

# Day 3

## primary key vs unique key

primary key cannot be null, unique key can

only 1 primary key, multiple unique key

primary key gives clustered index, unique key gives unclustered.

## constraints

Not null

between A and B

default()

IDENTITY(seed, increment)

## Window functions

rank over

```
--Rank: When ranking, numbers jump if there is a tiebreaker
--Dense_Rank: Numbers dont jump on ties
--Row_Number: just counters by the row number

Select p.ProductId, p.ProductName, p.UnitPrice,
Rank() Over(Order By UnitPrice asc) "Rank",
Dense_Rank() Over(Order By UnitPrice asc) [Dense],
Row_Number() Over(Order By UnitPrice asc) rownum
From Products p
```

partition

```
--Partition By: "In Relation to each"
Select c.ContactName, c.Country, Count(o.OrderId) [NumofOrd],
Rank() Over( Partition By c.Country Order By Count(o.OrderId)) as "Rank"
From Customers c Join Orders o on c.CustomerId = o.CustomerId
Group By c.ContactName, c.Country
```

	ContactName	Country	NumofOrd	Rank
4	Georg Pippis	Austria	10	1
5	Roland Mendel	Austria	30	2
6	Catherine Dewey	Belgium	7	1
7	Pascale Cartrain	Belgium	12	2
8	Pedro Alfonso	Brazil	5	1
9	Anabela Domingues	Brazil	6	2
10	Aria Cruz	Brazil	7	3
11	Bernardo Batista	Brazil	9	4
12	Paula Parente	Brazil	9	4
13	André Fonseca	Brazil	9	4
14	Janete Limeira	Brazil	11	7
15	Lúcia Carvalho	Brazil	13	8

Row number

```
Select c.ContactName, c.Country,
Row_Number() Over( Partition By c.Country Order By c.Country) as "Rank"
From Customers c Join (Select distinct CustomerId From Orders) o on o.CustomerId = c.CustomerId
Order By 2, 3
```

View: virtual table

```
CREATE VIEW SalesEmployees AS
SELECT ID, Salary
FROM Employees
WHERE Dept = 'Sales';
```

CTE: common table expressions

CTE can be used only in the query right after the CTE declared

```

SQLQuery1.sql - (L...-PB3VU74\Lynn (73))* X
--CTE: common table expressions
with orderCTE
as (
  select count(o.orderId) [Total Count], o.CustomerId
  From Orders o
  Group By o.CustomerId
)
Select * From OrderCTE
Select * From OrderCTE

```

113 %

Results Messages Execution plan

(89 rows affected)

(1 row affected)

Msg 208, Level 16, State 1, Line 270  
Invalid object name 'OrderCTE'.

Completion time: 2023-03-23T15:35:14.7584973-04:00

```

SQLQuery1.sql - (L...-PB3VU74\Lynn (73))* X
with orderCTE
as (
  select count(o.orderId) [Total Count], o.CustomerId
  From Orders o
  Group By o.CustomerId
)
usingOrderCTE as
(
  select [Total Count], CustomerID, 'I have referenced the previous cte' [Used]
  From OrderCTE
)
ohno as(
  select o.[Total Count], uo.CustomerId
  From OrderCte o join UsingOrderCTE uo on o.CustomerId = uo.CustomerId
  select * From UsingOrderCte
)

```

recursive cte

```

With EmpHieCte as
(
  select EmployeeId, FirstName, ReportsTo, 1 [level]
  From Employees
  Where ReportsTo is Null
  Union All
  select e.EmployeeId, e.FirstName, e.ReportsTo, Cte.[level] + 1
  From Employees e Join EmpHieCte cte on e.ReportsTo = cte.EmployeeId
)
select * From EmpHieCte

```

prevent sql injection

```
select * From EmpHieCte  
  
Declare @idpara varchar(30)  
set @idpara = '10248; Select * From Information_Schema.Tables'  
  
Select ShipAddress, ShippedDate, OrderId  
From Orders o  
Where OrderId = 10248; Select * From Information_Schema.Tables
```