Module 7: MySQL Select Data

- SELECT statement is
- The syntax is:
  - **SELECT column_list**
  - **[FROM table_source]**
  - **[WHERE search_condition]**
  - **[ORDER BY order_by_list]**
  - **[LIMIT row_limit]**
- SELECT clause is used for specifying the columns
- FROM clause is to specify the name of base table
- WHERE clause is to filter the rows in the base table depending on the search condition which must be a boolean expression
- ORDER BY is for sorting the data retrieved in a specified sequence else the sequence would be same as they appear in the base table
- LIMIT clause is to specify the number of rows else all rows would be returned.

1) Select all columns:

- * is used for

2) Select few columns:

- Column names can be used

3) Column alias:

- By default MySQL gives a column in a result set the same name as the column name in the base table.
- If the column is based on calculated value, it's assigned a name based on expression for the value.
- To assign a column alias, we code the column specification followed by AS keyword and the new name.

- If the new name has spaces then put it in "" or ''.

4) Column arithmetic:

- An arithmetic expression that calculates some value.
- In arithmetic expression we can use :
  - (H) → *, /, DIV, % (L to R)
  - (L) → +, - (L to R)
- We can use parenthesis to change the order of precedence

5) Column concat:

- ❑ CONCAT function can be use to join or concatenate strings which can be any combination of characters.
- ❑ The syntax is : CONCAT(string1 [, string2] …)

6) Distinct data:

- ❑ To eliminate duplicate rows, we can include the DISTINCT keyword.

WHERE clause:

→ ❑ Where clause is used with SELECT statement to

- ❑ With where clause we can use the following operators:
  - ❑ 1. Comparison operators: =, <>/!= , <, <=, > and >=
  - ❑ 2. Logical operators: AND, OR and NOT
  - ❑ 3. Membership operator: IN and BETWEEN
  - ❑ 4. Pattern operators: LIKE

1) Comparison Operators:

- ❑ Comparison operators (=, <>/!= , <, <=, > and >=) can be used to compare any two expressions.
- ❑ The syntax of comparison operator :
  - ❑ **Where expression_1 operator expression_2**
    - ❑ If the result of comparison is true,
      - ❑ the row being tested is included in the set.
    - ❑ If the result of comparison is false or null,
      - ❑ the row is not included in the result set.
- ❑ **How comparison works:**
  - ❑ Numeric literals are used without quotes and String & date are enclosed in quotes.
  - ❑ Character comparison is case – insensitive.
  - ❑ If we compare null using comparison operator then the result is always null.(to test for null use IS NULL clause).

2) Logical Operators:

- ❑ Logical operators (AND and OR) are used to create compound conditions that consist of two or more conditions.
  - ❑ AND operator is used to specify the search that must satisfy both the conditions.
  - ❑ OR operator is used to specify the search that must satisfy atleast one condition.
  - ❑ NOT operator is to negate a condition.
- ❑ Precedence between them is NOT, AND and OR. To change the level we can use ().

3) Membership Operator:

- **IN Operator:**
    - IN is used to test whether an expression is equal to a value in a list of expressions.
    - List of expressions can be written in any order without affecting the result set.
- **BETWEEN Operator:**
    - BETWEEN is used to test whether an expression falls within range of values (both the lower and upper range are inclusive).
    - The lower limit must be coded first and upper limit as second else MySQL returns an empty set.

4. Pattern Operator:

- LIKE operator is to retrieve rows that match a string pattern called a mask➔ which determines which values in the columns satisfy the condition.
- LIKE wildcards:
    - % ➔ matches any string of zero or more characters.
    - _ ➔ matches a single character.

Order by:

- → ORDER BY clause specifies how
- The syntax is :
    - **ORDER BY expression [ASC | DESC]   [, expression [ASC | DESC]] …**
- We can sort by one or more columns and we can sort each column in either ascending (ASC) or descending(DESC) sequence. ASC is default.
- How order by works:
    - In ASC sort, special characters appear first, followed by numbers and then letters.
    - NULL always appear first in any sort sequence.
    - ORDER BY clause can use numbers to specify the columns to use for sorting. 1 represents first column, 2 represents second column, and so on.
    - ORDER BY clause can include any valid expression and column alias also.

LIMIT:

- → LIMIT clause is to limit the

- The syntax is:
    - **LIMIT [offset,] row_count**
- Offset specifies the first row to return, where offset of first row is 0.
- Row_count specifies the maximum row count beginning with first row.
- Typically LIMIT is used along with ORDER BY clause.

MCQs

1. In select statement what does * stands for

Options:

A. all columns of the table are to be returned.

B. all records meeting the full criteria are to be returned.

C. all records with even partial criteria met are to be returned.

D. None of the above is correct.

Solution:

---

2) Which of the SQL statements is correct? (Select two)

Options:

A. SELECT Username AND Password FROM Users

B. SELECT Username, Password FROM Users

C. SELECT Username, Password WHERE Username = 'user1'

D. SELECT *, Username, Password FROM Users

Solution:

---

3. Which statement reports on unique JOB_ID values from the EMPLOYEES table? (Choose all that apply.)

Options:

A. SELECT JOB_ID FROM EMPLOYEES;

B. SELECT UNIQUE JOB_ID FROM EMPLOYEES;

C. SELECT DISTINCT JOB_ID, EMPLOYEE_ID FROM EMPLOYEES;

D. SELECT DISTINCT JOB_ID FROM EMPLOYEES;

Solution:

---

4) To give a temporary name to a table, or a column in a table for more readability, what is used?

Options:

1. SQL Wildcards
2. SQL aliases
3. SQL LIKES
4. SQL Comments

Solution:

---

5) Find all the cities whose humidity is 89

Options:

A. SELECT city WHERE humidity = 89;

B. SELECT city FROM weather WHERE humidity = 89;

C. SELECT humidity = 89 FROM weather;

D. SELECT city FROM weather;

Solution:

6. Which of the following statements contains an error?

Options:

a) SELECT * FROM emp
WHERE empid = 10003;
	b) SELECT empid
	FROM emp
	WHERE empid = 10006;

c) Select empid from emp;	d) SELECT empid
	WHERE empid = 1009 AND
	lastname = 'GELLER';

Solution:

7. Which operator tests column for the absence of data?

Options:

A. EXISTS operator B. NOT operator
C. IS NULL operator D. None of these

Solution:

8. Which operator performs pattern matching?

Options:

A. BETWEEN operator B. LIKE operator
C. EXISTS operator D. None of these

Solution:

9. How to select all data from student table starting the name from letter 'r'

Options:

A. SELECT * FROM student WHERE name LIKE 'r%';
B. SELECT * FROM student WHERE name LIKE '%r%';
C. SELECT * FROM student WHERE name LIKE '%r';
D. SELECT * FROM student WHERE name LIKE '_r%';

Solution:

10. What is the meaning of LIKE '%0%0%'

Options:

A. Feature begins with two 0's
B. Feature ends with two 0's
C. Feature has more than two 0's
D. Feature has two 0's in it, at any position

Solution:

11. Find the names of the cities with temperature and condition whose condition is neither sunny nor cloudy

Options:

A. SELECT city, temperature, condition FROM weather WHERE condition NOT IN ('sunny', 'cloudy');

B. SELECT city, temperature, condition FROM weather WHERE condition NOT BETWEEN ('sunny', 'cloudy');

C. SELECT city, temperature, condition FROM weather WHERE condition IN ('sunny', 'cloudy');

D. SELECT city, temperature, condition FROM weather WHERE condition BETWEEN ('sunny', 'cloudy');

Solution:

12.Which of the following query is correct for using comparison operators in SQL

Options:

A. SELECT name, course_name FROM student WHERE age>50 and <80;

B. SELECT name, course_name FROM student WHERE age>50 and age <80;

C. SELECT name, course_name FROM student WHERE age>50 and WHERE age<80;

D. None of these

Solution:

---

13. Find the name of those cities with temperature and condition whose condition is either sunny or cloudy but temperature must be greater than 70F

Options:

A. SELECT city, temperature, condition FROM weather WHERE condition = 'sunny' AND condition = 'cloudy' OR temperature > 70;

B. SELECT city, temperature, condition FROM weather WHERE condition = 'sunny' OR condition = 'cloudy' OR temperature > 70;

C. SELECT city, temperature, condition FROM weather WHERE condition = 'sunny' OR condition = 'cloudy' AND temperature > 70;

D. SELECT city, temperature, condition FROM weather WHERE condition = 'sunny' AND condition = 'cloudy' AND temperature > 70;

Solution:

---

14. Which two of the following conditions are equivalent to each other?

Options:

A. WHERE SALARY <=5000 AND SALARY >=2000

B. WHERE SALARY IN (2000,3000,4000,5000)

C. WHERE SALARY BETWEEN 2000 AND 5000

D. WHERE SALARY > 2000 AND SALARY < 5000

E. WHERE SALARY >=2000 AND <=5000

Solution:

15. Which two of the following conditions are equivalent to each other?
Options:

A. WHERE COMMISSION_PCT IS NULL
B. WHERE COMMISSION_PCT = NULL
C. WHERE COMMISSION_PCT IN (NULL)
D. WHERE NOT(COMMISSION_PCT IS NOT NULL)

Solution:

Q16) Wrong statement about ORDER BY keyword is
Options:

1. Used to sort the result-set in ascending or descending order
2. The ORDER BY keyword sorts the records in ascending order by default.
3. To sort the records in ascending order, use the ASC keyword.
4. To sort the records in descending order, use the DECENDING keyword.

Solution:

---

17. Find cities in the increasing order of temperature. (Select two)
Options:

A. SELECT city FROM weather ORDER BY temperature;
B. SELECT city, temperature FROM weather;
C. SELECT city, temperature FROM weather ORDER BY temperature;
D. SELECT city, temperature FROM weather ORDER BY city;

Solution:

---

18. With SQL, how can you return all the records from a table named "Persons" sorted descending by "FirstName" ?
Options:

a) SELECT * FROM Persons SORT BY 'FirstName' DESC
b) SELECT * FROM Persons ORDER FirstName DESC
c) SELECT * FROM Persons SORT 'FirstName' DESC
d) SELECT * FROM Persons ORDER BY FirstName DESC

Solution: