USE HelloWorld;

DROP TABLE IF EXISTS Employee;

CREATE TABLE Employee

(

EMPID INT PRIMARY KEY NOT NULL,

NAME VARCHAR(50) NOT NULL,

COUNTRY VARCHAR(50) NOT NULL,

SALARY DECIMAL(8,2) NOT NULL,

JOINING\_DATE DATETIME NOT NULL,

DEPARTMENT VARCHAR(50) NOT NULL

);

-- CREATING ANOTHER TABLE - Employee\_personal

-- Question 16 - Create a table Employee\_personal

DROP TABLE IF EXISTS Employee\_personal;

CREATE TABLE Employee\_personal

(

EMPID INT PRIMARY KEY NOT NULL,

NAME VARCHAR(50) NOT NULL,

PLACE VARCHAR(50) NOT NULL,

GENDER VARCHAR(50) NOT NULL,

AVAILABLE VARCHAR(50) NOT NULL,

PHNO VARCHAR(50) NOT NULL,

DEPARTMENT VARCHAR(50) NOT NULL

);

INSERT INTO

Employee

(EMPID, NAME, COUNTRY, SALARY, JOINING\_DATE, DEPARTMENT)

VALUES

(001, 'Shippy', 'Jamaica', 100000, '2022-02-20 09:00:00', 'S/W Engg-A'),

(002, 'Christopher', 'Jamaica', 800000, '2022-06-11 09:00:00', 'S/W Engg'),

(003, 'Alanzo', 'India', 300000, '2022-02-20 09:00:00', 'S/W Engg'),

(004, 'Ezra', 'Jamaica', 500000, '2022-02-20 09:00:00', 'S/W Engg-C'),

(005, 'Grant', 'India', 500000, '2022-06-11 09:00:00', 'S/W Engg'),

(006, 'Olando', 'Jamaica', 200000, '2022-06-11 09:00:00', 'S/W Engg'),

(007, 'Camille', 'India', 705000, '2022-01-20 09:00:00', 'S/W Engg-B'),

(008, 'Donald', 'India', 900000, '2022-04-11 09:00:00', 'S/W Engg');

INSERT INTO

Employee\_personal

(EMPID, NAME, PLACE, GENDER, AVAILABLE, PHNO, DEPARTMENT)

VALUES

(001, 'Shippy', 'Kingston', 'Male', 'Yes', '(876) 987-8346', 'S/W Engg-A'),

(002, 'Christopher', 'Montego Bay', 'Male', 'No', '(876) 785-6346', 'S/W Engg'),

(003, 'Alanzo', 'New Delhi', 'Male', 'Yes', '(876) 864-5346', 'S/W Engg'),

(004, 'Ezra', 'Saint James', 'Male', 'Yes', '(876) 879-5346', 'S/W Engg-C'),

(005, 'Grant', 'Chennai', 'Male', 'No', '(876) 956-8346', 'S/W Engg'),

(006, 'Olando', 'Portmore', 'Male', 'Yes', '(876) 849-4346', 'S/W Engg'),

(007, 'Camille', 'Coimbatore', 'Female', 'No', '(876) 987-5346', 'S/W Engg-B'),

(008, 'Donald', 'Udumalper', 'Male', 'No', '(876) 895-4346', 'S/W Engg');

-- SQL Query

SELECT \*

FROM Employee;

SELECT \*

FROM Employee\_personal;

-- Question 1 - Write an SQL query to fetch “NAME” from Employee table

SELECT NAME

FROM Employee;

-- Question 2 - Write an SQL query to fetch “NAME” from Employee table in upper case.

SELECT UPPER(NAME) AS 'Upper Case Names'

FROM Employee;

-- Question 3 - Write an SQL query to fetch unique values of DEPARTMENT.

SELECT DISTINCT DEPARTMENT

FROM Employee;

-- Question 4 - Write an SQL query to print the first three characters of  NAME from table.

SELECT SUBSTRING(NAME, 1, 3)

FROM Employee;

-- Question 5 - Write an SQL query to find the position of the alphabet (‘a’) in the first name column ‘Alanzo’ from the table.

SELECT NAME, CHARINDEX('a', 'Alanzo') AS 'Index of a', LEN('Alanzo')-LEN(REPLACE('Alanzo','a','')) AS 'Number of a'

FROM Employee

WHERE NAME='Alanzo';

SELECT NAME, CHARINDEX('a', 'Ezra') AS 'Position At'

FROM Employee

WHERE NAME='Ezra';

-- Question 6 - Change the column name from “SALARY” to “Basic Pay”

EXEC sp\_rename 'Employee.SALARY', 'BASIC\_PAY', 'COLUMN';

-- Question 7 - Write an SQL query to add column like “DA”, “Deductions”, “Gross Pay”, “Net Pay”.

ALTER TABLE Employee

ADD DA DECIMAL(8,2),

DEDUCTIONS DECIMAL(8,2),

GROSS\_PAY DECIMAL(8,2),

NET\_PAY DECIMAL(8,2);

-- Question 8 - DA = 2% of Basic Pay

UPDATE Employee

SET DA = 0.02 \* BASIC\_PAY;

-- Question 9 - Deductions = 3% of Basic Pay

UPDATE Employee

SET DEDUCTIONS = 0.03 \* BASIC\_PAY;

-- Question 10 - Gross Pay = Basic Pay + DA

UPDATE Employee

SET GROSS\_PAY = BASIC\_PAY + DA;

-- Question 11 - Net Pay = Gross Pay - Deductions

UPDATE Employee

SET NET\_PAY = GROSS\_PAY - DEDUCTIONS;

-- Question 12 - Display highest paid employee name

SELECT NAME, NET\_PAY

FROM Employee

WHERE NET\_PAY = (SELECT MAX(NET\_PAY)

FROM Employee);

-- Question 13 - Display lowest paid employee name

SELECT NAME, NET\_PAY

FROM Employee

WHERE NET\_PAY = (SELECT MIN(NET\_PAY)

FROM Employee);

-- Question 14 - Display average basic pay of employees

SELECT AVG(BASIC\_PAY) AS 'Average BASIC\_PAY'

FROM Employee;

-- Question 15 - Display the average DA

SELECT AVG(DA) AS 'Average DA'

FROM Employee;

-- Question 16 - See Above for Table creation and insert

-- QUERIES FOR JOINING

-- Question 17 - Display the Place and Phone number of all employees.

SELECT

e.EMPID, e.NAME, ep.PLACE, ep.PHNO

FROM

Employee e

INNER JOIN Employee\_personal ep

ON e.NAME = ep.NAME

ORDER BY

NAME DESC;

-- Question 18 - Display the Place and Country of employees with availability “Yes”

SELECT

e.NAME, e.COUNTRY, ep.PLACE, ep.AVAILABLE

FROM

Employee e

INNER JOIN Employee\_personal ep

ON e.EMPID = ep.EMPID

WHERE

ep.AVAILABLE = 'Yes'

ORDER BY

NAME DESC;

-- Question 19 - Display all the name and Salary of location of employee in “Coimbatore”

SELECT

e.NAME, e.BASIC\_PAY, e.DEPARTMENT, ep.PLACE

FROM

Employee e

INNER JOIN Employee\_personal ep

ON e.EMPID = ep.EMPID

WHERE

ep.PLACE = 'Coimbatore'

ORDER BY

NAME DESC;

-- Question 20 - Display the phone number, gender and name of employee with designation “S/W Engg “

SELECT

e.NAME, ep.PHNO, ep.GENDER, e.DEPARTMENT

FROM

Employee e

INNER JOIN Employee\_personal ep

ON e.NAME = ep.NAME

WHERE

e.DEPARTMENT = 'S/W Engg'

ORDER BY

NAME DESC;

-- Question 21 - Write a query to Filter Data using where clause gender = “Female”

SELECT

NAME, GENDER

FROM

Employee\_personal

WHERE

GENDER = 'Female';

-- Question 22 - Sort the Employee Name using the ORDER BY Clause.

SELECT

NAME

FROM

Employee

ORDER BY NAME;

-- Question 23 - Display the TOP (5) high salaried employees – display their name, salary and department.

SELECT TOP 5

NAME, BASIC\_PAY, DEPARTMENT

FROM

Employee

ORDER BY BASIC\_PAY DESC;

-- Question 24 - Retrieve the elements like available and Employee name using OFFSET - FETCH

SELECT

e.NAME,

ep.AVAILABLE

FROM

Employee e

FULL OUTER JOIN Employee\_personal ep

ON e.NAME = ep.NAME AND e.EMPID = ep.EMPID

ORDER BY

e.NAME

OFFSET 4 ROWS

FETCH NEXT 4 ROWS ONLY;

-- Question 25 - Use the following Aggregate functions and frame a query from the table - Employee :

-- i. SUM()

-- ii. COUNT()

-- iii. AVG()

-- iv. MIN()

-- v. MAX()

-- vi. FIRST\_VALUE()

-- vii. LAST\_VALUE()

SELECT SUM(BASIC\_PAY) AS sum\_of\_basic\_pay

FROM Employee;

SELECT COUNT(EMPID) count\_of\_employee

FROM Employee;

SELECT AVG(BASIC\_PAY) AS average\_of\_basic\_pay

FROM Employee;

SELECT MIN(BASIC\_PAY) AS minimum\_of\_basic\_pay

FROM Employee;

SELECT MAX(BASIC\_PAY) AS maximum\_of\_basic\_pay

FROM Employee;

SELECT \*, FIRST\_VALUE(BASIC\_PAY) OVER (ORDER BY BASIC\_PAY) AS first\_value\_of\_basic\_pay

FROM Employee;

SELECT NAME, BASIC\_PAY, LAST\_VALUE(BASIC\_PAY) OVER (PARTITION BY BASIC\_PAY ORDER BY BASIC\_PAY) AS last\_value\_of\_basic\_pay

FROM Employee;

-- Question 26 - Use the following Scalar functions and frame a query from the table - Employee\_personal :

-- i. LCASE()

-- ii. UCASE()

-- iii. LEN()

-- iv. MID is SUBSTRING() in sql server

-- v. ROUND()

-- vi. NOW() is SYSDATETIME() in sql server

-- vii. FORMAT()

SELECT UPPER(NAME) AS upper\_case\_of\_name

FROM Employee\_personal;

SELECT LOWER(NAME) AS lower\_case\_of\_name

FROM Employee\_personal;

SELECT NAME, LEN(NAME) AS length\_of\_name

FROM Employee\_personal;

SELECT NAME, SUBSTRING(NAME, 1, 4) AS substring\_of\_name

FROM Employee\_personal;

SELECT AVG(e.BASIC\_PAY) AS average\_basic\_pay, ROUND(AVG(e.BASIC\_PAY), 2) AS rounded\_basic\_pay

FROM Employee e

INNER JOIN Employee\_personal ep

ON e.NAME = ep.NAME;

-- SYS.DATETIME() is the current date and time

SELECT SYSDATETIME() AS current\_date\_time

FROM Employee\_personal

WHERE NAME = 'Ezra';

-- FORMAT() is a function that returns the formatted date and time

SELECT FORMAT(SYSDATETIME(), 'dd-MM-yyyy hh:mm:ss') AS formatted\_date\_time

FROM Employee\_personal;

-- 27. Update the following columns to the table

-- Employee (i) (iii) and Employee\_Personal (ii) (iii)

-- i. Rank 1,2,3,1,2,3,1,2

-- ii. Grade A,B,C,D,A,B,C,D

-- iii. Project Platform – Java , Python, Oracle, Flutter, Springboard, Oracle, Flutter, Springboard

ALTER TABLE Employee\_personal

ADD Rank INT,

Grade CHAR(1),

Project\_Platform VARCHAR(50);

UPDATE Employee\_personal

SET Rank = 1, Grade = 'A', Project\_Platform = 'Java'

WHERE NAME = 'Shippy';

UPDATE Employee\_personal

SET Rank = 2, Grade = 'B', Project\_Platform = 'Python'

WHERE NAME = 'Christopher';

UPDATE Employee\_personal

SET Rank = 3, Grade = 'C', Project\_Platform = 'Oracle'

WHERE NAME = 'Alanzo';

UPDATE Employee\_personal

SET Rank = 4, Grade = 'D', Project\_Platform = 'Flutter'

WHERE NAME = 'Ezra';

UPDATE Employee\_personal

SET Rank = 5, Grade = 'E', Project\_Platform = 'Springboard'

WHERE NAME = 'Grant';

UPDATE Employee\_personal

SET Rank = 6, Grade = 'F', Project\_Platform = 'Oracle'

WHERE NAME = 'Olando';

UPDATE Employee\_personal

SET Rank = 7, Grade = 'G', Project\_Platform = 'Flutter'

WHERE NAME = 'Camille';

UPDATE Employee\_personal

SET Rank = 8, Grade = 'H', Project\_Platform = 'Springboard'

WHERE NAME = 'Donald';

-- 28. Apply the Rank (), Dense\_rank() and Row\_number() function in all possible ways in both table.

SELECT \*

FROM Employee\_personal;

-- To check the changes

-- RANK()

SELECT

NAME, Rank, Grade, Project\_Platform,

RANK() OVER (

ORDER BY Project\_Platform

) AS rank\_of\_project\_platform

FROM Employee\_personal;

-- RANK() with OUTER JOIN

SELECT

ep.NAME, Rank, Grade, Project\_Platform,

RANK() OVER (

PARTITION BY Project\_Platform

ORDER BY Project\_Platform

) AS rank\_of\_project\_platform

FROM Employee\_personal ep

FULL OUTER JOIN Employee e

ON ep.NAME = e.NAME;

-- DENSE\_RANK()

SELECT

NAME, Rank, Grade, Project\_Platform,

DENSE\_RANK() OVER (

ORDER BY Project\_Platform

) AS rank\_of\_project\_platform

FROM Employee\_personal;

-- ROW\_NUMBER()

SELECT

NAME, Rank, Grade, Project\_Platform, DEPARTMENT,

ROW\_NUMBER() OVER (

PARTITION BY DEPARTMENT

ORDER BY NAME

) AS rank\_of\_row\_number

FROM Employee\_personal;

-- Question 29 - Implement the following Set Operators in both the tables(write syntax and example with existing tables)

-- i. UNION

SELECT

EMPID, NAME, DEPARTMENT

FROM Employee

UNION

SELECT

EMPID, NAME, DEPARTMENT

FROM Employee\_personal;

-- ii. UNION ALL

SELECT

EMPID, NAME, DEPARTMENT

FROM Employee

UNION ALL

SELECT

EMPID, NAME, DEPARTMENT

FROM Employee\_personal;

-- iii. INTERSECT

SELECT

EMPID, NAME, DEPARTMENT

FROM Employee

INTERSECT

SELECT

EMPID, NAME, DEPARTMENT

FROM Employee\_personal;

-- iv. MINUS (i.e. EXCEPT in sql server)

SELECT

EMPID, NAME, DEPARTMENT

FROM Employee

EXCEPT

SELECT

EMPID, NAME, DEPARTMENT

FROM Employee\_personal;

-- Question 30