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?

I prefer subqueries because although subqueries might be a bit slower it has better readability.

1. What is CTE and when to use it?

Common table expression is a temporary named result set, it can help create a recursive query, substitute view, improve readability.

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

The table variable is a special type of the local variable that helps to store data temporarily.

The scope is within its batch, procedure or function.

They are created in the tempdb database.

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

DELETE is used to delect specific rows in a table while TRUNCATE is used to delete every data row but the column row in a table, meaning it deletes every data but still keeps the structure of the table. TRUNCATE is faster because DELETE needs to scan every row and delect the rows that meet the requirement while TRUNCATE just delete every row without judgement, it makes less use of the transaction log.

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

Identity column of a table is a column whose value increases automatically. The value in an identity column is created by the server. A user generally cannot insert a value into an identity column. Identity column can be used to uniquely identify the rows in the table.

Delete retains the identity and does not reset it to the seed value.

Truncate command reset the identity to its seed value.

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

The functionality is the same but truncate is faster.

# Write queries for following scenarios

All scenarios are based on Database NORTHWND.

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

select distinct City from Customers

intersect

select distinct City from Employees

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

select distinct City from Customers

where City not in(select distinct City from Employees)

* 1. Do not use sub-query

select distinct City from Customers

except

select distinct City from Employees

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

select Products.ProductID,Products.ProductName, sum(Quantity) as 'total order quantity'from Products left join [Order Details]on Products.ProductID =[Order Details].ProductID

group by Products.ProductID,Products.ProductName

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

select City,sum(Quantity) as 'total products ordered'

from Customers left join Orders on Customers.CustomerID=Orders.CustomerID join [Order Details] on Orders.OrderID=[Order Details].OrderID

group by City

1. List all Customer Cities that have at least two customers.
   1. Use union

select City,count(CustomerID) as count

from Customers

group by City

having count(CustomerID)=2

union

select City,count(CustomerID) as count

from Customers

group by City

having count(CustomerID)>2

* 1. Use sub-query and no union

select distinct City

from Customers where City not in

(select City from

(select City,count(CustomerID) as count from Customers

group by City

having count(CustomerID)<=1) a

)

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

select City,Count([Order Details].ProductID) as 'type of products'from

Customers left join Orders on Customers.CustomerID=Orders.CustomerID

left join [Order Details] on Orders.OrderID = [Order Details].OrderID

group by City

having Count([Order Details].ProductID)>=2

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

select distinct Customers.CustomerId, ContactName from

Customers left join Orders on Customers.CustomerID=Orders.CustomerID

left join [Order Details] on Orders.OrderID = [Order Details].OrderID

where ShipCity is not null and City <>ShipCity

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

with cte as

(

select \* from

(

select [Order Details].ProductID, Customers.City, sum(Quantity) as 'Total', rank() over(partition by ProductID Order by sum(Quantity) desc) as 'Rank'

from Customers join Orders on Customers.CustomerID=Orders.CustomerID join [Order Details] on Orders.OrderID=[Order Details].OrderID

group by [Order Details].ProductID, Customers.City

)a

where Rank=1

)

select a.ProductID, [average price],City from

(select top 5 ProductID, avg(UnitPrice) as 'average price' from [Order Details]

group by ProductID

order by sum(Quantity) desc) a left join cte on a.ProductID=cte.ProductID

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

select distinct City from Employees

Where City not in(

select distinct City from Orders join Customers on Orders.CustomerID=Customers.CustomerID)

* 1. Do not use sub-query

Select distinct City from Employees

except

Select distinct City from Orders join Customers on Orders.CustomerID=Customers.CustomerID

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)

select Top 1 City from Employees join Orders on Employees.EmployeeID=Orders.EmployeeID

group by City

having City in(

select Top 1 Customers.City as City from

Customers join Orders on Customers.CustomerID=Orders.CustomerID join [Order Details] on Orders.OrderID=[Order Details].OrderID

group by Customers.City

Order by sum(quantity) desc)

Order by count(Orders.OrderID) desc

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

Find duplicate rows using GROUP BY clause or ROW\_NUMBER() function then use delete where rownumber>1

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.

Select empid from Employee

where empid not in(

select mgrid from Employee)

1. 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 deptname, [count of employee]from

(

Select CAST(deptname AS NVARCHAR(100)) deptname, count(empid) as 'count of employee', rank() over (order by count(empid) desc) as r from Dept left join Employee on Dept.depid=Employee.depid

group by CAST(deptname AS NVARCHAR(100))

) a

where r=1

order by 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.

with cte as

(

select deptname,empid ,salary, rank() over(partition by (Dept.depid) order by salary desc) as rank

from Employee join Dept on Employee.depid=dept.depid

)

select deptname,empid ,salary from cte where rank<4

order by CAST(deptname AS NVARCHAR(100))

GOOD LUCK.