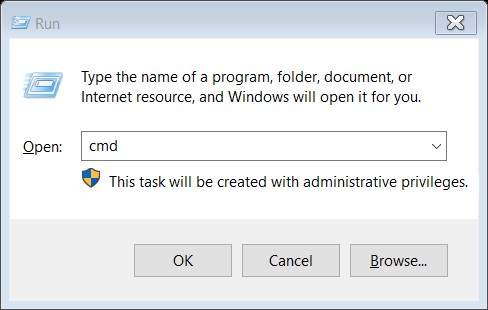
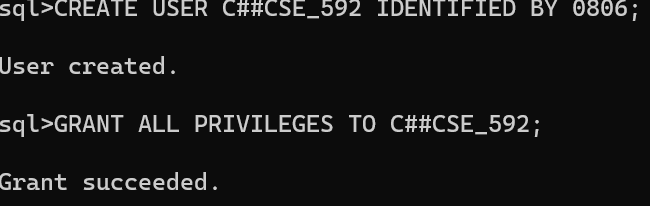
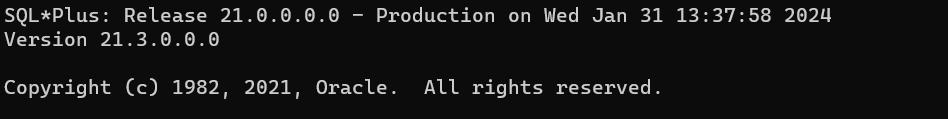
1. Open the command prompt Press WIN+R , type cmd



1. Create user with your id number and grant all privileges.



1. Now sign in with the new user.



1. DDL COMMANDS

Write SQL queries to CREATE TABLES for various databases using DDL commands (i.e. CREATE, ALTER, DROP, TRUNCATE).

CREATE TABLE

Syntax:

CREATE TABLE tablename (

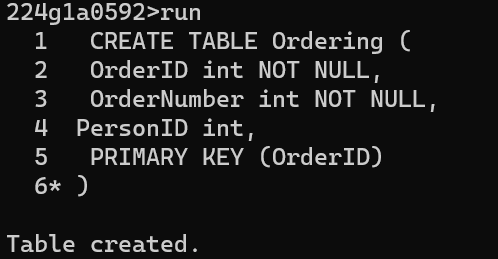
column1 data\_ type [constraint]

[, column2 data\_ type [constraint] ] [,

PRIMARY KEY (column1 [, column2]) ]

[, FOREIGN KEY (column1 [, column2]) REFERENCES tablename] [,CONSTRAINT constraint]);

Example:



ALTER TABLE

Syntax 1:

ALTER TABLE tablename

{ADD | MODIFY} (column\_name data\_type [ {ADD|MODIFY}

Column\_name data\_type]);

Syntax 2;

ALTER TABLE tablename

ADD constraint [ADD constraint];

Syntax 3:

ALTER TABLE tablename

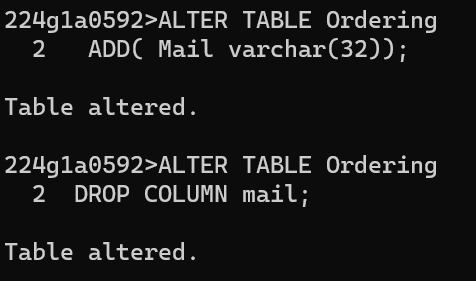
DROP {PRIMARY KEY | COLUMN column\_name | CONSTRAINT constraint\_name);

Syntax 4:

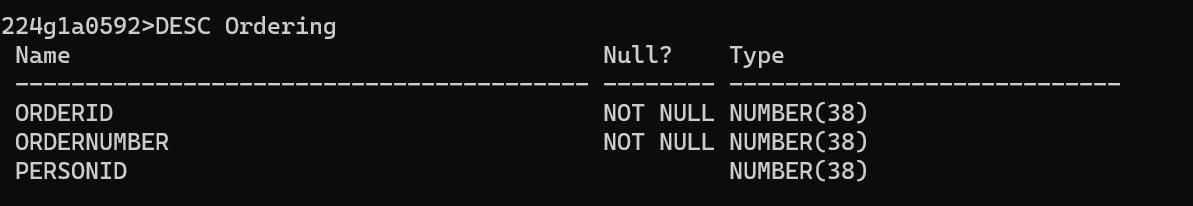
ALTER TABLE tablename

ENABLE CONSTRAINT constraint\_name;

Example:



DESC Orders:

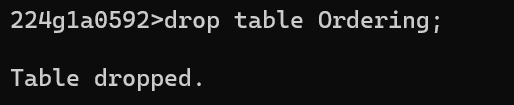


DROP TABLE

Syntax:

DROP TABLE table\_name;

Example:



TRUNCATE TABLE

Syntax:

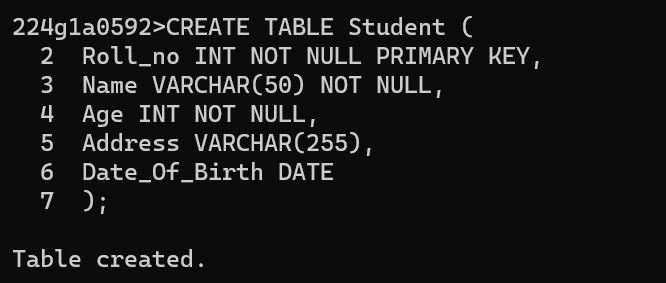
TRUNCATE TABLE table\_name;

Example:



2.DML COMMANDS

Write SQL queries to MANIPULATE TABLES for various databases using DML commands (i.e. INSERT, SELECT, UPDATE, DELETE,)



INSERT

Syntax:

INSERT INTO tablename

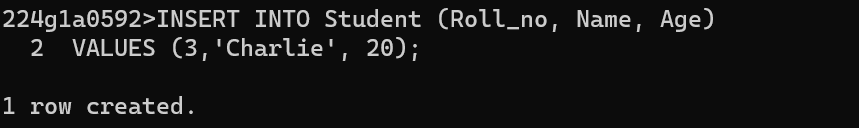
VALUES (value1,value2,...,valuen);

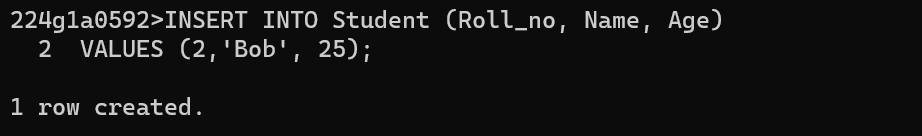
Syntax 2:

INSERT INTO tablename

(column1, column2,…,column) VALUES (value1, value2,...,valuen);

Example:





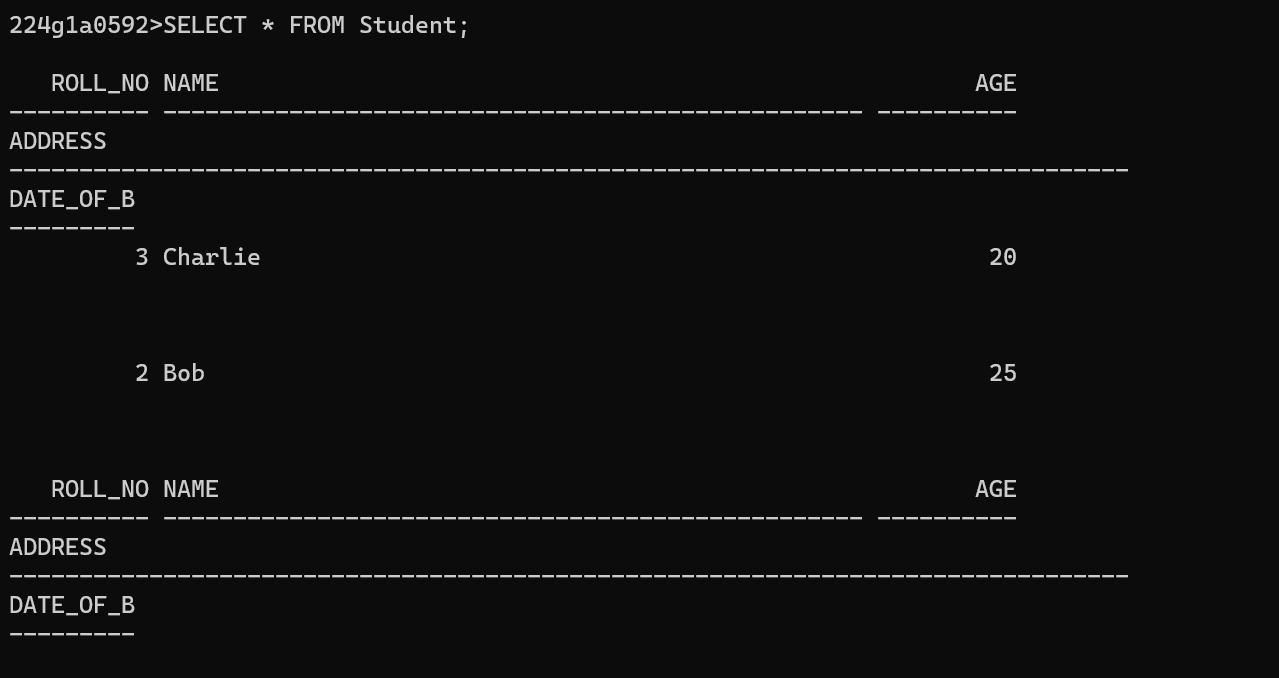
SELECT

Syntax:

SELECT \*

DROM <table\_name>;

Example:

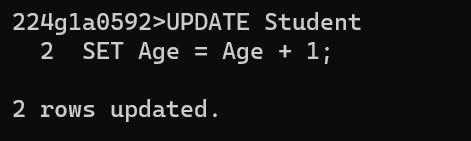


UPDATE

Syntax:

UPDATE table\_name SET [column\_name1= value\_1, column\_name2= value\_2,...]

WHERE CONDITION;

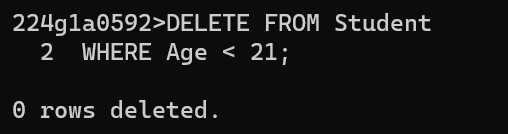


DELETE

Syntax:

DELETE FROM table\_Name WHERE condition;

Example:

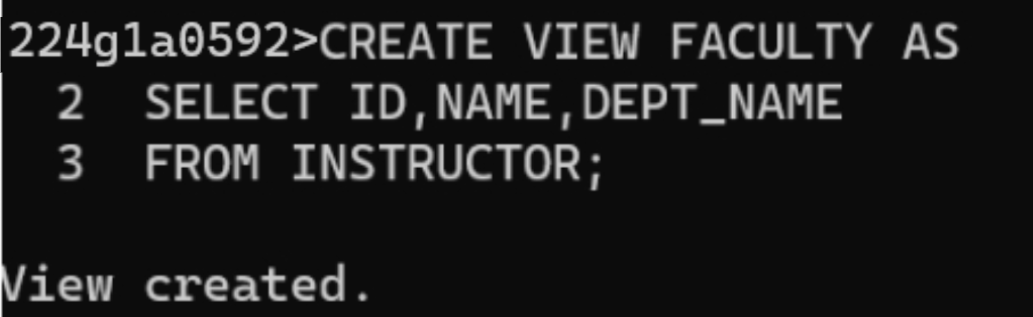


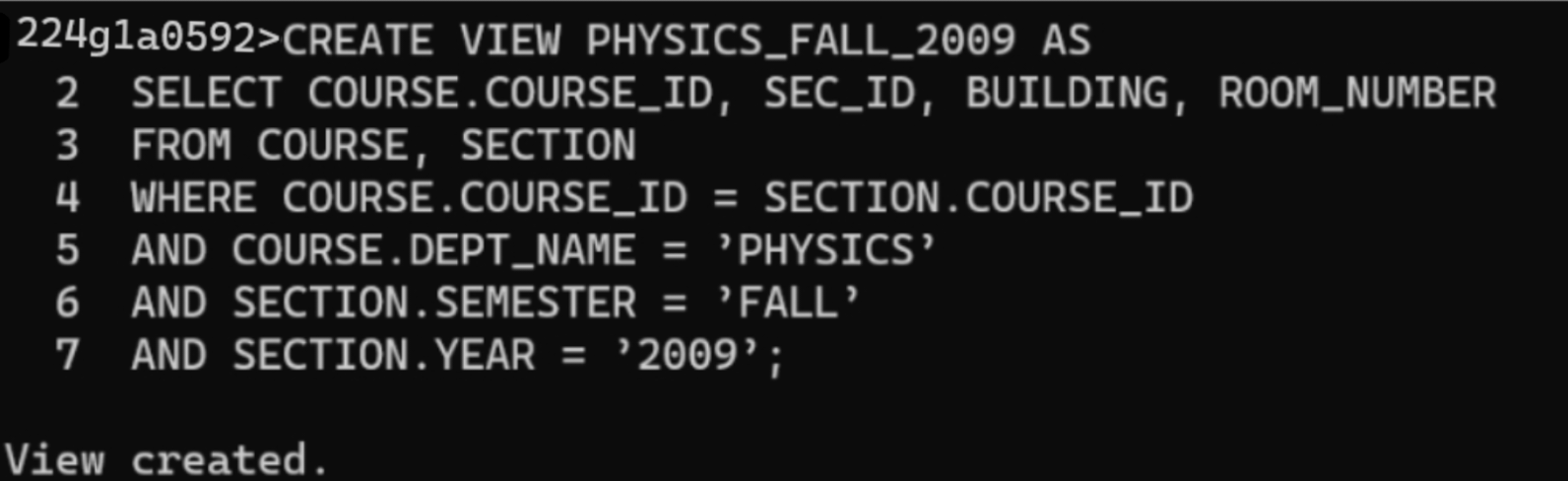
3.VIEWS

Write SQL queries to create VIEWS for various databases (i.e. CREATE VIEW, UPDATE VIEW, ALTER VIEW, and DELETE VIEW).

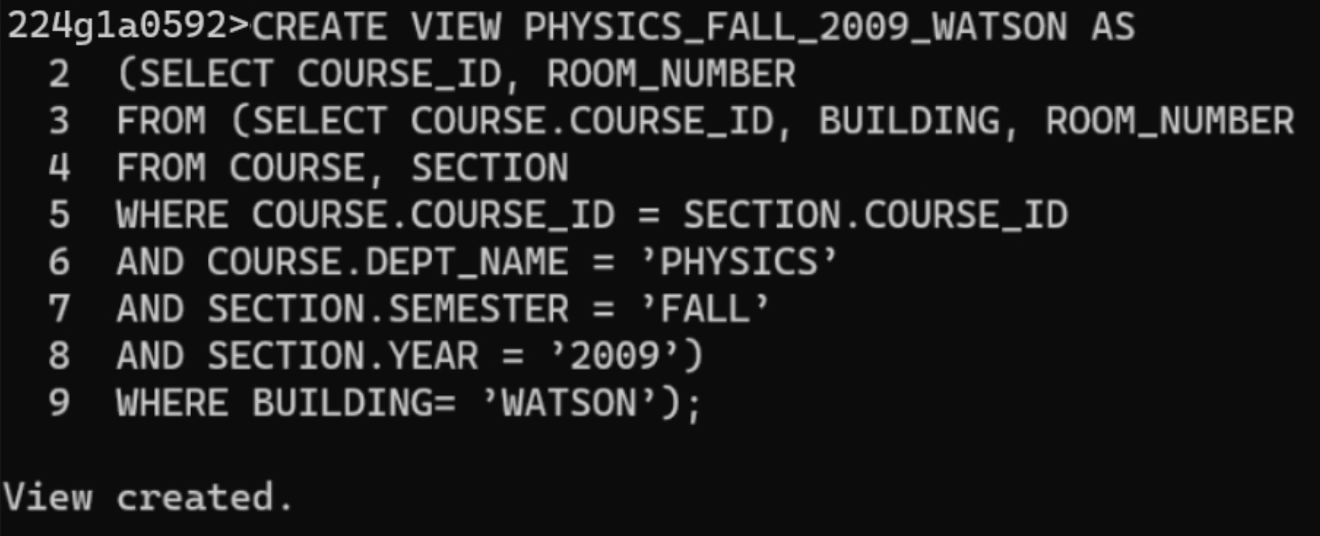
View syntax:

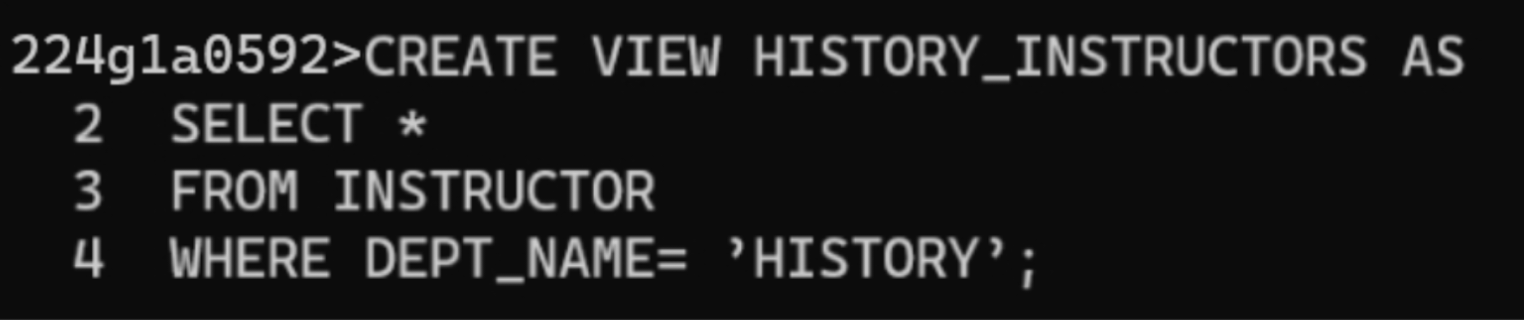
CREATE VIEW VIEW\_NAME AS <QUERY EXPRESSION>

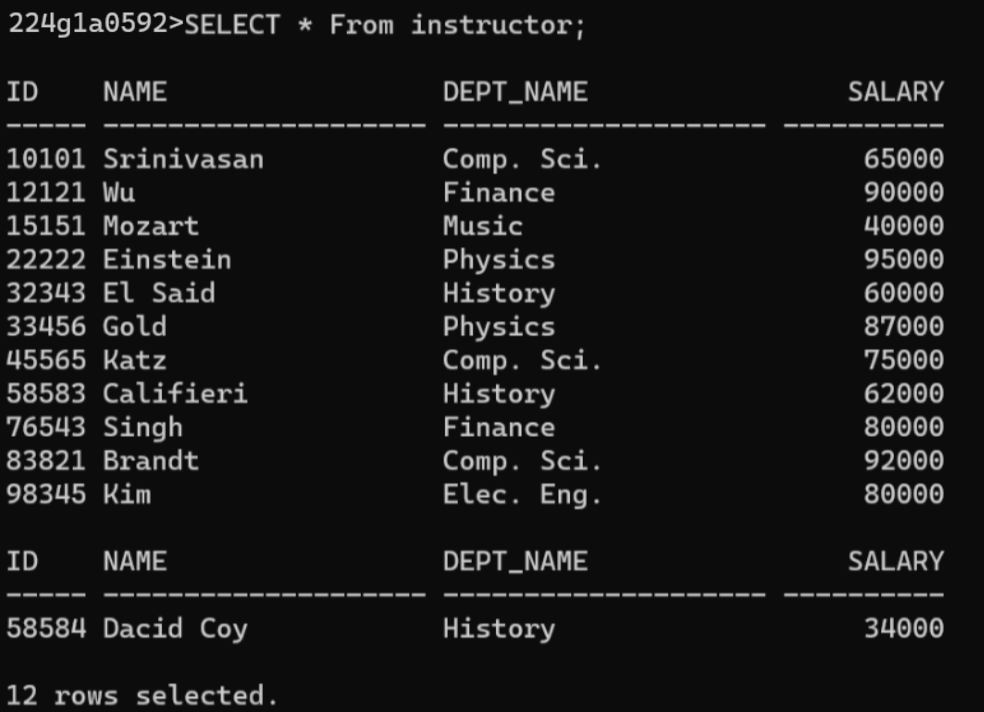




An equivalent relation of view without using view as original relation

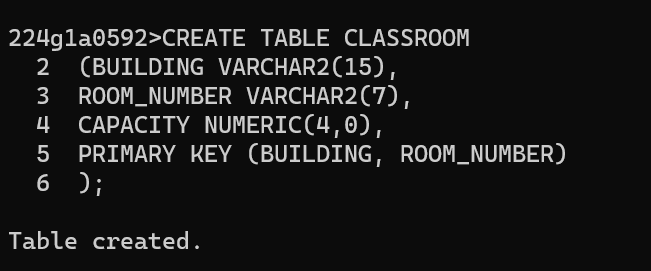


Commands to insert, Delete and update view 

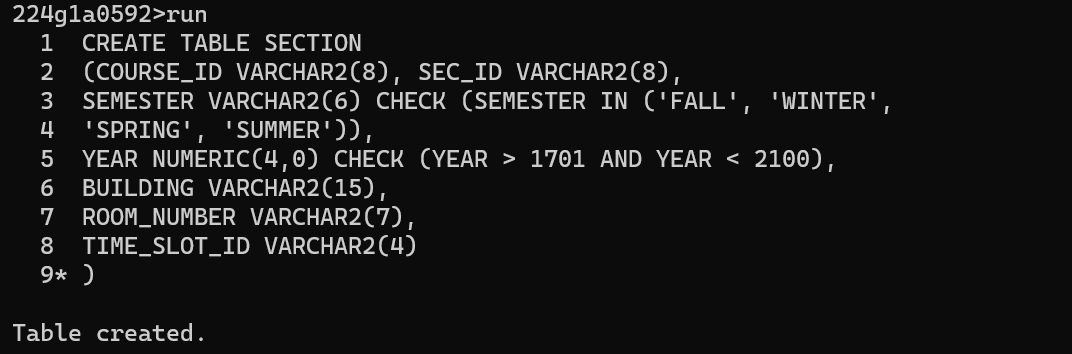


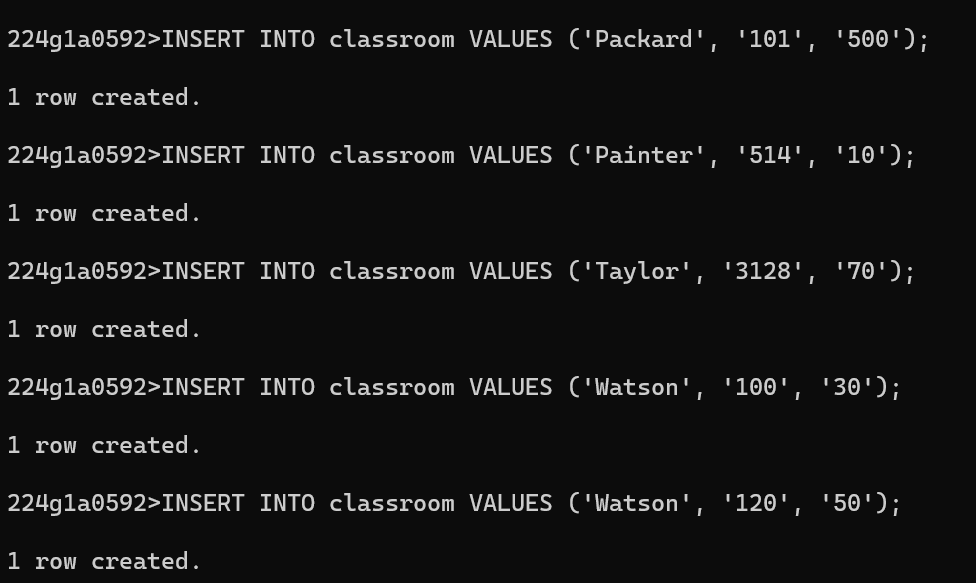
1. RELATIONAL SET OPERATIONS

Write SQL queries to perform RELATIONAL SET OPERATIONS (i.e. UNION, UNION ALL, INTERSECT, MINUS, CROSS JOIN, NATURAL JOIN)

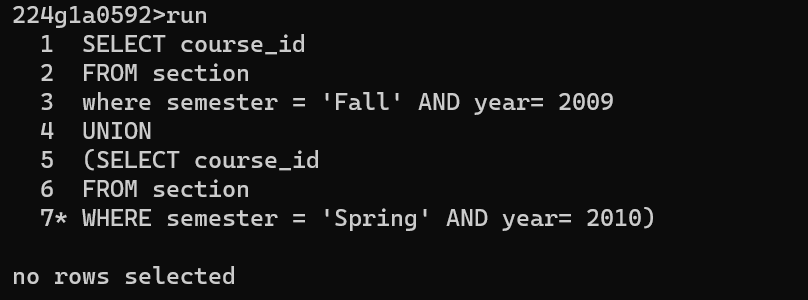


CREATE TABLE

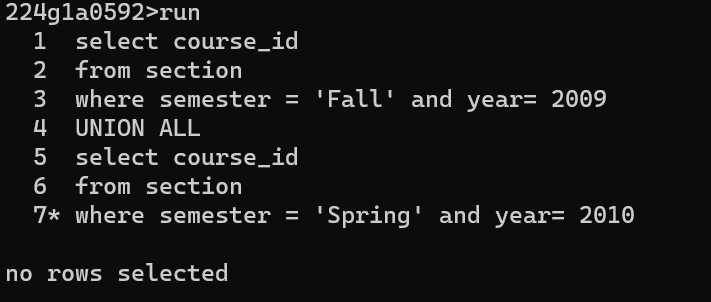




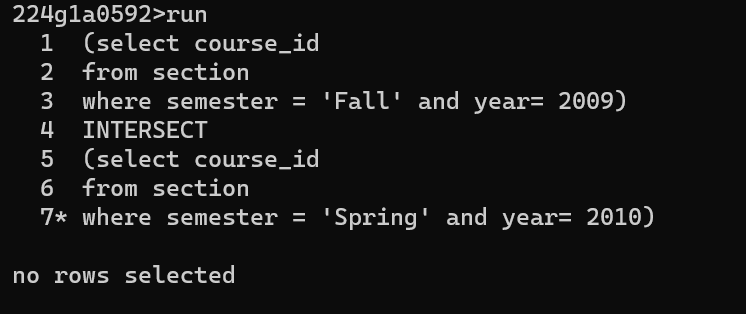
Union operation:



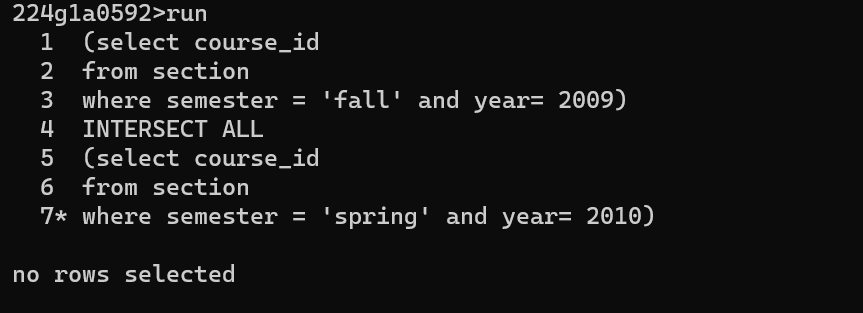
Union all Operation:



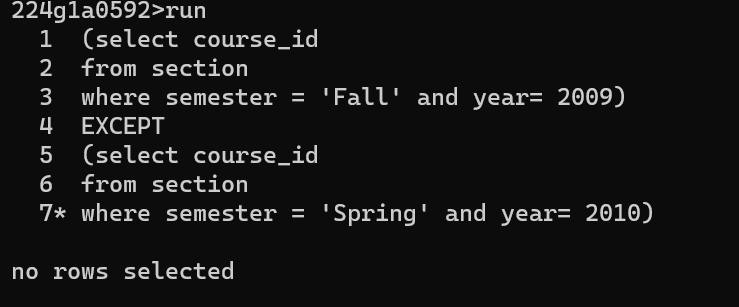
Intersect Operation:



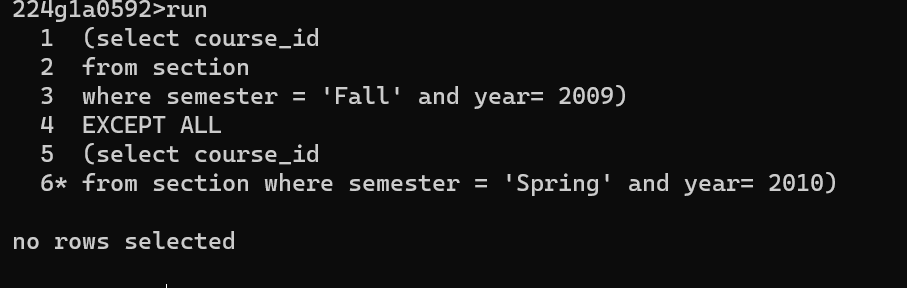
Intersect all operation:



except or minus operation:

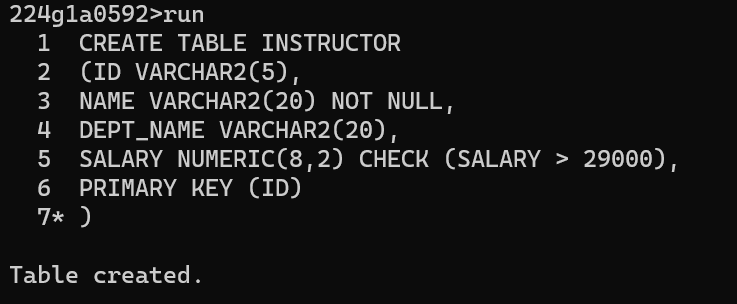


except all or minus all operations:



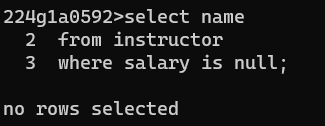
5.SPECIAL OPERATIONS

Write SQL queries to perform SPECIAL OPERATIONS (i.e. ISNULL, BETWEEN, LIKE, IN, EXISTS).



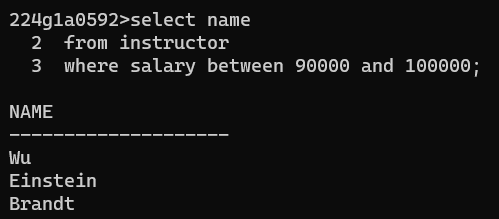


IS NULL

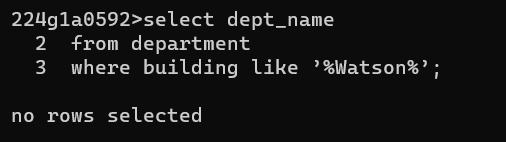




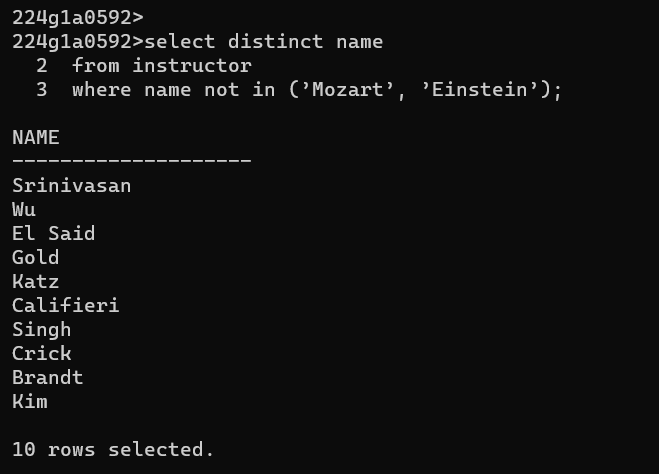
BETWEEN



LIKE

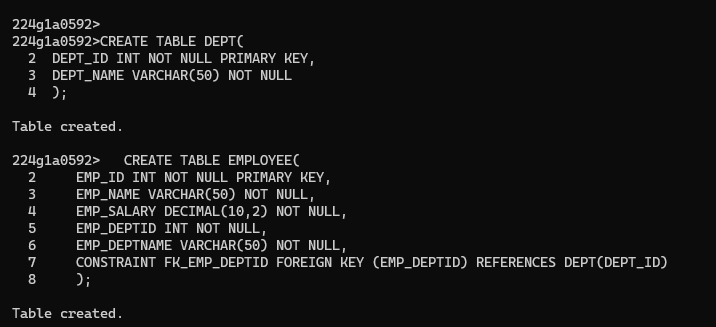


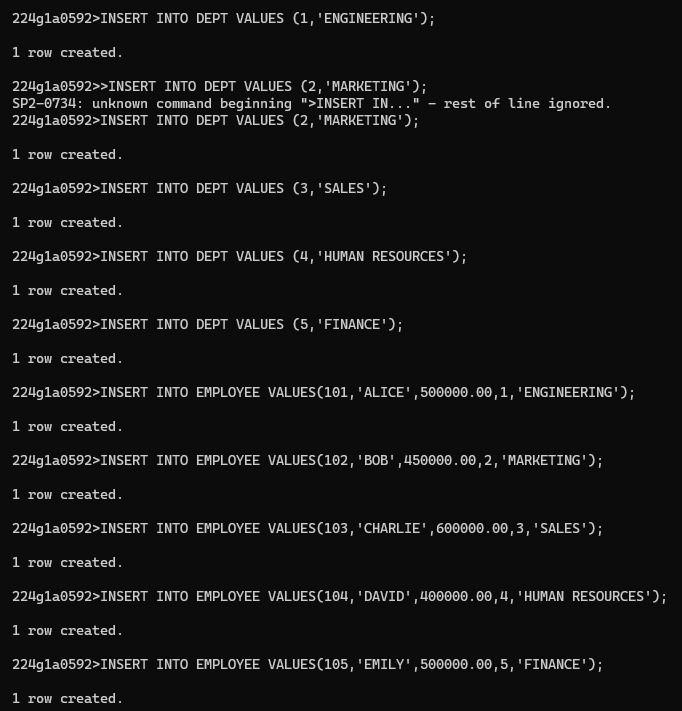
IN



6.JOIN OPERATIONS

Write SQL queries to perform JOIN OPERATIONS (i.e. CONDITIONAL JOIN, EQUI JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN)

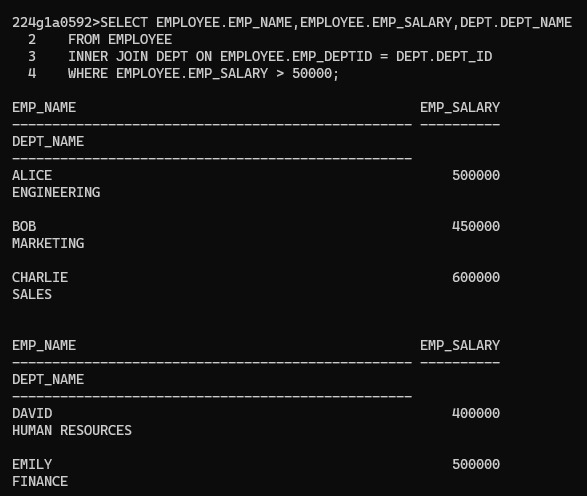




**Natural JOIN**



**CONDITIONAL JOIN**



**OUTER JOINS**



**left outer join operation.**

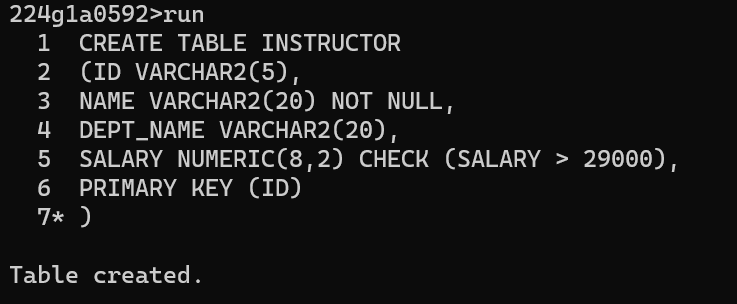


**RIGHT OUTER JOIN**

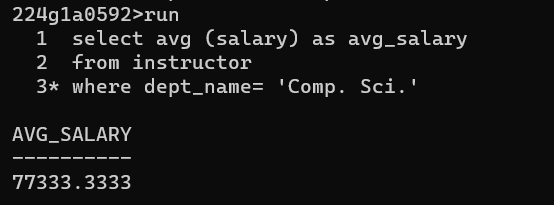


7. AGGREGATE OPERATIONS

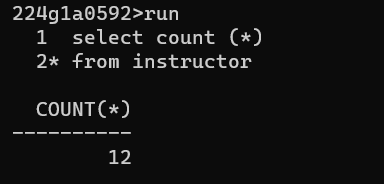
Write SQL queries to perform AGGREGATE OPERATIONS (i.e. SUM, COUNT, AVG, MIN, MAX).



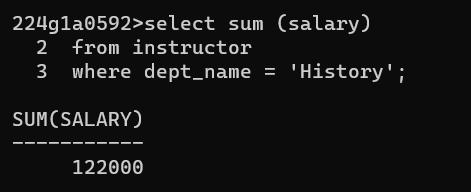
AVG



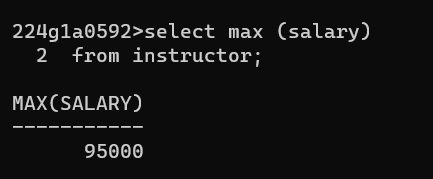
COUNT



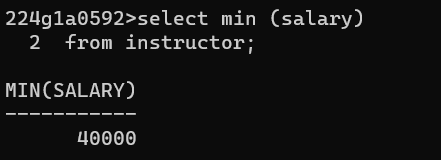
SUM



MAX



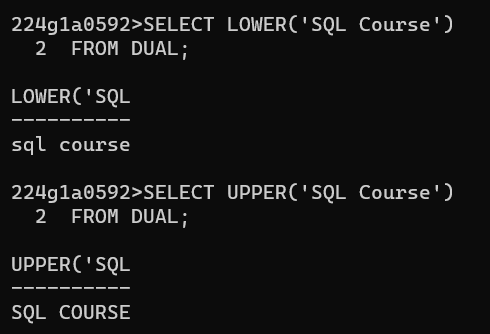
MIN

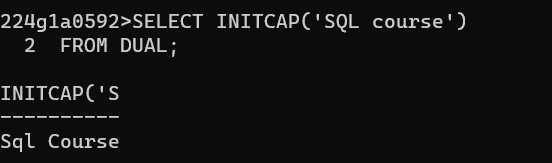


8. ORACLE BUILT-IN FUNCTIONS

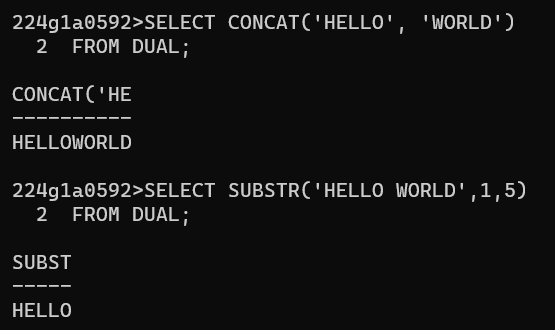
Write SQL queries to perform ORACLE BUILT-IN FUNCTIONS (i.e. DATE, TIME)

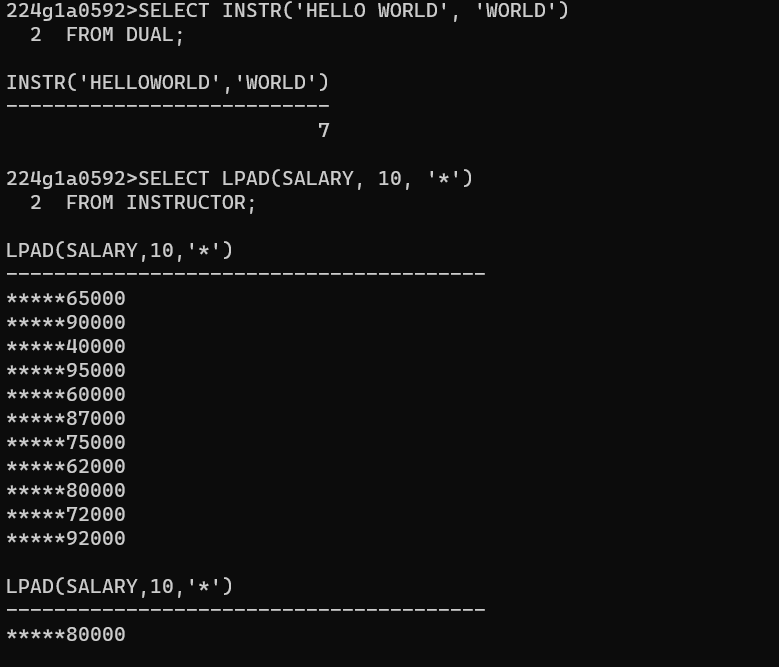
case-conversion functions:



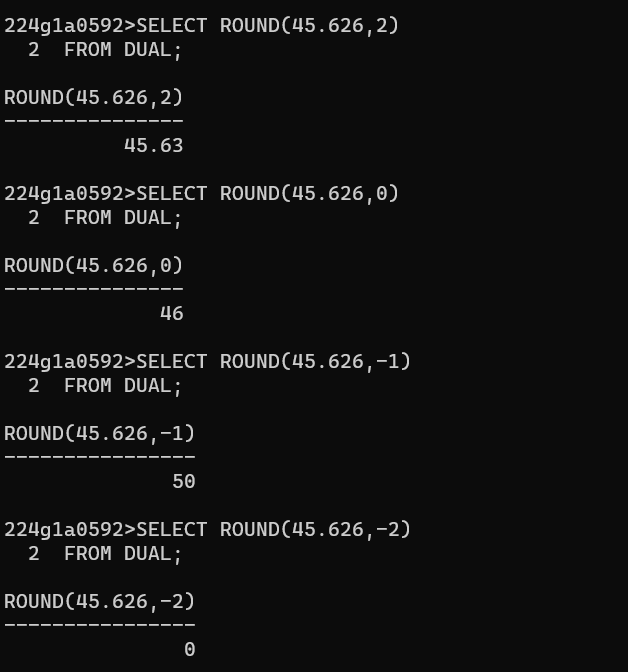


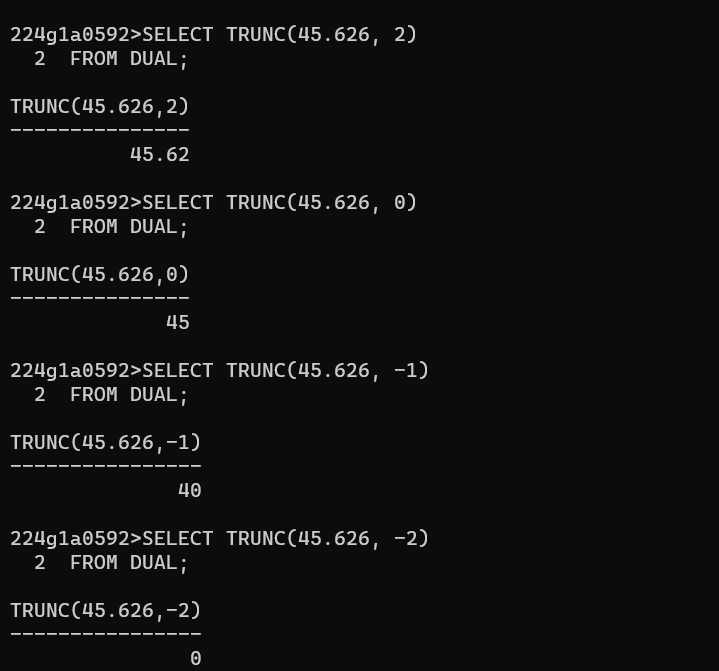
character manipulation functions:





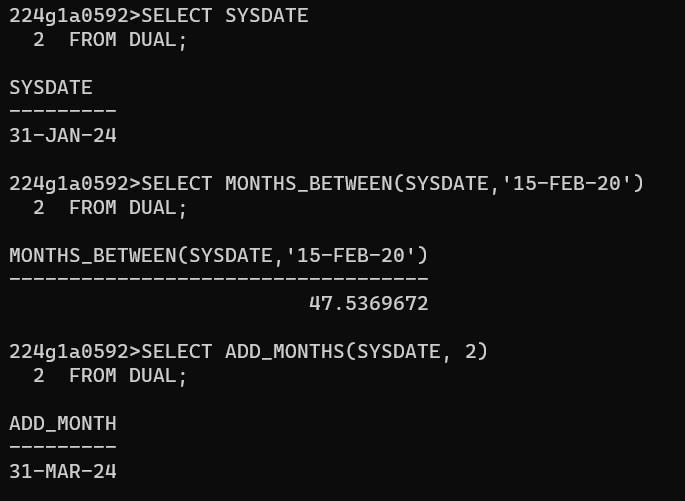
Number Functions:

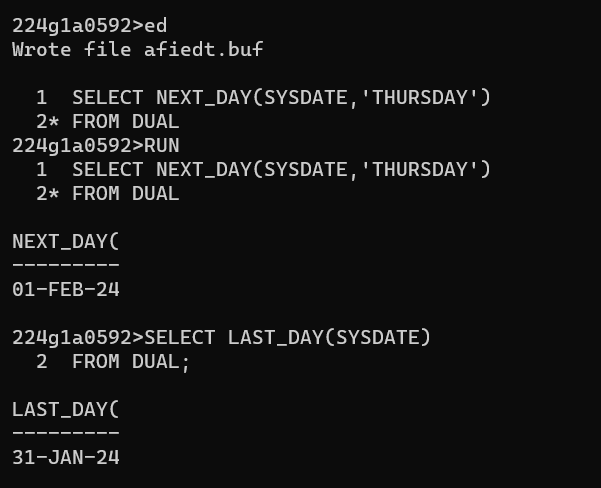




Date functions:

DATE FUNCTIONS

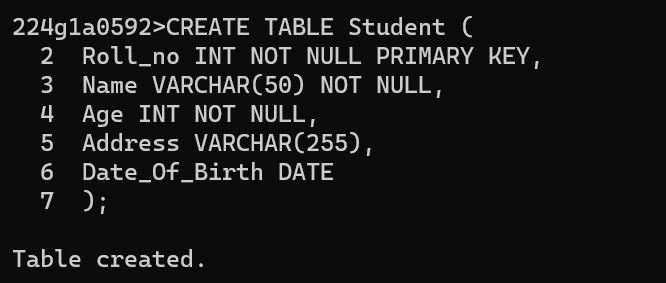




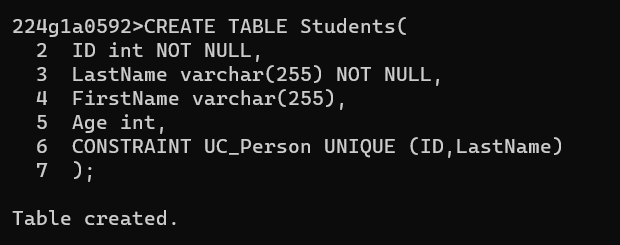
9.KEY CONSTRAINTS

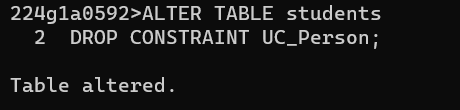
Write SQL queries to perform KEY CONSTRAINTS (i.e. PRIMARY KEY, FOREIGN KEY, UNIQUE NOT NULL, CHECK, DEFAULT)

NOT NULL COnstraint Example

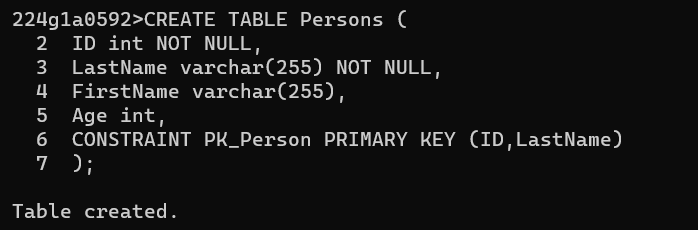


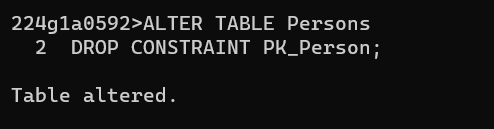
UNIQUE CONSTRAINT Example



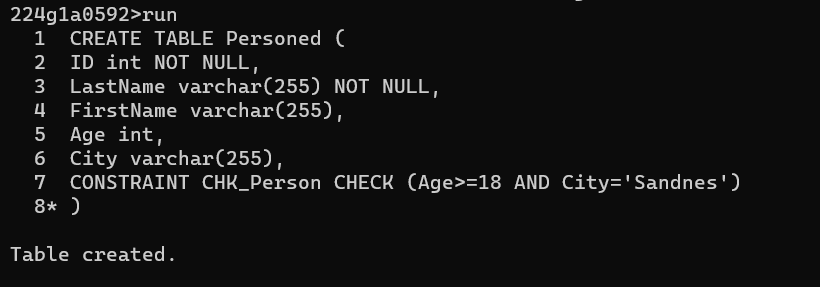


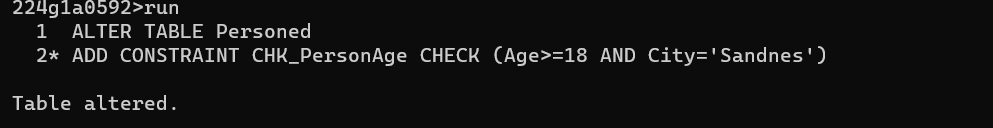
PRIMARY KEY CONSTRAINT Example:

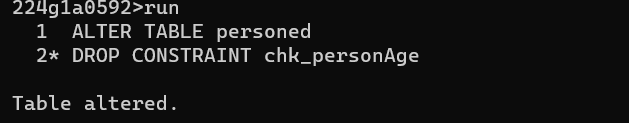




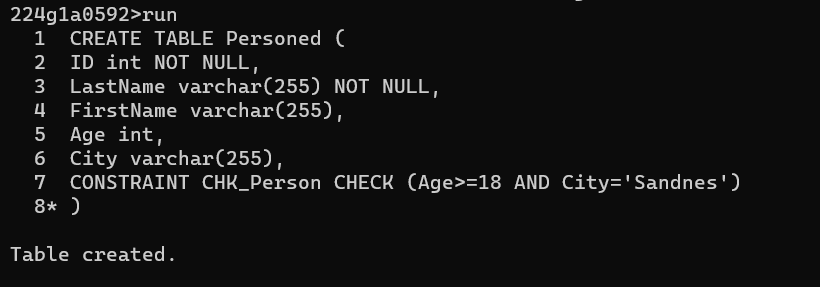
CHECK CONSTRAINTS Example:

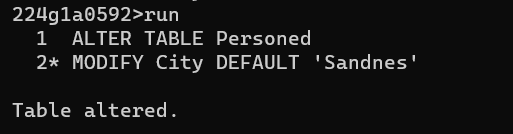


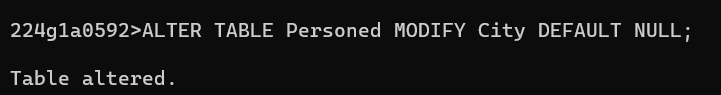




DEFAULT CONSTRAINTS Example:

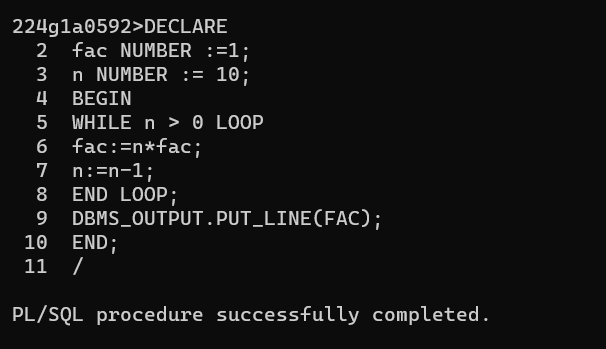






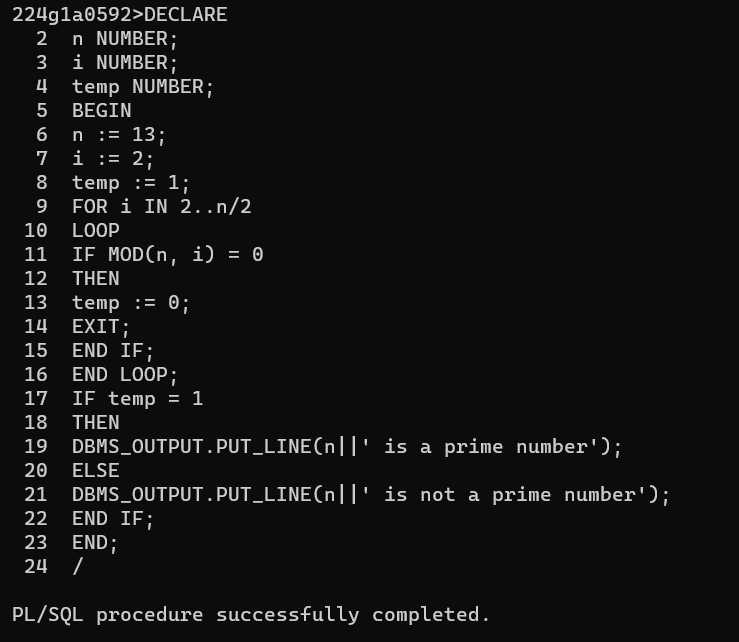
10. FACTORIAL

Write a PL/SQL program for calculating the factorial of a given number



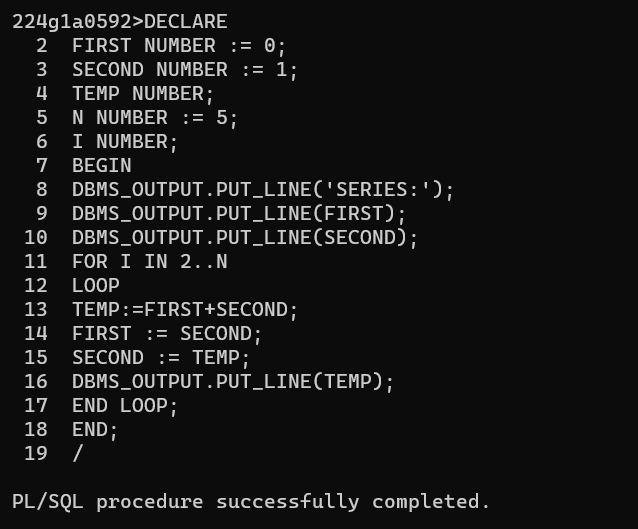
11.PRIME NUMBER OR NOT

Write a PL/SQL program for finding the given number is prime number or not.



12.FIBONACCI

Write a PL/SQL program for displaying the Fibonacci series up to an integer



13.STORED PROCEDURE

Write PL/SQL program to implement Stored Procedure on table.

SYNTAX:

CREATE [OR REPLACE] PROCEDURE procedure\_name

[ (parameter [,parameter]) ]

(IS | AS)

[declaration\_section]

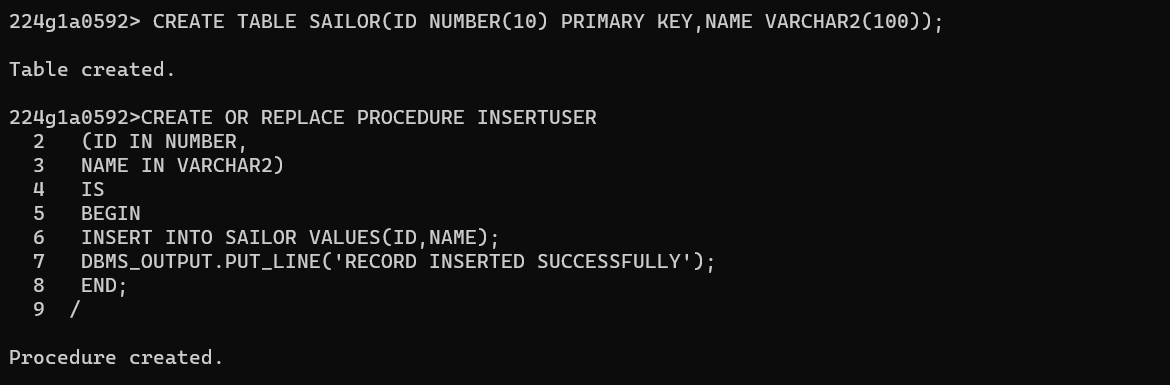
BEGIN

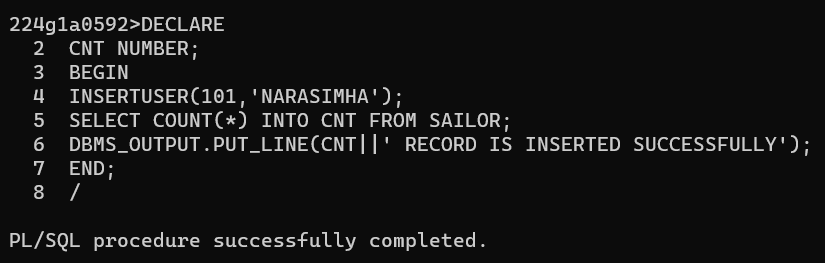
executable\_section

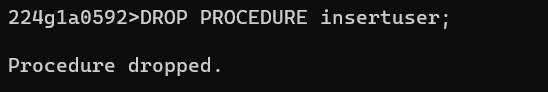
[EXCEPTION exception\_section]

END [procedure\_name];

Example:







14.STORED FUNCTION

Write PL/SQL program to implement Stored Function on table.

SYNTAX:

CREATE [OR REPLACE] FUNCTION function\_name

[ (parameter [,parameter]) ]

RETURN return\_datatype

(IS | AS)

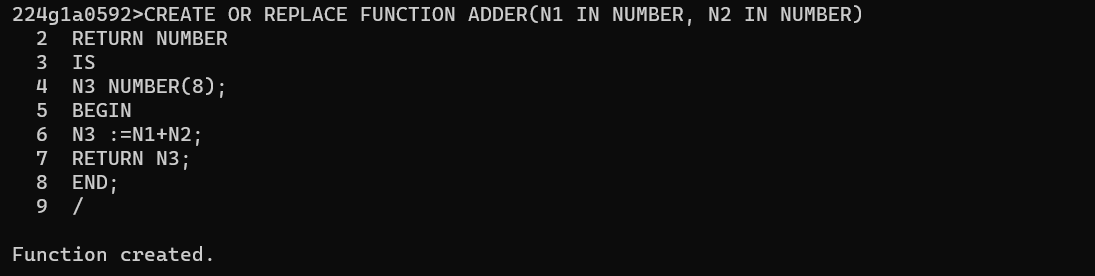
[declaration\_section]

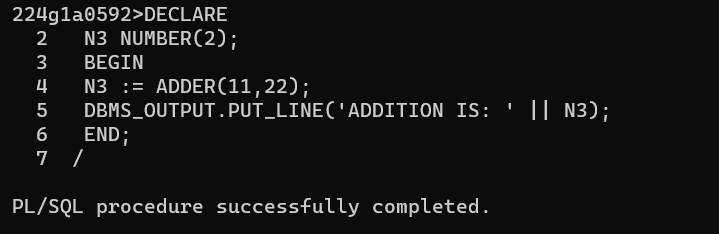
BEGIN executable\_section

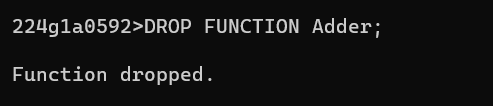
[EXCEPTION exception\_section]

END [procedure\_name];

Example:







15.IMPLEMENT TRIGGER

Write PL/SQL program to implement Trigger on table

Syntax:

CREATE [OR REPLACE ] TRIGGER TRIGGER\_NAME

{BEFORE | AFTER | INSTEAD OF }

{INSERT [OR] | UPDATE [OR] | DELETE}

[OF COL\_NAME]

ON TABLE\_NAME

[REFERENCING OLD AS O NEW AS N]

[FOR EACH ROW]

WHEN (CONDITION)

DECLARE

DECLARATION-STATEMENTS

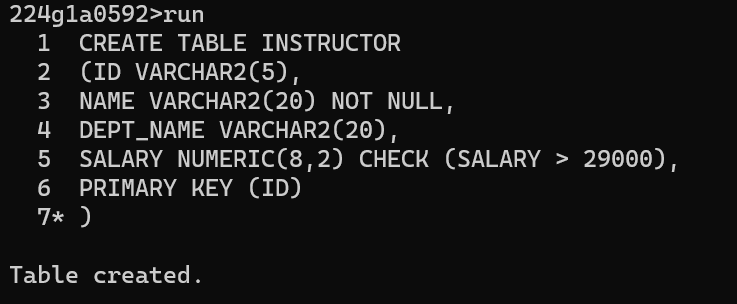
BEGIN

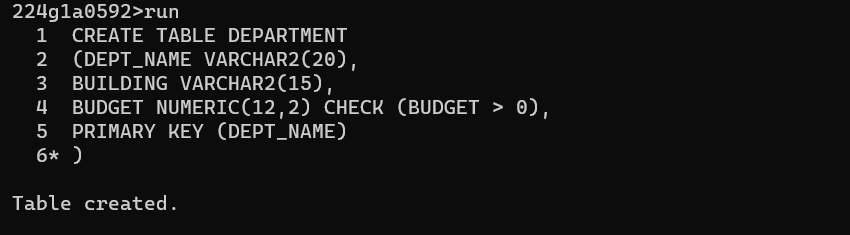
EXECUTABLE-STATEMENTS

EXCEPTION

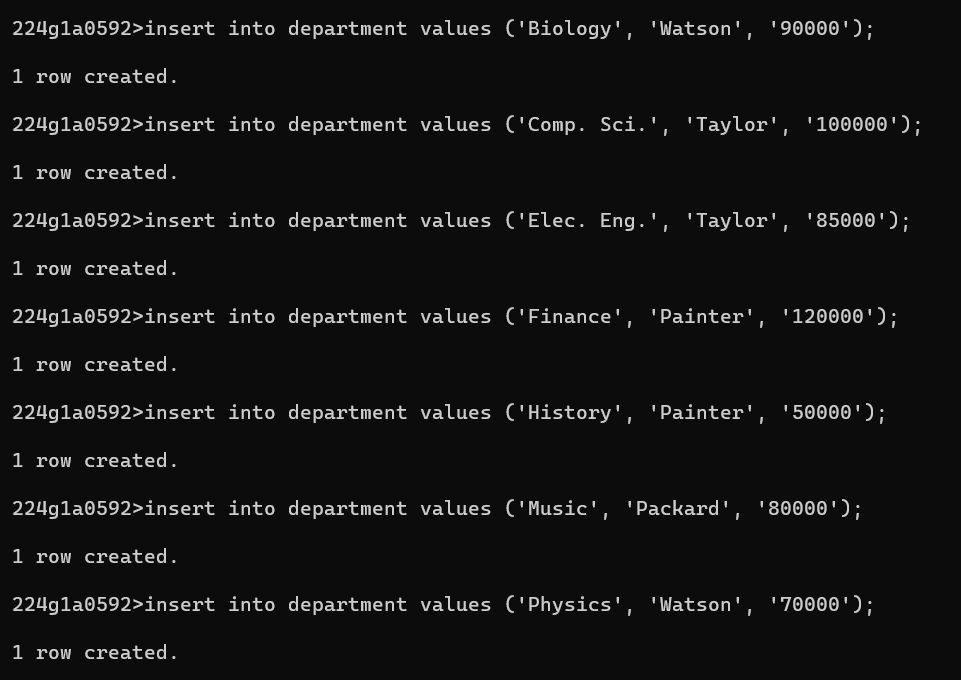
EXCEPTION-HANDLING-STATEMENTS

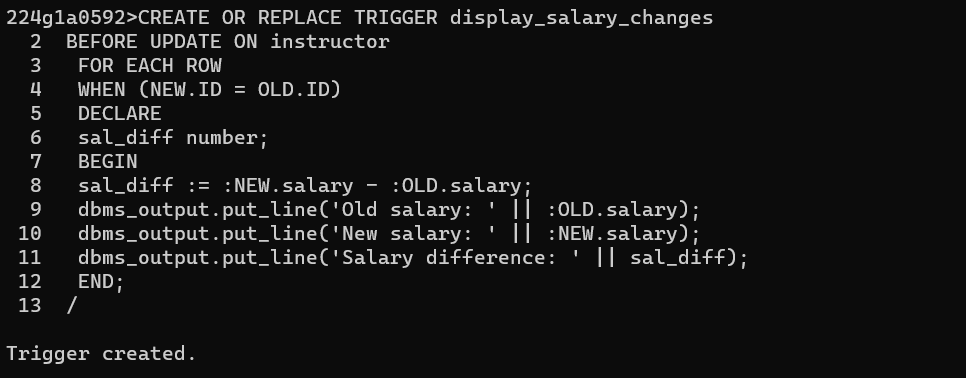
END;

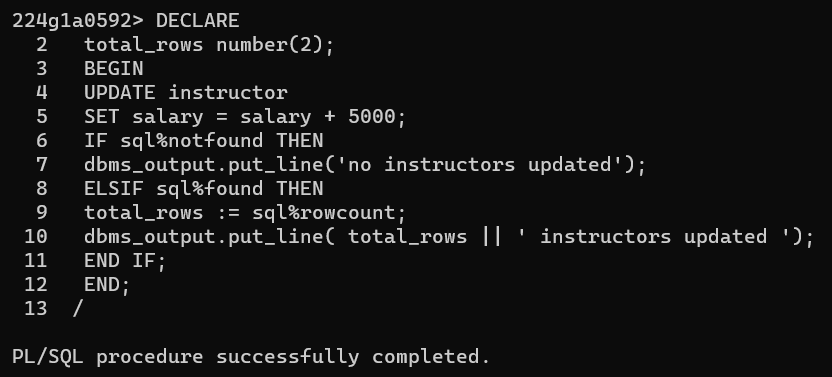












16.IMPLEMENT CURSOR

Write PL/SQL program to implement Cursor on table

Declare the cursor:

SYNTAX:

CURSOR cursor\_name IS select\_statement;

Open the cursor

SYNTAX:

OPEN cursor\_name;

Fetch the cursor

SYNTAX:

FETCH cursor\_name INTO variable\_list;

Close the cursor:

SYNTAX:

Close cursor\_name;

