# How the result is generated for Select … From … Where? ELIMINATION!

1. A SQL query is performed in the order of FROM WHERE SELECT rather than select from where.

FROM: First all the tables are joined in the Cartesian manner

WHERE: Every row is considered if it satisfies the condition in Where clause

SELECT: Only selected columns are viewed

1. Where, Group by eliminate rows while Select filter which columns to be shown
2. Functions apply on columns; they don’t restrict rows.

# A note about GROUP BY

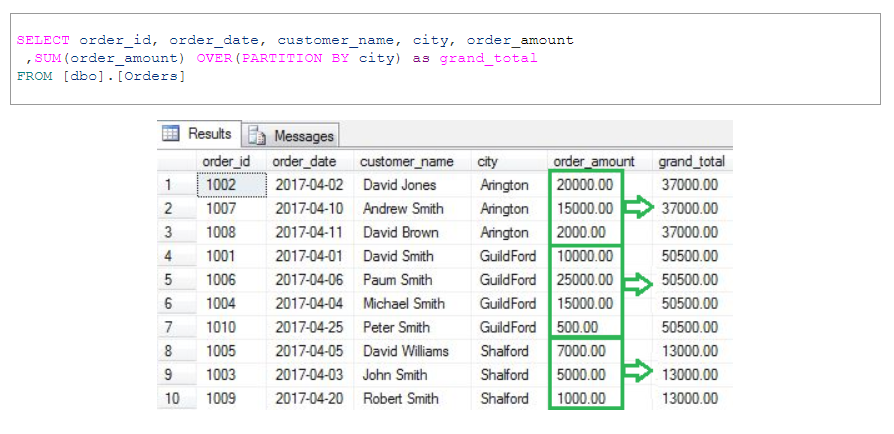
1. GROUP BY can have more than 1 field.  
   Group By X: put all rows with same X value in a group

Group By X, Y: put all rows with same both X, Y value in a group

1. If you use GROUP BY, then fields in SELECT must be the fields in the GROUP BY. Select can contain Aggregate functions on fields that are NOT in GROUP BY.
2. Condition for GROUP BY is given in HAVING, not in WHERE
3. Conditions in Having are on the fields in GROUP BY or on aggregate functions on the fields in GROUP BY.

Having cannot contain fields not in GROUP BY; otherwise use WHERE.

# Why is it called WINDOW functions?



# OVER PARTITION BY vs GROUP BY

Over partition by is similar to Group by.

Example: Consider the following “Employee” table

|  |  |  |  |
| --- | --- | --- | --- |
| Employee | | | |
| ID | EmployeeName | EmployeeAge | DepartmentName |
| 1 | Michael | 50 | Accounting |
| 2 | Frances | 60 | Accounting |
| 3 | Daisy | 20 | Accounting |
| 4 | Dennis | 40 | Software |
| 5 | Andrei | 25 | Software |
| 6 | Carol | 25 | Software |
| 7 | Peter | 40 | Human resource |
| 8 | Larry | 35 | Human resource |
| 9 | Robert | 35 | Human resource |
| 10 | Leonard | 22 | Marketing |
| 11 | Andrew | 24 | Marketing |

**Group by:**

**Exp1**: count the employees of each department that has more than 2 employees

Select DepartmentName, count(EmployeeName) as Count

From Employee

Group by DeparmentName

Having count(EmployeeName) > 2

|  |  |
| --- | --- |
| Group by | |
| DepartmentName | Count |
| Accounting | 3 |
| Software | 3 |
| Human resource | 3 |

(Note: it doesn’t compile if use “where” instead of “having”

~~Select DepartmentName, count(EmployeeName) as Count~~

~~From Employee~~

**~~Where Count > 2~~**

~~Group by DeparmentName~~)

Exp2: Counting number of employees older than 25 for each department

Select DepartmentName, count(EmployeeAge) as X

From Employee

where EmployeeAge > 25

Group by DepartmentName

|  |  |
| --- | --- |
| Group by | |
| DepartmentName | Count |
| Accounting | 2 |
| Software | 1 |
| Human resource | 3 |

(Note: it doesn’t compile if use “Having” instead of “Where” since EmployeeAge must be in Group by

Select DepartmentName, count(EmployeeAge) as X

From Employee

Group by DepartmentName

Having EmployeeAge > 25)

**Over partition by:**

**Exp1**: count the employees of each department that has more than 2 employees

The following doesn’t work

Select EmployeeName, DepartmentName, count(EmployeeName) over (partition by DepartmentName) as x

from Employee

where X > 2

Exp2: Counting number of employees older than 25 for each department

Select EmployeeName, DepartmentName, count(EmployeeName) over (partition by DepartmentName) as Count

From Employee

Where EmployeeAge > 25

|  |  |  |  |
| --- | --- | --- | --- |
| Over partition by | | | |
| EmployeeName | EmployeeAge | DepartmentName | Count |
| Michael | 50 | Accounting | 2 |
| Frances | 60 | Accounting | 2 |
| Dennis | 40 | Software | 1 |
| Peter | 40 | Human resource | 3 |
| Larry | 35 | Human resource | 3 |
| Robert | 35 | Human resource | 3 |

The most important difference is “Aggregate vs Analytic”.

In the above example, for Group By, only groups plus some properties of the groups are viewed. For Over partition by, individual elements of groups and properties of groups are displayed so there must be repetition.

Viewing group is called aggregate while viewing elements is called analytic.

# For every group, list only some elements

Exp: still with Employee table above, for each department, listing the oldest employee

Subquery:

Select DepartmentName, EmployeeName

From Employee

Where EmployeeAge = (Select MAX(EmployeeAge)

From Employee as tempEmployee

Where tempEmployee.DepartmentName = Employee.DepartmentName)

Over partition by

select \*

from (select DepartmentName, EmployeeName, EmployeeAge,

row\_number() over (partition by DepartmentName order by EmployeeAge DESC) as rank

from Employee) as extendedEmployee

where extendedEmployee.rank = 1

# with tableName as (select from where)

“with” must be followed by Select/Insert/Update command and it’s used only ONCE.

“with” cannot be followed by “with”. If want so, put multiple “with” in one command, separated by comma.

With name1 as (…), name2 as (…)

# In FROM clause, user-defined tables must have name

# JOIN, be it left join, outer join, inner join can multiply the number of records