--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 desc

--3) Print the number of titles published by every author

select count(title) 'No of titles published', a.au\_id 'Author id'

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

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

group by a.au\_id

--4) print the author name and title name

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

from titles t join titleauthor ta on ta.title\_id=t.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 'Publisher name', avg(advance) 'Average advance' from publishers p

join titles t 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 pub\_name 'Publisher name', CONCAT(au\_fname, ' ', au\_lname) 'Author name', title Title, s.qty\*t.price 'Sale amount'

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

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

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

join publishers p on p.pub\_id=t.pub\_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 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 employee working on it' from publishers p

join employee e on e.pub\_id=p.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\_fname, ' ', au\_lname) 'Author name' from authors a

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

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

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

where pub\_name='Algodata Infosystems'

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

select \* from employee e

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

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 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 tblEmployee

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

name varchar(30),

age int ,

phone varchar(8) not null,

gender varchar(1)

)

create table tblSalary

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

basic float,

HRA float,

DA float,

deductions float

)

create table tblEmployeeSalary

(transaction\_number int primary key,

employee\_id int references tblEmployee(id),

salary\_id int references tblSalary(id),

salDate datetime

)

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

alter table tblEmployee

add email varchar(100)

--Insert few records in all the tables

insert into tblEmployee(name, age, phone, gender) values ('LP',23, '87654321', 'F')

insert into tblEmployee(name, age, phone, gender) values ('timmy',23, '12345678', 'M')

insert into tblSalary(basic, HRA, DA, deductions) values (500,500,500,90)

insert into tblSalary(basic, HRA, DA, deductions) values (234,546777,24324,90)

insert into tblEmployeeSalary values (1, 100, 1,'1991-06-09 00:00:00.000')

insert into tblEmployeeSalary values (2, 101, 101,'1992-06-09 00:00:00.000')

--Create a procedure which will print the total salary of employee by taking the employee id and date

--total = Basic+HRA+DA-deductions

create proc proc\_TotalSalary(@empId int, @date datetime)

as

begin

declare @total float, @basic float, @hra float, @da float, @deduction float

set @basic = (select basic from tblSalary s join tblEmployeeSalary es on es.salary\_id=s.id where es.employee\_id=@empId and salDate=@date)

set @hra = (select hra from tblSalary s join tblEmployeeSalary es on es.salary\_id=s.id where es.employee\_id=@empId and salDate=@date)

set @da = (select da from tblSalary s join tblEmployeeSalary es on es.salary\_id=s.id where es.employee\_id=@empId and salDate=@date)

set @deduction = (select deductions from tblSalary s join tblEmployeeSalary es on es.salary\_id=s.id where es.employee\_id=@empId and salDate=@date)

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

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

end

exec proc\_TotalSalary 100, '1991-06-09 00:00:00.000'

--q1LP Create a procudure which will calculate the average salary of an employee taking his ID

-- each salary just find avg

create proc proc\_AvgSalaryOfEmployee(@empId float)

as

begin

declare @salary float

set @salary = (select avg(Basic+HRA+DA-deductions) 'Avg salary' from tblSalary where id=@empID)

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

end

exec proc\_AvgSalaryOfEmployee 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 proc\_CalculateTaxPayable(@empID int)

as

begin

declare @total float, @tax float

set @total = (select SUM(Basic+HRA+DA-deductions) from tblSalary where id=@empID)

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

if(@total < 100000)

set @tax = 0

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

set @tax = 5

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

set @tax = 6

if(@total >= 350000)

set @tax = 7.5

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 proc proc\_SumOfBasicHraDa(@basic float, @hra float, @da float)

as

begin

declare @total float

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

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

end

exec proc\_SumOfBasicHraDa 5,5,5

--15) Create a cursor that will pick up every employee and print his details --just simple query with cursor

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

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

DECLARE cur\_employee CURSOR FOR select e.\* , (basic+hra+ da-deductions) 'Total Salary', basic, hra, da, deductions

from tblEmployee e join tblEmployeeSalary es on es.employee\_id=e.id

join tblSalary s on s.id=es.salary\_id

--correct

select e.\* , (basic+hra+ da-deductions) 'Total Salary', basic, hra, da, deductions

from tblEmployee e join tblEmployeeSalary es on es.employee\_id=e.id

join tblSalary s on s.id=es.salary\_id

select \* from tblEmployee

select \* from tblSalary

select \* from tblEmployeeSalary

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

-- needs more fixing and testing.

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

//this solution is not completely correct, but it is my first attempt. more test cases need to be tested with

public class GetMax {

public static void main(String[] args) {

// TODO Auto-generated method stub

List<String> list = new ArrayList<String>();

list.add("1 97");

list.add("2");

list.add("1 20");

list.add("2");

list.add("1 26");

list.add("1 20");

list.add("2");

list.add("3");

list.add("1 91");

list.add("3");

getMax(list);

}

public static List<Integer> getMax(List<String> operations) {

// Write your code here

List<Integer> list = new ArrayList<Integer>();

String s = "";

for(int i = 0; i<operations.size(); i++){

if(operations.get(i).charAt(0) == '1'){

//add

//chec 3 char and more

if(operations.get(i).charAt(2) != ' '){

//has 2nd var

for(int j=2; j<operations.get(2).length(); j++){

s += Character.toString(operations.get(i).charAt(j));

System.out.println(Character.toString(operations.get(i).charAt(j)));

}

list.add(Integer.valueOf(s));

s = "";

}

}

if(operations.get(i).charAt(0) == '2') {

//delete

list.remove(list.get(list.size()-1));

}

if(operations.get(i).charAt(0) == '3') {

//print max

System.out.println("max: "+ Collections.max(list));

}

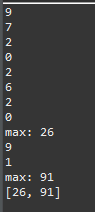
}

System.out.println(list);

return list;

}

}



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

-- could not think of a way, so i looked at the solution and understood it