Order By : Sorts the output based on a (Set) of attribute(s)

```
Select ...
from ...
where ...
order by Col1, Col2, ..., ColK
```

Default is ascending

[desc] for descending order

Find the names, Salary of the Professors in the 'CS' Dep., Sorted by Salaries

```
Select Professor.name , Salary
from    Professor, Department
where
        Professor.DepID = Department.ID
        and
        Department.name = 'CS'
order by Salary desc
```

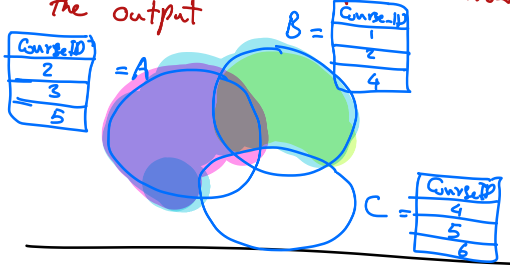
Set Operations

on query outputs with The Same Schema (Columns)

- Union : Union of two Tables
- Intersect : Intersection of " "
- Except : Subtraction of " "

```
(Select Course_ID from Section
   where Semester = 'Fall' and Year = 2025)
UNION
( "   "   "   "
   "  Semester = 'Spring'  "   " )
EXCEPT
( "   "   "   "
   "  Semester = 'Fall' and Year = 2024)
```

* Because it is a Set operation, It removes the duplicates from the output



A =

Course_ID
2
3
5

B =

Course_ID
1
2
4

C =

Course_ID
4
5
6

```
(Select Course.name from Course)
Except
( Select Course.name
    from Section , Course
    where
        Section.CourseID = Course.ID
        and
        Semester = 'Fall' and
        Year = 2025 )
```

Courses that were not offered in Fall 2025

Null ~~value~~

Common Mistake

email = null X

↖ Cannot be used for null

{ is null
  is not null

- Find the names of the ~~or~~ Professors who are a department head but has no phone number

```
Select Professor. name
from Professor , Department
where
    Professor.ID = DeptHead.ID
    and
    Phone IS NULL
```

Aggregation Operations.

- COUNT : The number of rows
- SUM : The Sum of values
- AVG : " Average " "
- MIN : " minimum " "
- MAX : " Maximum " "

- The number of Students who registered for 'CS48032o'

```
Select COUNT(*)
From Section
where CRN = 'CS48032o'
```

- The name & avg Salary of the departments

```
Select name, AVG(Salary)
from Professors
```

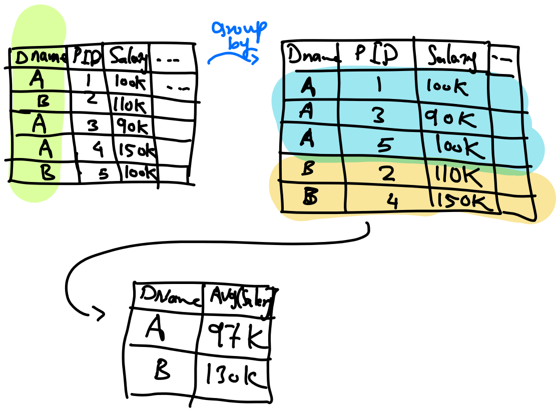
$\boxed{?}$

? : goal: to Partition the rows based on Some Columns & then apply the aggregate for each group Separately.

Group by [Attributes]

having [group by Conditions]

```
Select Dep.name, AVG(Salary)
from Professors , Department
where
    Professor.DepID = Department.ID
GROUP BY
    Department. name
```



Dname	PID	Salary	...
A	1	100k	...
B	2	110k	
A	3	90k	
A	4	150k	
B	5	100k	

Dname	PID	Salary	...
A	1	100k	
A	3	90K	
A	5	100K	
B	2	110K	
B	4	150K	

DName	AVG(Salary)
A	97K
B	130k

Find the names & Avg. Salaries of Departments with Avg. Salary less than 100k

Note: The Condition on Avg Salary must apply AFTER the group by operation

Select Dep.name, AVG(Salary)
from Professors, Department
where
Professor.DepID = Department.ID
GROUP BY
Department.name
HAVING AVG(Salary) < 100k

Nested Queries

Select $A_1, \dots, A_K$
From $T_1, T_2, \dots, T_m$
where $P_1, P_2, \dots P_\ell$

Each of A_i, T_j, or P_ℓ Can be a Subquery.

How to replace P_i with a Subquery?

A_i [Operation] Subquery

[Operations] =
- Set membership
  - IN, NOT IN
- Set Comparisons
  - $\begin{bmatrix} > \\ < \\ = \end{bmatrix} \begin{bmatrix} \text{Some} \\ \text{all} \end{bmatrix}$
  - Exists: checks for emptiness
  - UNIQUE: " " Duplicates

Set membership:

IN: if Some value belongs to the output set of the Subquery

e.g.,

CourseID IN (Select...)

True if belongs to set
False otherwise

– The Course IDs offered in Fall 2017 and Spring 2018

```
Select CourseID
from Course
where
    Semester = 'Fall' and
    Year = 2017
    and
  CourseID IN (
      Select Course ID
      From Course
      where
          Semester = 'Spring'
          and
          Year = 2018  )
```

– Set Comparison

– $\begin{bmatrix} > \\ < \\ = \end{bmatrix} \begin{bmatrix} \text{Some} \\ \text{all} \end{bmatrix}$

– Some : returns TRUE if at least one row of the set satisfies the condition

– all : " " only if all rows satisfy the condition

$\text{Salary} > \text{Some}\left(\begin{array}{|c|}\hline 100 \\ \hline 200 \\ \hline 150 \\ \hline \end{array}\right)$

||

150 : True bcz. 150 > 100

$\text{Salary} > \text{all}\left(\begin{array}{|c|}\hline 100 \\ \hline 200 \\ \hline 150 \\ \hline \end{array}\right)$

||

150 : False bcz.

$150 \ngtr 200$

Select name
from Professor
where
Salary ~~>~~ (Select Avg(Salary)
from Professor)

ERROR

An Int Variable

A Table

Pi

```
Select   name
from     Professor
where
  Salary > Some( Select Avg(Salary)
                 from Professor)
```

EXISTS { True if The Sub query output is not Empty
False otherwise

```
- Select ....
  from   ..
  where
  [NOT] EXISTS ( Query )
```

- Find all CourseIDs that were taught both in 2017 & 2025

```
- Select T.Course ID
  From   Section as T
  where
        Year = 2017
        and
        Exists ( Select *
                 From Section as S
                 where Year = 2025
                 and
                 S.CourseID = T.CourseID)
```

UNIQUE : { True if the Query output does not include Duplicates
False otherwise

- Find CourseID Courses that were offered Exactly once in Fall 2025. ✓

```
- Select   Course ID
  from     Section
  group by CourseID
  Having   Count (CourseID) = 1
```

```
Select   CourseID
from     Course
where
    UNIQUE(
       Select  CourseID
       from    Section
       where
               Course.CourseID =
                        Section.CourseID
          and
          Semester = 'Fall'
          and
          Year = 2025 )
```

Section

Course ID	Se_	Year
CS480	Fall	2025
CS401	Fall	2025
CS480	Spring	2025
CS401	Spring	2025
CS480	Fall	2025

→

CourseID
CS480
CS480

Unique → False

Insert : to add a new Row to a Table

```
INSERT INTO [Table] ([Columns])
            Values ([Value])
```

- Insert into Department (name, address)

```
values ('Computer Science',
        '500 Taylor St')
```

Update : to update the values of Existing rows

```
UPDATE [Table]
Set    [Updates]
where  [Conditions]
```

- e.g., give a 10% raise to all Prof. that theirs Salary is less than 50k.

```
-Update Professors
 Set    Salary = 1.1 * Salary
 where  Salary < 50000
```

Delete : to remove rows from a Table

```
DELETE [Table]
WHERE [Condition]
```

- e.g., delete all Courses offered by the CS dep.

```
DELETE Course
where
   CourseID Like 'CS%'
```

```
delete Course
 where
   DepID IN (Select ID
             from Department
             where
               name = 'CS')
```