--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,

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,

empId int constraint fk\_empSal references Employee(empId),

salId int constraint fl\_emplSal1 references Salary(salId),

tDate datetime 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 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,'2022-01-20 00:00:00.000')

insert into employeeSalary values (2,101,101,'2022-01-21 00:00:00.000')

insert into employeeSalary values (3, 102,201,'2022-01-19 00:00:00.000')

--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\_CalTotalSal(@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 and es.tDate = @date)

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

end

exec proc\_CalTotalSal 102, '2022-01-19 00:00:00.000'

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

create proc AvgSalCalculate(@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 AvgSalCalculate 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 calculateTax(@empId int, @date datetime)

as

begin

declare

@total float,

@taxRate float,

@tax float

set @total = ( select s.salBasic + s.hra + s.da - s.deductions from employee e

join employeeSalary es

on e.empId = es.empId join Salary s

on s.salId = es.salId

where e.empId = @empId and es.tDate = @date)

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

if(@total <100000)

set @taxRate = 0

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

set @taxRate = 0.05

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

set @taxRate = 0.06

else

set @taxRate = 0.075

print 'Tax Percentage: '+ cast(@taxRate as varchar(20)) +'%'

print 'Total tax payable: '+ cast(@tax as varchar(20))

end

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

create function fnSumofSalary(@SalBasic float, @hra float, @da float)

returns float

as

begin

declare

@total float

set @total = @SalBasic + @hra + @da

return @total

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)

declare

@empId int,

@empName varchar(50),

@empAge int,

@empPhone varchar(15),

@empGender varchar(10)

declare cur\_emp cursor for select \* from employee

open cur\_emp

fetch next from cur\_emp into @empId, @empName, @empAge, @empPhone,

@empGender

while(@@FETCH\_STATUS = 0)

begin

print 'Employee ID : '+ cast(@empId as varchar(20))

print 'Employee Name : '+ @empName

print 'Employee Age : '+ cast(@empAge as varchar(20))

print 'Phone : '+ @empPhone

print 'Gender : '+ @empGender

declare

@transNo int,

declare @transNo int, @salId int, @tDate datetime

declare cur\_transDetails cursor for select transNo, salId, tDate

from employeeSalary

where empId = @empId

open cur\_transDetails

fetch next from cur\_transDetails into @transNo, @salId, @tDate

while(@@FETCH\_STATUS = 0)

begin

print 'Transaction No : '+ CAST(@transNo as varchar(10))

print 'Salary ID : '+ CAST(@salId as varchar(10))

print 'Transaction Date : '+ CAST(@tDate as varchar(20))

fetch next from cur\_transDetails into @transNo , @salId, @tDate

end

close cur\_transDetails

deallocate cur\_transDetails

fetch next from cur\_emp into @empId, @empName, @empAge, @empPhone, @empGender

end

close cur\_emp

deallocate cur\_emp