-- day3 19/01/2022 exercises

use pubs

-- 1) Select the author firstname and lastname

select au\_fname, au\_lname

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(au\_fname,' ',au\_lname) as 'Author', count(title\_id) as 'No. of Published Titles'

from titleauthor ta join authors a

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

group by concat(au\_fname,' ',au\_lname)

-- 4) print the author name and title name

select concat(au\_fname,' ',au\_lname) as 'Author', title

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

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

order by 'Author'

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

select pub\_name, round(avg(advance),2) 'Average Advance'

from titles t join publishers p

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

group by pub\_name

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

select pub\_name, concat(au\_fname,' ',au\_lname) as 'Author', title, qty\*price as 'Sale Amount'

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

select \* from titles

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

select title, 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(fname,' ',lname) as 'Employee Name', pub\_name

from employee e join publishers p

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

order by pub\_name, 'Employee Name'

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

select pub\_name, count(e.pub\_id) 'No. of Employees'

from employee e join publishers p

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

group by pub\_name

having count(e.pub\_id) > 2

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

select concat(au\_fname,' ',au\_lname) as 'Author'

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

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

where pub\_id = (select pub\_id

                from publishers

                where pub\_name = 'Algodata Infosystems')

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

select concat(fname,' ',lname) as 'Employee Name'

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 Employees Id

Salary\_id reference Salary Id,

Date)

\*/

create database myDB

use myDB

create table Employee (

    emp\_id      int identity(100,1) primary key,

    name        varchar(50),

    age         int,

    phone       varchar(10) not null,

    gender      varchar(1))

create table Salary (

    sal\_id      int identity(100,100) primary key,

    basic       float,

    hra         float,

    da          float,

    deductions  float)

create table EmpSalary (

    trans\_num   int,

    emp\_id      int constraint fk\_empSal foreign key references Employee(emp\_id),

    sal\_id      int references Salary(sal\_id),

    trans\_date  datetime,

    primary key (emp\_id, sal\_id, trans\_date))

-- 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 values('John Doe', 24, '98764321', 'M', 'johndoe@mail.com')

insert into Employee values('Jonathan Tan', 22, '96784123', 'M', 'jontan@mail.com')

insert into Salary values(4000, 800, 400, 200)

insert into Salary values(6000, 1000, 600, 200)

insert into Salary values(8000, 1000, 600, 300)

insert into EmpSalary values(1, 100, 100, '25-DEC-2020')

insert into EmpSalary values(2, 101, 300, '25-DEC-2020')

insert into EmpSalary values(3, 100, 200, '25-JAN-2021')

insert into EmpSalary values(4, 101, 300, '25-JAN-2021')

-- 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\_TotalSalary (@emp\_id int,

                              @date datetime)

as

begin

    declare

        @basic float,

        @da float,

        @hra float,

        @deductions float,

        @total float

    select @basic = basic,

           @da = da,

           @hra = hra,

           @deductions = deductions

    from EmpSalary e join Salary s

    on e.sal\_id = s.sal\_id

    where emp\_id = @emp\_id and trans\_date = @date

    set @total = @basic + @da + @hra - @deductions

    print 'For Employee ID: ' + cast(@emp\_id as varchar)

    print '    Date: ' + cast(@date as varchar)

    print '-'

    print 'Basic pay: $' + cast(@basic as varchar)

    print 'Dearness allowance: $' + cast(@da as varchar)

    print 'House rent allowance: $' + cast(@hra as varchar)

    print 'Deductions: $' + cast(@deductions as varchar)

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

    print 'Total salary: $' + cast(@total as varchar)

end

exec proc\_TotalSalary 100,'25-DEC-2020'

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

create proc proc\_AvgSalary (@emp\_id int)

as

begin

    declare

        @basic float,

        @da float,

        @hra float,

        @deductions float,

        @avg float,

        @count int

    select @basic = sum(basic),

           @da = sum(da),

           @hra = sum(hra),

           @deductions = sum(deductions),

           @count = count(emp\_id)

    from EmpSalary e join Salary s

    on e.sal\_id = s.sal\_id

    where emp\_id = @emp\_id

    set @avg = (@basic + @da + @hra - @deductions) / @count

    print 'Average salary: $' + cast(@avg as varchar)

end

exec proc\_AvgSalary 100

-- 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\_CalculateTax (@emp\_id int)

as

begin

    declare

        @basic float,

        @da float,

        @hra float,

        @deductions float,

        @total float,

        @tax float

    select @basic = sum(basic),

           @da = sum(da),

           @hra = sum(hra),

           @deductions = sum(deductions)

    from EmpSalary e join Salary s

    on e.sal\_id = s.sal\_id

    where emp\_id = @emp\_id

    set @total = @basic + @da + @hra - @deductions

    if @total < 100000

        set @tax = @total \* 0

    if 100000 < @total and @total < 200000

        set @tax = @total \* 0.05

    if 200000 < @total and @total < 350000

        set @tax = @total \* 0.06

    if 350000 < @total

        set @tax = @total \* 0.075

    print 'Total tax payable: $' + cast(@tax as varchar)

end

exec proc\_CalculateTax 100

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

create function fn\_SumOfSalary (@basic float,

                                @hra float,

                                @da float)

returns float

as

begin

    declare @sum float

    set @sum = @basic + @hra + @da

    return @sum

end

select dbo.fn\_SumOfSalary(1,2,3)

-- 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,

        @name       varchar(50),

        @age        int,

        @phone      varchar(10),

        @gender     varchar(1),

        @email      varchar(100)

declare cur\_emp cursor for select \* from Employee

open cur\_emp

fetch next from cur\_emp into @empid, @name, @age, @phone, @gender, @email

    while (@@FETCH\_STATUS=0)

        begin

            print ''

            print 'Employee ID : ' + cast(@empid as varchar)

            print 'Name        : ' + @name

            print 'Age         : ' + cast(@age as varchar)

            print 'Phone       : ' + @phone

            print 'Gender      : ' + @gender

            print 'Email       : ' + @email

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

                declare @transnum int,

                        @transdate datetime,

                        @basic float,

                        @hra float,

                        @da float,

                        @deductions float,

                        @total float

                declare cur\_trans cursor for select trans\_num, trans\_date, basic, hra, da, deductions

                                             from EmpSalary es join Salary s

                                             on es.sal\_id = s.sal\_id

                                             where emp\_id = @empid

                open cur\_trans

                fetch next from cur\_trans into @transnum, @transdate, @basic, @hra, @da, @deductions

                while (@@FETCH\_STATUS=0)

                    begin

                        set @total = @basic + @hra + @da - @deductions

                        print '    Transaction ID : ' + cast(@transnum as varchar)

                        print '    Date           : ' + cast(@transdate as varchar)

                        print '    Salary         : $' + cast(@total as varchar)

                        print '        --'

                        print '        Basic                : $' + cast(@basic as varchar)

                        print '        House rent allowance : $' + cast(@hra as varchar)

                        print '        Dearness allowance   : $' + cast(@da as varchar)

                        print '        Deductions           : $' + cast(@deductions as varchar)

                        print ''

                        fetch next from cur\_trans into @transnum, @transdate, @basic, @hra, @da, @deductions

                    end

                close cur\_trans

                deallocate cur\_trans

            fetch next from cur\_emp into @empid, @name, @age, @phone, @gender, @email

        end

    close cur\_emp

    deallocate cur\_emp

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

-- in python 3

def getMax(operations):

    for i in operations:

        op = i.split()

        q = op[0]

        if q == '1':

            stack.append(op[1])

        if q == '2':

            stack.pop(0)

        if q == '3':

            print(max(stack))

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

-- using python 3

def ZeroSumSubArray (arr):

    for i in range(len(arr)+1):

        for j in range(i):

            if sum(arr[j:i]) == 0:

                return True

    return False