# AAnswer 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?

I’d prefer to use joins because it executes faster.

1. What is CTE and when to use it?

CTE stands for common table expression. It is used to reference a temporary named result set.

1. What are Table Variables? What is their scope and where are they created in SQL Server?

Table Variables are local variables that store data temporarily. The scope ends at the end of the batch. Table Variables are created in the tempdb database.

1. What is the difference between DELETE and TRUNCATE? Which one will have better performance and why?

TRUNCATE removes all rows in a table by deallocating the pages that are used to store the data whereas DELETE removes rows one at a time. TRUNCATE has better performance because it doesn’t scan every record.

1. What is Identity column? How does DELETE and TRUNCATE affect it?

An Identity column is generated by the database. Its value increase automatically. DELETE doesn’t reset identity columns but TRUNCATE does.

1. What is difference between “delete from table\_name” and “truncate table table\_name”?

Delete is a DML command and truncate is a DDL command.

# Write queries for following scenarios

All scenarios are based on Database NORTHWND.

1. List all cities that have both Employees and Customers.

SELECT c.City FROM Customers c WHERE c.City IN (SELECT e.City FROM Employees e)

1. List all cities that have Customers but no Employee.
   1. Use sub-query

SELECT c.City FROM Customers c WHERE c.City NOT IN (SELECT e.City FROM Employees e)

* 1. Do not use sub-query

SELECT c.City FROM Customers c LEFT OUTER JOIN Employees e ON c.City = e.City WHERE e.City IS NULL;

1. List all products and their total order quantities throughout all orders.

SELECT ProductID, SUM(Quantity) FROM [Order Details] GROUP BY ProductID;

1. List all Customer Cities and total products ordered by that city.

SELECT c.City, COUNT(o.OrderID) FROM Customers c LEFT JOIN Orders o ON c.City = o.ShipCity WHERE o.ShipCity IS NOT NULL GROUP BY c.City;

1. List all Customer Cities that have at least two customers.
   1. Use union
   2. Use sub-query and no union

SELECT City FROM (SELECT City, COUNT(CustomerID) AS CNT FROM Customers GROUP BY City) dt WHERE CNT >= 2

1. List all Customer Cities that have ordered at least two different kinds of products.

SELECT ShipCity FROM Orders o JOIN [Order Details] od ON o.OrderID = od.OrderID GROUP BY o.ShipCity HAVING COUNT(od.ProductID) > 1;

1. List all Customers who have ordered products, but have the ‘ship city’ on the order different from their own customer cities.

SELECT DISTINCT c.CustomerID FROM Orders o JOIN Customers c ON o.CustomerID = c.CustomerID WHERE o.ShipCity != c.City;

1. List 5 most popular products, their average price, and the customer city that ordered most quantity of it.

SELECT ProductID, AvgPrice, City FROM (SELECT TOP 5 ProductID, COUNT(ProductID) CNT, AVG(od.UnitPrice) AvgPrice, od.OrderID FROM [Order Details] od GROUP BY od.ProductID, od.OrderID ORDER BY CNT DESC) dt JOIN Orders o ON dt.OrderID = o.OrderID JOIN Customers c ON o.CustomerID = c.CustomerID GROUP BY ProductID, AvgPrice, City

1. List all cities that have never ordered something but we have employees there.
   1. Use sub-query

SELECT e.City FROM Employees e WHERE e.City NOT IN (SELECT o.ShipCity FROM Orders o)

* 1. Do not use sub-query

SELECT e.City FROM Employees e LEFT JOIN Orders o ON e.City = o.ShipCity WHERE o.ShipCity IS NULL

1. 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)
2. How do you remove the duplicates record of a table?

We can use GROUP BY and HAVING COUNT(\*) > 1 to find duplicates.

12. Sample table to be used for solutions below- Employee (empid integer, mgrid integer, deptid integer, salary money) Dept (deptid integer, deptname varchar(20))

Find employees who do not manage anybody.

SELECT e1.empid FROM Employee e1 LEFT JOIN Employee e2 ON e1.mgrid = e2.empid WHERE e2.empid IS 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.

SELECT d.deptname, COUNT(e.empid) FROM Employee e JOIN Dept d ON e.deptid = d.deptid GROUP BY e.deptid, d.deptname HAVING COUNT(e.empid) = (SELECT MAX(cnt) FROM (SELECT d.deptname, COUNT(e.empid) cnt FROM Employee e JOIN Dept d ON e.deptid = d.deptid GROUP BY e.deptid, d.deptname) ORDER BY d.deptname);

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.

SELECT d.deptname, e.emptid, e.salary, RANK() OVER(PARTITION BY d.deptid ORDER BY(e.salary) DESC) AS RNK FROM Employees e JOIN Dept d ON e.deptid = d.deptid WHERE RNK <= 3 GROUP BY d.deptname, e.emptid, e.salary