line('there are multiple cars available with the given data');

when others then

dbms\_output.put\_line('there is some error occured .Please check');

end;

procedure CarRented(v\_cust\_id customer.cust\_id%type) as

v\_car\_id vehicle.car\_id%type;

v\_req\_date booking\_request.req\_date%type;

prev\_rented\_date date;

last\_rented\_car\_id vehicle.car\_id%type;

counter number default 0;

cursor cars\_rented is

select car\_id,req\_date from booking\_request where cust\_id=v\_cust\_id;

begin

open cars\_rented;

loop

exit when cars\_rented%notfound;

fetch cars\_rented into v\_car\_id,v\_req\_date;

if (counter=0) then

prev\_rented\_date := v\_req\_date;

last\_rented\_car\_id := v\_car\_id;

counter := counter+1;

elsif prev\_rented\_date<v\_req\_date then

last\_rented\_car\_id :=v\_car\_id;

end if;

end loop;

close cars\_rented;

dbms\_output.put\_line('The last car rented by customer '||v\_cust\_id|| 'is : '||last\_rented\_car\_id);

exception

when no\_data\_found then

dbms\_output.put\_line('There is no customer available with the given data');

when too\_many\_rows then

dbms\_output.put\_line('there are multiple customers available with the given data');

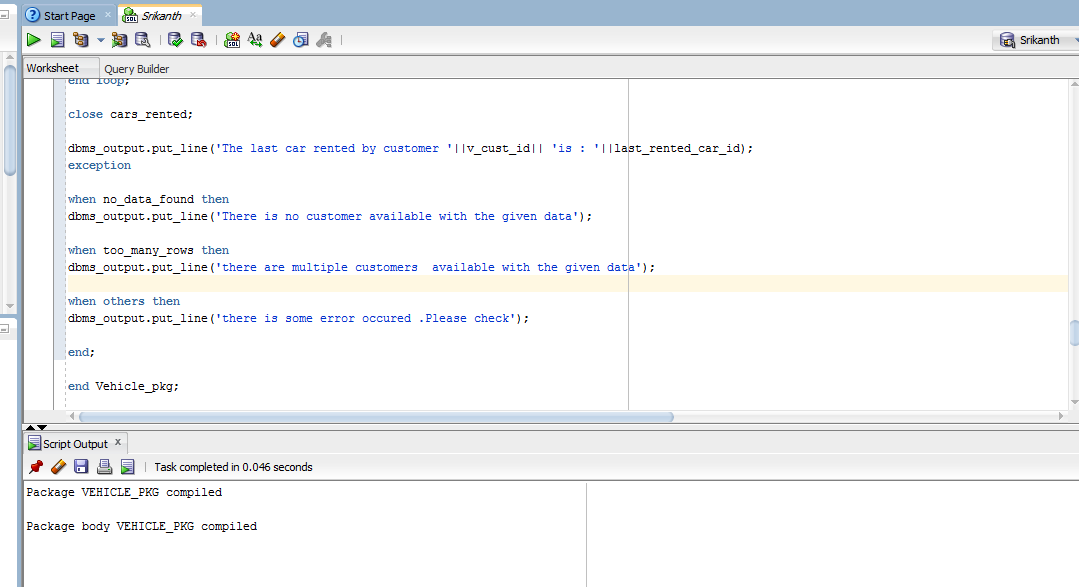
when others then

dbms\_output.put\_line('there is some error occured .Please check');

end;

end Vehicle\_pkg;

**Output:**



**15 d) Creating *Accident\_pkg***

**Program :**

create or replace package Accident\_pkg as

function CarAccident(v\_car\_id vehicle.car\_id%type) return varchar2;

function CustomerAccident(v\_cust\_id customer.cust\_id%type) return boolean;

end Accident\_pkg;

create or replace package body Accident\_pkg as

function CarAccident(v\_car\_id vehicle.car\_id%type) return varchar2 as

v\_cust\_name varchar2(50);

v\_cell\_phone customer.cell\_phone%type;

v\_cust\_address varchar2(100);

v\_accident\_date accident.accident\_date%type;

v\_accident\_time accident.accident\_time%type;

v\_accident\_place accident.place%type;

v\_ext\_of\_damage accident.ext\_of\_damage%type;

v\_cost\_of\_damage accident.cost\_of\_damage%type;

result varchar2(500);

counter number default 0;

v\_last\_accident\_date accident.accident\_date%type;

v\_temp\_accident\_date accident.accident\_date%type;

v\_booking\_id accident.booking\_id%type;

v\_cust\_id booking\_request.cust\_id%type;

v\_temp\_car\_id vehicle.car\_id%type;

v\_temp\_car\_id2 vehicle.car\_id%type;

v\_booking\_id\_temp accident.booking\_id%type;

req\_accident\_id booking\_request.booking\_id%type;

cursor accident\_ids is

select booking\_id from accident;

cursor accident\_ids\_temp is

select booking\_id from accident;

begin

open accident\_ids;

loop

fetch accident\_ids into v\_booking\_id;

exit when accident\_ids%notfound;

select car\_id into v\_temp\_car\_id from booking\_request where booking\_id=v\_booking\_id;

select accident\_date into v\_last\_accident\_date from accident where booking\_id=v\_booking\_id;

if (v\_temp\_car\_id=v\_car\_id) then

counter := counter+1;

open accident\_ids\_temp;

loop

fetch accident\_ids\_temp into v\_booking\_id\_temp;

exit when accident\_ids\_temp%notfound;

select car\_id into v\_temp\_car\_id2 from booking\_request where booking\_id=v\_booking\_id\_temp;

select accident\_date into v\_temp\_accident\_date from accident where booking\_id=v\_booking\_id\_temp;

if (v\_temp\_car\_id2=v\_car\_id and v\_last\_accident\_date<=v\_temp\_accident\_date) then

select booking\_id into req\_accident\_id from accident where booking\_id=v\_booking\_id\_temp;

end if;

end loop;

close accident\_ids\_temp;

select cust\_id into v\_cust\_id from booking\_request where booking\_id=req\_accident\_id;

select accident\_date,accident\_time,place,ext\_of\_damage,cost\_of\_damage into

v\_accident\_date,v\_accident\_time,v\_accident\_place,v\_ext\_of\_damage,v\_cost\_of\_damage from accident where booking\_id=req\_accident\_id;

select f\_name||' '||m\_name||' '||l\_name,cell\_phone,flatno||','||street\_name||', '||city||', '||state||' ,'||zipcode

into v\_cust\_name,v\_cell\_phone,v\_cust\_address from customer where cust\_id=v\_cust\_id;

result := 'Customer Name : '||v\_cust\_name||' , cell phone is : '||v\_cell\_phone||' Address is : '||v\_cust\_address||

' ,accident date is : '||v\_accident\_date||' , accident time is : '||v\_accident\_time||' , accident place is : '||v\_accident\_place

||' , extent of damage is : '||v\_ext\_of\_damage||' , cost of damage is : '||v\_cost\_of\_damage;

end if;

end loop;

close accident\_ids;

if(counter=0) then

result := 'There is no accident involved with this car';

end if;

return result;

exception

when no\_data\_found then

dbms\_output.put\_line('There is no vehicle available with the given data');

when too\_many\_rows then

dbms\_output.put\_line('there are multiple vehicles available with the given data');

when others then

dbms\_output.put\_line('Error occured in updating customer table');

end;

function CustomerAccident(v\_cust\_id customer.cust\_id%type) return boolean as

v\_booking\_id booking\_request.booking\_id%type;

req\_cust\_id customer.cust\_id%type;

counter number default 0;

cursor accident\_ids is

select booking\_id from accident;

begin

open accident\_ids;

loop

fetch accident\_ids into v\_booking\_id;

exit when accident\_ids%notfound;

select cust\_id into req\_cust\_id from booking\_request where booking\_id=v\_booking\_id;

if(v\_cust\_id=req\_cust\_id) then

counter := counter+1;

end if;

end loop;

close accident\_ids;

if(counter >0) then

return true;

else

return false;

end if;

exception

when no\_data\_found then

dbms\_output.put\_line('There is no vehicle available with the given data');

when too\_many\_rows then

dbms\_output.put\_line('there are multiple vehicles available with the given data');

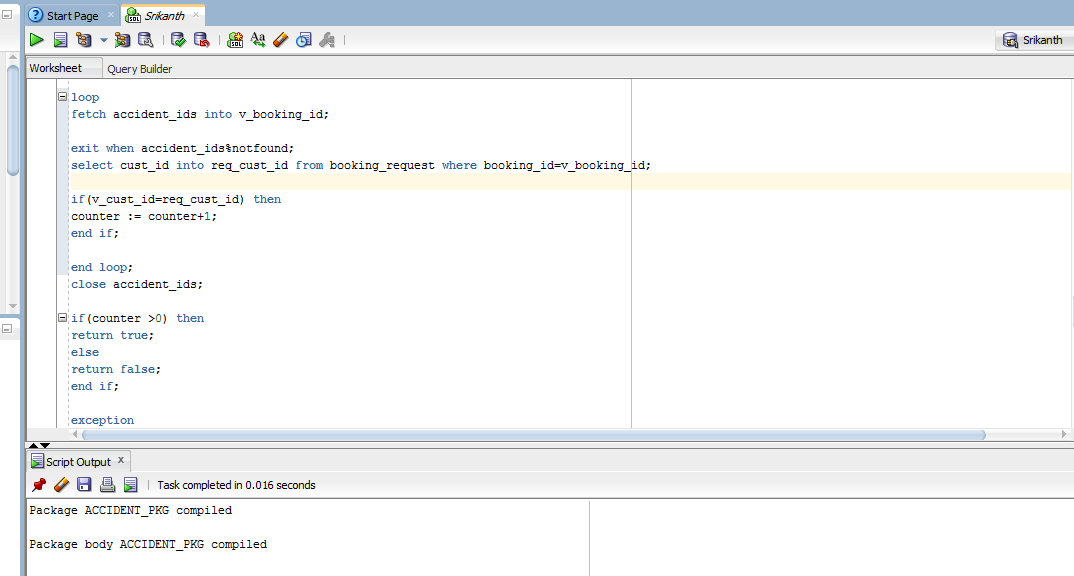
when others then

dbms\_output.put\_line('Error occured in updating customer table');

end;

end Accident\_pkg;

**Output:**



**15 e) Creating subprogram *BirthdaySub***

**Program :**

create table BIRTHDAYFILE(f\_name varchar2(50),l\_name varchar2(50),email\_id varchar2(50),address varchar2(100));

create or replace procedure BirthdaySub(given\_date date) as

v\_cust\_id customer.cust\_id%type;

v\_cust\_bday customer.birth\_date%type;

v\_compare\_date date;

dif number default 0;

v\_f\_name customer.f\_name%type;

v\_l\_name customer.l\_name%type;

v\_email\_id customer.email\_id%type;

v\_address varchar2(100);

temp\_string varchar2(50);

no\_of\_rows number;

cursor cust\_bday is

select cust\_id,birth\_date from customer;

begin

select count(\*) into no\_of\_rows from birthdayfile;

if(no\_of\_rows>0) then

delete from birthdayfile;

end if;

open cust\_bday;

loop

fetch cust\_bday into v\_cust\_id,v\_cust\_bday;

exit when cust\_bday%notfound;

temp\_string := to\_char(v\_cust\_bday,'dd')||'/'||to\_char(v\_cust\_bday,'mm')||'/'||to\_char(given\_date,'yyyy');

v\_compare\_date := to\_date(temp\_string,'dd-mm-yyyy');

dif := v\_compare\_date - given\_date;

if( abs(dif) <= 15) then

select f\_name,l\_name,email\_id,flatno||' ,'||street\_name||' ,'||city||' ,'||state||' ,'||zipcode into

v\_f\_name,v\_l\_name,v\_email\_id,v\_address from customer where cust\_id=v\_cust\_id;

dbms\_output.put\_line('First Name is : '||v\_f\_name||' ,last name is : '||v\_l\_name||' ,email id is : '||v\_email\_id||' ,address is : '||v\_address);

insert into birthdayfile values(v\_f\_name,v\_l\_name,v\_email\_id,v\_address);

end if;

end loop;

close cust\_bday;

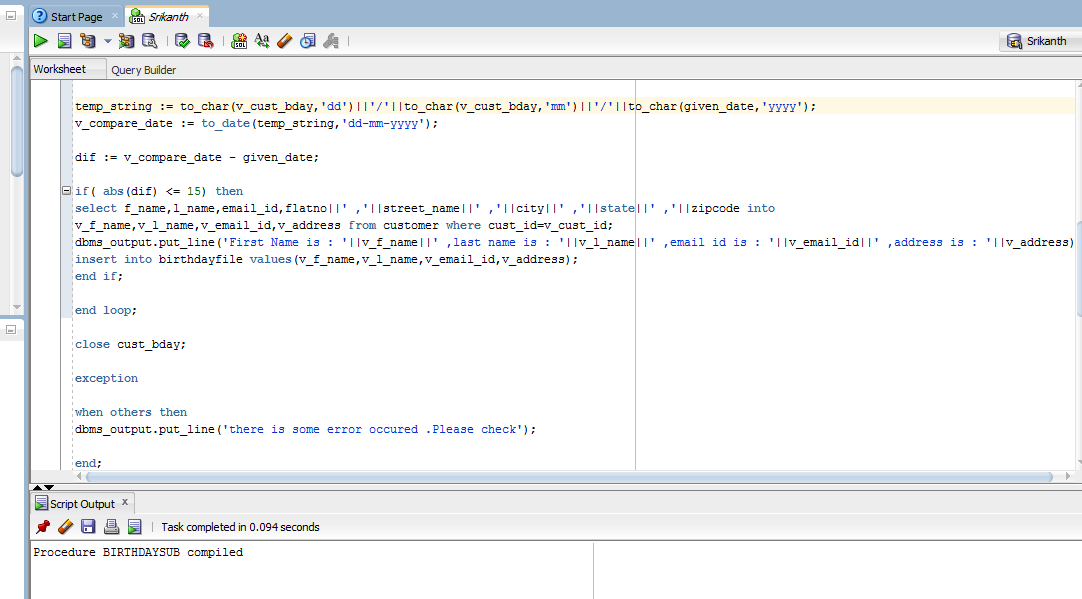
exception

when others then

dbms\_output.put\_line('there is some error occured .Please check');

end;

**Output:**



**15 f) Creating subprogram *Branch\_manager\_Info***

**Program :**

create or replace procedure branch\_mgr\_info is

v\_office\_id office.office\_id%type;

v\_office\_name office.name%type;

v\_mgr\_id emp.emp\_id%type;

v\_mgr\_hired\_date emp.hiring\_date%type;

cursor branch\_cursor is

select office\_id from office;

begin

open branch\_cursor;

loop

fetch branch\_cursor into v\_office\_id;

exit when branch\_cursor%notfound;

select a.office\_id,a.name,a.gm\_id,b.hiring\_date into v\_office\_id,v\_office\_name,v\_mgr\_id,v\_mgr\_hired\_date from office a,emp b where a.office\_id=v\_office\_id and a.gm\_id=b.emp\_id;

dbms\_output.put\_line('Office ID is : '||v\_office\_id);

dbms\_output.put\_line('Office name is : '||v\_office\_name);

dbms\_output.put\_line('Manager Id is : '||v\_mgr\_id);

dbms\_output.put\_line('Manager Hiring Date is : '||v\_mgr\_hired\_date);

dbms\_output.put\_line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

end loop;

close branch\_cursor;

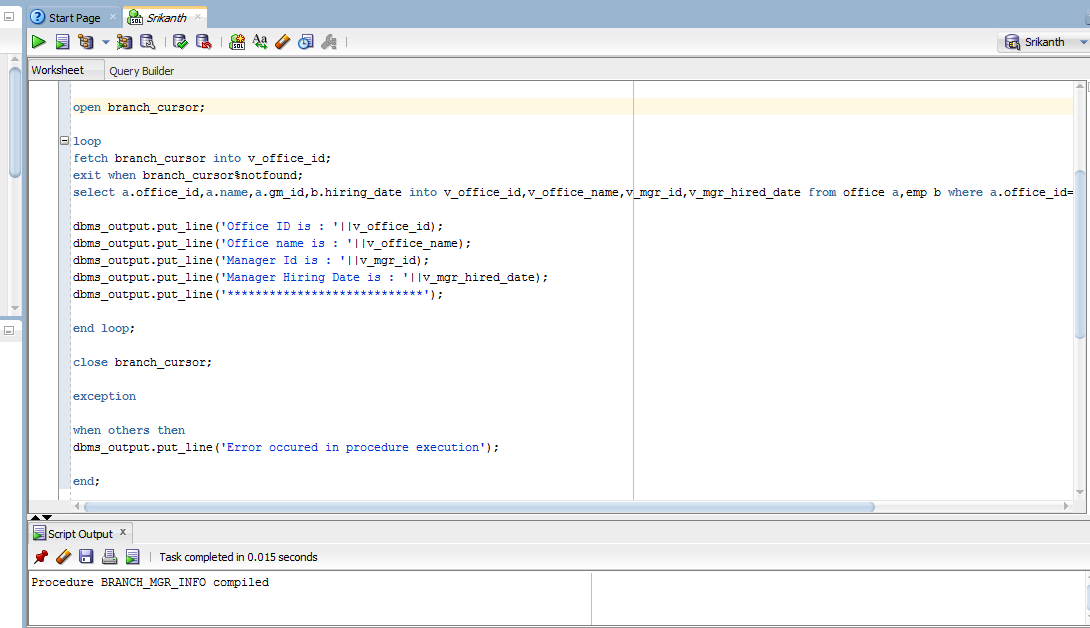
exception

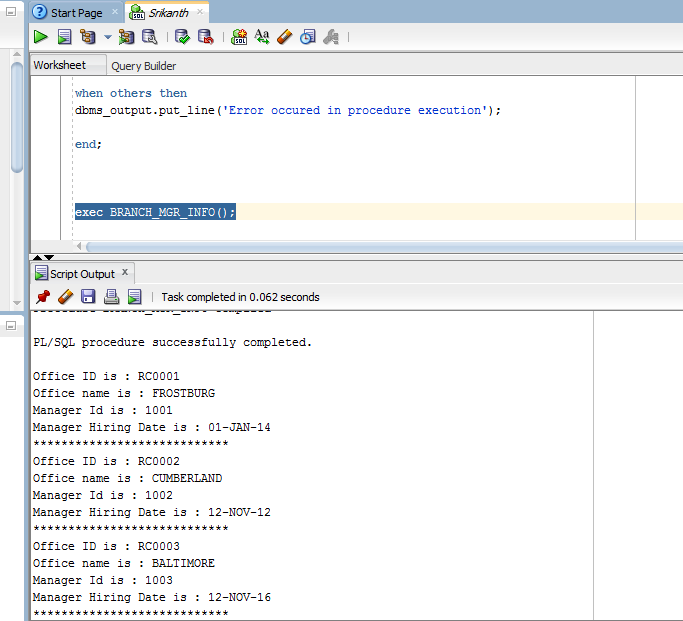
when others then

dbms\_output.put\_line('Error occured in procedure execution');

end;

**Output:**





**16 g) Creating *Employee\_History file* through *Triggers***

**Program :**

create or replace trigger emp\_del\_trigger after delete on emp for each row

declare

v\_emp\_rec emp%rowtype;

begin

v\_emp\_rec.emp\_id := :OLD.emp\_id;

v\_emp\_rec.office\_id := :OLD.office\_id;

v\_emp\_rec.f\_name := :OLD.f\_name;

v\_emp\_rec.m\_name := :OLD.m\_name;

v\_emp\_rec.l\_name := :OLD.l\_name;

v\_emp\_rec.gender := :OLD.gender;

v\_emp\_rec.flat\_no := :OLD.flat\_no;

v\_emp\_rec.street\_name := :OLD.street\_name;

v\_emp\_rec.city := :OLD.city;

v\_emp\_rec.state := :OLD.state;

v\_emp\_rec.zipcode := :OLD.zipcode;

v\_emp\_rec.ssn := :OLD.ssn;

v\_emp\_rec.salary := :OLD.salary;

v\_emp\_rec.tax\_deduction := :OLD.tax\_deduction;

v\_emp\_rec.dob := :OLD.dob;

v\_emp\_rec.marital\_status := :OLD.marital\_status;

v\_emp\_rec.spouse\_name := :OLD.spouse\_name;

v\_emp\_rec.hiring\_date := :OLD.hiring\_date;

v\_emp\_rec.divorce\_status := :OLD.divorce\_status;

v\_emp\_rec.previous\_name := :OLD.previous\_name;

v\_emp\_rec.no\_of\_years\_at\_office := :OLD.no\_of\_years\_at\_office;

v\_emp\_rec.no\_of\_years\_at\_company := :OLD.no\_of\_years\_at\_company;

v\_emp\_rec.last\_degree := :OLD.last\_degree;

v\_emp\_rec.no\_of\_dependents := :OLD.no\_of\_dependents;

v\_emp\_rec.home\_ph\_no := :OLD.home\_ph\_no;

v\_emp\_rec.mgr\_id := :OLD.mgr\_id;

insert into employee\_history values(v\_emp\_rec.emp\_id,v\_emp\_rec.office\_id,v\_emp\_rec.f\_name,v\_emp\_rec.m\_name,v\_emp\_rec.l\_name,v\_emp\_rec.gender,v\_emp\_rec.flat\_no,

v\_emp\_rec.street\_name,v\_emp\_rec.city,v\_emp\_rec.state,v\_emp\_rec.zipcode,v\_emp\_rec.ssn,v\_emp\_rec.salary,v\_emp\_rec.tax\_deduction,

v\_emp\_rec.dob,v\_emp\_rec.marital\_status,v\_emp\_rec.spouse\_name,v\_emp\_rec.hiring\_date,v\_emp\_rec.divorce\_status,v\_emp\_rec.previous\_name,

v\_emp\_rec.no\_of\_years\_at\_office,v\_emp\_rec.no\_of\_years\_at\_company,v\_emp\_rec.last\_degree,v\_emp\_rec.no\_of\_dependents,v\_emp\_rec.home\_ph\_no,

v\_emp\_rec.cell\_ph\_no,v\_emp\_rec.mgr\_id);

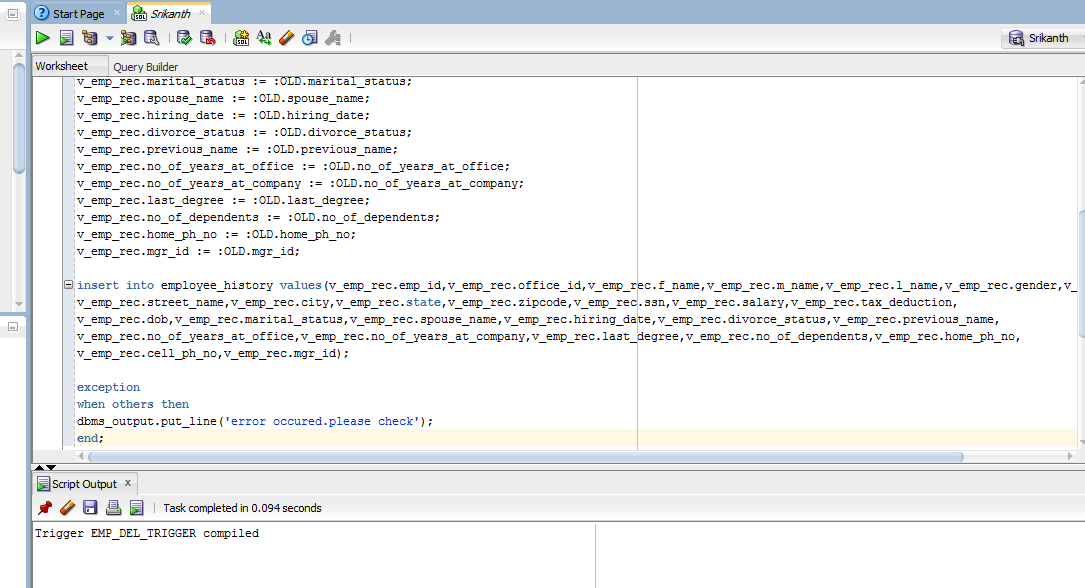
exception

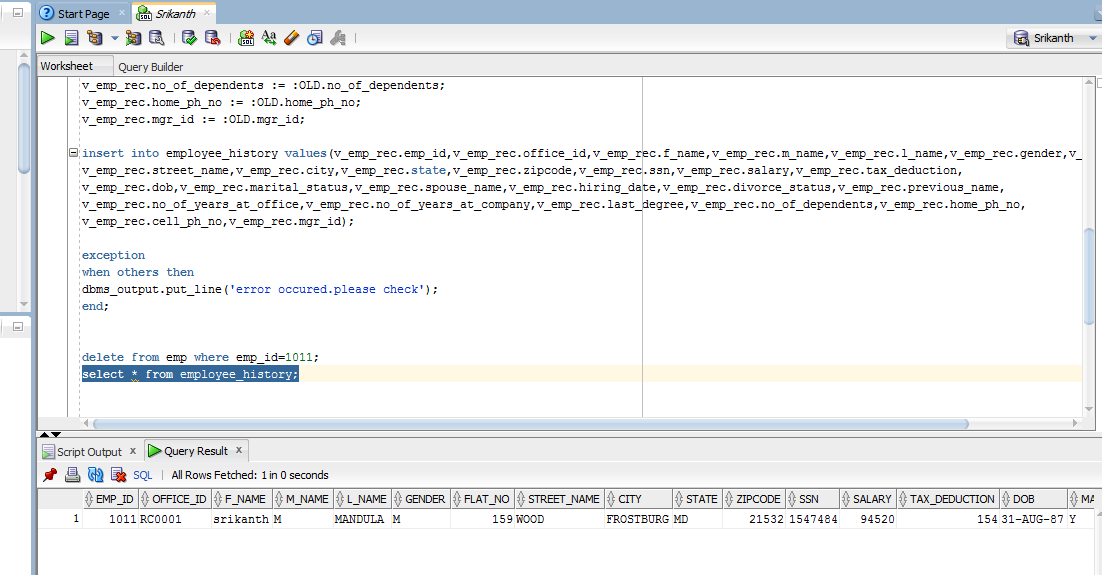
when others then

dbms\_output.put\_line('error occured.please check');

end;

**Output:**





**16 h) Creating Customer\_Log  *file* through *Triggers***

**Program :**

create table customer\_log(username varchar2(30),tran\_date date,nature\_of\_tran char);

create or replace trigger customer\_mod\_trigger after update or insert or delete on customer for each row

declare

v\_change char(1);

begin

if inserting then

v\_change := 'I';

elsif updating then

v\_change := 'U';

else

v\_change := 'D';

end if;

insert into customer\_log values(user,sysdate,v\_change);

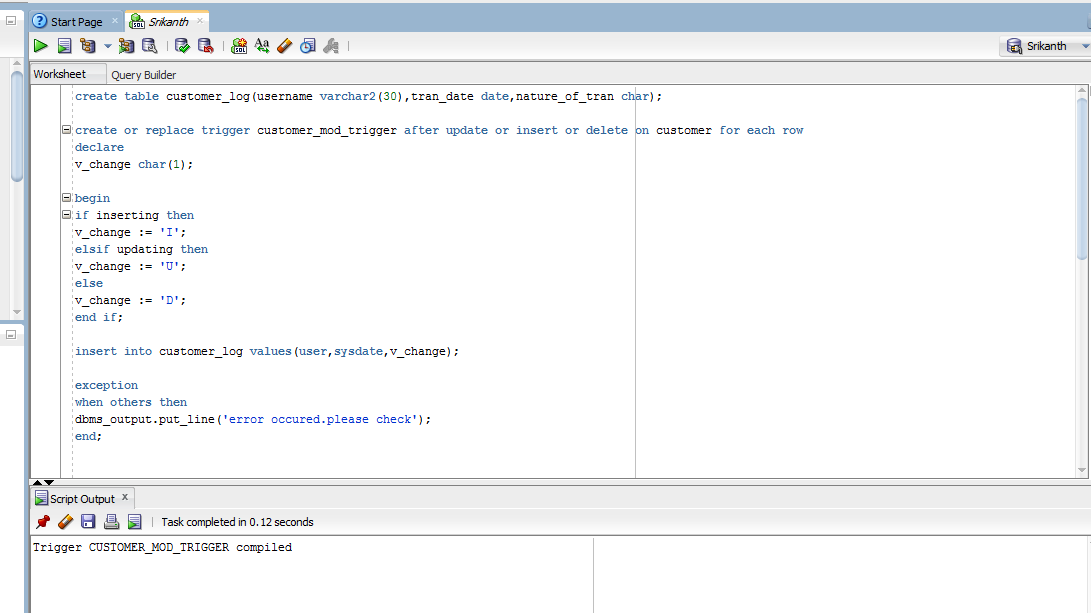
exception

when others then

dbms\_output.put\_line('error occured.please check');

end;

**Output:**



**16 i) Creating Accident\_Log  *file* through *Triggers***

**Program :**

create table accident\_log(cust\_name varchar2(30),client\_name varchar2(30),accident\_date date);

create or replace trigger accident\_trigger after insert on accident for each row

declare

v\_booking\_id booking\_request.booking\_id%type;

v\_cust\_name varchar2(50);

v\_accident\_date date;

v\_client\_name varchar2(30);

begin

v\_booking\_id := :NEW.booking\_id;

v\_accident\_date := :NEW.accident\_date;

select a.f\_name||' '||a.l\_name,b.f\_name||' '||b.l\_name into v\_cust\_name,v\_client\_name from customer a,client b,booking\_request c,vehicle d

where c.booking\_id= v\_booking\_id and c.cust\_id=a.cust\_id and c.car\_id=d.car\_id and d.clnt\_id=b.clnt\_id;

insert into accident\_log values(v\_cust\_name,v\_client\_name,v\_accident\_date);

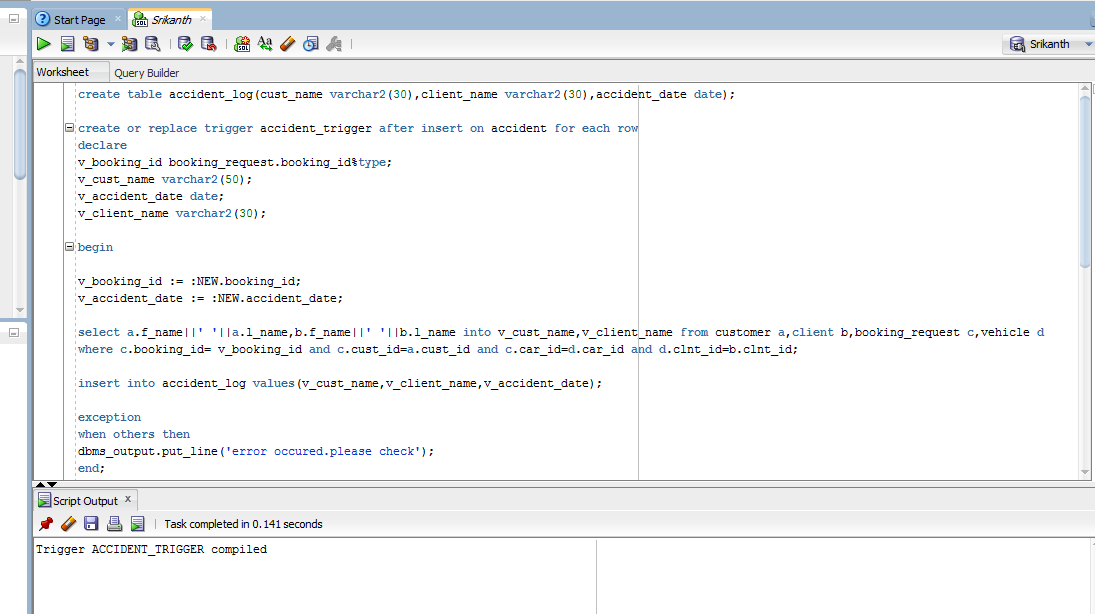
exception

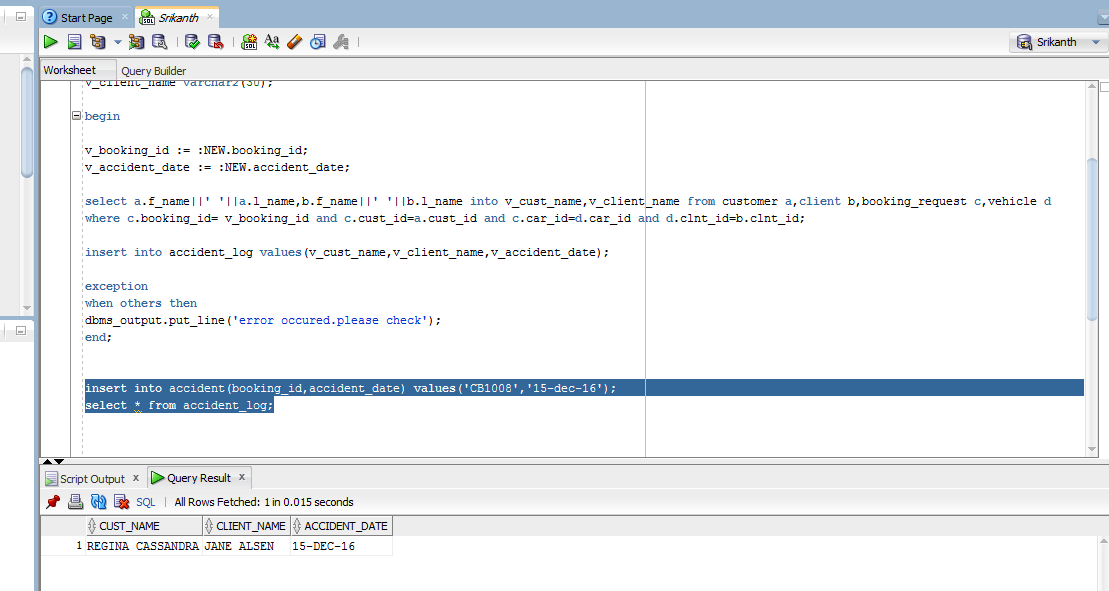
when others then

dbms\_output.put\_line('error occured.please check');

end;

**Output:**





**17) Error\_log file for errors in run time programs**

**Program :**

create or replace trigger accident\_trigger before insert on accident for each row

declare

v\_booking\_id booking\_request.booking\_id%type;

v\_cust\_name varchar2(50);

v\_accident\_date date;

v\_client\_name varchar2(30);

accident\_date\_null Exception;

begin

v\_booking\_id := :NEW.booking\_id;

v\_accident\_date := :NEW.accident\_date;

if(:NEW.accident\_date is null) then

raise accident\_date\_null;

end if;

select a.f\_name||' '||a.l\_name,b.f\_name||' '||b.l\_name into v\_cust\_name,v\_client\_name from customer a,client b,booking\_request c,vehicle d

where c.booking\_id= v\_booking\_id and c.cust\_id=a.cust\_id and c.car\_id=d.car\_id and d.clnt\_id=b.clnt\_id;

insert into accident\_log values(v\_cust\_name,v\_client\_name,v\_accident\_date);

exception

when others then

begin

insert into error\_log values('accident','accident\_trigger',sysdate,'after each row');

end;

end;

create or replace trigger customer\_mod\_trigger after update or insert or delete on customer for each row

declare

v\_change char(1);

begin

if inserting then

v\_change := 'I';

elsif updating then

v\_change := 'U';

else

v\_change := 'D';

end if;

insert into customer\_log values(user,sysdate,v\_change);

exception

when others then

insert into error\_log values('customer','customer\_mod\_trigger',sysdate,'after each row');

end;

end;

**Output:**

