AGGREGATE FUNCTIONS AND JOINs

1. COUNT

What is an aggregate function?

Aggregate functions are functions that take a collection of values as input and return a single value.
COUNT (Count) / **MIN** (Minimum) / **MAX** (Maximum) / **SUM** (Total) / **AVG** (Average)

Note : **NULL** means no data and is a special value. It shows us that a piece of information is unknown or missing or not applicable.

Introduction

Syntax ⇨ SELECT COUNT(column_name) FROM table_name;

Note : AS is used to rename a column or table with an alias.
Syntax ⇨ SELECT COUNT(first_name) AS count_of_students FROM student_info;

COUNT (DISTINCT …)

COUNT(…) counts the duplicate rows as separate rows. So, we need to use COUNT(DISTINCT …) function.
Syntax ⇨ SELECT COUNT(DISTINCT first_name) AS given_name FROM student_info;

We can also combine COUNT(DISTINCT) or COUNT() functions with WHERE clause.
Syntax ⇨ SELECT COUNT(*) AS state FROM student_info WHERE state = “Adana”;

Check Yourself Questions:

Please write a query to return the number of students as "number_of_students" from West Virginia State.

Answer: (penalty regime: 0, 10, 20, … %)

```
1  SELECT COUNT(*) AS number_of_students FROM student_info WHERE
       state = "West Virginia";
```

Please write a query to return the number of female students as "number_of_students" from Virginia State.

Answer: (penalty regime: 0, 10, 20, … %)

```
1  SELECT COUNT(*) AS number_of_students FROM student_info WHERE
       state = "Virginia" AND gender = "F";
```

Please write a query to return distinct number of job titles as "number_of_titles" in the company.

Answer: (penalty regime: 0, 10, 20, … %)

```
1  SELECT COUNT(DISTINCT job_title) AS number_of_titles FROM
       employees;
```

2. MIN and MAX

MIN Function

MIN function returns the minimum value in the selected column.
Syntax ⇨ SELECT MIN(column_name) FROM table_name;

Note : The MIN function ignores the NULL values. Thus, it retrieves the non-NULL minimum value in the selected column.

Example: Who gets paid the lowest wage in the company?
Syntax ⇨ SELECT MIN(salary) AS lowest_salary FROM employees;

Example: Display the female employee who gets paid the lowest.
Syntax ⇨ SELECT MIN(salary) AS lowest_salary FROM employees WHERE gender = 'Female' ORDER BY salary LIMIT 1;

Example: Display the earliest hired employee's date.
Syntax ⇨ SELECT MIN(hire_date) AS earliest_date FROM employees;

Max Function

MAX function returns the maximum value in the selected column.
Syntax ⇨ SELECT MAX(column_name) FROM table_name;

Example: Find the highest wage in the company.
Syntax ⇨ SELECT MAX(salary) AS highest_salary FROM employees;

Note: We can also combine MAX function with WHERE clause, ORDER BY and LIMIT as we did in the MIN function.

Check Yourself Questions:

Please write a query to return the earliest start date as 'earliest_date"' among the female students.

Answer: (penalty regime: 0, 10, 20, ... %)

```
1 SELECT MIN(start_date) AS earliest_date FROM student_info;
```

Please rewrite the previous query to return the earliest start date among the female students without using the MIN function. Your query should return just a single column. (Note: Please don't use AS keyword in the query)

Answer: (penalty regime: 0, 10, 20, ... %)

```
1 SELECT start_date FROM student_info ORDER BY start_date LIMIT 1
    ;
```

Please write a query to return the newest start date as 'newest_date' among the male students.

Answer: (penalty regime: 0, 10, 20, ... %)

```
1 SELECT MAX(start_date) AS newest_date FROM student_info WHERE
    gender = "M";
```

Please rewrite the previous query to return the newest start date among the male students. Your query should return just a single column. Please don't use AS keyword and MAX function in the query.

Answer: (penalty regime: 0, 10, 20, ... %)

```
1 SELECT start_date FROM student_info WHERE gender = "M" ORDER BY
    start_date DESC LIMIT 1;
```

3. SUM and AVG

SUM Function

SUM function returns the sum of a numeric column.
Syntax ⇨ SELECT SUM(column_name) FROM table_name;

Example: Let's calculate the total amount of the salary of the employees in the company.
Syntax ⇨ SELECT SUM(salary) AS total_salary FROM employees;

Example: We can find the total amount of salary of female employees.
Syntax ⇨ SELECT SUM(salary) AS total_salary FROM employees WHERE gender = 'Female';

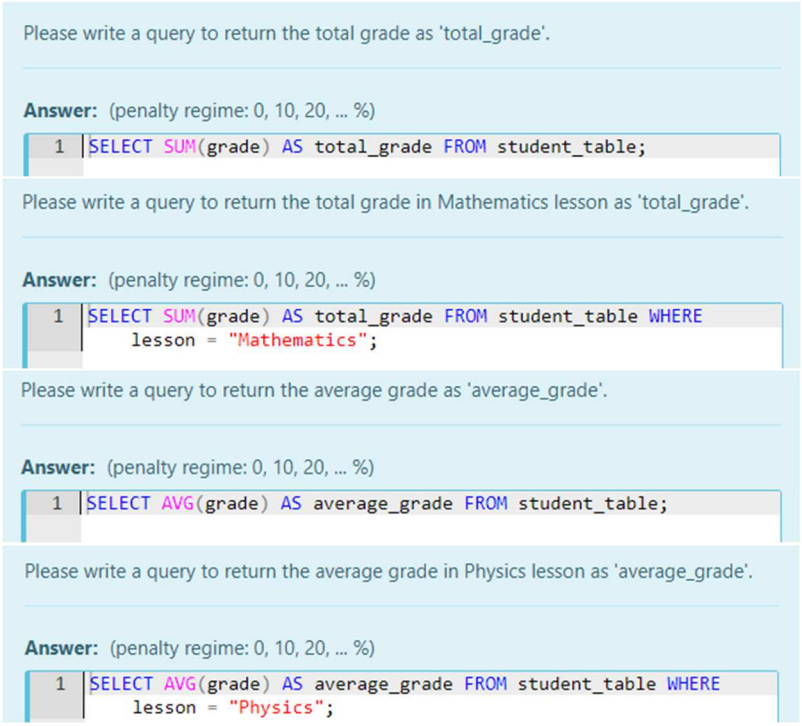
AVG Function

AVG function calculates the average of a numeric column.
Syntax ⇨ SELECT AVG(column_name) FROM table_name;

Example: What is the average salary of the employees?
Syntax ⇨ SELECT AVG(salary) AS average_salary FROM employees;

Example: Let's find the average salary amongst male employees.
Syntax ⇨ SELECT AVG(salary) AS average_salary FROM employees WHERE gender = 'Male';

Check Yourself Questions:



4. Aggregating with Grouping

GROUP BY Clause

We may need to apply the aggregate function to a multiple group of rows. In such cases, we use GROUP BY statement. Used with **COUNT**, **MAX**, **MIN**, **SUM**, **AVG**.
Syntax ⇨ SELECT column_1, aggregate_function(column_2) FROM table_name GROUP BY column_1;

Note: The GROUP BY clause should come after the WHERE clause.

GROUP BY with COUNT Function

Example: Returns the number of employees per gender.
Syntax ⇨ SELECT gender, COUNT(gender) FROM employees GROUP BY gender;

Example: What is the number of employees working as a data scientist? Break down it by gender.
Syntax ⇨ SELECT gender, COUNT(job_title) FROM employees WHERE job_title = “Data Scientist” GROUP BY gender;

GROUP BY with MIN&MAX Functions

Example: Find the minimum salaries of each gender group using the MIN function.
Syntax ⇨ SELECT gender, MIN(salary) AS min_salary FROM employees GROUP BY gender;

Note: Similarly, we can find the maximum salaries of each group using the MAX function.

Note2 : If we wanna sort the result, we can use the ORDER BY clause. But ORDER BY follows GROUP BY.

Example: Sort the maximum salaries for each group in descending order
Syntax ⇨ SELECT gender, MAX(salary) AS max_salary FROM employees GROUP BY gender ORDER BY max_salary DESC; --ORDER BY gender, ORDER BY MAX(salary) also true.

GROUP BY with SUM&AVG Functions

Example: Calculate the total salaries of each group (gender).
Syntax ⇨ SELECT gender, SUM(salary) AS total_salary FROM employees GROUP BY gender;

Example: Find the average salaries of each group using the AVG function.
Syntax ⇨ SELECT gender, AVG(salary) AS average_salary FROM employees GROUP BY gender;

Note : You may also use the ORDER BY clause to sort the salaries in descending or ascending order.

Example: Sort the total salaries for each group in descending order.
Syntax ⇨ SELECT gender, SUM(salary) AS total_salary FROM employees GROUP BY gender ORDER BY total_salary DESC; -- ORDER BY gender, ORDER BY SUM(salary) also true.

Check Yourself Questions:

Please write a query to return the total salary of each department. The result table's column headers will be dept_name and total_salary.

Answer: (penalty regime: 0, 10, 20, ... %)

```
1  SELECT dept_name, SUM(salary) AS total_salary FROM department
       GROUP BY dept_name;
```

5. Inner JOIN

Introduction

It creates a new result table from two or more tables. INNER JOIN returns a table that contains only matched rows that meet the specified join conditions.
Syntax ⇨ SELECT columns
FROM table_A
INNER JOIN table_B ON join_conditions;

A join condition generally takes the following form: *table_A.column* = *table_B.column*.

Note: Three or more tables can be combined using the INNER JOIN clause
Syntax ⇨ SELECT columns
FROM table_A
INNER JOIN table_B
ON join_conditions1 AND join_conditions2
INNER JOIN table_C
ON join_conditions3 OR join_conditions4

Interview Question:
Q: What is inner join?
A: Retrieves records that have matching values in both tables involved in the join. This is the widely used join for queries.

Example: Suppose you want to join the "employees" table and the "departments" table. In this example, an INNER JOIN has been created that is based on the *emp_id* columns in the two tables:

Syntax ⇨

```
SELECT
        employees.emp_id,
        employees.first_name,
        employees.last_name,
        departments.dept_name,
        departments.dept_id
FROM employees
INNER JOIN departments
ON employees.emp_id = departments.emp_id;
```

Check Yourself Questions:

Please write a query to return the departments of the employees who have a salary higher than 80000. Show just first name, last name, salary and department of the employees.

Answer: (penalty regime: 0, 10, 20, ... %)

```
SELECT
    employees.first_name,
    employees.last_name,
    employees.salary,
    departments.dept_name
FROM employees
INNER JOIN departments
ON employees.emp_id = departments.emp_id
WHERE salary > 80000;
```

Please write a query to return the departments of female employees. Show just first name, last name, salary and department of the employees.

Answer: (penalty regime: 0, 10, 20, ... %)

```
SELECT
    employees.first_name,
    employees.last_name,
    employees.salary,
    departments.dept_name
FROM employees
INNER JOIN departments
ON employees.emp_id = departments.emp_id
WHERE gender = "Female";
```

6. Left JOIN

Left JOIN

All the records of the left table and the common records of the right table are returned in the query. If no matching rows are found in the right table during the JOIN operation, these values are assigned as **NULL**.

Syntax ⇨ SELECT columns FROM table_A LEFT JOIN table_B ON join_conditions

Note: LEFT JOIN and LEFT OUTER JOIN keywords are exactly the same. OUTER keyword is *optional*.

Syntax ⇨ SELECT columns FROM table_A LEFT OUTER JOIN table_B ON join_conditions

Tips:

- If no match is found for a particular row, `NULL` is returned.

Example: Suppose you want to join the "employees" table and the "departments" table. In this example, an INNER JOIN has been created that is based on the *emp_id* columns in the two tables:

Syntax ⇨

```
SELECT
    employees.emp_id,
    employees.first_name,
    employees.last_name,
    departments.dept_name,
    departments.dept_id
FROM employees
LEFT JOIN departments
  ON employees.emp_id = departments.emp_id;
```

Check Yourself Questions:

Using the Left JOIN method, write a query to return the departments of all male employees in the "employees" table. Show just first name, last name, department and department ID of the employees.

Answer: (penalty regime: 0, 10, 20, ... %)

```
SELECT
    employees.first_name,
    employees.last_name,
    departments.dept_name,
    departments.dept_id
  FROM employees
  LEFT JOIN departments
  ON employees.emp_id = departments.emp_id
  WHERE gender = "Male";
```