USE pubs

--1) Select the author firtname and last name

select au\_fname, au\_lname from authors

select CONCAT(au\_fname, ' ', au\_lname) 'Full Name' from authors

--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 titlespublished by every author

select CONCAT(a.au\_fname, ' ', a.au\_lname) 'Full Name', COUNT(t.title) 'Titles Published'

from authors a

join titleauthor ta on a.au\_id = ta.au\_id

join titles t on ta.title\_id = t.title\_id

group by CONCAT(a.au\_fname, ' ', a.au\_lname)

--4) print the author name and title name

select CONCAT(a.au\_fname, ' ', a.au\_lname) 'Full Name', t.title

from authors a

join titleauthor ta on a.au\_id = ta.au\_id

join titles t on ta.title\_id = t.title\_id

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

select p.pub\_name, AVG(t.advance) 'Average Advance' from publishers p

join titles t on p.pub\_id = t.pub\_id

group by p.pub\_name

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

select

p.pub\_name,

CONCAT(a.au\_fname, ' ', a.au\_lname) 'Author Name',

t.title,

SUM(s.qty \* t.price) 'Sale Amount'

from publishers p

join titles t on p.pub\_id = t.pub\_id

join titleauthor ta on t.title\_id = ta.title\_id

join authors a on ta.au\_id = a.au\_id

join sales s on t.title\_id = s.title\_id

group by p.pub\_name, CONCAT(a.au\_fname, ' ', a.au\_lname), t.title

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

select price from titles where title like '%s'

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

select 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 from employee e

join publishers p on e.pub\_id = p.pub\_id

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

select p.pub\_name, COUNT(e.emp\_id) 'Number of Employees' from publishers p

join employee e on p.pub\_id = e.pub\_id

group by p.pub\_name

having COUNT(e.emp\_id) > 2

--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 ta.au\_id = a.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'

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

select \* from employee where pub\_id =

(

select pub\_id from publishers where pub\_name = 'Algodata Infosystems'

)

--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 DB19012022QNS

USE DB19012022QNS

create table Employee

(

ID int primary key identity(100, 1),

Name varchar(50),

Age int,

Phone varchar(30) not null,

Gender varchar(1)

)

create table Salary

(

ID int primary key identity(1, 100),

Basic float,

HRA float,

DA float,

Deductions float

)

create table EmployeeSalary

(

Transaction\_Number int primary key identity(1, 1),

Employee\_ID int references Employee(ID),

Salary\_ID int references Salary(ID),

Date datetime

)

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

alter table Employee

add Email varchar(100)

--Insert few records in all the tables

insert into Employee(Name, Age, Phone, Gender, Email) values('John', 20, '96587421', 'M', 'email1@email.com')

insert into Employee(Name, Age, Phone, Gender, Email) values('Mary', 34, '94652846', 'F', 'email2@email.com')

insert into Employee(Name, Age, Phone, Gender, Email) values('Jane', 26, '+6586545652', 'F', 'email3@email.com')

insert into Employee(Name, Age, Phone, Gender, Email) values('Joey', 36, '91235468', 'M', 'email4@email.com')

insert into Salary(Basic,HRA,DA,Deductions) values(10000, 5000, 3000, 1500)

insert into Salary(Basic,HRA,DA,Deductions) values(20000, 10000, 6000, 3000)

insert into Salary(Basic,HRA,DA,Deductions) values(40000, 20000, 12000, 6000)

insert into Salary(Basic,HRA,DA,Deductions) values(80000, 40000, 24000, 12000)

select \* from Employee

select \* from Salary

insert into EmployeeSalary(Employee\_ID, Salary\_ID,Date) values(100, 1, '2010-12-11')

insert into EmployeeSalary(Employee\_ID, Salary\_ID,Date) values(101, 101, '2010-8-10')

insert into EmployeeSalary(Employee\_ID, Salary\_ID,Date) values(102, 201, '2007-11-09')

insert into EmployeeSalary(Employee\_ID, Salary\_ID,Date) values(102, 101, '2005-02-25')

insert into EmployeeSalary(Employee\_ID, Salary\_ID,Date) values(103, 301, '2005-09-08')

insert into EmployeeSalary(Employee\_ID, Salary\_ID,Date) values(103, 201, '2004-04-04')

select \* from EmployeeSalary

--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 proc\_PrintTotalSalary(@Employee\_ID int, @Date datetime)

as

begin

declare

@Basic float,

@HRO float,

@DA float,

@Deductions float,

@SalaryID int,

@Total float,

@EmployeeName varchar(50)

set @SalaryID =

(

select Salary\_ID from EmployeeSalary where Employee\_ID = @Employee\_ID and Date = @Date

)

set @Basic = (select Basic from Salary where ID = @SalaryID)

set @HRO = (select HRA from Salary where ID = @SalaryID)

set @DA = (select DA from Salary where ID = @SalaryID)

set @Deductions = (select Deductions from Salary where ID = @SalaryID)

set @EmployeeName = (select Name from Employee where ID = @Employee\_ID)

set @Total = @Basic + @HRO + @DA - @Deductions

Print'Total Salary of ' + @EmployeeName + ' is $' + CAST(@Total as varchar(50)) + ' as of ' + CAST(@Date as varchar(50))

end

select \* from EmployeeSalary

exec proc\_PrintTotalSalary 103, '2004-04-04'

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

create proc proc\_PrintAverageSalary(@Employee\_ID int)

as

begin

declare

@Average float,

@EmployeeName varchar(50)

set @Average =

(

select AVG(s.Basic + s.DA + s.HRA - s.Deductions) from EmployeeSalary es

join Salary s on es.Salary\_ID = s.ID

where es.Employee\_ID = @Employee\_ID

)

set @EmployeeName = (select Name from Employee where ID = @Employee\_ID)

Print'Average Salary of ' + @EmployeeName + ' is $' + CAST(@Average as varchar(50))

end

exec proc\_PrintAverageSalary 103

--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 proc\_CalculateTaxPayable(@Employee\_ID int)

as

begin

declare

@Total float,

@EmployeeName varchar(50),

@Payable float

set @EmployeeName = (select Name from Employee where ID = @Employee\_ID)

set @Total =

(

select SUM(s.Basic + s.DA + s.HRA - s.Deductions) from EmployeeSalary es

join Salary s on es.Salary\_ID = s.ID

where es.Employee\_ID = @Employee\_ID

)

if(@Total < 100000)

set @Payable = 0

if(@Total >= 100000 and @Total < 200000)

set @Payable = @Total \* 0.05

if(@Total >= 200000 and @Total < 350000)

set @Payable = @Total \* 0.06

if(@Total >= 350000 )

set @Payable = @Total \* 0.075

Print'Tax Payable of ' + @EmployeeName + ' is $' + CAST(@Payable as varchar(50))

end

exec proc\_CalculateTaxPayable 100

exec proc\_CalculateTaxPayable 101

exec proc\_CalculateTaxPayable 102

exec proc\_CalculateTaxPayable 103

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

create function fn\_SumOfValues(@basic float, @da float,@hra float)

returns float

as

begin

return @basic + @da + @hra

end

select \*, dbo.fn\_SumOfValues(s.Basic, s.DA, s.HRA) 'Sum of Basic, DA and HRA' from Salary s

--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

@EmpID int,

@EmpName varchar(50),

@Age int,

@Phone varchar(30),

@Gender varchar(1),

@Email varchar(100)

declare cur\_emp cursor for select \* from Employee

open cur\_emp

fetch next from cur\_emp into @EmpID, @EmpName, @Age, @Phone, @Gender, @Email

while(@@FETCH\_STATUs = 0)

begin

print'Employee Number ' + CAST(@EmpID as varchar(20))

print'Name ' + @EmpName

print'Age ' + CAST(@Age as varchar(20))

print'Phone ' + @Phone

print'Gender ' + @Gender

print'Email ' + @Email

print'-----------------------------------------------------'

declare

@ESDate datetime,

@TotalSalary float

declare cur\_sal cursor for select es.Date, dbo.fn\_SumOfValues(s.Basic, s.DA, s.HRA) - s.Deductions 'Total Salary' from EmployeeSalary es join Salary s on es.Salary\_ID = s.ID where es.Employee\_ID = @EmpID

open cur\_sal

fetch next from cur\_sal into @ESDate,@TotalSalary

while(@@FETCH\_STATUS = 0)

begin

print' Date ' + CAST(@ESDate as varchar(100))

print' Total Salary $' + CAST(@TotalSalary as varchar(100))

print'-----------------------------------------------------'

fetch next from cur\_sal into @ESDate,@TotalSalary

end

close cur\_sal

DEALLOCATE cur\_sal

fetch next from cur\_emp into @EmpID, @EmpName, @Age, @Phone, @Gender, @Email

end

close cur\_emp

DEALLOCATE cur\_emp

--16) <https://www.hackerrank.com/challenges/maximum-element/problem>

--17) <https://www.geeksforgeeks.org/find-if-there-is-a-subarray-with-0-sum/>