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

--1) Select the author firtname and last name

select concat(au\_fname, ' ', au\_lname) 'Author 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 titles published by every author

select au\_id, count(title\_id)'Number of titles'

from titleauthor

group by au\_id

--4) print the author name and title name

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

from titles t join titleauthor ta

on t.title\_id = ta.title\_id join authors au

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

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

select \* from titles

select p.pub\_name, avg(t.advance) 'Average advance'

from titles t join publishers p

on t.pub\_id = p.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(au.au\_fname, ' ', au.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 sales s

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

on ta.title\_id = t.title\_id join authors au

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

--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(e.fname, ' ',e.lname) 'Employee Name', p.pub\_name

from publishers p join employee e

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 p.pub\_name, count(e.emp\_id) 'Number of employee'

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(au.au\_fname, ' ', au.au\_lname) 'Author Name', p.pub\_name

from publishers p join titles t

on p.pub\_id = t.pub\_id join sales s

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

on ta.title\_id = t.title\_id join authors au

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

where p.pub\_name = 'Algodata Infosystems'

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

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

from publishers p join employee e

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(

emp\_id int not null identity(100, 1) primary key,

emp\_name varchar(50),

emp\_age int,

phone char(11) not null,

gender char(1)

)

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

--Basic,Dearness allowance(da), House Rent Allowance(hra),deductions)

create table Salary(

sal\_id int not null identity(1, 100) primary key,

sal\_basic float,

hra float,

da float,

deductions float

)

--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 table EmployeeSalary(

trans\_no int not null primary key,

emp\_id int references Employee(emp\_id),

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

date datetime 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 (emp\_name, emp\_age, phone, gender, email) values ('Johnson', 18, '1231231123', 'M', 'johnson@email.com')

insert into employee (emp\_name, emp\_age, phone, gender, email) values ('Albert Liew', 22, '54564567', 'F', 'liew@email.com')

insert into employee (emp\_name, emp\_age, phone, gender, email) values ('Michael Lim', 65, '678687897', 'M', 'lim@email.com')

insert into salary (sal\_basic, hra, da, deductions) values (2000, 200, 200, 100)

insert into salary (sal\_basic, hra, da, deductions) values (3500, 300, 200, 250)

insert into salary (sal\_basic, hra, da, deductions) values (4500, 300, 200, 300)

insert into salary (sal\_basic, hra, da, deductions) values (224500, 300, 200, 300)

insert into EmployeeSalary (trans\_no, emp\_id, sal\_id, date) values (1, 100, 1, '2021-09-14 00:00:00.000')

insert into EmployeeSalary (trans\_no, emp\_id, sal\_id, date) values (2, 101, 101, '2021-03-12 00:00:00.000')

insert into EmployeeSalary (trans\_no, emp\_id, sal\_id, date) values (3, 102, 201, '2021-11-29 00:00:00.000')

insert into EmployeeSalary (trans\_no, emp\_id, sal\_id, date) values (4, 101, 201, '2021-04-29 00:00:00.000')

insert into EmployeeSalary (trans\_no, emp\_id, sal\_id, date) values (5, 101, 301, '2021-04-27 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\_totalSalaryCalculator(@emp\_id int, @date datetime)

as

begin

declare

@total float

set @total = (select sum(s.sal\_basic + s.hra + s.da - s.deductions) total\_salary from employee e join employeesalary es

on e.emp\_id = es.emp\_id join salary s

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

where e.emp\_id = @emp\_id and es.date = @date)

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

end

exec proc\_totalSalaryCalculator 101, '2021-03-12 00:00:00.000'

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

create proc procAverageSalaryCalculator(@emp\_id int)

as

begin

declare

@averageSalary float,

@totalSalary float,

@timesOfSalary int

set @totalSalary = (select sum(s.sal\_basic + s.hra + s.da - s.deductions) total\_salary from employee e join employeesalary es

on e.emp\_id = es.emp\_id join salary s

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

where e.emp\_id = @emp\_id)

set @timesOfSalary = (select count(es.emp\_id) from employee e join employeesalary es

on e.emp\_id = es.emp\_id join salary s

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

where e.emp\_id = @emp\_id

group by es.emp\_id)

set @averageSalary = @totalSalary / @timesOfSalary

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

print 'Total times of salary released: ' + cast(@timesOfSalary as varchar(20)) + ' times'

print 'Average Salary: ' + cast(@averageSalary as varchar(20))

end

exec procAverageSalaryCalculator 101

--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 taxCalculator(@emp\_id int, @date datetime)

as

begin

declare

@total float,

@taxRate float,

@tax float

set @total = (select s.sal\_basic + s.hra + s.da - s.deductions from employee e join employeesalary es

on e.emp\_id = es.emp\_id join salary s

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

where e.emp\_id = @emp\_id and es.date = @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

set @tax = @total \* @taxRate

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

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

end

exec taxCalculator 101, '2021-04-27 00:00:00.000'

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

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

returns float

as

begin

declare

@total float

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

return @total

end

select dbo.fnSumofSalary(200000, 2000, 200) 'Sum of Salary'

--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

@e\_emp\_id int,

@e\_emp\_name varchar(50),

@e\_emp\_age int,

@e\_phone char(11),

@e\_gender char(1),

@e\_email varchar(100)

declare cur\_employee cursor for select \* from employee

open cur\_employee

fetch next from cur\_employee into @e\_emp\_id, @e\_emp\_name, @e\_emp\_age, @e\_phone, @e\_gender, @e\_email

while(@@FETCH\_STATUS = 0)

begin

print '====================================================='

print 'Employee ID : '+ cast(@e\_emp\_id as varchar(20))

print 'Employee Name : '+ cast(@e\_emp\_name as varchar(20))

print 'Employee Age : '+ cast(@e\_emp\_age as varchar(20))

print 'Phone : '+ cast(@e\_phone as varchar(20))

print 'Gender : '+ cast(@e\_gender as varchar(20))

print 'Email : '+ cast(@e\_email as varchar(20))

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

declare

@es\_trans\_no int,

@es\_emp\_id int,

@es\_sal\_id int,

@es\_date datetime,

@s\_sal\_basic float,

@s\_hra float,

@s\_da float,

@s\_deduction float,

@sum\_salary float

declare cur\_transDetails cursor for select es.trans\_no, es.emp\_id, es.sal\_id, es.date, s.sal\_basic, s.hra, s.da, s.deductions

from employeesalary es join salary s

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

where emp\_id = @e\_emp\_id

open cur\_transDetails

fetch next from cur\_transDetails into @es\_trans\_no, @es\_emp\_id, @es\_sal\_id, @es\_date, @s\_sal\_basic, @s\_hra, @s\_da, @s\_deduction

while(@@FETCH\_STATUS = 0)

begin

print 'Transaction Number : ' +cast(@es\_trans\_no as varchar(20))

print 'Transaction Date : ' +cast(@es\_date as varchar(20))

print 'Employee ID : ' +cast(@es\_emp\_id as varchar(20))

print 'Salary ID : ' +cast(@es\_sal\_id as varchar(20))

print 'Basic Salary : ' +cast(@s\_sal\_basic as varchar(20))

print 'Hourse Rent Allowance(HRA) : ' +cast(@s\_hra as varchar(20))

print 'Dearness Allowance(DA) : ' +cast(@s\_da as varchar(20))

print 'Deductions : ' +cast(@s\_deduction as varchar(20))

print '====================================================='

fetch next from cur\_transDetails into @es\_trans\_no, @es\_emp\_id, @es\_sal\_id, @es\_date, @s\_sal\_basic, @s\_hra, @s\_da, @s\_deduction

end

close cur\_transDetails

deallocate cur\_transDetails

fetch next from cur\_employee into @e\_emp\_id, @e\_emp\_name, @e\_emp\_age, @e\_phone, @e\_gender, @e\_email

end

close cur\_employee

deallocate cur\_employee