

SQL

Recap

- *SELECT*
- *DISTINCT*
- *WHERE*
- *OPERATORS*
- *AND, OR NOT*
- *ORDER BY*
- *LIMIT, OFFSET*
- *AGGREGATE FUNCTIONS*

Contents

- LIMIT
- LIKE OPERATOR
- WILDCARDS
- String Functions
- Aliases
- IN, BETWEEN, ANY

Table → students

id	first_name	last_name	city	roll	cgpa
1	Alice	Johnson	Dhaka	101	3.5
2	Bob	Brown	Dhaka	102	3.75
3	Charlie	Chopra	Barishal	103	3.4
4	David	Das	Khulna	104	4.00
5	Emily	Khan	Rangpur	105	3.00

LIMIT

- **LIMIT** clause is used to *specify the number of records* to return from a query.
- It is commonly used with *SELECT* statements.
- The LIMIT clause can also be used with an optional *OFFSET* to skip a certain number of rows before starting to return the results.

```
SELECT column_name(s)  
FROM table_name  
WHERE condition  
LIMIT number;
```

Example

--This will return the first 3 rows from the students table.

```
SELECT * FROM students  
LIMIT 3;
```

--This will skip the first 2 rows and return the next 3 rows.

```
SELECT * FROM students  
LIMIT 3 OFFSET 2;
```

LIKE Operator

- The **LIKE** operator is used in a **WHERE** clause to search for a *specified pattern* in a column.
- There are two *wildcards* often used in conjunction with the LIKE operator:
 - The percent sign (%) represents zero, one, or multiple characters
 - The underscore sign (__) represents one, single character
- **Syntax**

```
SELECT column_name  
FROM table_name  
WHERE column_name LIKE pattern;
```

Examples

```
SELECT * FROM students  
WHERE city LIKE 'L_nd__';
```

--Return all students from a city that starts with 'L' followed by one wildcard character, then 'nd' and then two wildcard characters:

```
SELECT * FROM students  
WHERE city LIKE '%L%';
```

--Output: Return all students from a city that *contains* the letter 'L':

Like Operator with (% and _ wildcards)

LIKE Operator	Description
WHERE name LIKE 'a%'	Finds any values that start with "a"
WHERE name LIKE '%a'	Finds any values that end with "a"
WHERE name LIKE '%or%'	Finds any values that have "or" in any position
WHERE name LIKE '_r%'	Finds any values that have "r" in the second position
WHERE name LIKE 'a_ %'	Finds any values that start with "a" and are at least 2 characters in length
WHERE name LIKE 'a__ %'	Finds any values that start with "a" and are at least 3 characters in length
WHERE name LIKE 'a%o'	Finds any values that start with "a" and ends with "o"

Wildcard Characters

- A *wildcard* character is used to substitute one or more characters in a string.
- Wildcard characters are used with the *LIKE operator*.
- The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

Symbol	Description
%	Represents zero or more characters
_	Represents a single character
[]	*Represents any single character within brackets
^	*Represents any character not in the brackets
-	*Represents any single character within the specified range

* Not supported in MySQL

Exercises on LIKE

- Find the students whose name start with 'A'.
- Find the students whose name end with 'a'.
- Find students whose name contain 'ch'.
- Find students whose name is 5 characters long and starts with 'S'.
- Find teachers whose city starts with 'New' but ends with any character.

Exercises on LIKE

- Find the students whose name starts with 'A'.
 - `SELECT * FROM students WHERE name LIKE 'A%';`
- Find the students whose name ends with 'a'.
 - `SELECT * FROM students WHERE name LIKE '%a';`
- Find students whose name contains 'ch'.
 - `SELECT * FROM students WHERE name LIKE '%ch%';`
- Find students whose name is 5 characters long and starts with 'S'.
 - `SELECT * FROM students WHERE name LIKE 'S_____';`
- Find teachers whose city starts with 'Dhak' but ends with any character.
 - `SELECT * FROM teachers WHERE city LIKE 'Dhak_';`

String Functions

- **CONCAT():** Concatenates two or more strings into one.
 - `SELECT CONCAT(first_name, last_name) FROM students.`
- **LENGTH():** Returns the length of a string.
 - `SELECT name, LENGTH(city) FROM students;`
- **LOWER():** Converts a string to lowercase.
 - `SELECT name, lower(city) FROM students;`
- **UPPER():** Converts a string to uppercase.
 - `SELECT name, upper(city) FROM students;`

String Functions

- **LEFT():** Returns the leftmost characters from a string.
 - `SELECT LEFT(first_name,3) FROM students;`
- **RIGHT():** Returns the rightmost characters from a string.
 - `SELECT RIGHT(last_name,3) FROM students;`
- **LPAD():** Pads the left side of a string with another string until it reaches the specified length.
 - `SELECT LPAD(roll, 6, '0' FROM students;`
`[101→ 000101]`
- **RPAD():** Pads the right side of a string with another string until it reaches the specified length.
 - `SELECT RPAD(city, 7, '*') FROM students;`
`[dhaka→dhaka**]`
- **REPLACE():** Replaces all occurrences of a substring within a string.
 - `SELECT name, REPLACE(city, 'Dhaka', 'Dhaka City')`
`FROM students;`
`[Dhaka→Dhaka City]`

Aliases

- SQL **aliases** are used to give a *table, or a column* in a table, a **temporary** name.
- Aliases are often used to make column names more readable.
- An alias only exists for the duration of that query.
- An alias is created with the **AS** keyword.

```
SELECT id AS student_ID  
FROM students;
```

```
SELECT id student_ID  
FROM students;
```

```
SELECT CONCAT(first_name, last_name) AS name  
FROM students.
```

Exercises

- Retrieve all students who live in Dhaka.
- Find students whose email ends with '@example.com'.
- List all students whose first name and city starts with 'A' .
- Retrieve the first name, last name, and email number, and display the email column as Contact Email and both name as Name.
- Retrieve the students' names in uppercase.
- Display the roll numbers padded with first three letter of your department.
- Retrieve students with replacement of all phone numbers starting with 019 to 017.
- Find students whose roll number is greater than 102.
- Find distinct cities where students live.
- Retrieve students whose last name contains 's'.