use pubs

--1) Select the author firtname and last name

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 count(title\_id) 'Number of Titles Published', au\_id from titleauthor group by au\_id

--4) print the author name and title name

select t.title, concat(a.au\_fname,' ',a.au\_lname)'Author Name'

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

join authors a on a.au\_id=ta.au\_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, (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 a.au\_id=ta.au\_id

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

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

select price, title 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 p.pub\_name ,concat(e.fname,' ',e.lname)'Employee Name'

from publishers p

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

--10) print the publisher name and number of employees working 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 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'

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

select concat(e.fname,' ',e.lname)'Employee Name'

from publishers p

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

where p.pub\_name = 'Algodata Infosystems'

-----------------------------------------------------------------------

use master

create database day3Assignment

--13)Create the following tables

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

create table tblEmp

(empID int identity(100,1) primary key,

empName varchar(20),

age int,

phone int not null,

gender varchar(5))

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

create table tblSalary

(salaryID int identity(1,100) primary key,

basic int,

hra int,

da int ,

deduction int)

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

---- Salary\_id reference Salary Id,Date)

create table tblEmployeeSalary

(transaction\_num int primary key,

empID int references tblEmp(empID) unique,

salaryID int references tblSalary(salaryID) unique,

date Date unique)

-----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

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

alter table tblEmp

add email varchar(100)

---Insert few records in all the tables

insert into tblEmp values(103,'Max',25,90875665,'Male','abc@yahoo.com')

insert into tblEmp values(153,'Ruby',35,90872332,'Female','abc@gmail.com')

insert into tblEmp values(132,'Ken',45,90871991,'Male','abc@hotmail.com')

insert into tblSalary values(300,3200,5000,340,100,200)

insert into tblSalary values(500,2200,7000,900,300,100)

insert into tblSalary values(600,6200,9000,600,900,300)

insert into tblEmployeeSalary values(1,103,300,'2017-06-15')

insert into tblEmployeeSalary values(2,153,500,'2017-06-15')

insert into tblEmployeeSalary values(3,153,600,'2017-07-15')

---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(@empID int, @date Date)

as

begin

Declare

@total int,

@basic int,

@hra int,

@da int,

@d int

set @basic = (select basic from tblEmployeeSalary es

join tblEmp e on e.empID = es.empID

join tblSalary s on s.salaryID = es.salaryID

where empID = @empID)

set @hra = (select hra from tblEmployeeSalary es

join tblEmp e on e.empID = es.empID

join tblSalary s on s.salaryID = es.salaryID

where empID = @empID)

set @da = (select da from tblEmployeeSalary es

join tblEmp e on e.empID = es.empID

join tblSalary s on s.salaryID = es.salaryID

where empID = @empID)

set @d = (select deduction from tblEmployeeSalary es

join tblEmp e on e.empID = es.empID

join tblSalary s on s.salaryID = es.salaryID

where empID = @empID)

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

print 'Total Salary: ' + cast(@total as varchar(20))

end

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

create proc proc\_avgSalary(@empID int)

as

begin

Declare

@salary int,

@basic int,

@hra int,

@da int,

@d int,

@avg int,

@countTotal int,

@c int

set @countTotal = (select count(empID) from tblEmployeeSalary group by empID)

set @c = 0

while(@c<(@countTotal+1))

begin

set @basic = (select sum(s.basic) from tblEmployeeSalary es

join tblEmp e on e.empID = es.empID

join tblSalary s on s.salaryID = es.salaryID

group by e.empID)

set @hra = (select sum(s.hra) from tblEmployeeSalary es

join tblEmp e on e.empID = es.empID

join tblSalary s on s.salaryID = es.salaryID

group by e.empID)

set @da = (select sum(s.da) from tblEmployeeSalary es

join tblEmp e on e.empID = es.empID

join tblSalary s on s.salaryID = es.salaryID

group by e.empID)

set @d = (select sum(s.d) from tblEmployeeSalary es

join tblEmp e on e.empID = es.empID

join tblSalary s on s.salaryID = es.salaryID

group by e.empID)

set @salary = @basic+@hra+@da-@d

set @c = @c +1

end

set @avg = @salary/@countTotal

print 'Average Salary: ' + cast(@avg 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 proc\_calcTax(@empID int)

as

begin

Declare

@salary int,

@basic int,

@hra int,

@da int,

@d int,

@avg int,

@tax float,

@countTotal int,

@c int

set @countTotal = (select count(empID) from tblEmployeeSalary group by empID)

set @c = 0

while(@c<(@countTotal+1))

begin

set @basic = (select sum(s.basic) from tblEmployeeSalary es

join tblEmp e on e.empID = es.empID

join tblSalary s on s.salaryID = es.salaryID

group by e.empID)

set @hra = (select sum(s.hra) from tblEmployeeSalary es

join tblEmp e on e.empID = es.empID

join tblSalary s on s.salaryID = es.salaryID

group by e.empID)

set @da = (select sum(s.da) from tblEmployeeSalary es

join tblEmp e on e.empID = es.empID

join tblSalary s on s.salaryID = es.salaryID

group by e.empID)

set @d = (select sum(s.d) from tblEmployeeSalary es

join tblEmp e on e.empID = es.empID

join tblSalary s on s.salaryID = es.salaryID

group by e.empID)

set @salary = @basic+@hra+@da-@d

set @c = @c +1

end

if (@salary <= 100000)

set @tax = @salary

else if(@salary <= 200000)

set @tax = @salary\*(5/100)

else if(@salary <= 350000)

set @tax = @salary\*(6/100)

else

set @tax = @salary\*(7.5/100)

print 'Tax Payable: ' + cast(@tax as varchar(20))

end

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

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

returns float

as

begin

Declare

@total float,

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

return @total

end

--15) Create a cursor that will pick up every employee and print his details (difficult,cannot make it, need more time pls~)

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

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

use pubs

declare

DECLARE cur\_emp CURSOR FOR select \* from employee

OPEN cur\_emp

FETCH NEXT FROM cur\_emp INTO

while(@@FETCH\_STATUS =0)

begin

declare @amount float, @status varchar(20)

FETCH NEXT FROM cur\_emp INTO

end

CLOSE cur\_emp

DEALLOCATE cur\_emp