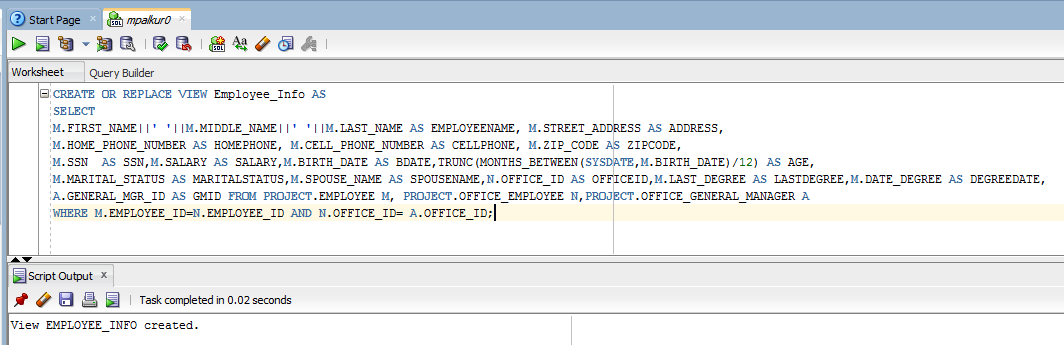
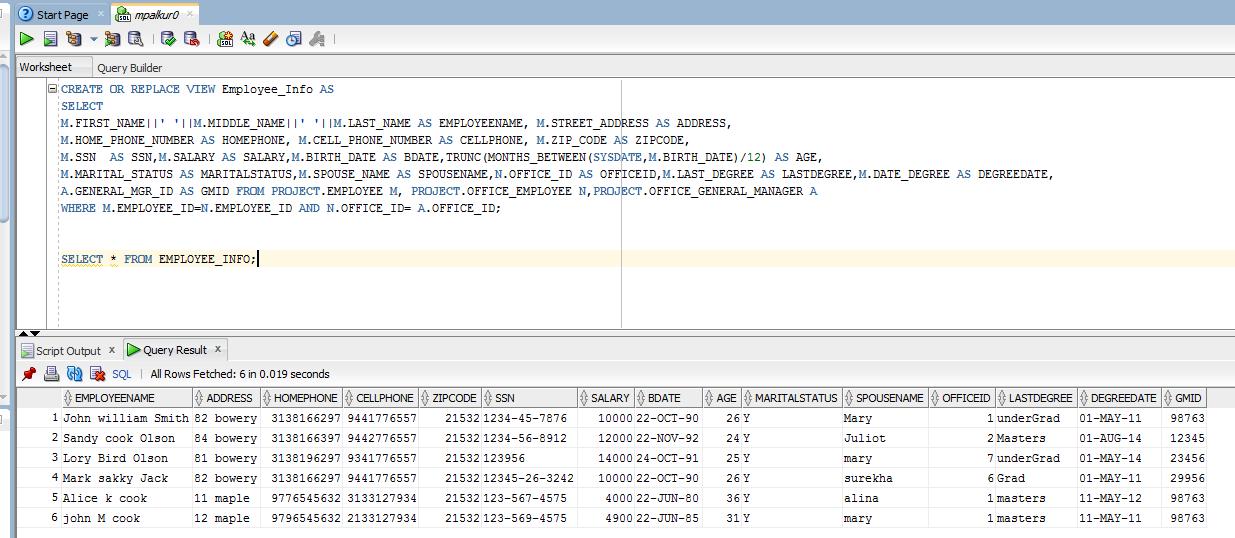
DATABASE PROJECT (FINAL)

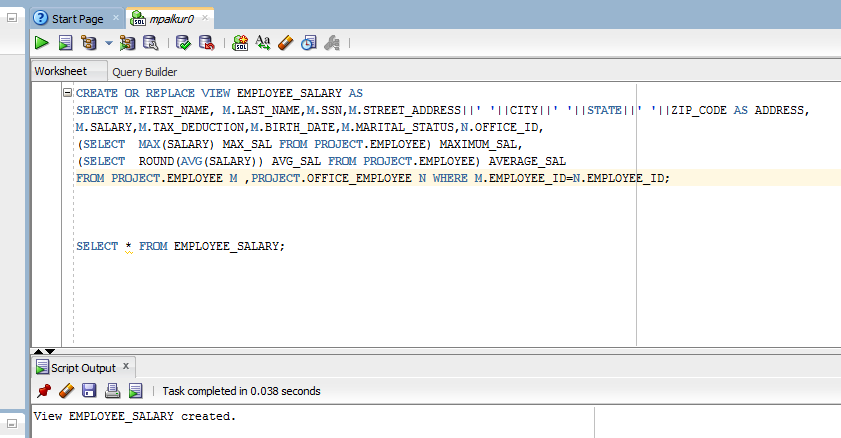
12. (A) EMPLOYEE\_INFO: -

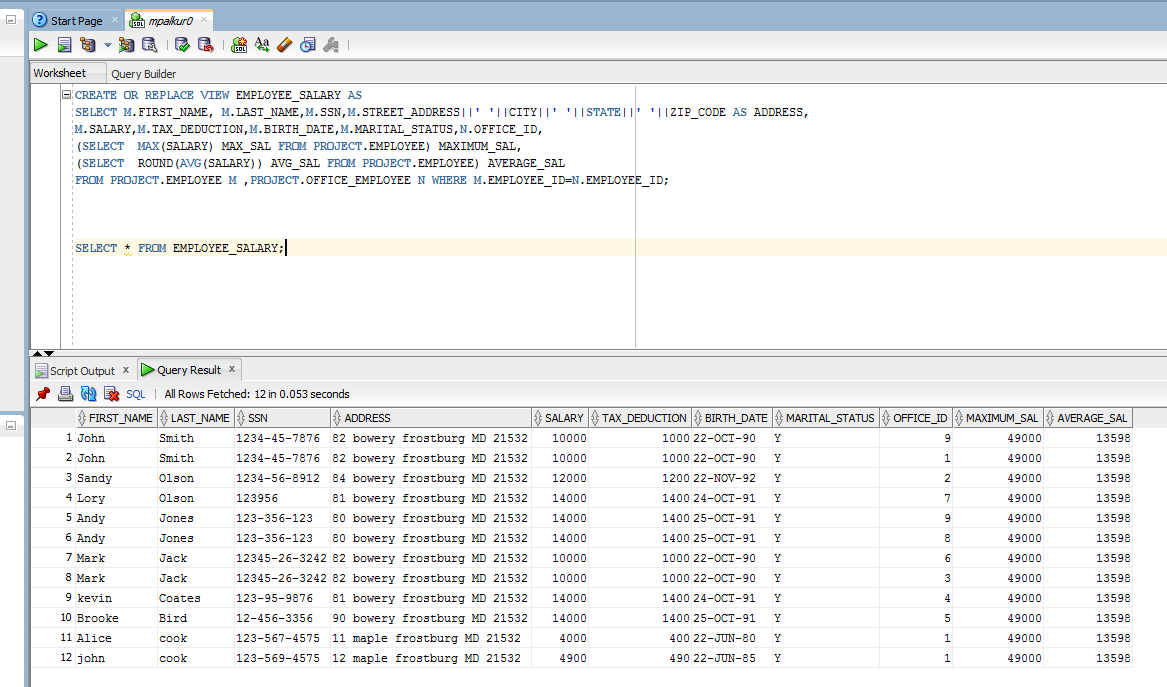
1.



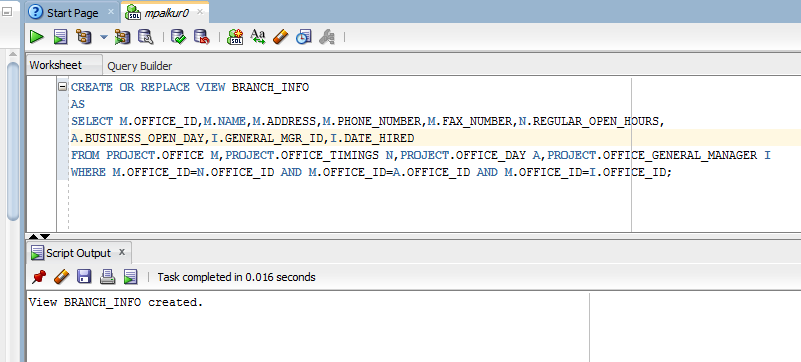


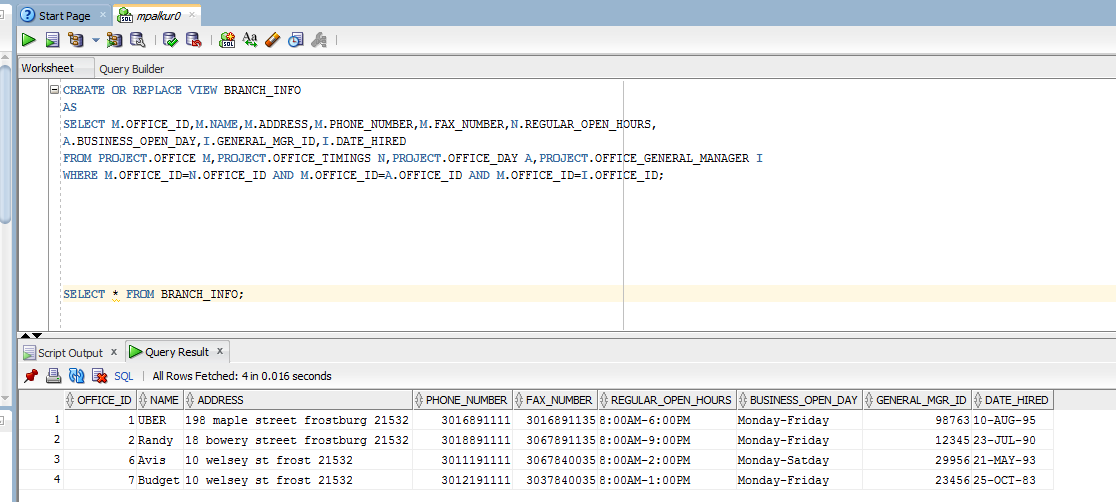
12. (B) EMPLOYEE\_SALARY: -





12. (C) BRANCH\_INFO: -





12. (D) VALUED\_CUSTOMERS: -

CREATE OR REPLACE VIEW VALUED\_CUSTOMERS

AS

SELECT C\_NAME,C\_AGE,CELL\_NUMBER,C\_ADDRESS,ZIPCODE,C\_EMAIL,C\_LICENSE\_NUMBER,LICENSE\_ISSUED\_STATE,NO\_OF\_TIMES\_CAR\_RENTED FROM

(SELECT CUSTOMER\_ID,FIRST\_NAME||' '||LAST\_NAME AS C\_NAME,(TO\_CHAR(SYSDATE,'yyyy')-TO\_CHAR(BIRTH\_DATE,'yyyy')) AS C\_AGE,

CELL\_PHONE AS CELL\_NUMBER,STREET\_ADDRESS||' '||CITY||' '||STATE AS C\_ADDRESS,ZIP\_CODE AS ZIPCODE,EMAIL AS C\_EMAIL,

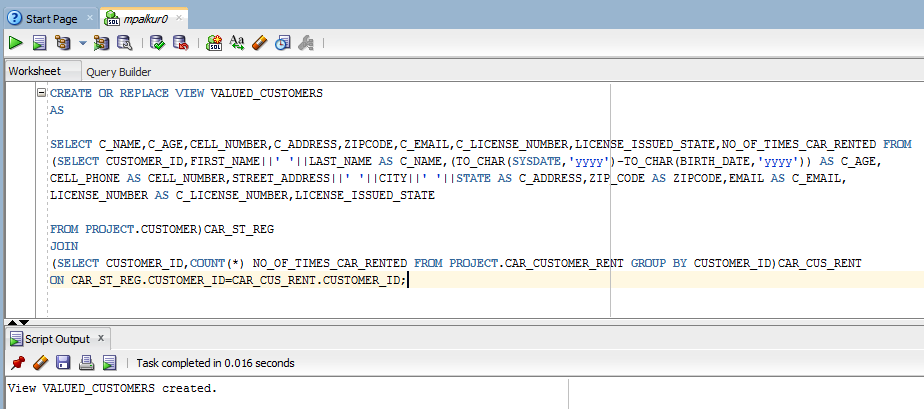
LICENSE\_NUMBER AS C\_LICENSE\_NUMBER,LICENSE\_ISSUED\_STATE

FROM PROJECT.CUSTOMER)CAR\_ST\_REG

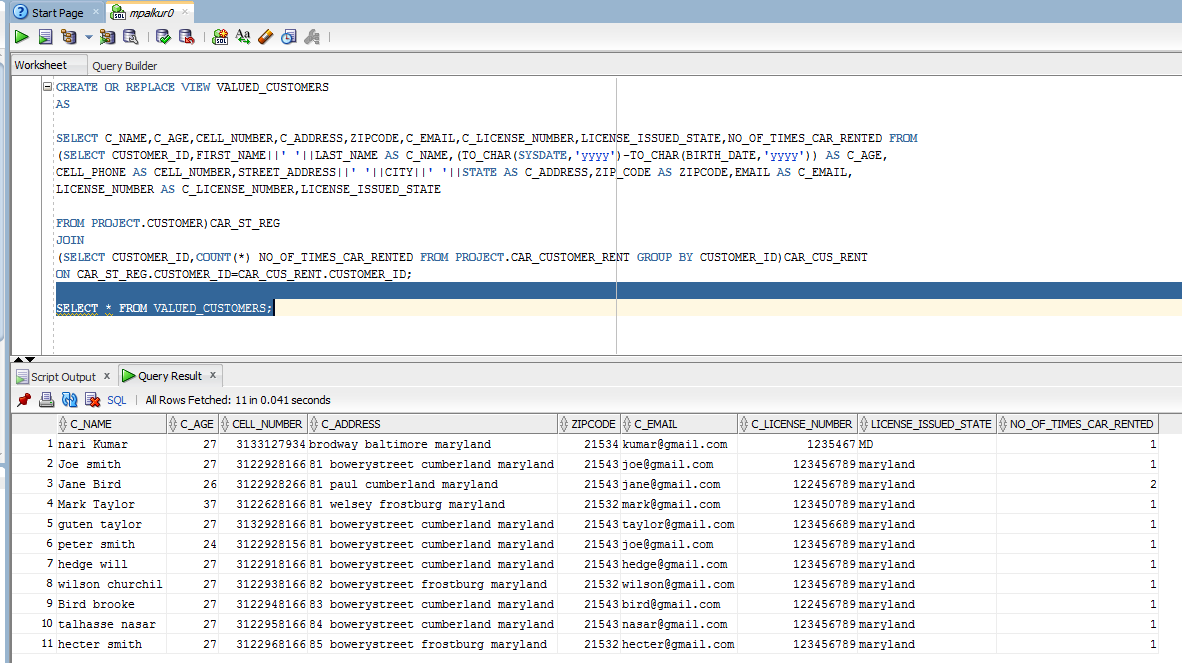
JOIN

(SELECT CUSTOMER\_ID,COUNT(\*) NO\_OF\_TIMES\_CAR\_RENTED FROM PROJECT.CAR\_CUSTOMER\_RENT GROUP BY CUSTOMER\_ID)CAR\_CUS\_RENT

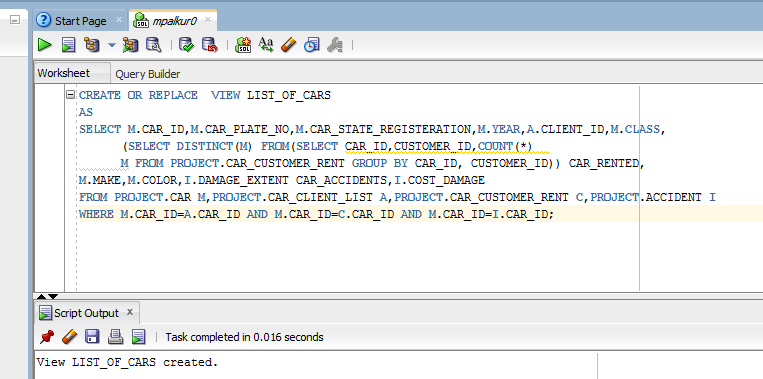
ON CAR\_ST\_REG.CUSTOMER\_ID=CAR\_CUS\_RENT.CUSTOMER\_ID;

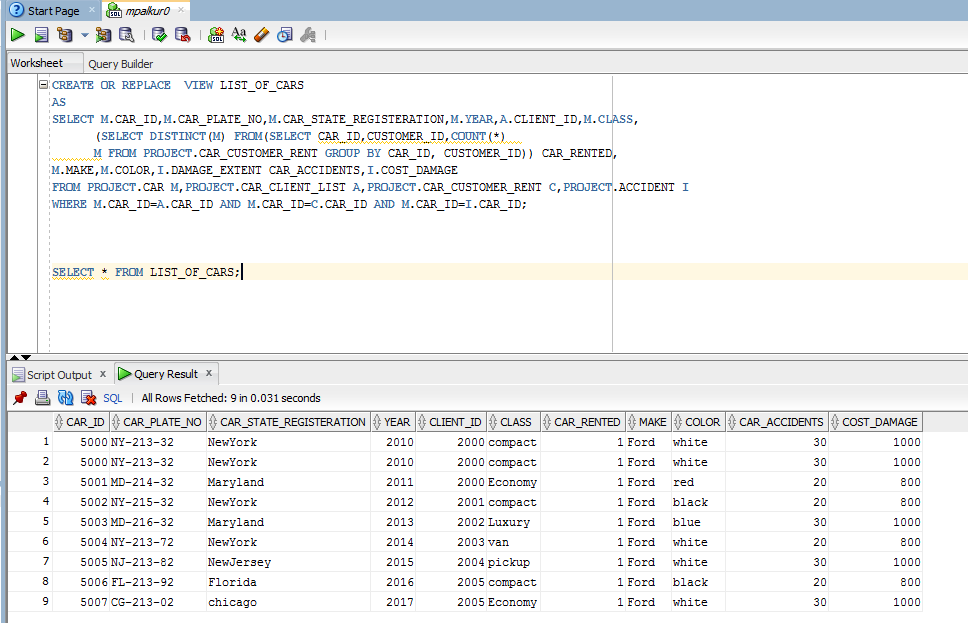


SELECT \* FROM VALUED\_CUSTOMERS;



12. (E) LIST\_OF\_CARS: -





12. (F) LIST\_OF\_TRANSACTIONS: -

CREATE OR REPLACE VIEW LIST\_OF\_TRANSACTIONS

AS

SELECT

(SELECT (END\_DATE - START\_DATE) FROM PROJECT.CAR\_CLIENT\_LIST) DAY\_CAR\_USED ,

M.PAYMENT\_AMOUNT,A.DAILY\_PRICE,

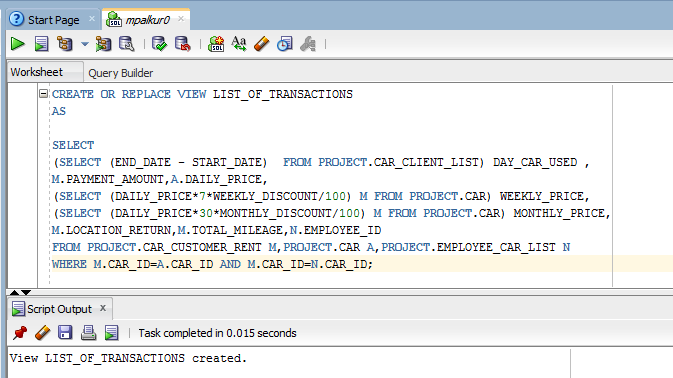
(SELECT (DAILY\_PRICE\*7\*WEEKLY\_DISCOUNT/100) M FROM PROJECT.CAR) WEEKLY\_PRICE,

(SELECT (DAILY\_PRICE\*30\*MONTHLY\_DISCOUNT/100) M FROM PROJECT.CAR) MONTHLY\_PRICE,

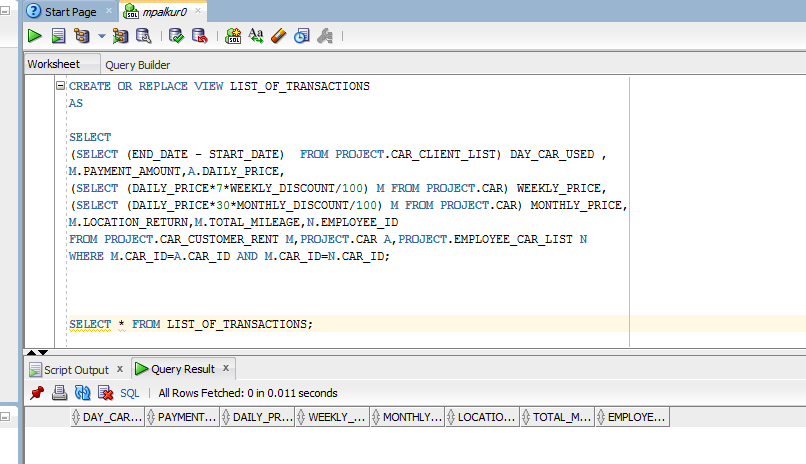
M.LOCATION\_RETURN,M.TOTAL\_MILEAGE,N.EMPLOYEE\_ID

FROM PROJECT.CAR\_CUSTOMER\_RENT M,PROJECT.CAR A,PROJECT.EMPLOYEE\_CAR\_LIST N

WHERE M.CAR\_ID=A.CAR\_ID AND M.CAR\_ID=N.CAR\_ID;



SELECT \* FROM LIST\_OF\_TRANSACTIONS;



12. (G) STATISTICS\_BY\_BRANCH: -

CREATE OR REPLACE VIEW STATISTICS\_BY\_BRANCH\_YEAR AS

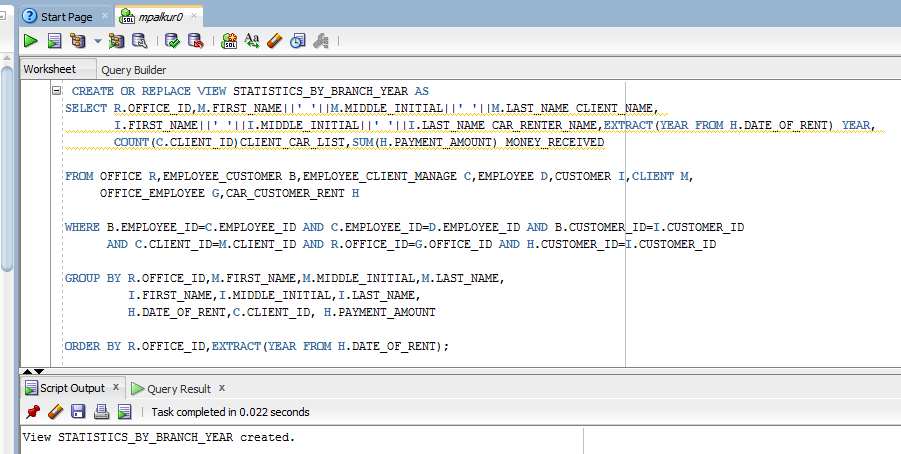
SELECT R.OFFICE\_ID,M.FIRST\_NAME||' '||M.MIDDLE\_INITIAL||' '||M.LAST\_NAME CLIENT\_NAME, I.FIRST\_NAME||' '||I.MIDDLE\_INITIAL||' '||I.LAST\_NAME CAR\_RENTER\_NAME,EXTRACT(YEAR FROM H.DATE\_OF\_RENT) YEAR,COUNT(C.CLIENT\_ID)CLIENT\_CAR\_LIST,SUM(H.PAYMENT\_AMOUNT) MONEY\_RECEIVED

FROM OFFICE R,EMPLOYEE\_CUSTOMER B,EMPLOYEE\_CLIENT\_MANAGE C,EMPLOYEE D,CUSTOMER I,CLIENT M,OFFICE\_EMPLOYEE G,CAR\_CUSTOMER\_RENT H

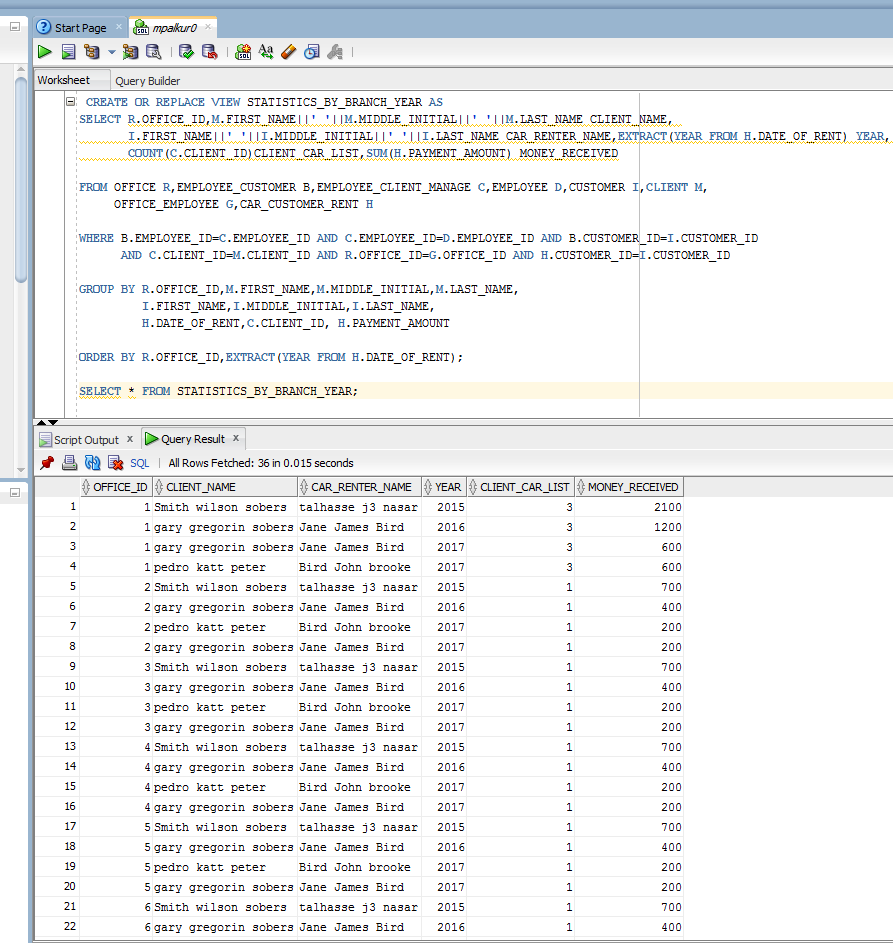
WHERE B.EMPLOYEE\_ID=C.EMPLOYEE\_ID AND C.EMPLOYEE\_ID=D.EMPLOYEE\_ID AND B.CUSTOMER\_ID=I.CUSTOMER\_ID AND C.CLIENT\_ID=M.CLIENT\_ID AND R.OFFICE\_ID=G.OFFICE\_ID AND H.CUSTOMER\_ID=I.CUSTOMER\_ID

GROUP BY R.OFFICE\_ID,M.FIRST\_NAME,M.MIDDLE\_INITIAL,M.LAST\_NAME, I.FIRST\_NAME,I.MIDDLE\_INITIAL,I.LAST\_NAME, H.DATE\_OF\_RENT,C.CLIENT\_ID, H.PAYMENT\_AMOUNT

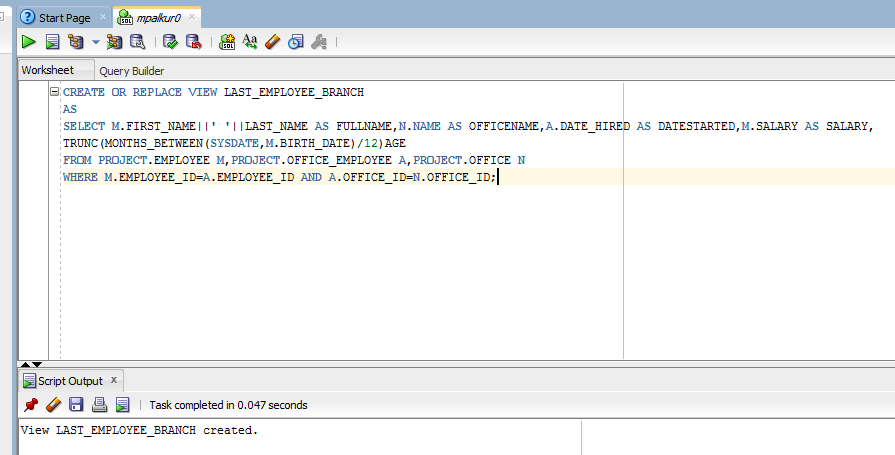
ORDER BY R.OFFICE\_ID,EXTRACT(YEAR FROM H.DATE\_OF\_RENT);

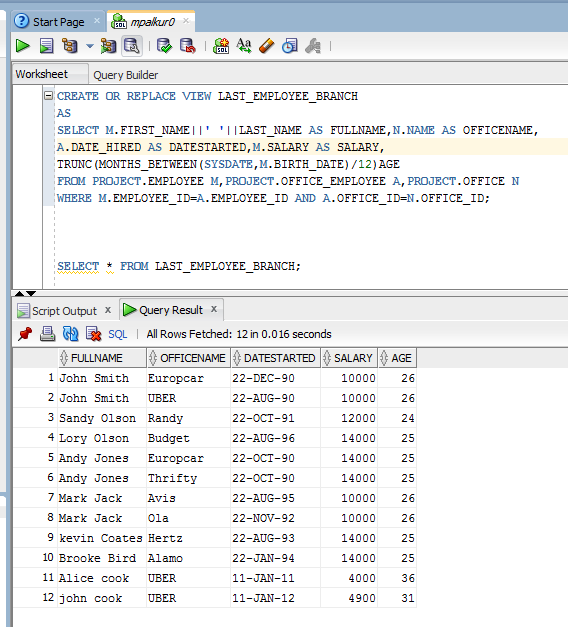


SELECT \* FROM STATISTICS\_BY\_BRANCH\_YEAR;



12. (H) LAST\_EMPLOYEE\_BRANCH : -





12. (I) MANAGERS: -

CREATE OR REPLACE FORCE VIEW MANAGERS AS

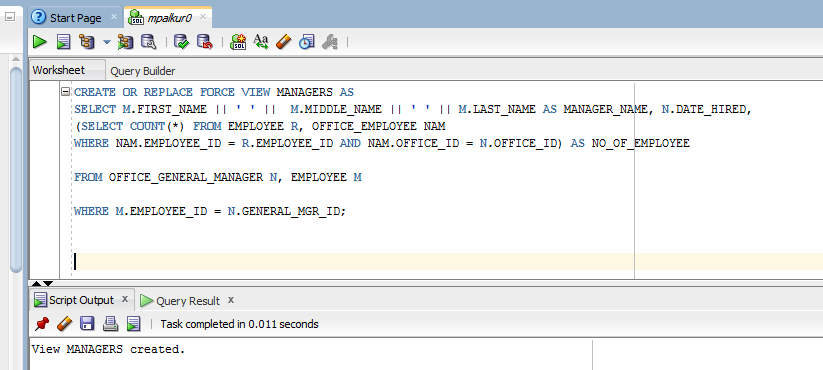
SELECT M.FIRST\_NAME || ' ' || M.MIDDLE\_NAME || ' ' || M.LAST\_NAME AS MANAGER\_NAME, N.DATE\_HIRED,

(SELECT COUNT (\*) FROM EMPLOYEE R, OFFICE\_EMPLOYEE NAM

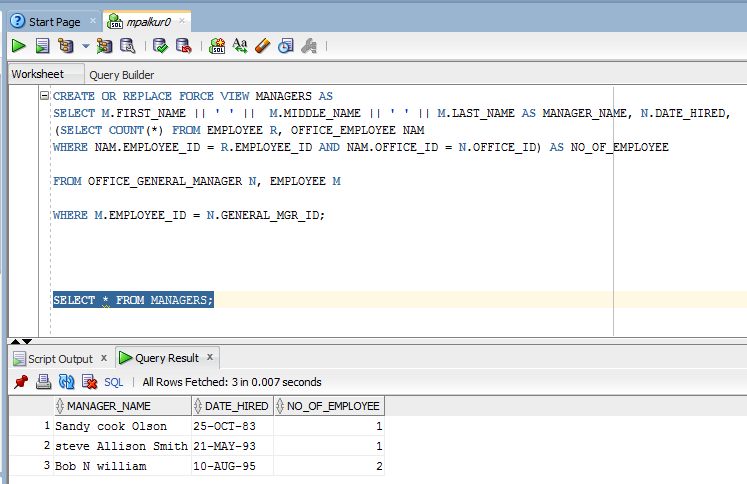
WHERE NAM.EMPLOYEE\_ID = R.EMPLOYEE\_ID AND NAM.OFFICE\_ID = N.OFFICE\_ID) AS NO\_OF\_EMPLOYEE

FROM OFFICE\_GENERAL\_MANAGER N, EMPLOYEE M

WHERE M.EMPLOYEE\_ID = N.GENERAL\_MGR\_ID;



SELECT \* FROM MANAGERS;



**12. (J)**

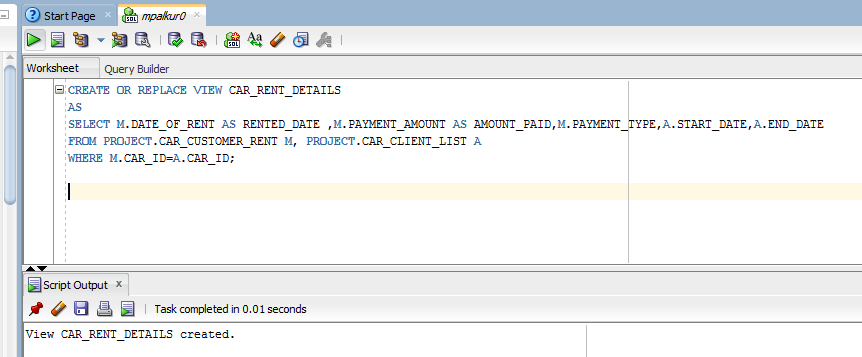
CREATE OR REPLACE VIEW CAR\_RENT\_DETAILS

AS

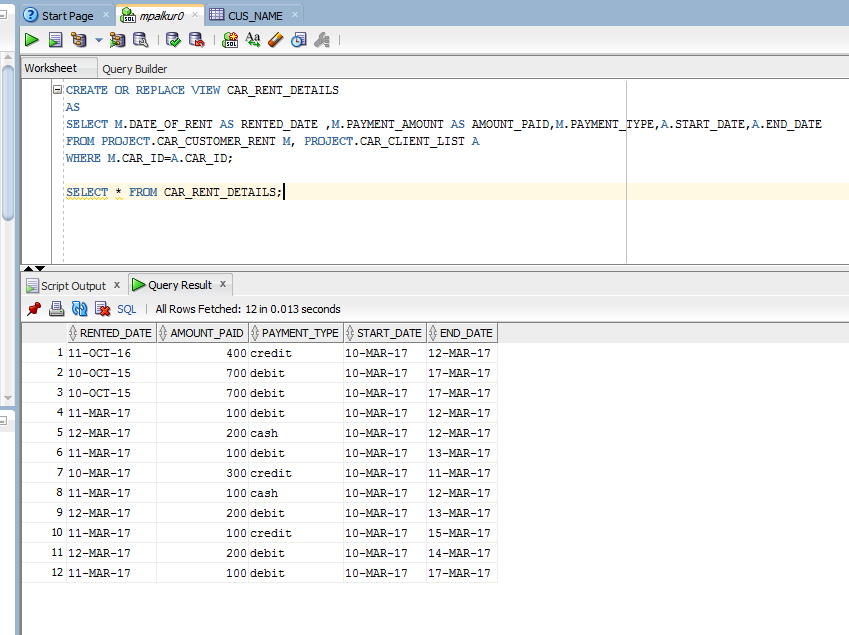
SELECT M.DATE\_OF\_RENT AS RENTED\_DATE, M.PAYMENT\_AMOUNT AS AMOUNT\_PAID, M.PAYMENT\_TYPE, A.START\_DATE, A.END\_DATE

FROM PROJECT.CAR\_CUSTOMER\_RENT M, PROJECT.CAR\_CLIENT\_LIST A

WHERE M.CAR\_ID=A.CAR\_ID;



SELECT \* FROM CAR\_RENT\_DETAILS;



**12. (K**)

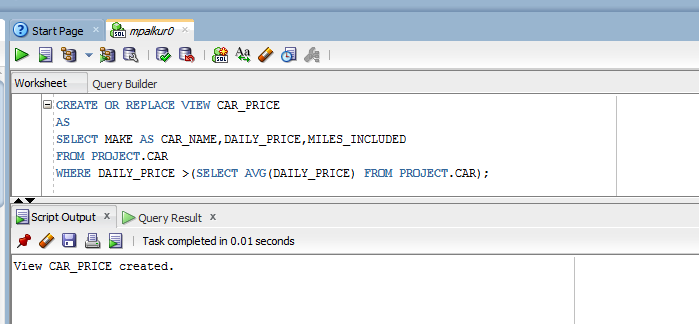
CREATE OR REPLACE VIEW CAR\_PRICE

AS

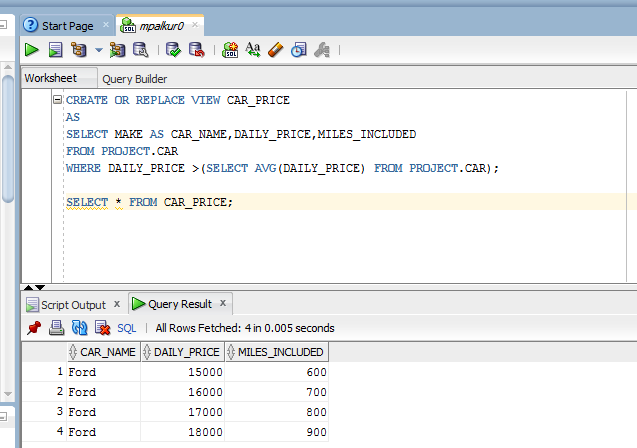
SELECT MAKE AS CAR\_NAME, DAILY\_PRICE, MILES\_INCLUDED

FROM PROJECT.CAR

WHERE DAILY\_PRICE > (SELECT AVG (DAILY\_PRICE) FROM PROJECT.CAR);



SELECT \* FROM CAR\_PRICE;



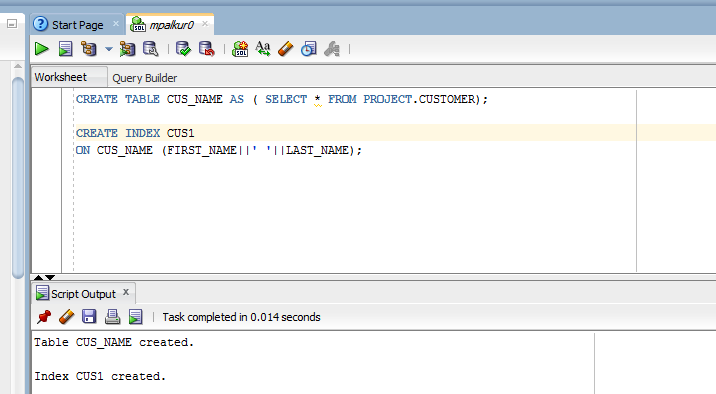
***13. INDEX: -***

A.

CREATE TABLE CUS\_NAME AS ( SELECT \* FROM PROJECT.CUSTOMER);

CREATE INDEX CUS1

ON CUS\_NAME (FIRST\_NAME||' '||LAST\_NAME);

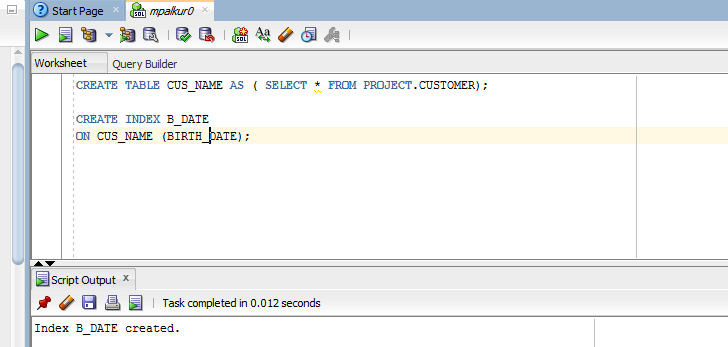


b. BIRTHDATE: -

CREATE TABLE CUS\_NAME AS ( SELECT \* FROM PROJECT.CUSTOMER);

CREATE INDEX B\_DATE

ON CUS\_NAME (BIRTH\_DATE);



***14. SEQUENCES: -***

**A.**

CREATE SEQUENCE ID\_GENERATOR

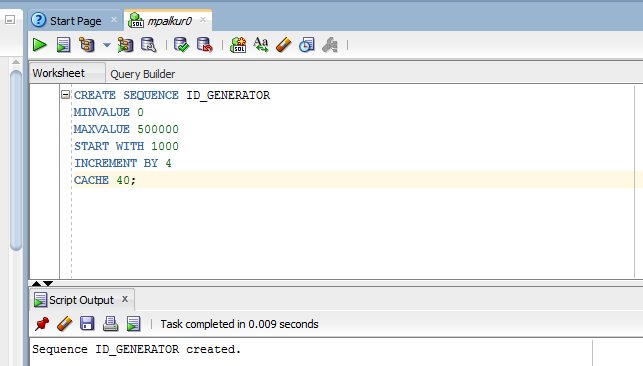
MINVALUE 0

MAXVALUE 500000

START WITH 1000

INCREMENT BY 4

CACHE 40;



**B.**

CREATE SEQUENCE TRANSACTION\_ID

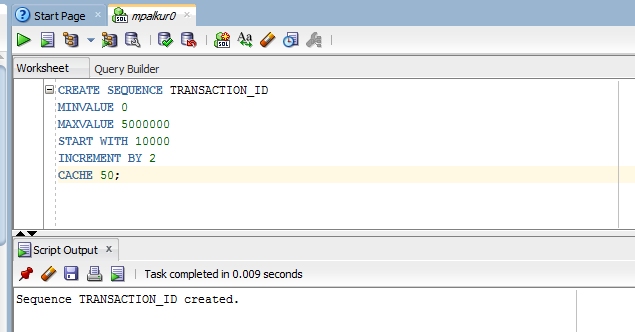
MINVALUE 0

MAXVALUE 5000000

START WITH 10000

INCREMENT BY 2

CACHE 50;



**C.**

CREATE SEQUENCE TRACK\_NUMBER

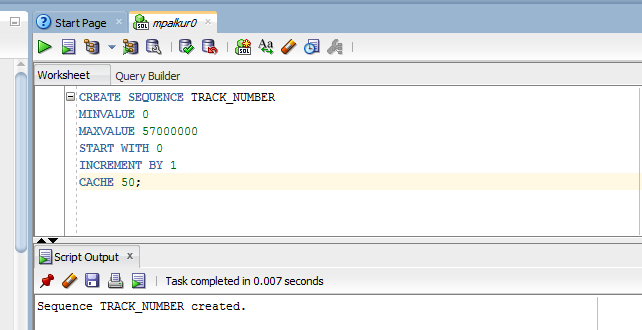
MINVALUE 0

MAXVALUE 57000000

START WITH 0

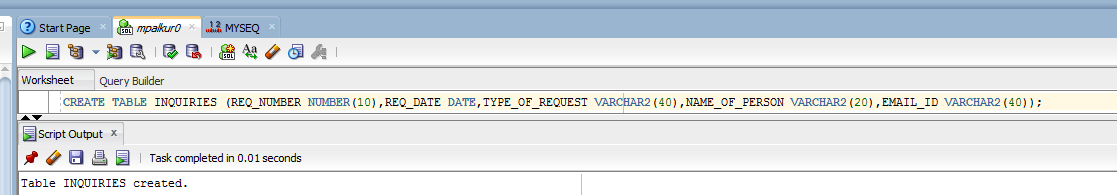
INCREMENT BY 1

CACHE 50;



**D.**

CREATE TABLE INQUIRIES (REQ\_NUMBER NUMBER (10), REQ\_DATE DATE, TYPE\_OF\_REQUEST VARCHAR2 (40), NAME\_OF\_PERSON VARCHAR2 (20), EMAIL\_ID VARCHAR2 (40));



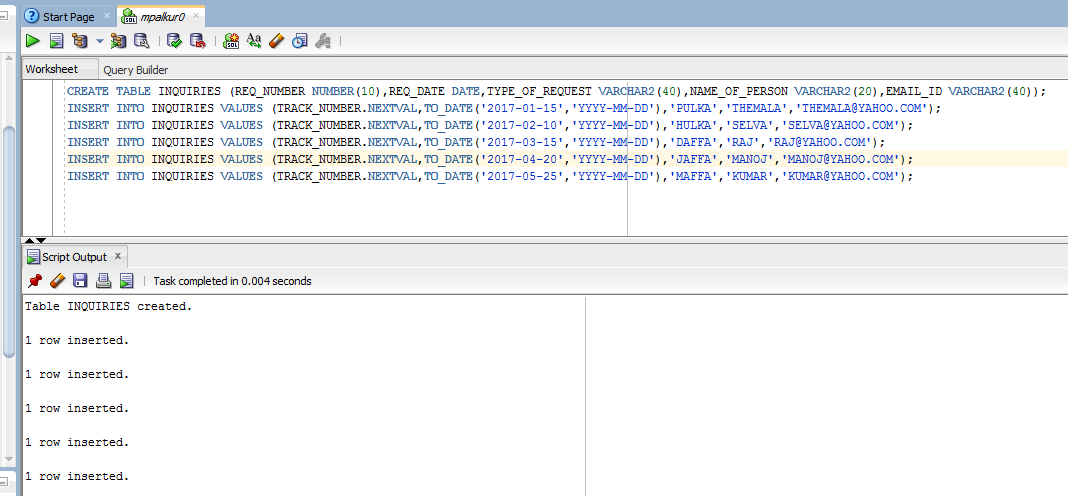
INSERT INTO INQUIRIES VALUES (TRACK\_NUMBER.NEXTVAL, TO\_DATE ('2017-01-15','YYYY-MM-DD'),'PULKA','THEMALA','THEMALA@YAHOO.COM');

INSERT INTO INQUIRIES VALUES (TRACK\_NUMBER.NEXTVAL, TO\_DATE ('2017-02-10','YYYY-MM-DD'),'HULKA','SELVA','SELVA@YAHOO.COM');

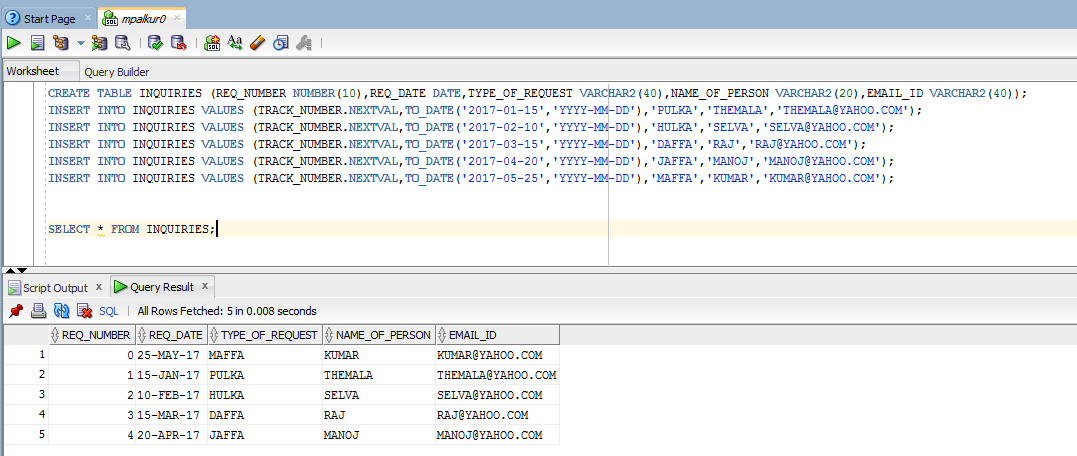
INSERT INTO INQUIRIES VALUES (TRACK\_NUMBER.NEXTVAL, TO\_DATE ('2017-03-15','YYYY-MM-DD'),'DAFFA','RAJ','RAJ@YAHOO.COM');

INSERT INTO INQUIRIES VALUES (TRACK\_NUMBER.NEXTVAL, TO\_DATE ('2017-04-20','YYYY-MM-DD'),'JAFFA','MANOJ','MANOJ@YAHOO.COM');

INSERT INTO INQUIRIES VALUES (TRACK\_NUMBER.NEXTVAL, TO\_DATE ('2017-05-25','YYYY-MM-DD'),'MAFFA','KUMAR','KUMAR@YAHOO.COM');



SELECT \* FROM INQUIRIES;



***15. SUBPROGRAMS & PACKAGES: -***

A. EMPINSERT\_PKG: -

CREATE OR REPLACE PACKAGE EMPINSERT\_PKG AS

PROCEDURE INSERT\_DATA

(EMPLOYEE\_ID EMPLOYEE.EMPLOYEE\_ID%TYPE,FIRST\_NAME EMPLOYEE.FIRST\_NAME%TYPE, MIDDLE\_NAME EMPLOYEE.MIDDLE\_NAME%TYPE,LAST\_NAME EMPLOYEE.LAST\_NAME%TYPE,GENDER EMPLOYEE.GENDER%TYPE,STREET\_ADDRESS EMPLOYEE.STREET\_ADDRESS%TYPE, CITYEMPLOYEE.CITY%TYPE,STATE EMPLOYEE.STATE%TYPE,ZIP\_CODE EMPLOYEE.ZIP\_CODE%TYPE,SSN EMPLOYEE.SSN%TYPE, SALARY EMPLOYEE.SALARY%TYPE,TAX\_REDUCTIONEMPLOYEE.TAX\_DEDUCTION%TYPE,BIRTH\_DATE EMPLOYEE.BIRTH\_DATE%TYPE,MARITAL\_STATUS EMPLOYEE.MARITAL\_STATUS%TYPE,SPOUSE\_NAME EMPLOYEE.SPOUSE\_NAME%TYPE,LAST\_DEGREE EMPLOYEE.LAST\_DEGREE%TYPE,

DATE\_DEGREE EMPLOYEE.DATE\_DEGREE%TYPE,NO\_OF\_DEPENDENTS EMPLOYEE.NO\_OF\_DEPENDENTS%TYPE,HOME\_PHONE\_NUM EMPLOYEE.HOME\_PHONE\_NUMBER%TYPE,CELL\_PHONE\_NUM EMPLOYEE.CELL\_PHONE\_NUMBER%TYPE);

PROCEDURE MARITAL\_STATUS\_UPDATE(A\_EMPLOYEE IN EMPLOYEE.EMPLOYEE\_ID%TYPE,A\_MARITAL\_STATUS EMPLOYEE.MARITAL\_STATUS%TYPE);

PROCEDURE LAST\_DEGREE\_UPDATE(A\_EMPLOYEE EMPLOYEE.EMPLOYEE\_ID%TYPE,A\_LAST\_DEGREE EMPLOYEE.LAST\_DEGREE%TYPE);

PROCEDURE CERTIFICATE\_UPDATE(A\_EMPLOYEE EMPLOYEE\_CERTIFICATES.EMPLOYEE\_ID%TYPE,A\_CERTIFICATE EMPLOYEE\_CERTIFICATES.LIST\_OF\_CERTIFICATES%TYPE);

PROCEDURE MANAGER\_UPDATE(MANAGER\_ID OFFICE\_GENERAL\_MANAGER.GENERAL\_MGR\_ID%TYPE,A\_EMPLOYEE EMPLOYEE.EMPLOYEE\_ID%TYPE);

END EMPINSERT\_PKG;

/

SET SERVEROUTPUT ON;

CREATE OR REPLACE PACKAGE BODY EMPINSERT\_PKG AS

PROCEDURE INSERT\_DATA

(EMPLOYEE\_ID EMPLOYEE.EMPLOYEE\_ID%TYPE,FIRST\_NAME EMPLOYEE.FIRST\_NAME%TYPE,MIDDLE\_NAME EMPLOYEE.MIDDLE\_NAME%TYPE,LAST\_NAME EMPLOYEE.LAST\_NAME%TYPE,GENDER EMPLOYEE.GENDER%TYPE,STREET\_ADDRESS EMPLOYEE.STREET\_ADDRESS%TYPE,CITY EMPLOYEE.CITY%TYPE,STATE EMPLOYEE.STATE%TYPE,ZIP\_CODE EMPLOYEE.ZIP\_CODE%TYPE,SSN EMPLOYEE.SSN%TYPE,SALARY EMPLOYEE.SALARY%TYPE,TAX\_REDUCTION EMPLOYEE.TAX\_DEDUCTION%TYPE,BIRTH\_DATE EMPLOYEE.BIRTH\_DATE%TYPE,MARITAL\_STATUS EMPLOYEE.MARITAL\_STATUS%TYPE,SPOUSE\_NAME EMPLOYEE.SPOUSE\_NAME%TYPE,LAST\_DEGREE EMPLOYEE.LAST\_DEGREE%TYPE,DATE\_DEGREE EMPLOYEE.DATE\_DEGREE%TYPE,NO\_OF\_DEPENDENTS EMPLOYEE.NO\_OF\_DEPENDENTS%TYPE,HOME\_PHONE\_NUM EMPLOYEE.HOME\_PHONE\_NUMBER%TYPE,

CELL\_PHONE\_NUM EMPLOYEE.CELL\_PHONE\_NUMBER%TYPE) AS

BEGIN

INSERT INTO EMPLOYEE (EMPLOYEE\_ID,FIRST\_NAME,MIDDLE\_NAME,LAST\_NAME,GENDER,STREET\_ADDRESS,CITY,STATE,ZIP\_CODE,SSN,SALARY,TAX\_DEDUCTION,BIRTH\_DATE,MARITAL\_STATUS,

SPOUSE\_NAME,LAST\_DEGREE,DATE\_DEGREE,NO\_OF\_DEPENDENTS,HOME\_PHONE\_NUMBER,CELL\_PHONE\_NUMBER)

VALUES(EMPLOYEE\_ID,FIRST\_NAME,MIDDLE\_NAME,LAST\_NAME,GENDER,STREET\_ADDRESS,CITY,STATE,ZIP\_CODE,SSN,SALARY,TAX\_REDUCTION,BIRTH\_DATE,MARITAL\_STATUS,

SPOUSE\_NAME,LAST\_DEGREE,DATE\_DEGREE,NO\_OF\_DEPENDENTS,HOME\_PHONE\_NUM,CELL\_PHONE\_NUM);

DBMS\_OUTPUT.PUT\_LINE('EMPLOYEE TABLE UPDATED');

END;

PROCEDURE MARITAL\_STATUS\_UPDATE(A\_EMPLOYEE IN EMPLOYEE.EMPLOYEE\_ID%TYPE,A\_MARITAL\_STATUS EMPLOYEE.MARITAL\_STATUS%TYPE) AS

BEGIN

UPDATE EMPLOYEE SET MARITAL\_STATUS = A\_MARITAL\_STATUS WHERE EMPLOYEE\_ID=A\_EMPLOYEE;

DBMS\_OUTPUT.PUT\_LINE('MARITAL STATUS UPDATED');

END;

PROCEDURE LAST\_DEGREE\_UPDATE(A\_EMPLOYEE EMPLOYEE.EMPLOYEE\_ID%TYPE,A\_LAST\_DEGREE EMPLOYEE.LAST\_DEGREE%TYPE) AS

BEGIN

UPDATE EMPLOYEE SET LAST\_DEGREE = A\_LAST\_DEGREE WHERE EMPLOYEE\_ID=A\_EMPLOYEE;

DBMS\_OUTPUT.PUT\_LINE('LAST DEGREE UPDATED');

END;

PROCEDURE CERTIFICATE\_UPDATE(A\_EMPLOYEE EMPLOYEE\_CERTIFICATES.EMPLOYEE\_ID%TYPE,A\_CERTIFICATE EMPLOYEE\_CERTIFICATES.LIST\_OF\_CERTIFICATES%TYPE) AS

BEGIN

UPDATE EMPLOYEE\_CERTIFICATES SET LIST\_OF\_CERTIFICATES = A\_CERTIFICATE WHERE EMPLOYEE\_ID=A\_EMPLOYEE;

DBMS\_OUTPUT.PUT\_LINE('EMPLOYEE CERTIFICATE UPDATED');

END;

PROCEDURE MANAGER\_UPDATE(MANAGER\_ID OFFICE\_GENERAL\_MANAGER.GENERAL\_MGR\_ID%TYPE,A\_EMPLOYEE EMPLOYEE.EMPLOYEE\_ID%TYPE) AS

BEGIN

UPDATE OFFICE\_GENERAL\_MANAGER SET GENERAL\_MGR\_ID = A\_EMPLOYEE WHERE GENERAL\_MGR\_ID = MANAGER\_ID;

DBMS\_OUTPUT.PUT\_LINE('EMPLOYEE MANAGER UPDATED');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('NO DATA FOUND');

WHEN TOO\_MANY\_ROWS THEN

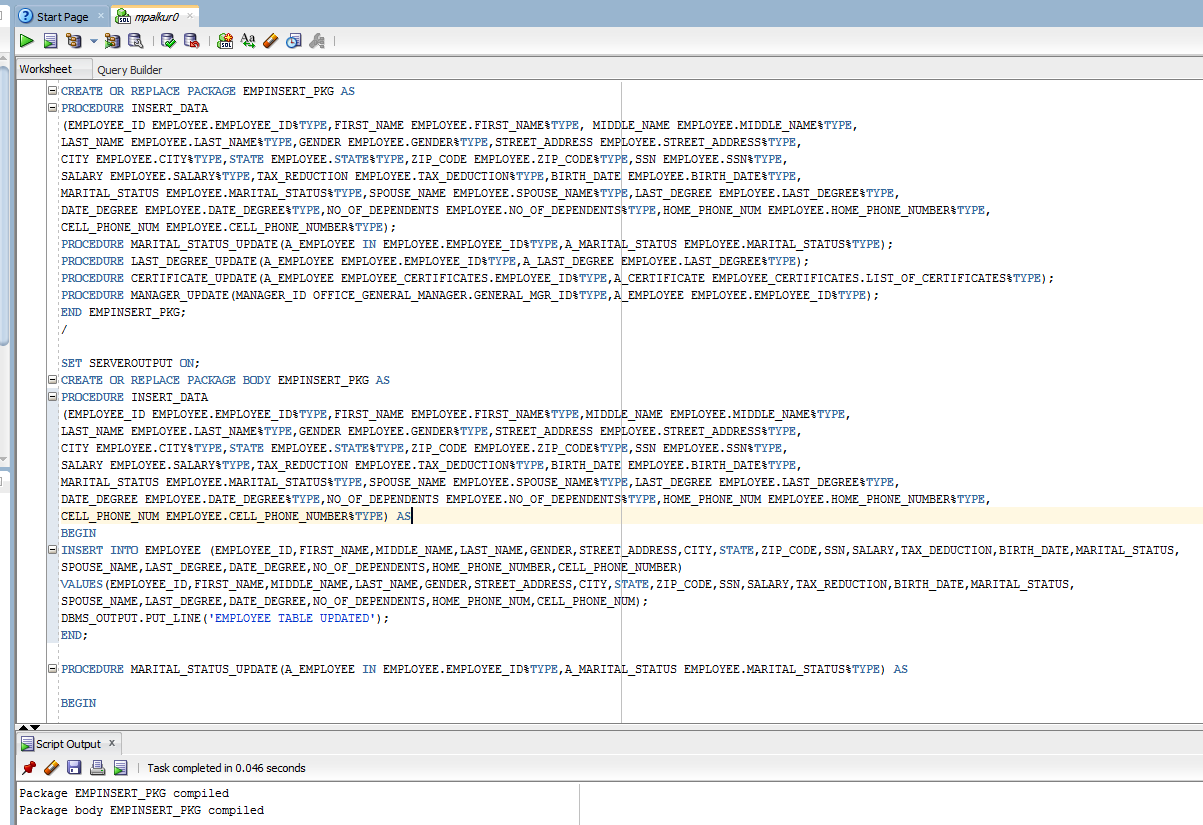
DBMS\_OUTPUT.PUT\_LINE('TOO MANY ROWS');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('ERROR');

END;

END EMPINSERT\_PKG;



**B. INSERT\_PKG**

CREATE OR REPLACE PACKAGE INSERT\_PKG AS

PROCEDURE INSERT\_DATA\_1

(CUSTOMERID CUSTOMER.CUSTOMER\_ID%TYPE,FIRST\_NAME CUSTOMER.FIRST\_NAME%TYPE,MIDDLE\_NAME CUSTOMER.MIDDLE\_INITIAL%TYPE,

LAST\_NAME CUSTOMER.LAST\_NAME%TYPE,STREET\_ADDRESS CUSTOMER.STREET\_ADDRESS%TYPE,CITY CUSTOMER.CITY%TYPE,

STATE CUSTOMER.STATE%TYPE,ZIP\_CODE CUSTOMER.ZIP\_CODE%TYPE,B\_DATE CUSTOMER.BIRTH\_DATE%TYPE,

EMAIL CUSTOMER.EMAIL%TYPE,LICENSEISSUED\_STATE CUSTOMER.LICENSE\_ISSUED\_STATE%TYPE,

LICENSE\_NO CUSTOMER.LICENSE\_NUMBER%TYPE,CR\_CARD\_NO CUSTOMER.CREDIT\_CARD\_NUMBER%TYPE,

EXPIRATION\_DATE CUSTOMER.EXPIRATION\_DATE%TYPE,CELL\_PHONENUMBER CUSTOMER.CELL\_PHONE%TYPE);

PROCEDURE INSERT\_NEW\_CAR\_OWNER(CAR\_ID CAR\_CLIENT\_LIST.CAR\_ID%TYPE,CLIENT\_ID CAR\_CLIENT\_LIST.CLIENT\_ID%TYPE,

AVAILABLE\_HRS CAR\_CLIENT\_LIST.AVAILABLE\_HRS%TYPE,START\_DATE CAR\_CLIENT\_LIST.START\_DATE%TYPE,

END\_DATE CAR\_CLIENT\_LIST.END\_DATE%TYPE);

PROCEDURE INFO\_UPDATE\_CUSTOMER(CUST\_ID CUSTOMER.CUSTOMER\_ID%TYPE,STREET CUSTOMER.STREET\_ADDRESS%TYPE);

PROCEDURE INFO\_UPDATE\_CAROWNER(EXISTING\_CAR\_ID CAR\_CLIENT\_LIST.CAR\_ID%TYPE,NEW\_CLIENT\_ID CAR\_CLIENT\_LIST.CLIENT\_ID%TYPE);

PROCEDURE INSERT\_ACCIDENT(CUSTOMER\_ID ACCIDENT.CUSTOMER\_ID%TYPE,CAR\_ID ACCIDENT.CAR\_ID%TYPE,DATE\_TIME ACCIDENT.DATE\_TIME%TYPE,

LOCATION\_1 ACCIDENT.LOCATION%TYPE,DAMAGE\_EXTENT ACCIDENT.DAMAGE\_EXTENT%TYPE,COST\_DAMAGE ACCIDENT.COST\_DAMAGE%TYPE,

POLICE\_REPORT ACCIDENT.POLICE\_REPORT%TYPE,SUMMARY\_1 ACCIDENT.SUMMARY\_POLICE\_REPORT%TYPE);

END INSERT\_PKG;

/

SET SERVEROUTPUT ON;

CREATE OR REPLACE PACKAGE BODY INSERT\_PKG AS

PROCEDURE INSERT\_DATA\_1

(CUSTOMERID CUSTOMER.CUSTOMER\_ID%TYPE,

FIRST\_NAME CUSTOMER.FIRST\_NAME%TYPE,MIDDLE\_NAME CUSTOMER.MIDDLE\_INITIAL%TYPE,LAST\_NAME CUSTOMER.LAST\_NAME%TYPE,

STREET\_ADDRESS CUSTOMER.STREET\_ADDRESS%TYPE,CITY CUSTOMER.CITY%TYPE,STATE CUSTOMER.STATE%TYPE,ZIP\_CODE CUSTOMER.ZIP\_CODE%TYPE,

B\_DATE CUSTOMER.BIRTH\_DATE%TYPE,EMAIL CUSTOMER.EMAIL%TYPE,LICENSEISSUED\_STATE CUSTOMER.LICENSE\_ISSUED\_STATE%TYPE,

LICENSE\_NO CUSTOMER.LICENSE\_NUMBER%TYPE,CR\_CARD\_NO CUSTOMER.CREDIT\_CARD\_NUMBER%TYPE,EXPIRATION\_DATE CUSTOMER.EXPIRATION\_DATE%TYPE,

CELL\_PHONENUMBER CUSTOMER.CELL\_PHONE%TYPE) AS

BEGIN

INSERT INTO CUSTOMER (CUSTOMER\_ID,FIRST\_NAME,MIDDLE\_INITIAL,LAST\_NAME,STREET\_ADDRESS,CITY,STATE,ZIP\_CODE,BIRTH\_DATE,EMAIL,

LICENSE\_ISSUED\_STATE,LICENSE\_NUMBER,CREDIT\_CARD\_NUMBER,EXPIRATION\_DATE,CELL\_PHONE)

VALUES(CUSTOMERID,FIRST\_NAME,MIDDLE\_NAME,LAST\_NAME,STREET\_ADDRESS,CITY,STATE,ZIP\_CODE,B\_DATE,EMAIL,LICENSEISSUED\_STATE,LICENSE\_NO,CR\_CARD\_NO,EXPIRATION\_DATE,CELL\_PHONENUMBER);

DBMS\_OUTPUT.PUT\_LINE('CUSTOMER TABLE UPDATED');

END;

PROCEDURE INSERT\_NEW\_CAR\_OWNER(CAR\_ID CAR\_CLIENT\_LIST.CAR\_ID%TYPE,CLIENT\_ID CAR\_CLIENT\_LIST.CLIENT\_ID%TYPE,

AVAILABLE\_HRS CAR\_CLIENT\_LIST.AVAILABLE\_HRS%TYPE,START\_DATE CAR\_CLIENT\_LIST.START\_DATE%TYPE,

END\_DATE CAR\_CLIENT\_LIST.END\_DATE%TYPE) AS

BEGIN

INSERT INTO CAR\_CLIENT\_LIST(CAR\_ID,CLIENT\_ID,AVAILABLE\_HRS,START\_DATE,END\_DATE)

VALUES(CAR\_ID,CLIENT\_ID,AVAILABLE\_HRS,START\_DATE,END\_DATE);

DBMS\_OUTPUT.PUT\_LINE('NEW CAR OWNER UPDATED');

END;

PROCEDURE INFO\_UPDATE\_CUSTOMER(CUST\_ID CUSTOMER.CUSTOMER\_ID%TYPE,STREET CUSTOMER.STREET\_ADDRESS%TYPE) AS

BEGIN

UPDATE CUSTOMER SET STREET\_ADDRESS = STREET WHERE CUSTOMER\_ID = CUST\_ID;

DBMS\_OUTPUT.PUT\_LINE('CUSTOMER INFO UPDATED');

END;

PROCEDURE INFO\_UPDATE\_CAROWNER(EXISTING\_CAR\_ID CAR\_CLIENT\_LIST.CAR\_ID%TYPE,NEW\_CLIENT\_ID CAR\_CLIENT\_LIST.CLIENT\_ID%TYPE) AS

BEGIN

UPDATE CAR\_CLIENT\_LIST SET CLIENT\_ID = NEW\_CLIENT\_ID WHERE CAR\_ID = EXISTING\_CAR\_ID;

DBMS\_OUTPUT.PUT\_LINE('CARQWNER UPDATED');

END;

PROCEDURE INSERT\_ACCIDENT(CUSTOMER\_ID ACCIDENT.CUSTOMER\_ID%TYPE,CAR\_ID ACCIDENT.CAR\_ID%TYPE,DATE\_TIME ACCIDENT.DATE\_TIME%TYPE,

LOCATION\_1 ACCIDENT.LOCATION%TYPE,DAMAGE\_EXTENT ACCIDENT.DAMAGE\_EXTENT%TYPE,COST\_DAMAGE ACCIDENT.COST\_DAMAGE%TYPE,

POLICE\_REPORT ACCIDENT.POLICE\_REPORT%TYPE,SUMMARY\_1 ACCIDENT.SUMMARY\_POLICE\_REPORT%TYPE) AS

BEGIN

INSERT INTO ACCIDENT(CUSTOMER\_ID,CAR\_ID,DATE\_TIME,LOCATION,DAMAGE\_EXTENT,COST\_DAMAGE,POLICE\_REPORT,SUMMARY\_POLICE\_REPORT)

VALUES (CUSTOMER\_ID,CAR\_ID,DATE\_TIME,LOCATION\_1,DAMAGE\_EXTENT,COST\_DAMAGE,POLICE\_REPORT,SUMMARY\_1);

DBMS\_OUTPUT.PUT\_LINE('NEW ACCIDENT UPDATED');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('NO DATA FOUND');

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('TOO MANY ROWS');

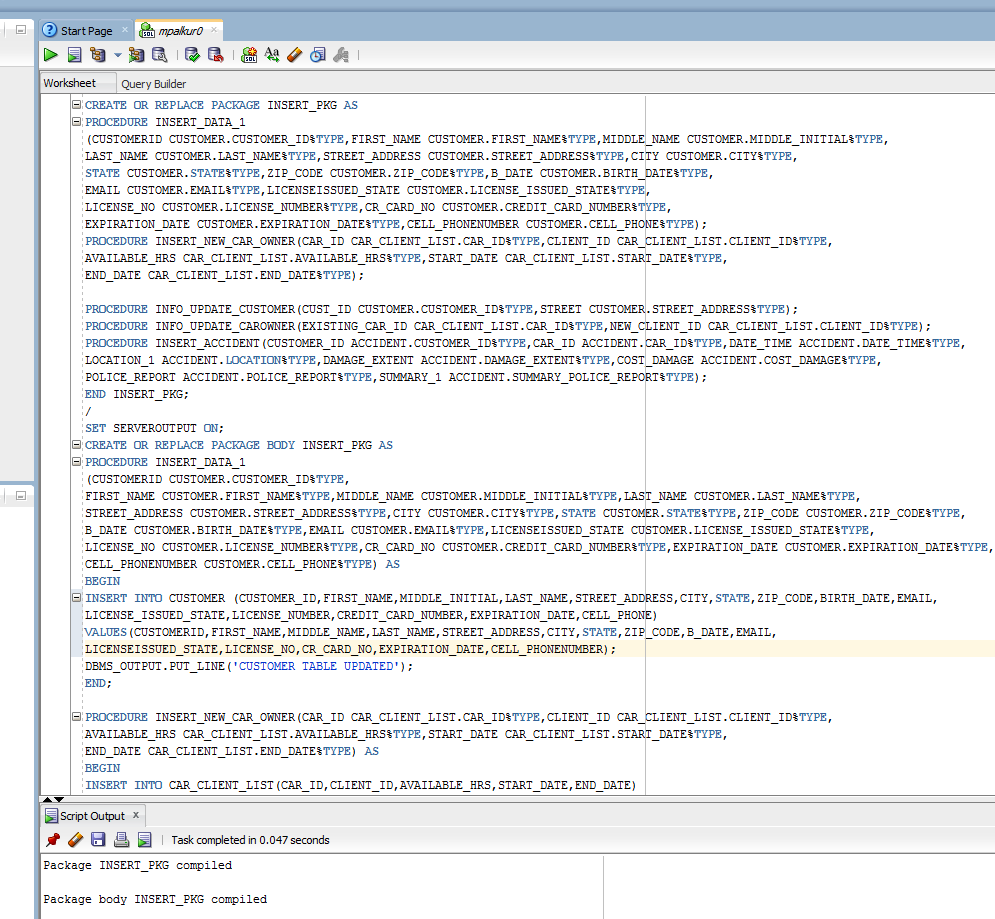
WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('ERROR');

END;

END INSERT\_PKG;

/



***16. Triggers: -***

G.

CREATE TABLE EMPLOYEE\_HISTORY1 AS SELECT \* FROM PROJECT.EMPLOYEE;

CREATE TABLE EMPLOYEE\_HISTORY1 (EMPLOYEE\_ID NUMBER,CITY VARCHAR2(20),STATE VARCHAR2(15),SSN NUMBER);

CREATE OR REPLACE TRIGGER EMPLOYEE\_HISTORY1

AFTER DELETE ON EMPLOYEE

for each row

BEGIN

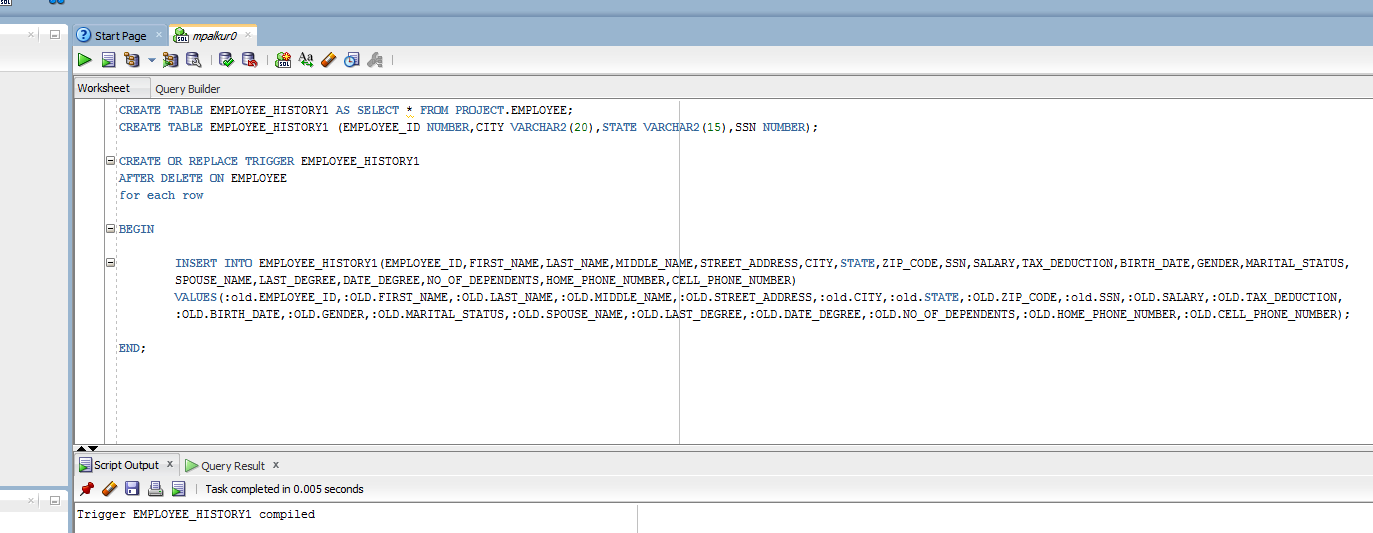
INSERT INTO EMPLOYEE\_HISTORY1(EMPLOYEE\_ID,FIRST\_NAME,LAST\_NAME,MIDDLE\_NAME,STREET\_ADDRESS,CITY,STATE,ZIP\_CODE,SSN,SALARY,TAX\_DEDUCTION,BIRTH\_DATE,GENDER,MARITAL\_STATUS,

SPOUSE\_NAME,LAST\_DEGREE,DATE\_DEGREE,NO\_OF\_DEPENDENTS,HOME\_PHONE\_NUMBER,CELL\_PHONE\_NUMBER)

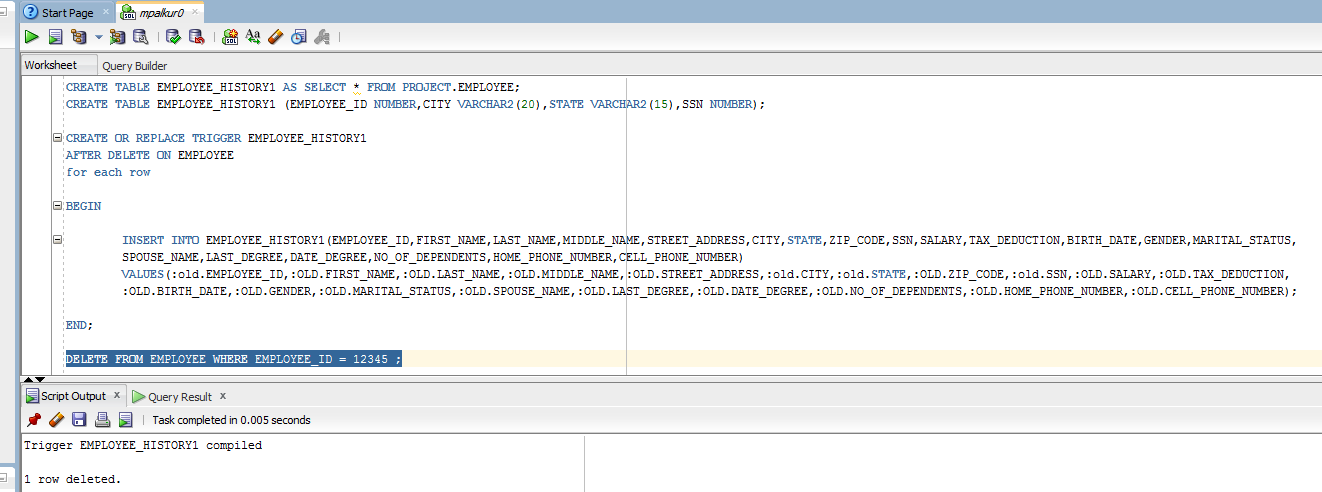
VALUES(:old.EMPLOYEE\_ID,:OLD.FIRST\_NAME,:OLD.LAST\_NAME,:OLD.MIDDLE\_NAME,:OLD.STREET\_ADDRESS,:old.CITY,:old.STATE,:OLD.ZIP\_CODE,:old.SSN,:OLD.SALARY,:OLD.TAX\_DEDUCTION,

: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);

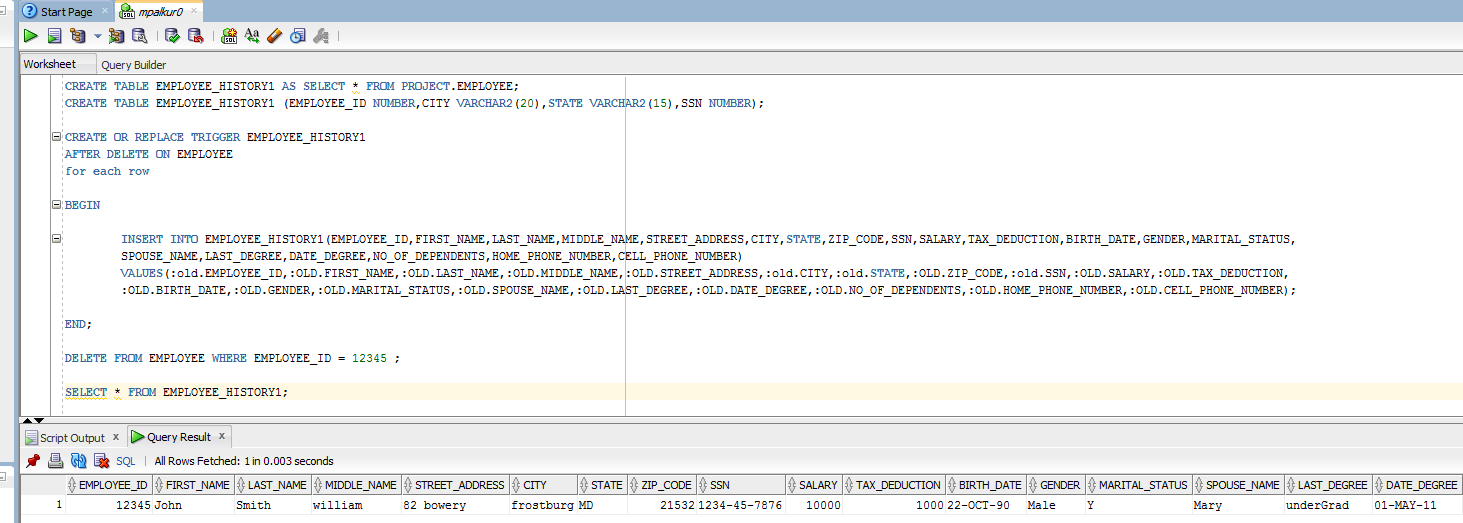
END;

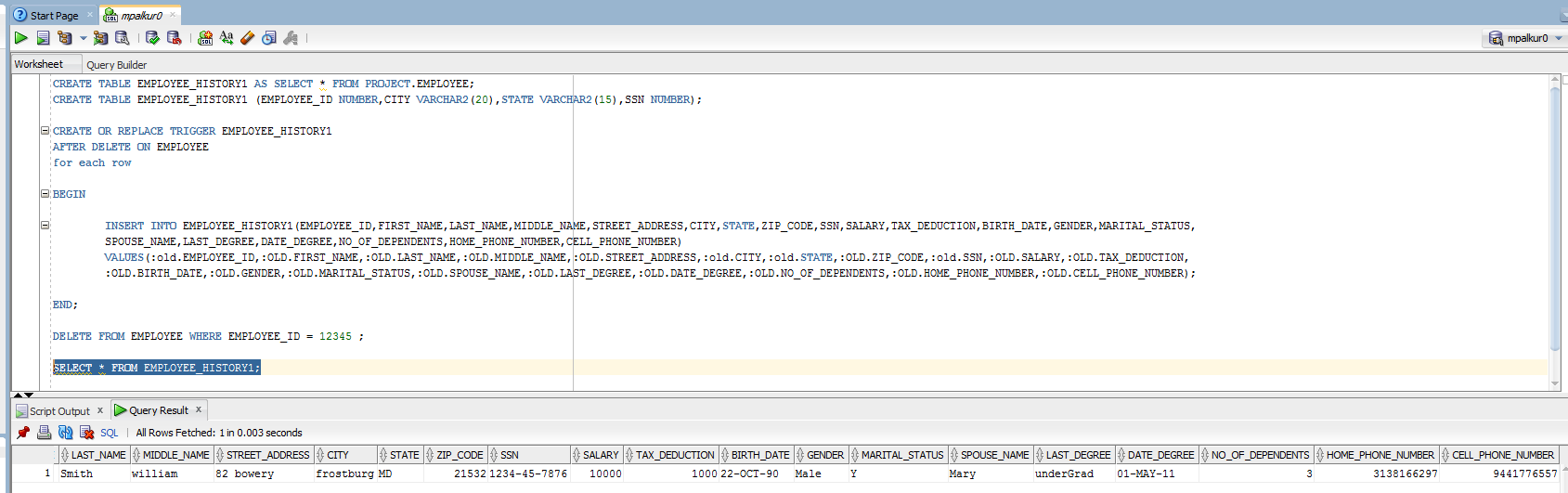


DELETE FROM EMPLOYEE WHERE EMPLOYEE\_ID = 12345;



SELECT \* FROM EMPLOYEE\_HISTORY1;





H. UPDATING: -

CREATE TABLE CUSTOMER\_LOG AS SELECT \* FROM PROJECT.CUSTOMER;

CREATE TABLE CUSTOMER AS SELECT \* FROM PROJECT.CUSTOMER

CREATE OR REPLACE TRIGGER CUSTOMER\_LOG

AFTER UPDATE ON CUSTOMER

for each row

BEGIN

INSERT INTO CUSTOMER\_LOG(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(:OLD.CUSTOMER\_ID,:OLD.FIRST\_NAME,:OLD.LAST\_NAME,:OLD.MIDDLE\_INITIAL,:OLD.BIRTH\_DATE,:OLD.CELL\_PHONE,

:OLD.STREET\_ADDRESS,:OLD.CITY,:OLD.STATE,:OLD.ZIP\_CODE,:OLD.EMAIL,:OLD.LICENSE\_ISSUED\_STATE,

:OLD.LICENSE\_NUMBER,:OLD.CREDIT\_CARD\_NUMBER,:OLD.EXPIRATION\_DATE,:OLD.SSN);

INSERT INTO CUSTOMER\_LOG(CUSTOMER\_ID,FIRST\_NAME,LAST\_NAME,MIDDLE\_INITIAL,CELL\_PHONE,BIRTH\_DATE,

STREET\_ADDRESS,CITY,STATE,ZIP\_CODE,EMAIL,LICENSE\_ISSUED\_STATE,

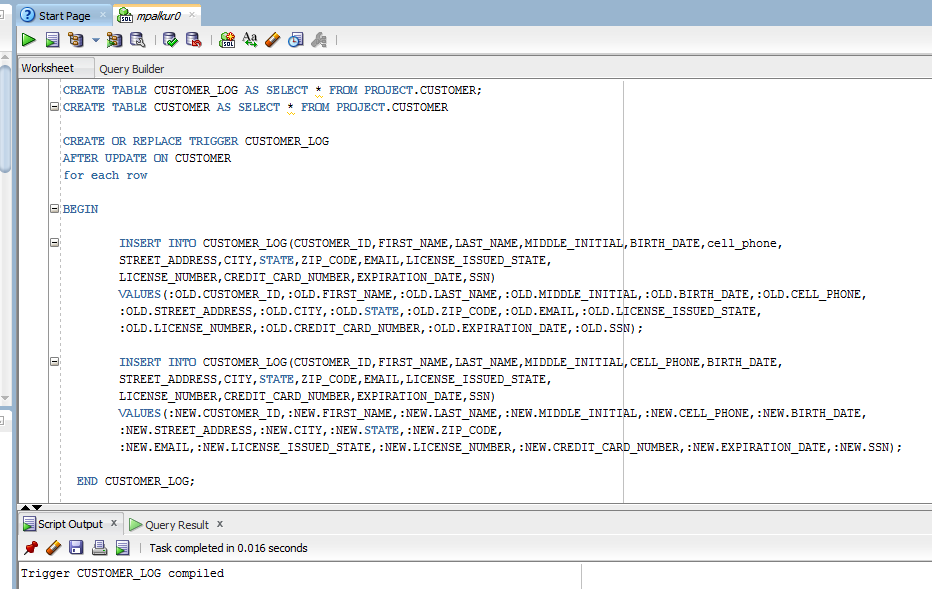
LICENSE\_NUMBER,CREDIT\_CARD\_NUMBER,EXPIRATION\_DATE,SSN)

VALUES(:NEW.CUSTOMER\_ID,:NEW.FIRST\_NAME,:NEW.LAST\_NAME,:NEW.MIDDLE\_INITIAL,:NEW.CELL\_PHONE,:NEW.BIRTH\_DATE,

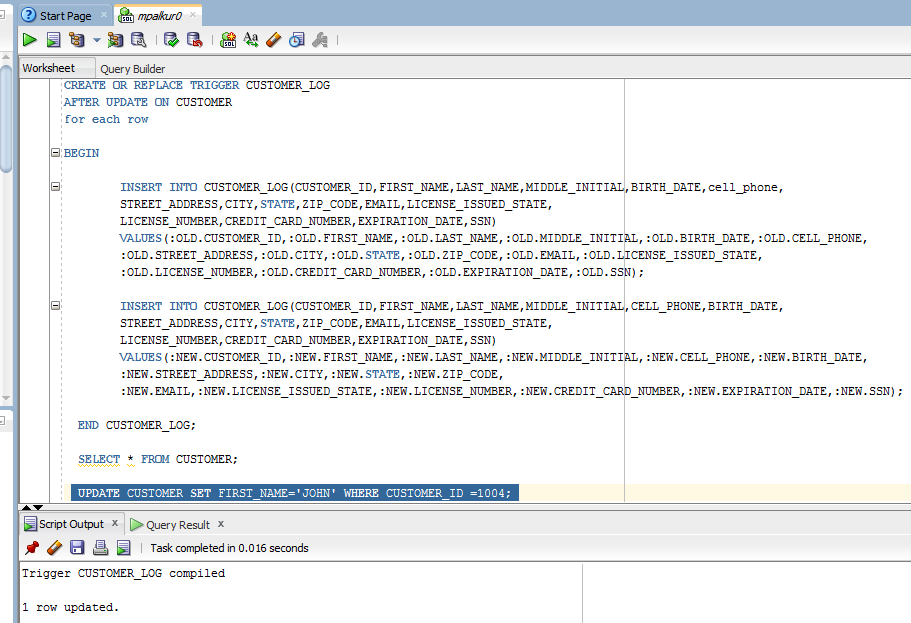
:NEW.STREET\_ADDRESS,:NEW.CITY,:NEW.STATE,:NEW.ZIP\_CODE,

:NEW.EMAIL,:NEW.LICENSE\_ISSUED\_STATE,:NEW.LICENSE\_NUMBER,:NEW.CREDIT\_CARD\_NUMBER,:NEW.EXPIRATION\_DATE,:NEW.SSN);

END CUSTOMER\_LOG;



UPDATE CUSTOMER SET FIRST\_NAME='JOHN' WHERE CUSTOMER\_ID =1004;



SELECT \* FROM CUSTOMER;

