--1) Select the author firtname and last name

select au\_fname 'first name', au\_lname 'last 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

--3) Print the number of titlespublished by every author

select count(title\_id) 'num of titles published', au\_id 'authord id'

from titleauthor

group by au\_id

--4) print the author name and title name

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

from authors a join [titleauthor] ta

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

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

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

select pub\_name, avg(advance) 'Avg advance' from publishers p

left join titles t

on p.pub\_id = t.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\_lname,' ',au\_fname)'author name', title, qty\*price 'sales amount'

from titles t

left join publishers p

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

left join titleauthor ta

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

join authors a

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

left join sales s

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

--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(fname, ' ', lname)'Employee name', pub\_name 'publisher name'

from employee e join [publishers] p

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

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

select pub\_name 'publisher name', count(emp\_id)'number of employees'

from employee e

join publishers p

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

group by pub\_name

having count(emp\_id)>2

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

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

from authors a join titleauthor ta

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

left join titles t

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

where t.pub\_id = (select pub\_id from publishers p where pub\_name =

'Algodata Infosystems')

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

select concat(lname,' ',fname)'Employee name' , pub\_name

from employee e

join publishers p

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

where p.pub\_name = 'Algodata Infosystems'

--14)Create the following tables

--Employee(id-identity starts in 100 inc by 1,

--Name,age, phone cannot be null, gender)

create table Employee(

empId int identity(100,1) primary key not null,

empName varchar(50),

empAge int,

empPhone varchar(15) not null,

empGender varchar(10)

)

--Salary(id-identity starts at 1 increments by 100,

--Basic,HRA,DA,deductions)

create table Salary

(

salId int identity(1,100) primary key not null,

salBasic float,

hra float,

da float,

deductions float

)

--EmployeeSalary(transaction\_number int,

--employee\_id-reference Employee's Id

--Salary\_id reference Salary Id,

--Date)

drop table employeeSalary

create table employeeSalary

(

transNo int primary key not null,

empId int references Employee(empId),

salId int references Salary(salId),

tDate datetime

)

--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 Employee

add email varchar(100)

--Insert few records in all the tables

insert into Employee values ('Phoebe toh','26','98765432','female','phoebe123@gmail.com')

insert into Employee values ('Brandon koh','23','91234567','male','brandonk@gmail.com')

insert into Employee values ('Tydus','18','91122334','male','tyduskzy@gmail.com')

insert into Salary values (2000,200,400,100)

insert into Salary values (3000,300,500,200)

insert into Salary values (4000,400,600,300)

insert into employeeSalary values (1,100,1,'2022-01-19')

insert into employeeSalary values (2,101,101,'2022-01-19')

insert into employeeSalary values (3, 102,201,'2022-01-19')

--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\_totalSal(@empId int, @date datetime)

as

begin

declare

@total float

set @total = (select sum(s.salBasic + s.hra + s.da - s.deductions)

totalSal from employee e join employeeSalary es

on e.empId = es.empId join Salary s

on s.salId = es.salId

where e.empId = @empId)

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

end

exec proc\_totalSal 102, '2022-01-19'

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

create proc AvgSal(@empId int)

as

begin

declare

@avgSal float,

@totalSal float,

@timesSalary int

set @totalSal = (select sum(s.salBasic + s.hra + s.da - s.deductions)

totalSal from employee e join employeeSalary es

on e.empId = es.empId join Salary s

on s.salId = es.salId

where e.empId = @empId)

set @timesSalary = (select count(es.empId) from employee e join employeeSalary es

on e.empId = es.empId join salary s

on s.salId = es.salId

where e.empId = @empId

group by es.empId)

set @avgSal = @totalSal / @timesSalary

print 'total salary :' + cast(@totalSal as varchar(20))

print 'total times of salary: ' + cast(@timesSalary as varchar(20))

print 'average salary: ' + cast(@avgSal as varchar(20))

end

exec AvgSal 102

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

--total - 100000 - 0%

--100000 > total < 200000 - 5%

--200000 > total < 350000 - 6%

--total > 350000 - 7.5%

create proc proc\_calculateTax(@empId int)

as

begin

declare

@count int

set @count = (select count(transNo) from employeeSalary where empId = @empId)

if(@count > 0)

begin

declare

@totalSalary float,

@totalBasic float,

@totalDeduction float,

@totalhra float,

@totalda float

set @totalBasic = (select sum(s.salBasic) from Salary s

inner join employeeSalary es

on s.salID = es.salID

and empID = @empID)

set @totalDeduction = (select sum(deductions) from employeeSalary es

join Salary s

on s.salID = es.salID

where empID = @empID)

set @totalhra = (select sum(hra) from employeeSalary es

join Salary s

on s.salID = es.salID

where empID = @empID)

set @totalda = (select sum(da) from employeeSalary es

join Salary s

on s.salID = es.salID

where empID = @empID)

set @totalSalary = (@totalBasic - @totalDeduction + @totalhra + @totalda)

print cast(@totalSalary as varchar(10))

if(@totalSalary <= 100000)

print 'Payable tax is : 0'

else if(@totalSalary > 100000 and @totalSalary <= 200000)

print 'Payable tax is : ' + cast(@totalSalary \* 0.05 as varchar(10))

else if(@totalSalary > 200000 and @totalSalary <= 350000 )

print 'Payable tax is : ' + cast(@totalSalary \* 0.06 as varchar(10))

else if(@totalSalary > 350000 )

print 'Payable tax is : ' + cast(@totalSalary \* 0.075 as varchar(10))

end

else

print 'Payable tax is : 0'

end

--15) Create a function that will take the basic,HRA and da returns the sum of the three

create function sumOfThree(@empId int, @date datetime)

returns float

as

begin

declare

@sum float,

@basic float,

@hra float,

@da float

set @basic = (select salBasic from Salary s

join employeeSalary es

on es.salID = s.salID

where es.tDate = @date and empID = @empID)

set @hra = (select hra from Salary s

join employeeSalary es

on es.salID = s.salID

where es.tDate = @date and empID = @empID)

set @da = (select da from Salary s

join employeeSalary es

on es.salID = s.salID

where es.tDate = @date and empID = @empID)

set @sum = @basic + @hra + @da

return @sum

end

--16) 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)