

## DAY-2 SQL

### (1) AGGREGATE FUNCTIONS →

These operate on multiple rows and returns a single result.

(1) `count()`

count numbers of rows

`select count(*) from employees;`

`sum()`

Adds up value

`select sum(salary)`  
`from employees;`

`avg()`

Average value.

`select avg(salary)`  
`from employees;`

`min()`

minimum value

`select min(salary)`  
`from employees;`

`max()`

maximum value.

`select max(salary)`  
`from employees.`

### (2) STRING FUNCTIONS →

`concat()`

combines strings

`select concat(first_name, ' ', last_name)`  
`from employees;`

`length()`

returns string length

`select length(name) from employees;`

`lower()`

converts to lowercase

`select lower(name) from employees;`

`upper()`

converts to uppercase

`select upper(name) from employees;`

`substring()`

Extracts parts of string

`select substring(name, 1, 3)`  
`from employees;`

### (3) Data FUNCTIONS →

`now()`

current timestamp

`Select Now();`

`curdate()`

current date.

`Select curdate();`

`datediff()`

diff in date.

`select datediff(curdate(),`  
`hiredate) from employees;`

`date_format()`

format date

`-en?`

`select date_format`  
`curdate(), '%Y-%m-%d')`  
`from employees;`

## Mathematical Functions

Round()

rounds number

Select Round(123.456, 2)  
→ 123.46

Floor()

rounds down

Select Floor(123.456);  
→ 123

Ceil()

rounds up

Select Ceil(123.456) - (2)

Abs()

absolute value

Select Abs(-10); -10

GroupBy → Groups rows based on column(s) and applies aggregate function.

Select department\_id, Count(\*)

From employees

Group By department\_id

HAVING → Used to filter aggregated results (like where, but for groups)

Select department\_id, Count(\*)

From employees

Group By department\_id

Having Count(\*) > 5;

where filters before grouping

having filters after.

ALIAS (AS) - Alias are used to rename columns or tables temporarily for readability

Select first\_name as name, salary as "Employee Salary"  
From employees;

Select e.name, d.name.

From employees AS e

Join departments AS d  
On e.department\_id = d.department\_id.

Filtrering mit IN, Between, like, is null

Date: 1

Page: 1

IN

Matches any value in a list

WHERE dep\_id IN  
110, 201301

Between

Between two values

WHERE salary  
BETWEEN 20000 AND 40000

like

Pattern matching

WHERE name like 'A-%'

is null

Checks for null value

WHERE manager\_id  
IS NULL

(\*) WHERE ... ORDER BY ... LIMIT ...

SELECT ... FROM ...

ORDER BY ... GROUP BY ...

SELECT ... FROM ... WHERE ...

WHERE ... AND ...

WHERE ... OR ...

GROUP BY ...

grouped with comma  
using WITH clause

WITH clause  
↳ breakdown and calculated column

↳ breakdown and calculated column (left) and