--------------- Part 5 CUBE ROLLUP AND GROUPING SET---------------

---1---

select City, Country, grouping\_id(City ,Country) ID

from Customers

group by cube(City,Country)

having grouping\_id(City ,Country) = 1

order by ID

---2---

select CompanyName, ROW\_NUMBER() OVER (partition by city order by city)

from Customers

---3---

with EMP AS

(

select ROW\_NUMBER() over(order by hiredate) ROW# , \*

from Employees

)

select top 5 \*

from EMP

where ROW#>3

---4a---

with prod AS

(

select ROW\_NUMBER() over(order by unitprice desc) ROW# , \*

from Products

)

select \*

from prod

where ROW# = 18

---4b---

select \*

from products

order by UnitPrice desc

offset 17 rows

fetch next 1 rows only

---5a---

select \*

from Customers

where CustomerID in (select CustomerID

from Orders

group by CustomerID

having count(\*)>=14)

---5b---

with ord AS

(

select ROW\_NUMBER() over(partition by customerid order by orderId desc) ROW# , \*

from orders

)

select distinct c.\*

from ord o join Customers c

on o.CustomerID = c. CustomerID

where ROW#>= 14

---7---

select \*

from Orders

where year(OrderDate) = 1997

order by OrderDate

offset 10 rows

fetch next 10 rows only

-----or------

go

with ord AS

(

select ROW\_NUMBER() over(order by orderdate) ROW# , \*

from Orders

where year(OrderDate) = 1997

)

select top 10 \*

from ord

where ROW# > 10

---8---

with cat as(

select ROW\_NUMBER() over(order by CategoryName) ROW# , \*

from Categories)

select p.\*

from Products p join cat c

on p.CategoryID = c.CategoryID

where ROW# = 2

---9---

with ord as(

select ROW\_NUMBER() over(partition by month(OrderDate) order by month(OrderDate) ) ROW# , \*

from Orders

where year(OrderDate) = 1996

)

select \*

from ord

where ROW# = 1

---11a---

create TABLE #table

(myday datetime,

totalcustomers int)

declare @rows int = 1

while @rows <=10000

begin

insert into #table

values

((SELECT DATEADD(day, (ABS(CHECKSUM(NEWID())) % 1090), '2008-01-01')) , (SELECT ABS(CHECKSUM(NEWID()) % 6000)))

SET @rows= @rows+1

PRINT (@rows)

END

--------------- Part 7 DATETIME ISSUES, ---------------

---1---

|  |  |  |  |
| --- | --- | --- | --- |
| DateType | From date | To Date | Storage |
| datetime | 1753-01-01 | 9999-12-31 | 8 Bytes |
| Datetime2(3) | 0001-01-01 | 9999-12-31 | 7 Bytes |
| date | 0001-01-01 | 9999-12-31 | 3 Bytes |
| Smalldatetime | 1900-01-01 | 2079-06-06 | 4 Bytes |
| Datetimeoffset(3) | 0001-01-01 | 9999-12-31 | 9 Bytes |

---2---

36 characters data type that can be used as unique value (key ID of the table). Because the filed is used as table ID in the table can be only one filed of that type. If converting character data type to uniqueidentifier data type all the values that bigger than 36 characters will be truncated.

---3---

declare @date datetime

set @date = getdate() -- for current date

--set @date = DATEADD(day, (ABS(CHECKSUM(NEWID())) % (datediff(dd, 0,getdate()))), '1900-01-01') -- for random date

---- A ----

select case when Year(@date)%100 = 0 then 'Leap year'

else case when Year(@date)%4 = 0 then 'Leap year'

else 'Regular year'

End

End AS 'IsLeap'

,@date AS 'Date'

---- B ----

declare @quater int = DATEDIFF(qq, 0, @date)

SELECT DATEADD(qq, @quater, 0) AS 'FirstDayofQ'

,datename(DW, DATEADD(qq, @quater, 0)) AS 'DayOfWeek'

,@date AS 'Date'

---- C ----

declare @quater2 datetime = DATEADD(qq, DATEDIFF(qq, 0, @date) +1, 0)

SELECT DATEADD(dd, -1, @quater2 ) AS 'LastDayofQ'

,datename(DW, DATEADD(dd, -1, @quater2 )) AS 'DayOfWeek'

,@date AS 'Date'

---- D ----

declare @Week int = DATEDIFF(ww, 0, @date)

SELECT datename(dw,@@DATEFIRST) as FirstDOW

,@@LANGUAGE as Language

,DATEADD(WW, @Week, 0) AS 'FirstDayofW'

,DATEADD(dd, -1, DATEADD(WW, @Week +1, 0) ) AS 'LastDayofW'

,@date AS 'Date'

---- E ----

select DATEDIFF(ww, dateadd(dd,1,EOMONTH ( @date, -1 )), @date)+1 AS 'NumberofWeek\_Month'

,DATEDIFF(ww, DATEADD(qq, DATEDIFF(qq, 0, @date), 0), @date)+1 AS 'NumberofWeek\_Quarter'

,DATEPART( wk, @date ) AS 'NumberofWeek\_Year'

,@date AS 'Date'

---- F ----

SELECT DATEDIFF(dd, DATEADD(dd, -1, DATEADD(qq, DATEDIFF(qq, 0, @date), 0) ), DATEADD(dd, -1, DATEADD(qq, DATEDIFF(qq, 0, @date)+1, 0) ) ) AS 'NumOfDays\_Quarter'

,DATEDIFF(dd, EOMONTH ( @date, - 1), EOMONTH ( @date)) AS 'NumOfDays\_Month'

,@date AS 'Date'

---- G ----

select EOMONTH ( @date) AS 'EndOfM'

,DATENAME(DW, EOMONTH ( @date)) AS 'DayOfWeek'

,@date AS 'Date'

---- H ----

select DATEDIFF(dd, @date, DATEADD(dd, -1, DATEADD(qq, DATEDIFF(qq, 0, @date)+1, 0) )) AS 'DaysUntilEnd\_Quoter'

,DATEDIFF(dd, @date, EOMONTH(@date)) AS 'DaysUntilEnd\_Month'

,DATEDIFF(dd, @date, DATEADD(dd, -1, DATEADD(WW, DATEDIFF(ww, 0, @date) +1, 0) )) AS 'DaysUntilEnd\_Week'

---4---

declare @language varchar(50)

,@NumOfMonth int

set @language = 'Thai'

set @NumOfMonth = 4

set LANGUAGE @language

SELECT DATENAME(MM, DATETIMEFROMPARTS ( YEAR(GETDATE()), @NumOfMonth, DAY(GETDATE()), 0, 0, 0, 0) )

SET LANGUAGE English

SELECT @@LANGUAGE