Assignment Day3 –SQL: Comprehensive practice

# Answer following questions

1. In SQL Server, assuming you can find the result by using both joins and subqueries, which one would you prefer to use and why?
2. A Join query is preferred in such cases as Join queries are faster than using subqueries.
3. What is CTE and when to use it?
4. A common table expression is a temporary result set that can be reference within other statements. It can be helpful to simplify queries when queries are too long and complex
5. What are Table Variables? What is their scope and where are they created in SQL Server?
6. Table variables are variable which help store data temporarily, the table variables has a scope within the batch, procedure or function and are stored in tempdb table.
7. What is the difference between DELETE and TRUNCATE? Which one will have better performance and why?
8. Truncate is a DDL statement whereas Delete is a DML statement. Truncate removes all the data from the table while delete statement can delete records based on conditions provided. Truncate has better performance as it does not have to search through the table to match conditions.
9. What is Identity column? How does DELETE and TRUNCATE affect it?
10. Identity column is a column whose values increase automatically. Truncate will reset the count on identity column while delete will not.
11. What is difference between “delete from table\_name” and “truncate table table\_name”?
12. If the table has an identity column, delete command will not reset the value of the identity column, while the truncate command will reset the identity column values as well.

# Write queries for following scenarios

All scenarios are based on Database NORTHWND.

1. List all cities that have both Employees and Customers.
2. select distinct c.City from Customers c where c.City in (select e.City from Employees e);
3. List all cities that have Customers but no Employee.
   1. Use sub-query
4. select distinct c.City from Customers c where c.City not in (select e.City from Employees e);
   1. Do not use sub-query
5. select distinct c.City from Customers c left join Employees e on c.City = e.City where e.City is Null;
6. List all products and their total order quantities throughout all orders.
7. select od.ProductID,count(ProductID) from orders o inner join [Order Details] od on o.OrderID = od.OrderID group by od.ProductID;
8. List all Customer Cities and total products ordered by that city.
9. select c.City,sum(od.Quantity) as "Total Products" from Orders o inner join Customers c on c.CustomerID=o.CustomerID inner join [Order Details] od on o.OrderID=od.OrderID group by c.City;
10. List all Customer Cities that have at least two customers.
    1. Use union
11. select City from Customers group by City having count(CustomerID)>=2;
    1. Use sub-query and no union
12. select c1.City from (select c.City from Customers c) c1 group by c1.City having COUNT(c1.City) >=2;
13. List all Customer Cities that have ordered at least two different kinds of products.
14. select distinct dt.City from (select c.City,od.ProductID, dense\_rank() over(partition by c.City order by od.productID) rnk from orders o inner join [Order Details] od on o.OrderID = od.OrderID inner join Customers c on o.CustomerID = c.CustomerID) dt where rnk >=2;
15. List all Customers who have ordered products, but have the ‘ship city’ on the order different from their own customer cities.
16. select distinct c.CompanyName from Orders o inner join Customers c on o.CustomerID = c.CustomerID where o.ShipCity<>c.City;
17. List 5 most popular products, their average price, and the customer city that ordered most quantity of it.
18. select dt.ProductID,dt.avgPrice,dt.City from (select DENSE\_RANK() over(partition by t1.ProductID order by t2.qtySold desc) as rnk ,t1.ProductID,t2.City,t1.avgPrice from (select top 5 od.ProductID,avg(od.UnitPrice) as "avgPrice" from Orders o inner join [Order Details] od on o.OrderID = od.OrderID group by od.ProductID order by count (od.ProductID) desc) t1 inner join ( select c.City,od.ProductID,sum(od.Quantity) as "qtySold" from Orders o inner join [Order Details] od on o.OrderID=od.OrderID inner join Customers c on c.CustomerID = o.CustomerID group by od.ProductID,c.City) t2 on t1.ProductID=t2.ProductID) dt where dt.rnk = 1;
19. List all cities that have never ordered something but we have employees there.
    1. Use sub-query
20. select e.City from Employees e where e.City not in ( select distinct c.City from Orders o inner join Customers c on o.CustomerID = c.CustomerID);
    1. Do not use sub-query
21. select e.City from orders o inner join customers c on o.CustomerID = c.CustomerID right join Employees e on c.City=e.City where c.City is null ;
22. List one city, if exists, that is the city from where the employee sold most orders (not the product quantity) is, and also the city of most total quantity of products ordered from. (tip: join sub-query)

**A.** select t1.city from (select e.City from Employees e where e.EmployeeID = (select top 1 EmployeeID from Orders group by EmployeeID order by count(OrderID) desc)) t1 inner join (select top 1 c.City from Orders o inner join [Order Details] od on o.OrderID=od.OrderID inner join Customers c on o.CustomerID = o.CustomerID group by c.City order by SUM(od.Quantity) desc) t2 on t1.City=t2.City;

11. How do you remove the duplicates record of a table?

**A.** duplicate records can be remove by using a combination of delete query and rank function with all the columns as parameters, any record with more than 1 rank can be deleted.

12. Sample table to be used for solutions below- Employee ( empid integer, mgrid integer, deptid integer, salary integer) Dept (deptid integer, deptname text)

Find employees who do not manage anybody.

**A.** select e.empid from employee e where e.empid not in (select mgrid from employee where mgrid is not null);

13. Find departments that have maximum number of employees. (solution should consider scenario having more than 1 departments that have maximum number of employees). Result should only have - deptname, count of employees sorted by deptname.

**A.** select d.deptname,t1.empCount from (select deptid,count(empid) as "empCount",DENSE\_RANK() over(order by count(empid) desc) as "rnk" from employee group by deptid) t1 inner join dept d on t1.deptid = d.deptid where rnk = 1;

14. Find top 3 employees (salary based) in every department. Result should have deptname, empid, salary sorted by deptname and then employee with high to low salary.

**A.** select d.deptname,t1.empid,t1.salary from (select deptid,empid,salary,DENSE\_RANK() over(partition by deptid order by salary desc) as "rnk" from employee ) t1 inner join dept d on t1.deptid=d.deptid where rnk <=3 order by d.deptname, salary desc;

GOOD LUCK.