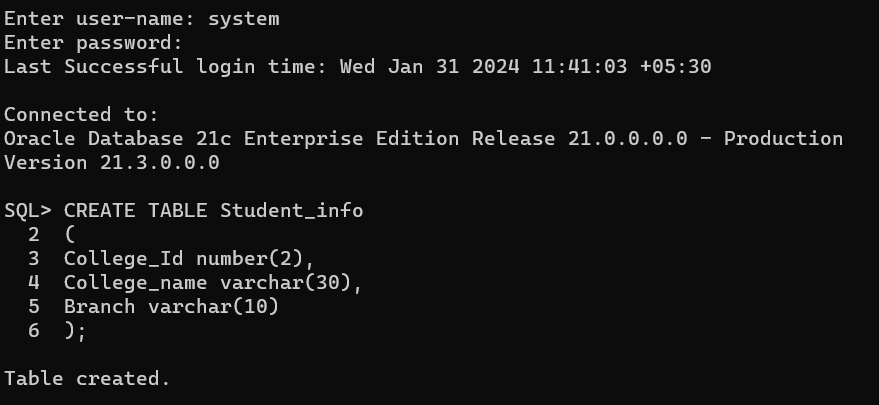
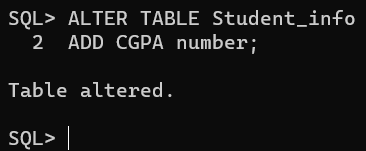
**DATABASE MANAGEMENT SYSTEM LAB**

**1.WRITE SQL QUERIES TO CREATE TABLE FOR VARIOUS DATABASE USING DDL COMMANDS(CREATE,ALTER,DROP,TRUNCATE).**

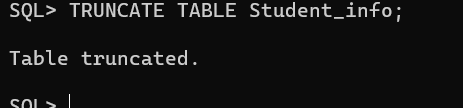
**AIM**: SQL QUERIES TO CREATE TABLE FOR VARIOUS DATABASE USING DDL COMMANDS(CREATE,ALTER,DROP,TRUNCATE)**.**

**CREATE COMMAND:**CREATE is a DDL command used to create databases, tables, triggers and other database objects.

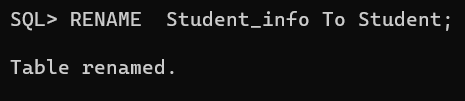
**ALTER1 COMMAND:**This command is used to add, delete or change columns in the existing table. The user needs to know the existing table name and can do add, delete or modify tasks easily.



**TRUNCATE COMMAND :**This command is used to remove all rows from the table, but the structure of the table still exists.

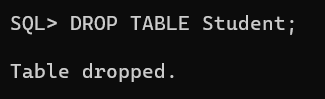


**RENAME COMMAND:** It is possible to change name of table with or without data in it using simple RENAME command.



**DROP COMMAND:**

This command is used to remove an existing table along with its structure from the Database.

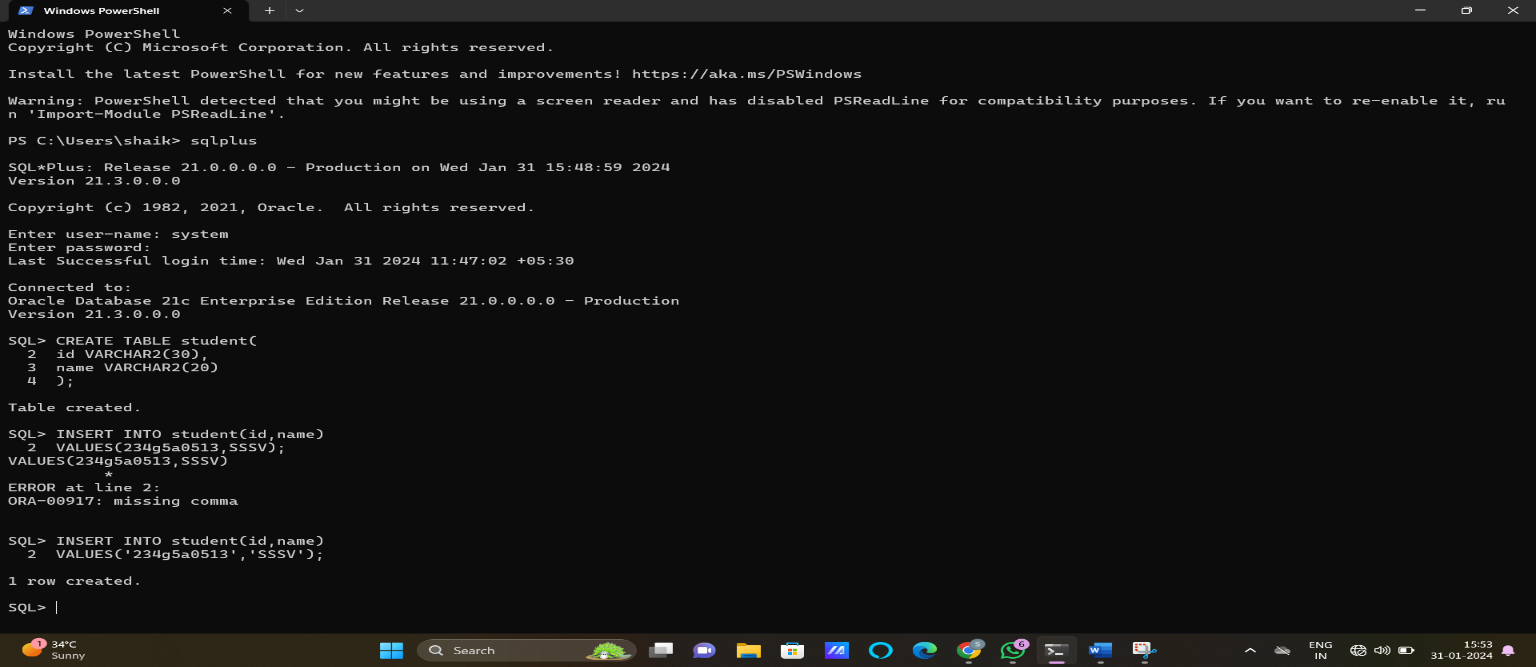


**CONCLUSION:** SQL QUERIES TO CREATE TABLE FOR VARIOUS DATABASE USING DDL COMMANDS(CREATE,ALTER,DROP,TRUNCATE)IS SUCCESSFULLY EXECUTED.

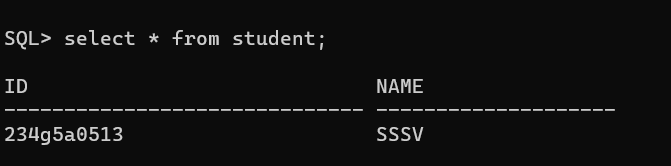
**2.WRITE SQL QUERIES TO MANIPULATE TABLES FOR VARIOUS DATABSES USING DML COMMANDS (INSERT,SELECT,UPDATE,DELETE).**

**AIM:** SQL QUERIES TO MANIPULATE TABLES FOR VARIOUS DATABSES USING DML COMMANDS (INSERT, SELECT, UPDATE, DELETE).

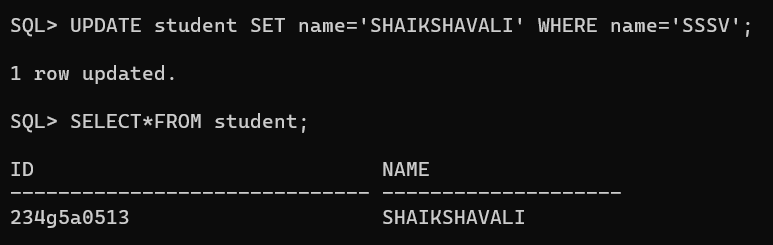
**INSERT COMMAND:** INSERT is another most important data manipulation command in Structured Query Language, which allows users to insert data in database tables.

****

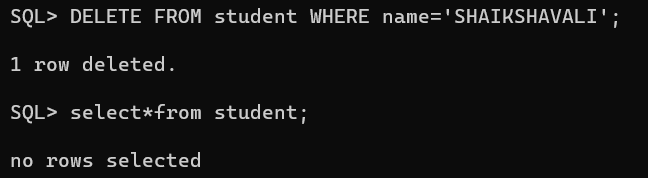
**SELECT COMMAND:** The SELECT command shows the records of the specified table. It also shows the particular record of a particular column by using the WHERE clause.

****

**UPDATE COMMAND:** UPDATE is another most important data manipulation command in Structured Query Language, which allows users to update or modify the existing data in database tables.

****

**DELETE COMMAND:** DELETE is a DML command which allows SQL users to remove single or multiple existing records from the database tables.

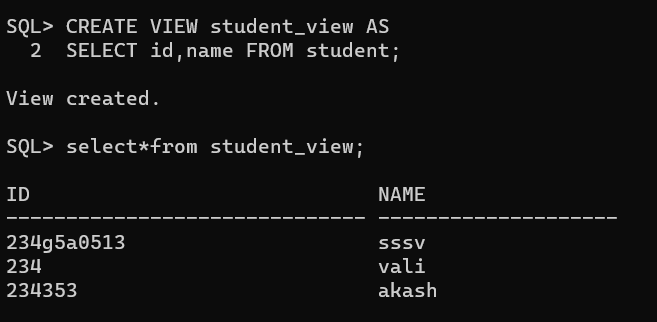
****

**CONCLUSION:** SQL QUERIES TO MANIPULATE TABLES FOR VARIOUS DATABSES USING DML COMMANDS (INSERT, SELECT, UPDATE, DELETE) IS SUCCESSFULLY EXCECUTED.

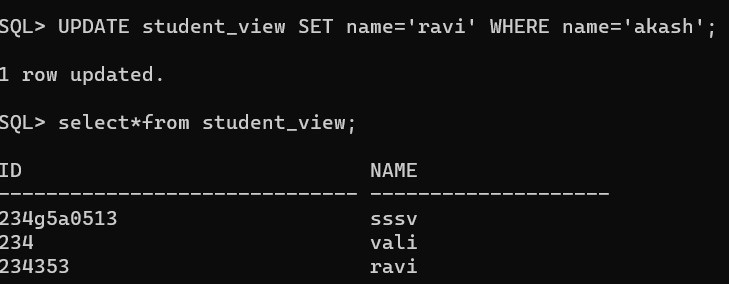
**3.WRITE SQL QUERIES TO CREATE VIEW FOR VARIOUS DATABASES (CREATE VIEW, UPDATE VIEW, ALTER VIEW, DELETE VIEW).**

**AIM:** SQL QUERIES TO CREATE VIEW FOR VARIOUS DATABASES (CREATE VIEW, UPDATE VIEW, ALTER VIEW, DELETE VIEW).

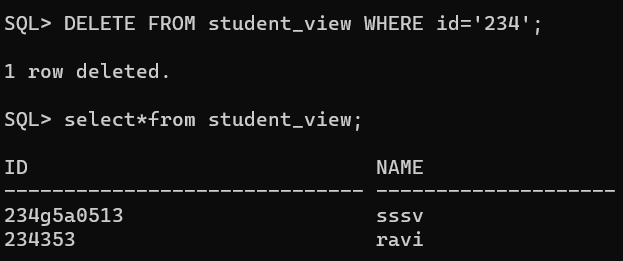
**CREATE VIEW:** A view can be created using the **CREATE VIEW** statement. We can create a view from a single table or multiple tables.

****

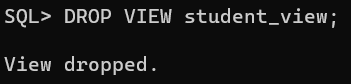
**UPDATE VIEW:** A view can be updated by using **UPDATE** statement.



**DELETE VIEW:** To delete a row in a view by using **DELETE** statement.



**DROP VIEW:** A view can be dropped by using **DROP** statement.

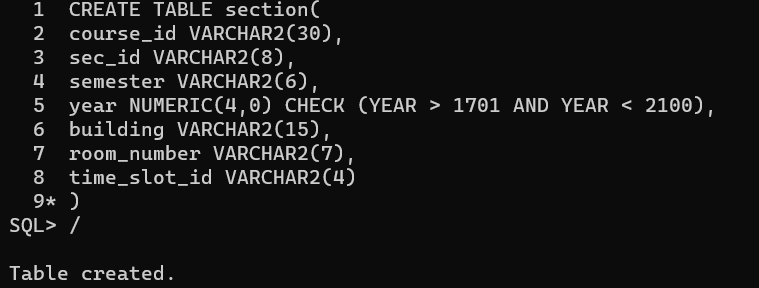


**CONCLUSION:** SQL QUERIES TO CREATE VIEW FOR VARIOUS DATABASES (CREATE VIEW, UPDATE VIEW, ALTER VIEW, DELETE VIEW) IS SUCCESSFULLY EXCECUTED.

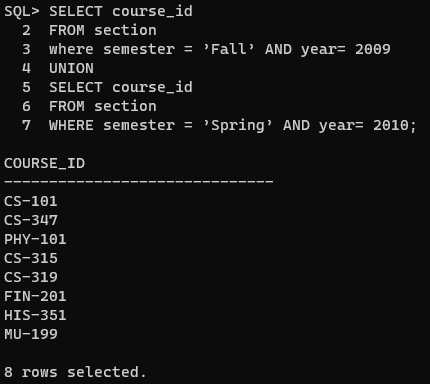
**4.WRITE SQL QUERIES TO PERFORM RELATIONAL SET OPERATIONS (UNION, UNION ALL, INTERSECT, MINUS).**

**AIM:**SQL QUERIES TO PERFORM RELATIONAL SET OPERATIONS (UNION, UNION ALL, INTERSECT, MINUS).

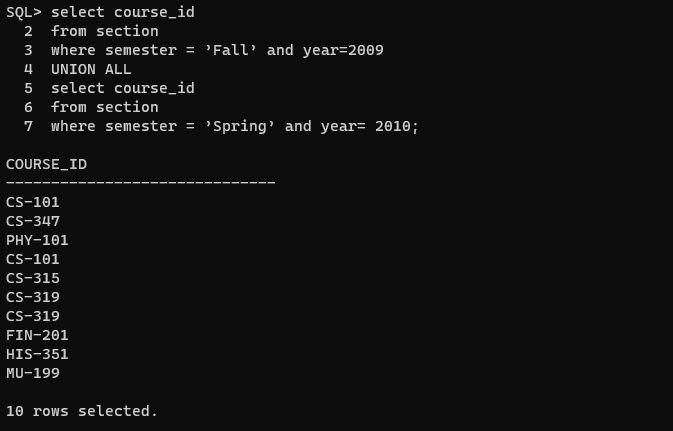
**CREATE TABLE:**

****

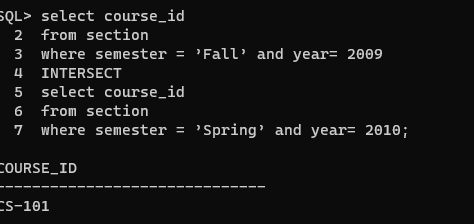
**UNION OPERATION:**

****

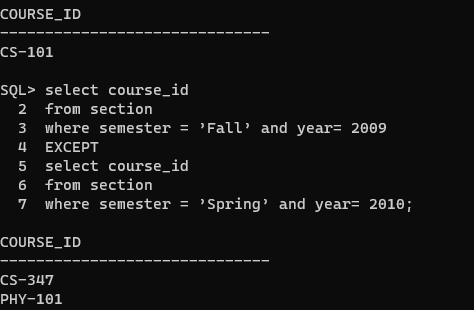
**UNION ALL OPERATION:**

****

**INTERSECT OPERATION:**

****

**MINUS OPERATION:**

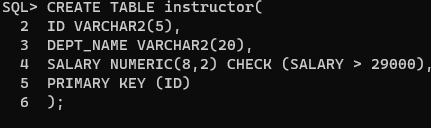
****

**CONCLUSION:** SQL QUERIES TO PERFORM RELATIONAL SET OPERATIONS (UNION, UNION ALL, INTERSECT, MINUS) IS SUCCESSFULLY EXCECUTED.

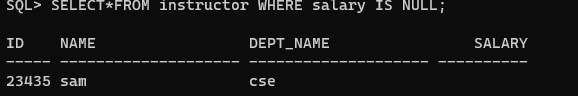
**5.WRITE SQL QUERIES TO PERFORM SPECIAL OPERATIONS (ISNULL, BETWEEN, IN, EXISTS).**

**AIM:** SQL QUERIES TO PERFORM SPECIAL OPERATIONS (ISNULL, BETWEEN, IN, EXISTS).

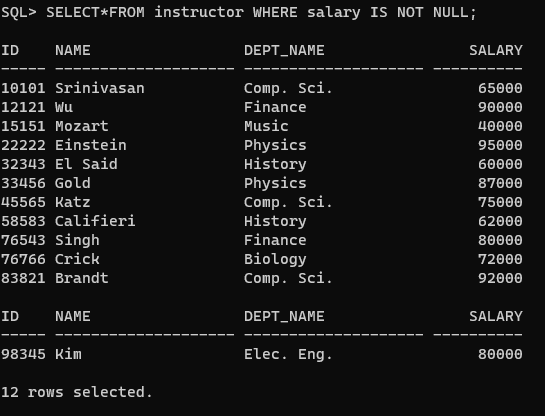
**CREATE A TABLE:**

****

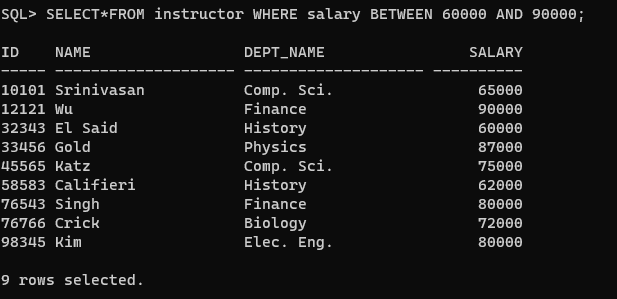
**IS NULL OPERATION:**

****

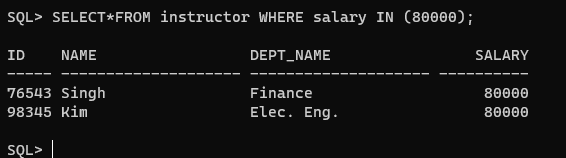
**IS NOT NULL:**

****

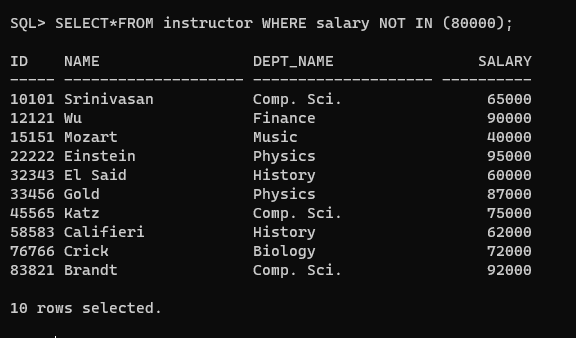
**BETWEEN OPERATION:**

****

**IN OPERATION:**

****

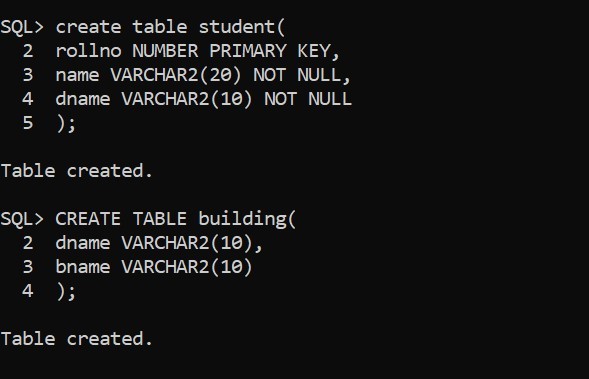
**NOT IN OPERATION:**

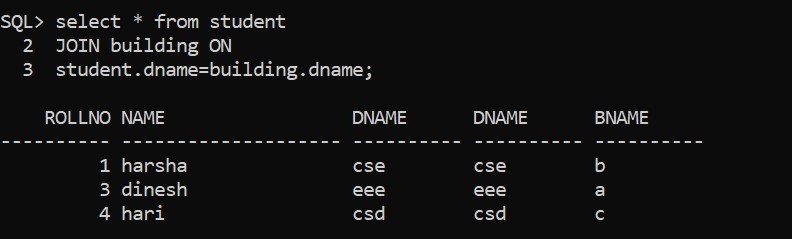
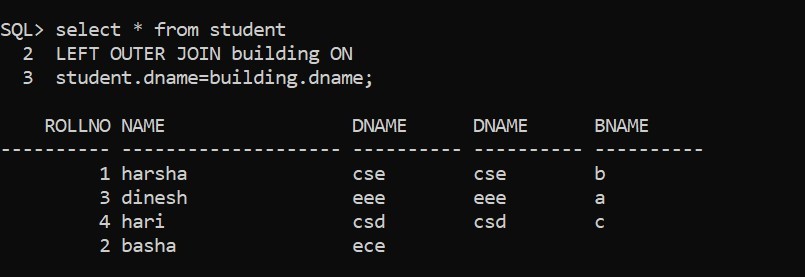
****

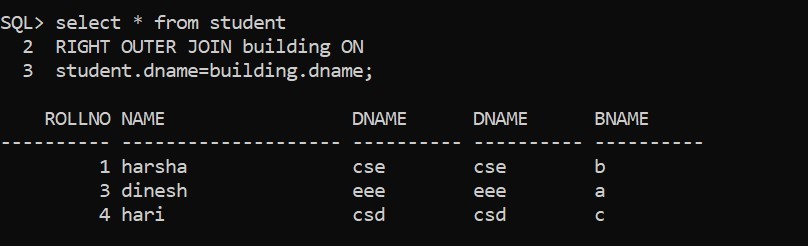
**CONCLUSION: :** SQL QUERIES TO PERFORM SPECIAL OPERATIONS (ISNULL, BETWEEN, IN, EXISTS) IS EXCECUTED SUCESSFULLY.

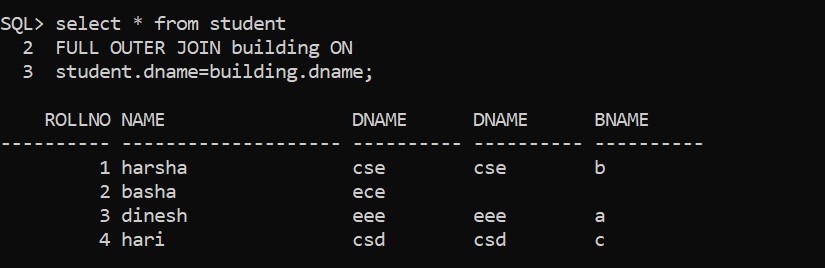
**6.WRITE SQL QUERIES TO PERFORM JOIN OPERATIONS**

**AIM:** SQL QUERIES TO PERFORM JOIN OPERATIONS

**CREATE TWO TABLES:**

**JOIN OPERATORS:**





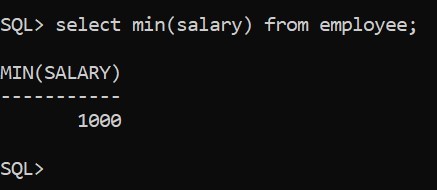
**CONCLUSION:** SQL QUERIES TO PERFORM JOIN OPERATIONS IS SUCCESSFULLY COMPLETED.

**7.WRITE SQL QUERIES TO PERFORM AGGREGATE OPERATIONS.**

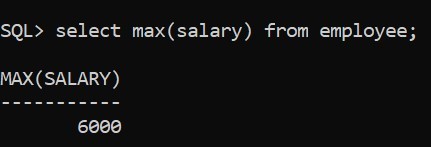
**AIM:** WRITE SQL QUERIES TO PERFORM AGGREGATE OPERATIONS.

**AGGREGATE FUNCTIONS (MIN, MAX, COUNT, SUM, AVG):**

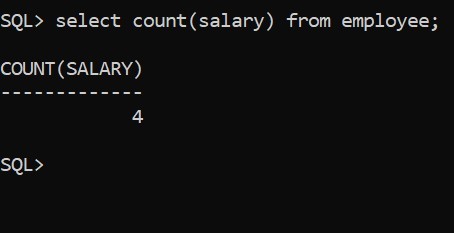
**MIN FUNCTION:**

**E:**

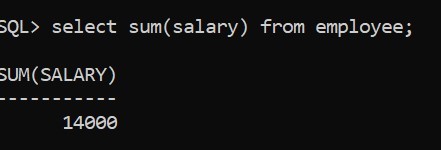
**MAX FUNCTION:**

**EX:**

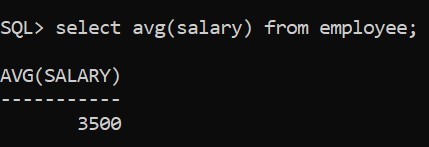
**COUNT FUNCTION:**

**EX:**

**SUM FUNCTION:**

**EX:**

**AVG FUNCTION:**

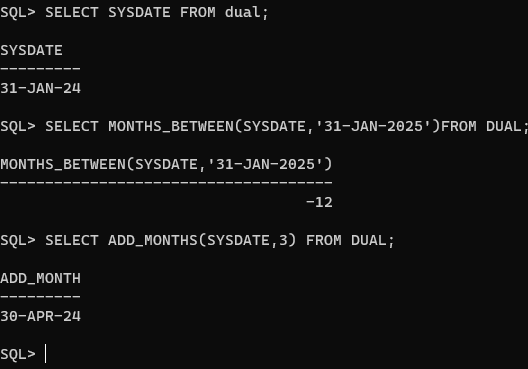
**EX:** 

**CONCLUSION:** SQL QUERIES TO PERFORM AGGREGATE OPERATIONS IS SUCCESSFULLY EXCECUTED.

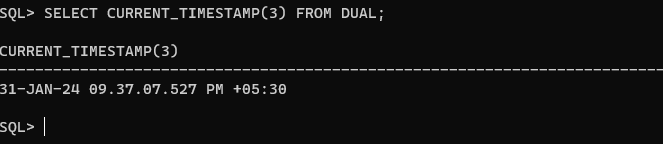
**8.WRITE SQL QUERIES TO PERFORM ORACLE BUILT-IN FUNCTIONS.**

**AIM:** SQL QUERIES TO PERFORM ORACLE BUILT-IN FUNCTIONS.

**DATE FUNCTION:**



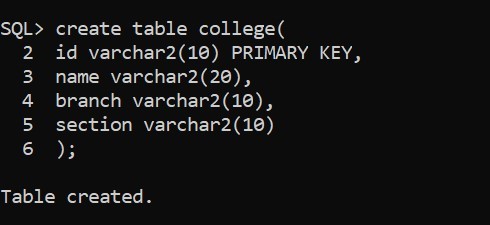
**TIME FUNCTION**:

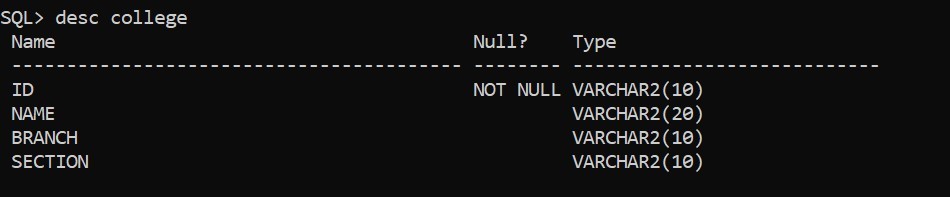


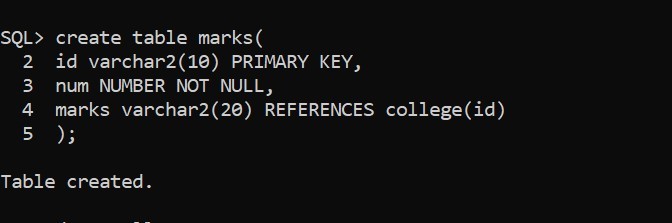
**CONCLUSION**: SQL QUERIES TO PERFORM ORACLE BUILT-IN FUNCTIONS IS EXCEUTED SUCCSESSFULLY.

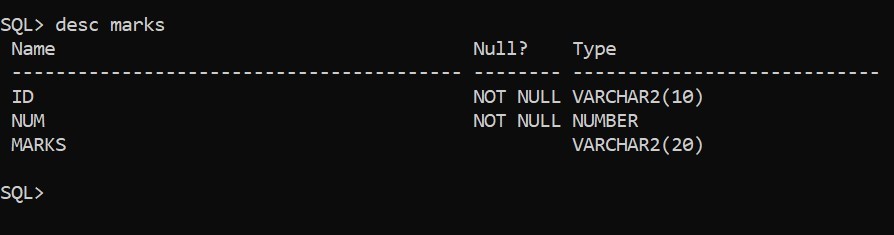
**9.WRITE SQL QUERIES TO PERFORM KEY CONSTRAINTS.**

**AIM:**  SQL QUERIES TO PERFORM KEY CONSTRAINTS.

**PRIMARY KEY:**

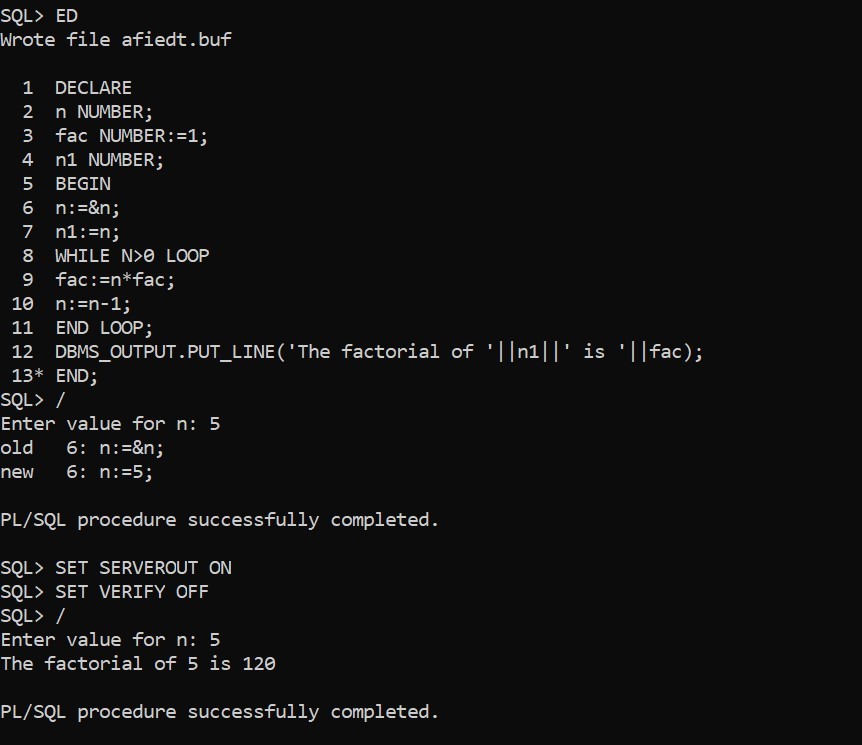


**FOREIGN KEY:**



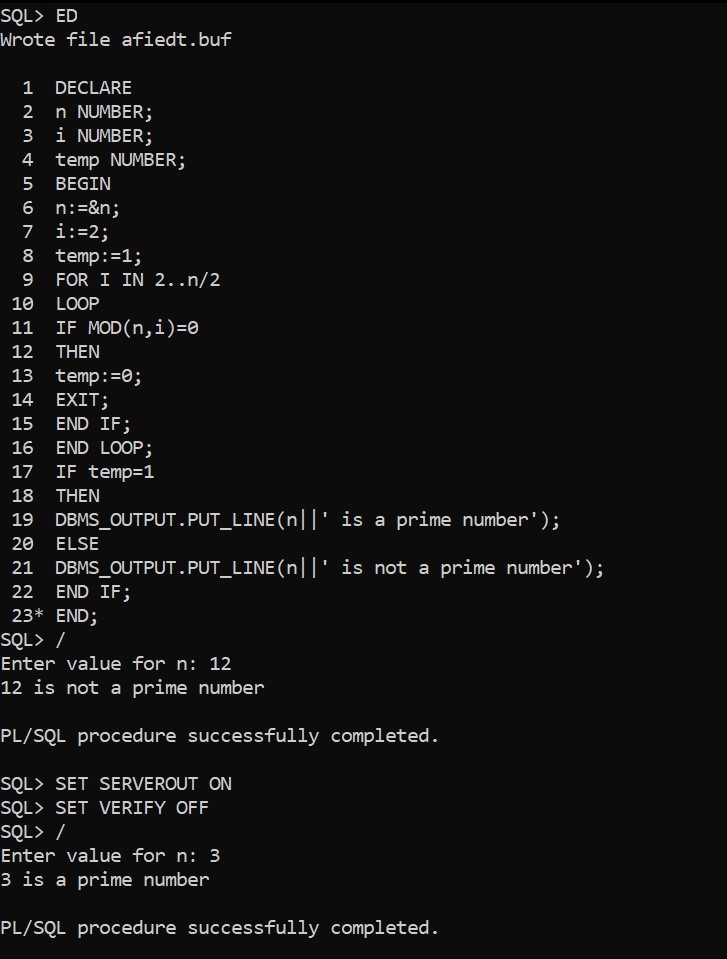
**CONCLUSION:** SQL QUERIES TO PERFORM KEY CONSTRAINTS IS SUCCESSFULLY EXCECUTED.

**10.WRITE A PL/SQL PROGRAM FOR CALCULATING THE FACTORIAL OF A GIVEN NUMBER**.

**AIM:** TO WRITE A PL/SQL PROGRAM FOR CALCULATING THE FACTORIAL OF A GIVEN NUMBER.

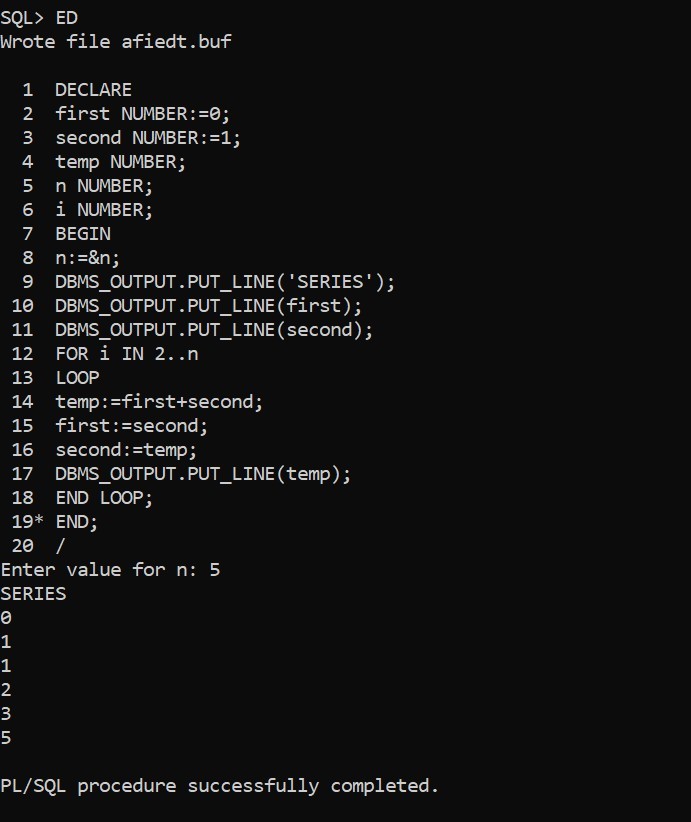
**CONCLUSION:** A PL/SQL PROGRAM FOR CALCULATING THE FACTORIAL OF A GIVEN NUMBER IS SUCCESSFULLY EXECUTED.

**11.WRITE PL/SQL PROGRAM TO FIND GIVEN NUMBER IS PRIME OR NOT.**

**AIM:** TO WRITE PL/SQL PROGRAM TO FIND GIVEN NUMBER IS PRIME OR NOT.

**CONCLUSION:** PL/SQL PROGRAM TO FIND GIVEN NUMBER IS PRIME OR NOT IS EXCECUTED SUCCESSFULLY.

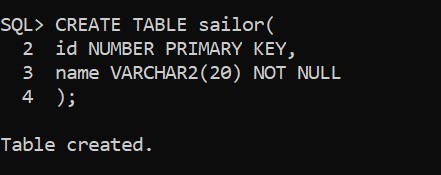
**11.WRITE A PL/SQL PROGRAM FOR DISPLAYING THE FIBONACCI SERIES UP TO AN INTEGER.**

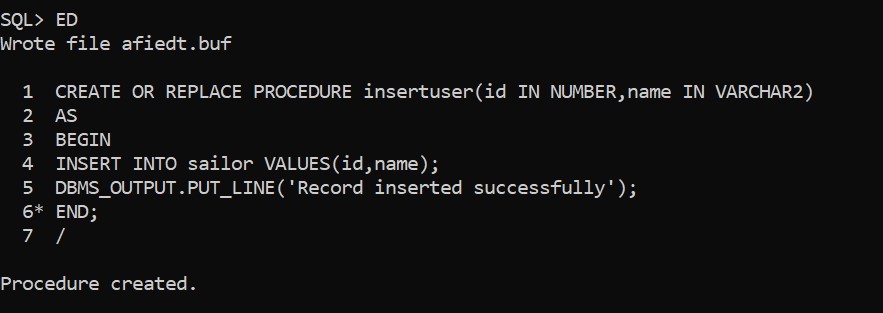
**AIM:** TO WRITE A PL/SQL PROGRAM FOR DISPLAYING THE FIBONACCI SERIES UP TO AN INTEGER.

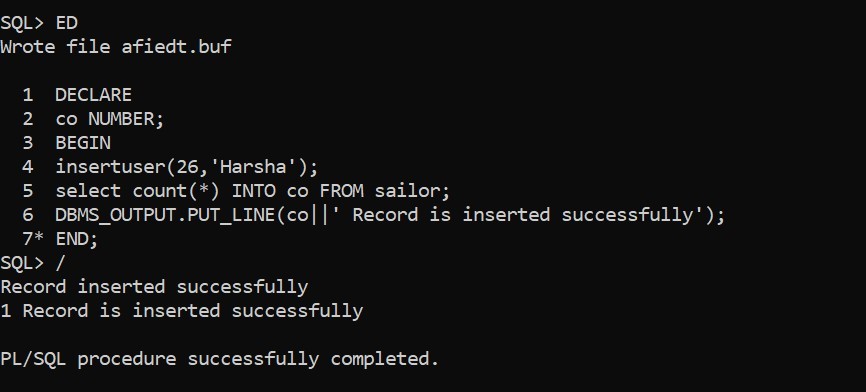
**CONCLUSION:** A PL/SQL PROGRAM FOR DISPLAYING THE FIBONACCI SERIES UP TO AN INTEGER IS SUCCESSFULLY COMPLETED.

**13.WRITE PL/SQL PROGRAM TO IMPLEMENT STORED PROCEDURE ON A TABLE.**

**AIM:** TO WRITE PL/SQL PROGRAM TO IMPLEMENT STORED PROCEDURE ON A TABLE.

**CREATE A TABLE:**

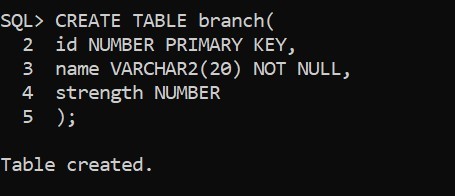
**CREATE A PROCEDURE:**

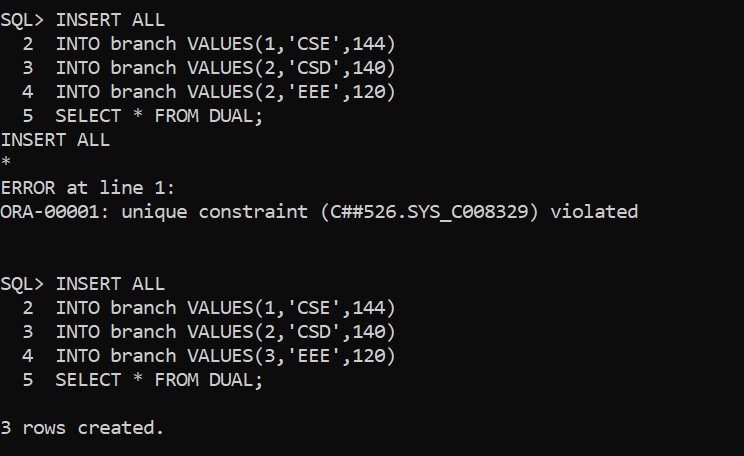


**CONCLUSION**: PL/SQL PROGRAM TO IMPLEMENT STORED PROCEDURE ON A TABLE IS EXECUTED SUCCESSFULLY.

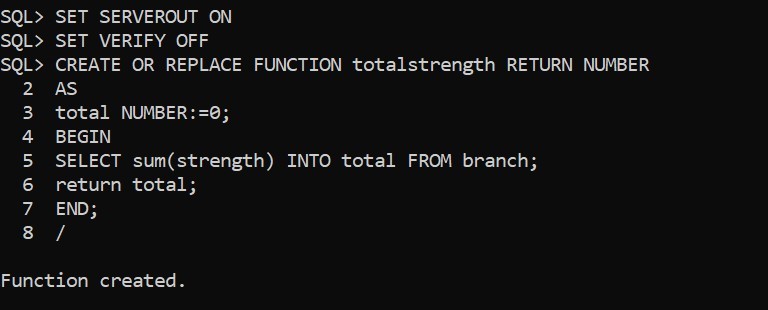
**14.WRITE PL/SQL PROGRAM TO IMPLEMENT STORED FUNCTION ON TABLE.**

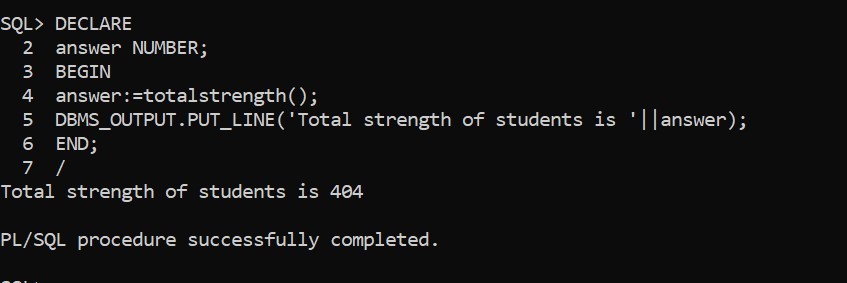
**AIM:** TO WRITE PL/SQL PROGRAM TO IMPLEMENT STORED FUNCTION ON TABLE.

**CREATE A TABLE:**



**CREATE A FUNCTION:**



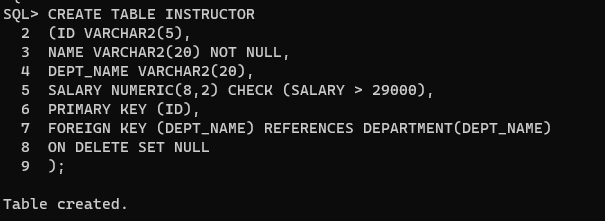


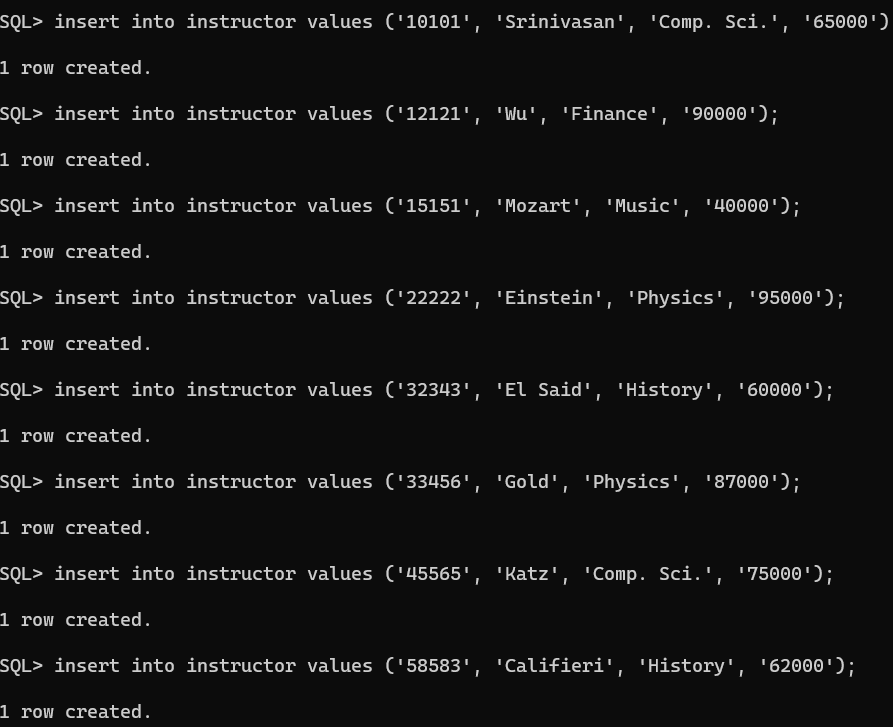
**CONCLUSION:** PL/SQL PROGRAM TO IMPLEMENT STORED FUNCTION ON TABLE IS EXECUTED SUCCESSFULLY.

**15.WRITE A PL/SQL PROGRAM ON TRIGGER.**

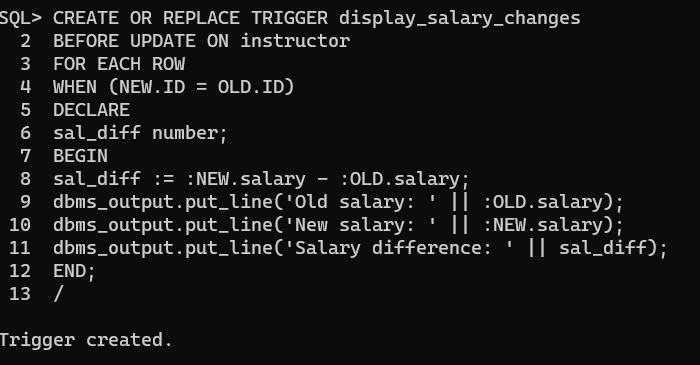
**AIM:** TO WRITE A PL/SQL PROGRAM ON TRIGGER.

**CREATE INSTRUCTOR TABLE:**

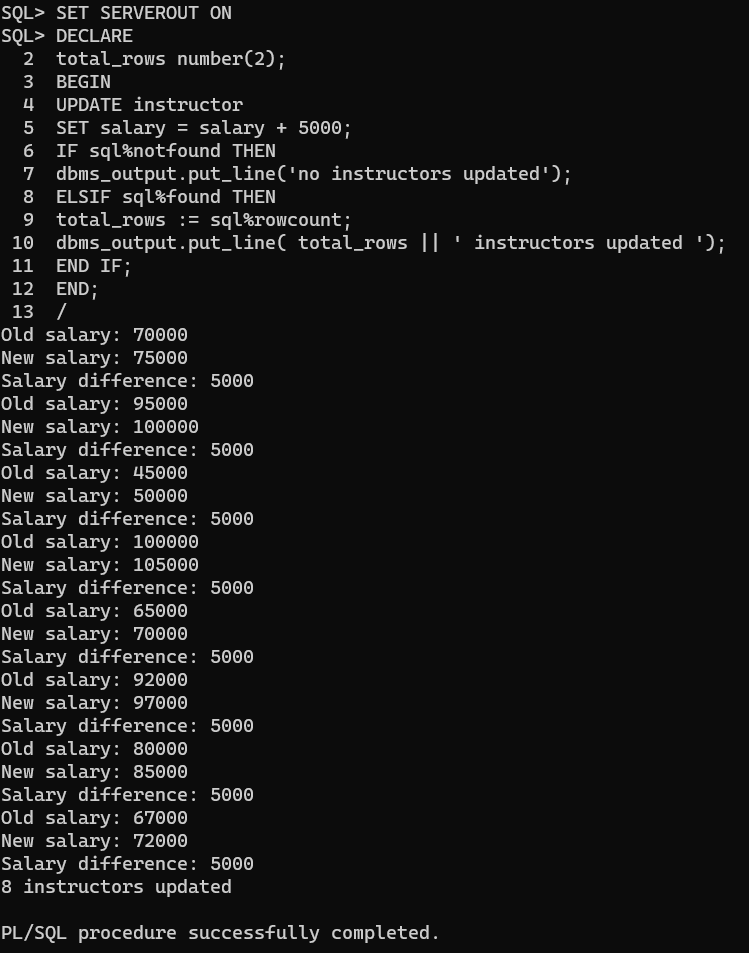
****

****

**CREATE A TRIGGER:**

****

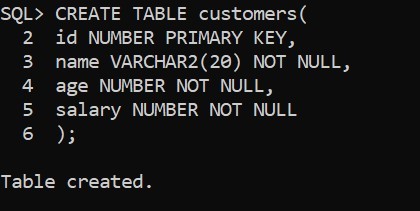
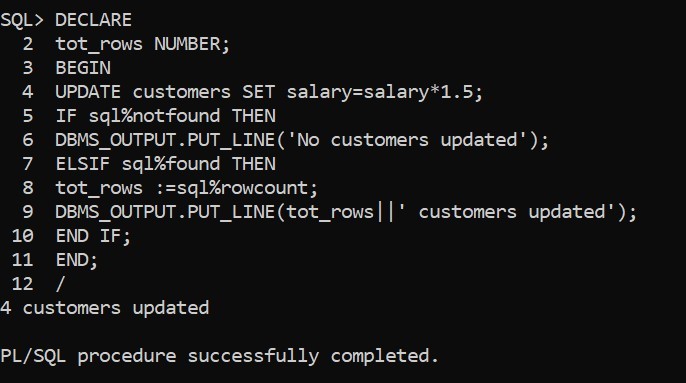
**PROGRAM:**

****

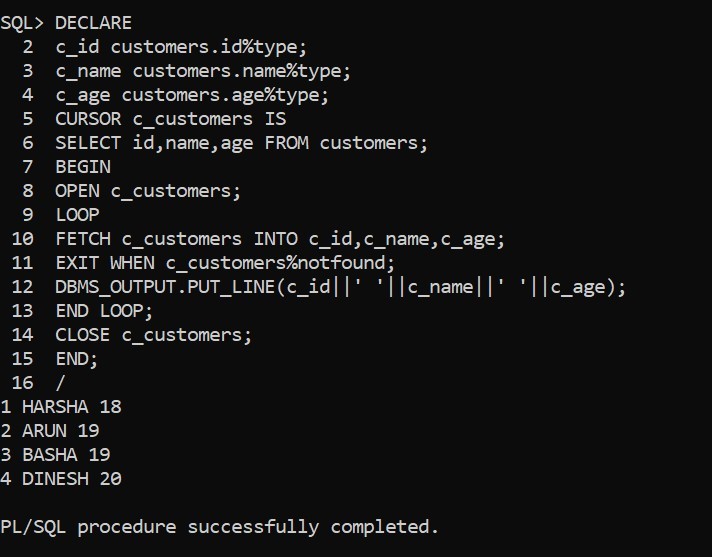
**CONCLUSION:** A PL/SQL PROGRAM ON TRIGGER IS SUCCESSFULLY EXECUTED.

**16.WRITE PL/SQL PROGRAM TO IMPLEMENT CURSORS ON A TABLE.**

**AIM:** TO WRITE PL/SQL PROGRAM TO IMPLEMENT CURSORS ON A TABLE.

**CREATE A TABLE:**

**PROGRAM**



**CONCLUSION**: PL/SQL PROGRAM TO IMPLEMENT CURSORS ON A TABLE IS EXECUTED SUCCESSFULLY.