USE pubs

--1) Select the author firstname and last name

SELECT au\_fname 'First Name', au\_lname 'Last Name' FROM authors

ORDER BY 1

SELECT CONCAT(au\_fname,' ', au\_lname) 'Author Full Name' FROM authors

ORDER BY 1

--2) Sort the titles by the title name in descending order and print all the details

SELECT \* FROM titles

ORDER BY title DESC

--3) Print the number of titles published by every author

SELECT TA.au\_id 'Author ID', COUNT(T.title) 'No. of Titles'

FROM titles T JOIN titleauthor TA

ON T.title\_id = TA.title\_id

GROUP BY TA.au\_id

--4) print the author name and title name

SELECT CONCAT(A.au\_fname,' ', A.au\_lname) 'Author Full Name', title 'Title Name'

FROM titles T JOIN titleauthor TA

ON T.title\_id = TA.title\_id

JOIN authors A ON A.au\_id = TA.au\_id

ORDER BY 1

--5) print the publisher name and the average advance for every publisher

SELECT P.pub\_name 'Publisher Name', AVG(T.advance) 'Average Advance'

FROM publishers P LEFT OUTER JOIN titles T

ON P.pub\_id = T.pub\_id

GROUP BY P.pub\_name

ORDER BY 1

--6) print the publishername, author name, title name and the sale amount(qty\*price)

SELECT P.pub\_name 'Publisher Name', CONCAT(A.au\_fname, ' ', A.au\_lname) 'Author Name',

T.title 'Title Name', S.qty\*T.price 'Sale Amount'

FROM authors A JOIN titleauthor TA ON A.au\_id = TA.au\_id

JOIN titles T ON T.title\_id = TA.title\_id

JOIN publishers P ON P.pub\_id = T.pub\_id

JOIN sales S ON S.title\_id = T.title\_id

ORDER BY 1

--7) print the price of all that titles that have name that ends with s

SELECT title 'Title', price 'Price' FROM titles

WHERE title LIKE '%s'

--8) print the title names that contain 'and' in it

SELECT title 'Title' FROM titles

WHERE title LIKE '%and%'

--9) print the employee name and the publisher name

SELECT CONCAT(E.fname, ' ', E.lname) 'Employee Name', P.pub\_name 'Publisher Name'

FROM publishers P JOIN employee E

ON P.pub\_id = E.pub\_id

ORDER BY 2, 1

--10) print the publisher name and number of employees woking in it if the publisher has more than 2 employees

SELECT P.pub\_name 'Publisher Name', COUNT(E.pub\_id) 'No. of Employee'

FROM publishers P JOIN employee E

ON P.pub\_id = E.pub\_id

GROUP BY P.pub\_name

HAVING COUNT(E.pub\_id) > 2

ORDER BY 1

--11) Print the author names who have published using the publisher name 'Algodata Infosystems'

SELECT CONCAT(A.au\_fname, ' ', A.au\_lname) 'Author Name' FROM authors A

JOIN titleauthor TA ON A.au\_id = TA.au\_id

JOIN titles T ON T.title\_id = TA.title\_id

JOIN publishers P ON P.pub\_id = T.pub\_id

WHERE P.pub\_name = 'Algodata Infosystems'

ORDER BY 1

--12) Print the employees of the publisher 'Algodata Infosystems'

SELECT \* FROM employee E JOIN publishers P

ON P.pub\_id = E.pub\_id

WHERE P.pub\_name = 'Algodata Infosystems'

ORDER BY 1

--13)Create the following tables

--Employee(id-identity starts in 100 inc by 1, Name,age, phone cannot be null, gender)

--Salary(id-identity starts at 1 increments by 100, Basic,HRA,DA,deductions)

--EmployeeSalary(transaction\_number int, employee\_id-reference Employee's Id, Salary\_id reference Salary Id,Date)

--PS - In the emeployee salary table transaction number is the primary key

--the combination of employee\_id, salary\_id and date should always be unique

--CREATE DATABASE dbSampleTutorialD3

USE dbSampleTutorialD3

CREATE TABLE Employee(

Emp\_id INT IDENTITY(100,1) PRIMARY KEY,

Emp\_name VARCHAR(20),

Emp\_age INT,

Emp\_phone VARCHAR(20) NOT NULL,

Emp\_gender VARCHAR(20) CHECK (Emp\_gender in ('Male', 'Female', 'Other')))

CREATE TABLE Salary(

Sal\_id INT IDENTITY(100,1) PRIMARY KEY,

Sal\_basic FLOAT,

Sal\_HRA FLOAT,

Sal\_DA FLOAT,

Sal\_deductions FLOAT)

CREATE TABLE EmployeeSalary(

Tran\_num INT IDENTITY(1,1) PRIMARY KEY,

Emp\_id INT CONSTRAINT fk\_empId FOREIGN KEY REFERENCES Employee(Emp\_id),

Sal\_id INT CONSTRAINT fk\_salId FOREIGN KEY REFERENCES Salary(Sal\_id),

Tran\_date DATE,

UNIQUE (Emp\_id, Sal\_id, Tran\_date)

)

--Add a column email-varchar(100) to the employee table

ALTER TABLE Employee ADD Emp\_email VARCHAR(100)

--Insert few records in all the tables

INSERT INTO Employee VALUES('Aria', 20, '0123456789', 'Female', 'aria20@email.com')

INSERT INTO Employee VALUES('Bob', 22, '0113456789', 'Male', 'bob22@email.com')

INSERT INTO Employee VALUES('Carson', 22, '0111456789', 'Other', 'iamacar@email.com')

INSERT INTO Salary VALUES(5000,1000,1500,500)

INSERT INTO Salary VALUES(15000,400,600,550)

INSERT INTO Salary VALUES(10000,500,1000,50)

INSERT INTO EmployeeSalary VALUES(100,100,'2021-10-01')

INSERT INTO EmployeeSalary VALUES(101,101,'2021-10-02')

INSERT INTO EmployeeSalary VALUES(102,102,'2021-10-03')

INSERT INTO EmployeeSalary VALUES(100,102,'2021-10-04')

--Create a procedure which will print the total salary of employee by taking the employee id and the date

--total = Basic+HRA+DA-deductions

CREATE PROC sp\_totalEmpSalary(

@Emp\_id INT,

@date DATE

)

AS

BEGIN

DECLARE

@Sal\_id INT,

@basic FLOAT,

@HRA FLOAT,

@DA FLOAT,

@deductios FLOAT,

@totalSal FLOAT

SET @Sal\_id = (SELECT Sal\_id FROM EmployeeSalary WHERE Emp\_id = @Emp\_id AND Tran\_date = @date)

SET @basic = (SELECT Sal\_basic FROM Salary WHERE Sal\_id = @Sal\_id)

SET @HRA = (SELECT Sal\_HRA FROM Salary WHERE Sal\_id = @Sal\_id)

SET @DA = (SELECT Sal\_DA FROM Salary WHERE Sal\_id = @Sal\_id)

SET @deductios = (SELECT Sal\_deductions FROM Salary WHERE Sal\_id = @Sal\_id)

SET @totalSal = @basic + @HRA + @DA - @deductios

PRINT 'Total Salary of Employee ID: ' + CAST(@Emp\_id AS VARCHAR(20)) + ' is ' + CAST(@totalSal AS VARCHAR(20))

END

--Create a procedure which will calculate the average salary of an employee taking his ID

CREATE PROC sp\_averageEmpSalary(

@Emp\_id INT

)

AS

BEGIN

DECLARE

@Sal\_count INT,

@Sal\_id INT,

@basic FLOAT,

@HRA FLOAT,

@DA FLOAT,

@deductios FLOAT,

@totalSal FLOAT,

@averageSAL FLOAT

SET @Sal\_count = (SELECT COUNT(Sal\_id) FROM EmployeeSalary WHERE Sal\_id IN (SELECT Sal\_id FROM EmployeeSalary WHERE Emp\_id = @Emp\_id))

IF (@Sal\_count > 1)

BEGIN

SET @basic = (SELECT SUM(Sal\_basic) FROM Salary WHERE Sal\_id IN (SELECT Sal\_id FROM EmployeeSalary WHERE Emp\_id = @Emp\_id))

SET @HRA = (SELECT SUM(Sal\_HRA) FROM Salary WHERE Sal\_id IN (SELECT Sal\_id FROM EmployeeSalary WHERE Emp\_id = @Emp\_id))

SET @DA = (SELECT SUM(Sal\_DA) FROM Salary WHERE Sal\_id IN (SELECT Sal\_id FROM EmployeeSalary WHERE Emp\_id = @Emp\_id))

SET @deductios = (SELECT SUM(Sal\_deductions) FROM Salary WHERE Sal\_id IN (SELECT Sal\_id FROM EmployeeSalary WHERE Emp\_id = @Emp\_id))

SET @totalSal = @basic + @HRA + @DA - @deductios

SET @averageSAL = @totalSal / @Sal\_count

END

ELSE

BEGIN

SET @Sal\_id = (SELECT Sal\_id FROM EmployeeSalary WHERE Emp\_id = @Emp\_id)

SET @basic = (SELECT Sal\_basic FROM Salary WHERE Sal\_id = @Sal\_id)

SET @HRA = (SELECT Sal\_HRA FROM Salary WHERE Sal\_id = @Sal\_id)

SET @DA = (SELECT Sal\_DA FROM Salary WHERE Sal\_id = @Sal\_id)

SET @deductios = (SELECT Sal\_deductions FROM Salary WHERE Sal\_id = @Sal\_id)

SET @totalSal = @basic + @HRA + @DA - @deductios

SET @averageSAL = @totalSal

END

PRINT 'Average Salary of Employee ID: ' + CAST(@Emp\_id AS VARCHAR(20)) + ' is ' + CAST(@averageSAL AS VARCHAR(20))

END

--Create a procedure which will catculate tax payable by employee

--Slabs as follows

--total < 100000 - 0%

--100000 < total < 200000 - 5%

--200000 < total < 350000 - 6%

--total > 350000 - 7.5%

CREATE PROC sp\_calculateTaxPay(

@Emp\_id INT)

AS

BEGIN

DECLARE

@Sal\_count INT,

@Sal\_id INT,

@basic FLOAT,

@HRA FLOAT,

@DA FLOAT,

@deductios FLOAT,

@totalSal FLOAT,

@taxPay FLOAT

SET @basic = (SELECT SUM(Sal\_basic) FROM Salary WHERE Sal\_id IN (SELECT Sal\_id FROM EmployeeSalary WHERE Emp\_id = @Emp\_id))

SET @HRA = (SELECT SUM(Sal\_HRA) FROM Salary WHERE Sal\_id IN (SELECT Sal\_id FROM EmployeeSalary WHERE Emp\_id = @Emp\_id))

SET @DA = (SELECT SUM(Sal\_DA) FROM Salary WHERE Sal\_id IN (SELECT Sal\_id FROM EmployeeSalary WHERE Emp\_id = @Emp\_id))

SET @deductios = (SELECT SUM(Sal\_deductions) FROM Salary WHERE Sal\_id IN (SELECT Sal\_id FROM EmployeeSalary WHERE Emp\_id = @Emp\_id))

SET @totalSal = @basic + @HRA + @DA - @deductios

IF (@totalSal < 10000) -- 0%

BEGIN

SET @taxPay = 0

END

ELSE IF (@totalSal >10000 AND @totalSal < 20000) -- 5%

BEGIN

SET @taxPay = 0.05\*@totalSal

END

ELSE IF (@totalSal >200000 AND @totalSal <350000) -- 6%

BEGIN

SET @taxPay = 0.06\*@totalSal

END

ELSE IF (@totalSal >350000) -- 7.5%

BEGIN

SET @taxPay = 0.075\*@totalSal

END

PRINT 'The Tax payable for the Employee ID: ' + CAST(@Emp\_id AS VARCHAR(20)) + ' is ' + CAST(@taxPay AS VARCHAR(20))

END

--testing execution

EXEC sp\_totalEmpSalary 100, '2021-10-01'

EXEC sp\_averageEmpSalary 100

EXEC sp\_calculateTaxPay 100

--14) Create a function that will take the basic,HRA and da returns the sum of the three

CREATE FUNCTION fn\_sumOfThree(

@basic FLOAT,

@da FLOAT,

@hra FLOAT)

RETURNS FLOAT

AS

BEGIN

RETURN @basic + @da + @hra

END

SELECT dbo.fn\_sumOfThree(10,10,10)

--15) Create a cursor that will pick up every employee and print his details

--then print all the entries for his salary in the employeesalary table.

--Also show the salary splitt up(Hint-> use the salary table)

DECLARE

@Empl\_id INT ,

@Empl\_name VARCHAR(20),

@Empl\_age INT,

@Empl\_phone VARCHAR(20),

@Empl\_gender VARCHAR(20),

@Empl\_email VARCHAR(100)

DECLARE cur\_employee CURSOR FOR SELECT \* FROM Employee

OPEN cur\_employee

FETCH NEXT FROM cur\_employee INTO @Empl\_id, @Empl\_name, @Empl\_age , @Empl\_phone , @Empl\_gender, @Empl\_email

WHILE (@@FETCH\_STATUS = 0)

BEGIN

PRINT 'Employee number: ' + CAST(@Empl\_id AS VARCHAR(20))

PRINT 'Employee name: ' + CAST(@Empl\_name AS VARCHAR(20))

PRINT 'Employee age: ' + CAST(@Empl\_age AS VARCHAR(20))

PRINT 'Employee phone: ' + CAST(@Empl\_phone AS VARCHAR(20))

PRINT 'Employee gender: ' + CAST(@Empl\_gender AS VARCHAR(20))

PRINT 'Employee email: ' + CAST(@Empl\_email AS VARCHAR(20))

PRINT '-----------------------------------------------------------'

DECLARE

@ES\_num INT,

@ES\_emp\_id INT,

@ES\_sal\_id INT,

@ES\_date DATE

DECLARE cur\_empSalary CURSOR FOR SELECT Tran\_num, Sal\_id, Tran\_date FROM EmployeeSalary WHERE Emp\_id = @Empl\_id

OPEN cur\_empSalary

FETCH NEXT FROM cur\_empSalary INTO @ES\_num ,@ES\_sal\_id ,@ES\_date

WHILE (@@FETCH\_STATUS = 0)

BEGIN

PRINT ' Transaction number: ' + CAST(@ES\_num AS VARCHAR(20))

PRINT ' Salary ID: ' + CAST(@ES\_sal\_id AS VARCHAR(20))

PRINT ' Transaction date: ' + CAST(@ES\_date AS VARCHAR(20))

PRINT '-----------------------------------------------------------'

DECLARE

@Sala\_basic FLOAT,

@Sala\_HRA FLOAT,

@Sala\_DA FLOAT,

@Sala\_deductions FLOAT

DECLARE cur\_salary CURSOR FOR SELECT Sal\_basic, Sal\_HRA, Sal\_DA, Sal\_deductions FROM Salary WHERE Sal\_id = @ES\_sal\_id

OPEN cur\_salary

FETCH NEXT FROM cur\_salary INTO @Sala\_basic ,@Sala\_HRA ,@Sala\_DA, @Sala\_deductions

WHILE (@@FETCH\_STATUS = 0)

BEGIN

PRINT ' Basic Salary: ' + CAST(@Sala\_basic AS VARCHAR(20))

PRINT ' HRA: ' + CAST(@Sala\_HRA AS VARCHAR(20))

PRINT ' DA: ' + CAST(@Sala\_DA AS VARCHAR(20))

PRINT ' Deductions: ' + CAST(@Sala\_deductions AS VARCHAR(20))

PRINT '-----------------------------------------------------------'

FETCH NEXT FROM cur\_salary INTO @Sala\_basic ,@Sala\_HRA ,@Sala\_DA, @Sala\_deductions

END

CLOSE cur\_salary

DEALLOCATE cur\_salary

FETCH NEXT FROM cur\_empSalary INTO @ES\_num,@ES\_sal\_id ,@ES\_date

END

CLOSE cur\_empSalary

DEALLOCATE cur\_empSalary

FETCH NEXT FROM cur\_employee INTO @Empl\_id, @Empl\_name, @Empl\_age , @Empl\_phone , @Empl\_gender, @Empl\_email

END

CLOSE cur\_employee

DEALLOCATE cur\_employee