Assignment 3

Set 1:

1. Joins are used whenever multiple rows are to be returned while subqueries are used to return a single datum or row set. The performance of join is usually better.
2. Cte is common table expressing and it’s used whenever recursion is required.
3. Table variables are similar to tables but the only difference is the lifespan of it. It can be created inside a stored procedure and functions but cannot be used outside of that scope.
4. Both delete records but delete removes one row at a time while truncate deletes entire rows from a table. Hence, truncate performs better and faster.
5. It’s a column including values created by the database in the database table. Truncate resets the identity value to original value while delete does not reset but increases
6. Both delete all rows but by using truncate, the changes cannot be restored.

Set 2:

Q1:

select distinct city from customers

where city in (select city from employees)

Q2a:

select distinct city from customers

where city not in (select city from employees)

Q2b:

select distinct c.city from customers c

left join Employees e

on c.city = e.city

Q3:

select c.customerid, c.companyname, c.contactname,

sum(od.quantity) as Quantity from customers c

left join orders o

on c.CustomerID=o.CustomerID

left join [Order Details] od

on o.orderid = od.OrderID

group by c.CustomerID, c.CompanyName, c.ContactName

Q4:

select c.city,

sum(od.quantity) as Quantity from customers c

left join orders o

on c.CustomerID=o.CustomerID

left join [Order Details] od

on o.orderid = od.OrderID

group by c.city

Q5a:

select un.city from customers un group by un.city

having count(un.city) > 2

union

select c.city from customers c group by c.city having count(c.city) =2

Q5b:

select distinct c.city from customers c

where c.city in (select u.city from Customers u group by

u.city having count(u.city) >= 2)

Q6:

select distinct c.City from Orders o inner join Customers c

on o.CustomerID = c.CustomerID

inner join [Order Details] r

on r.OrderID = o.OrderID

group by c.City, r.ProductID having count(r.ProductID) > 2

Q7:

select c.customerid, c.ContactName from Customers c where c.City not in

(select o.ShipCity from Orders o inner join Customers c on o.ShipCity = c.City)

Q8:

with cte\_order

as( select oc.ShipCity, oc.ProductID, oc.average, dense\_rank() over (partition by

oc.ProductID order by oc.number) rnk

from ( select top(5) od.ProductID, o.ShipCity, sum(quantity) number, avg(od.UnitPrice)

average from dbo.Orders o left join dbo.[Order Details] od on o.OrderID=od.OrderID

group by o.ShipCity, od.ProductID

order by number desc) oc)

select \* from cte\_order where rnk=1

Q9a:

select e.City from Employees e where e.City not in (

select c.City from Orders o inner join Customers c

on c.CustomerID = o.CustomerID)

Q9b:

select distinct e.City from Employees e

left join Customers c

on e.City = c.City where c.City is null

Q10:

select \* from

(select Top (1) e.City, count(o.OrderID) countOrder from Employees e inner join Orders o

on e.EmployeeID = o.EmployeeID group by e.City)dt1

inner join (

select Top (1) c.City, count(r.Quantity) countQuantity from [Order Details] r inner join

Orders d on r.OrderID = d.OrderID

inner join Customers c on c.CustomerID = d.CustomerID group by c.City)dt2

on dt1.City = dt2.City;

Q11:

Use delete to remove the duplicate rows which can be found using group by

Q12:

select deptname,empid,salary

from(select d.deptname, e.empid, e.salary, rank() over ( partition by e.deptid

order by e.salary desc ) as rnk

from dept d, employee e where d.deptid = e.deptid)

where rnk <= 3 order by deptname,rnk

Q13:

select countbydept.\*

from (select deptid, count(\*) as departCount from Employee

group by deptid order by departCount desc

limit 1) as maxcount

inner join

( select dept.id, dept.`name`, count(\*) as employeeCount

from Dept

inner join Employee on Employee.deptid = deptid group by deptid, deptname

) countbydept

Q14:

Select deptname,empid,salary

from (select d.deptname, e.empid, e.salary, rank() over (partition by e.deptid order by

e.salary desc) as rnk

from dept d, employee e

where d.deptid = e.deptid)

where rnk <= 3

order by deptname,rnk