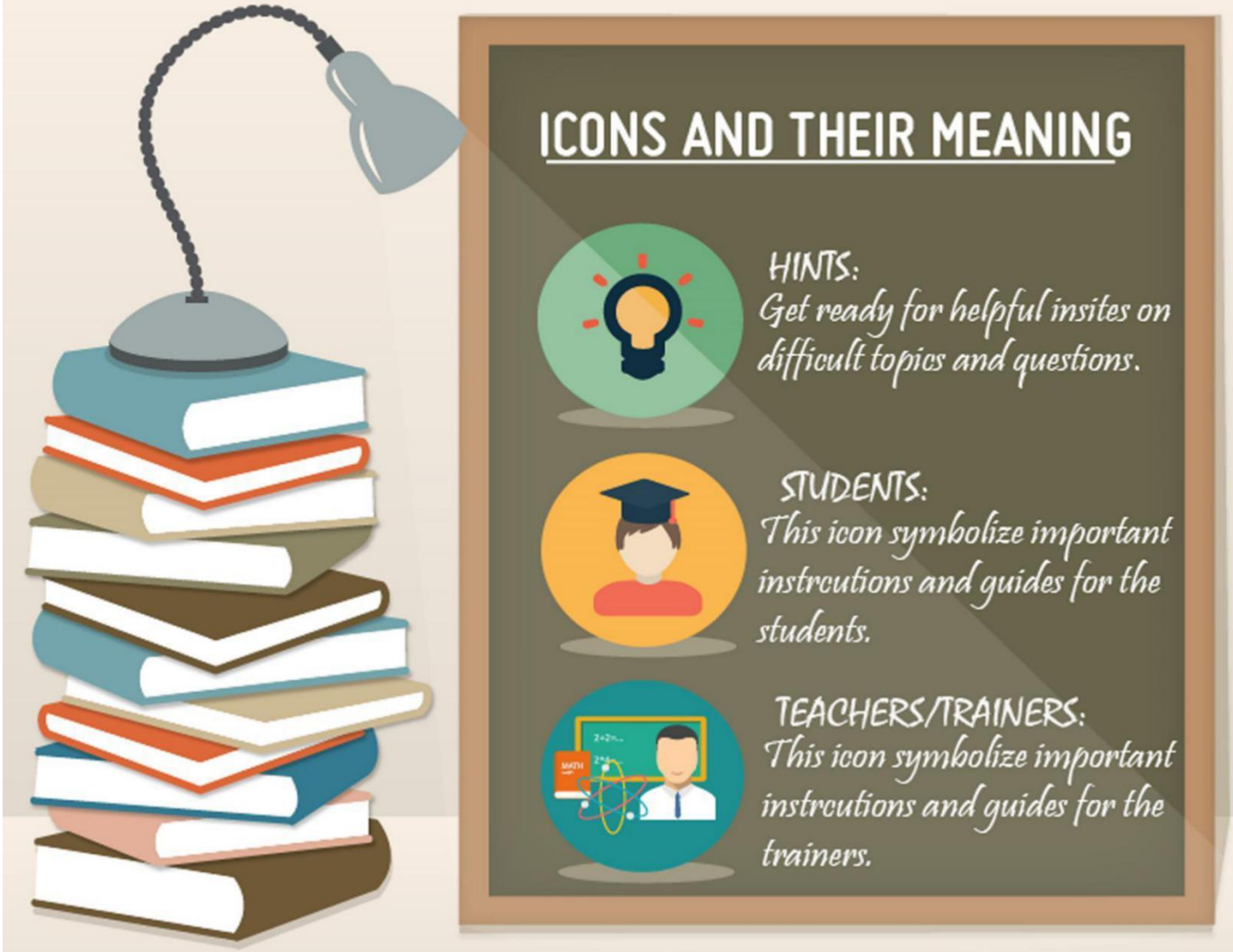
MySQL

MANUAL V8.3

MODULE CODE:

ANUDIP FOUNDATION


ICONS AND THEIR MEANING
HINTS:
Get ready for helpful insites on difficult topics and questions.
STUDENTS:
This icon symbolize important instrcutions and guides for the students.
TEACHERS/TRAINERS:
This icon symbolize important instrcutions and guides for the trainers.

Use of Where Clause

Objective: After completing this lesson you will be able to : * Gain an understanding of how to use the WHERE clause in MySQL * Utilize And and Or operators	**Materials Required:** 1. Computer 2. Internet access
Theory Duration: 120 minutes	**Practical Duration:** 0 minute
Total Duration: 120 minutes	

Chapter 22

Where clause

The WHERE clause in MySQL is used for filtering results. It enables users to define selection criteria for selecting a table's records.

Syntax

The code below has a SELECT command syntax with the WHERE clause for fetching the MySQL table's data -

SELECT field1, field2,...fieldN table_name1, table_name2...

[WHERE condition1 [AND [OR]] condition2.....

- One or more comma-separated tables can be used for including different conditions utilizing a WHERE clause. The WHERE clause is an optional component of the SELECT statement.
- Any condition using the WHERE clause can be specified by a user.
- The AND or OR operators are utilized for specifying multiple conditions.
- Using a WHERE clause with SQL's DELETE or UPDATE commands can help in specifying a condition.

The WHERE clause functions similar to an if condition. This clause can help to compare a value with available field values of a MySQL table. A row is returned if the given value matches the MySQL table's available field value.

Take a look at the operators which can be utilized with the WHERE clause.

Here the values of A and B are 20 and 30 respectively -

Operator	Description	Example
=	Used for verifying if two operand values are equal or not. Condition becomes true if yes.	(A = B) is not true.
!=	Checks whether two operand values are equal or not. Condition is true if values are not equal.	(A != B) is true.
>	Checks if the left operand value is greater than right operand value. Condition becomes true if yes.	(A > B) is not true.

<	Checks if the left operand value is less than the right operand value. Condition becomes true if yes.	(A < B) is true.
>=	Checks if the left operand value is greater than or equal to the right operand value. Condition becomes true if yes.	(A >= B) is not true.
<=	Checks if the left operand value is less than or equal to the right operand value. Condition becomes true if yes.	(A <= B) is true.

The WHERE clause is preferred for fetching selected table rows, especially in cases where the MySQL Join is being used.

Users generally search for records with a Primary Key for getting results faster.

If a condition does not match a record, no row is returned.

Retrieving Data from the Command Prompt

In the example given below, the SQL SELECT command can be used with the WHERE clause for retrieving chosen data from the table named tutorials_tbl.

Example -

The example provided returns all tutorials_tbl table records which have the author name John.

root@host# mysql -u root -p password;

Enter password:*******

mysql> use TUTORIALS;

Database changed

mysql> SELECT * from tutorials_tbl WHERE tutorial_author = 'John';

+-------------+---------------+-----------------+-----------------+

| tutorial_id | tutorial_title | tutorial_author | submission_date |

+-------------+---------------+-----------------+-----------------+

```
|       3       | HTML Tutorial  |       John       |     2009-07-15     |

+-------------+----------------+----------------+-----------------+

1 rows in set (0.01 sec)

mysql>
```

A comparison is not case sensitive unless a LIKE comparison is performed on a string. The BINARY keyword can be utilized for making the search case sensitive. Example query -

```
root@host# mysql -u root -p password;

Enter password:********

mysql> use TUTORIALS;

Database changed

mysql> SELECT * from tutorials_tbl \

   WHERE BINARY tutorial_author = 'sanjay';

Empty set (0.02 sec)

 mysql>
```

Utilizing a PHP Script for Fetching Data

The SQL SELECT command can be used with the WHERE CLAUSE in the mysql_query() PHP function. This function helps in executing the SQL command and the mysql_fetch_array() PHP function for retrieving all selected data. The function returns a row as a numeric array, associative array, or both. FALSE is returned if there are no more rows.

Example -

The example below returns all the tutorials_tbl table records for which the author name is John -

```
<?php

   $dbhost = 'localhost:3036';

   $dbuser = 'root';

   $dbpass = 'rootpassword';



   $conn = mysql_connect($dbhost, $dbuser, $dbpass);

    if(! $conn ) {
```

```
      die('Could not connect: ' . mysql_error());
   }
 $sql = 'SELECT tutorial_id, tutorial_title,
      tutorial_author, submission_date
      FROM tutorials_tbl
      WHERE tutorial_author = "John"';
    mysql_select_db('TUTORIALS');
   $retval = mysql_query( $sql, $conn );
    if(! $retval ) {
      die('Could not get data: ' . mysql_error());
   }
    while($row = mysql_fetch_array($retval, MYSQL_ASSOC)) {
      echo "Tutorial ID :{$row['tutorial_id']}  <br> ".
      "Title: {$row['tutorial_title']} <br> ".
      "Author: {$row['tutorial_author']} <br> ".
      "Submission Date : {$row['submission_date']} <br> ".
      "--------------------------------<br>";
   }
    echo "Data has been fetched successfully\n";
   mysql_close($conn);
?>
```

And Operator

MySQL AND operator

The AND operator is a MySQL operator for comparing two expressions. If both expressions are true the operator returns true.

Syntax:

AND, &&

The operator returns 1 in a case where all operands are not NULL or nonzero. It returns 0 if one or multiple operands are 0. A NULL value is returned otherwise.

Example - MySQL AND operator

The MySQL statement given below has all non-zero operands (6 and 6), both of which are not NULL. Hence, the logical AND operator returns a value of 1.

Code -

```
SELECT 6 && 6;
```

Sample Output:

```
MySQL> SELECT 6 && 6;
+--------+
| 6 && 6 |
+--------+
|      1 |
+--------+
1 row in set (0.00 sec)
```

MySQL AND operator with zero input Example

The MySQL statement given below has one non-zero and one zero operator (6 and 0). Hence, the logical AND operator returns the value of 0.

Code -

```
SELECT 6 && 0;
```

Sample Output:

```
MySQL> SELECT 6 && 0;
+--------+
| 6 && 0 |
```

```
+--------+
|      0 |
+--------+
```

1 row in set (0.00 sec)

MySQL AND operator with NULL input example -

The MySQL statement given below has one non-zero operator (6) and one NULL operator. Hence, the logical AND operator returns the NULL value.

Code -

```
SELECT 6 && NULL;
```

Sample Output:

```
MySQL> SELECT 6 && NULL;
+-----------+
| 6 && NULL |
+-----------+
|      NULL |
+-----------+
```

1 row in set (0.00 sec)

MySQL AND operator with zero and NULL input Example

The MySQL statement given below has one zero operand (0) and another NULL operand. Hence, the logical AND operator returns 0 as the value.

Code:

```
SELECT 0 && NULL;
```

Sample Output:

```
MySQL> SELECT 0 && NULL;

+-----------+
| 0 && NULL |
+-----------+
```

```
|           0 |
+-----------+
```

1 row in set (0.00 sec)

MySQL AND operator with both NULL inputs Example

The MySQL statement given below has two NULL operands. Hence, the operator returns NULL.

Code:

```
SELECT NULL && NULL;
```

Sample Output:

```
MySQL> SELECT NULL && NULL;
+--------------+
| NULL && NULL |
+--------------+
|         NULL |
+--------------+
```

1 row in set (0.00 sec)

Or Operator

MySQL OR operator

The MySQL OR operator is used for comparing two expressions. It returns TRUE if any of the two given expressions evaluate to TRUE.

Syntax:

OR, ||

If a statement has more than one logical operator, OR operators perform after the AND operator. Using parentheses lets a user modify the evaluation order of the AND and OR operators.

The operator returns a value of 1 if both the operands are non-NULL and one of them is also non-zero.

The operator returns a value of 0 when the two operands are non-NULL and one is zero.

The operator returns NULL of one of the operands is NULL while the other one is zero. It returns 1 when one operand is NULL and the other is non-zero.

The operator returns NULL when both of the operands are NULL.

Example: MySQL OR operator

The MySQL statement given below meets the condition that both operands are non-NULL and one among them is non-zero. Hence, a value of 1 is returned.

Code:

```
SELECT 6 || 6;
```

Sample Output:

```
MySQL> SELECT 6 || 6;
        +--------+
        | 6 || 6 |
        +--------+
        |      1 |
        +--------+
1 row in set (0.00 sec)
```

MySQL OR operator with at least one (zero) 0 example

The MySQL statement given below meets the condition that both operands are non-NULL, with one of them being zero. Hence, a value of 1 is returned.

Code:

```
SELECT 6 || 0;
```

Sample Output:

```
MySQL> SELECT 5 || 0;
        +--------+
        | 5 || 0 |
        +--------+

        |      1 |
        +--------+
1 row in set (0.00 sec)
```

MySQL OR operator example when both operands are zero(0)

In the MySQL statement given below, both operands are 0. Hence, the value of 0 is returned.

Code:

```
SELECT 0 || 0;
```

Sample Output:

```
MySQL> SELECT 0 || 0;
        +--------+
        | 0 || 0 |
        +--------+
        |      0 |
        +--------+
1 row in set (0.00 sec)
```

MySQL OR operator example with at least one NULL operand

The MySQL statement given below meets the condition that one operand is NULL and another is zero. Hence, a NULL value is returned.

SELECT 0 || NULL;

Sample Output:

```
MySQL> SELECT 0 || NULL;
        +-----------+
        | 0 || NULL |
        +-----------+
        |      NULL |
        +-----------+
1 row in set (0.01 sec)
```

MySQL OR operator example with NULL and non-zero operand

The MySQL statement given below meets the condition that one operand is NULL and the other is non-zero. Hence, the value 1 is returned.

Code:

```
SELECT 6 || NULL;
```

Sample Output:

```
MySQL> SELECT 6 || NULL;
                +-----------+
                | 6 || NULL |
                +-----------+
                |          1 |
                +-----------+
1 row in set (0.00 sec)
```

MySQL OR operator example with two NULL operands

The MySQL statement given below has two NULL operands. As a result, the NULL value is returned.

Code:

```
SELECT NULL || NULL;
```

Sample Output:

```
MySQL> SELECT NULL || NULL;
                +--------------+
                | NULL || NULL |
                +--------------+
                |          NULL |
                +--------------+
1 row in set (0.00 sec)
```

Instructions: The progress of students will be assessed with the exercises mentioned below.

MCQ

1. MySQL table records selection criteria can be defined with the _____ clause

a) SELECT

b) WHERE

c) SORT

d) none of the mentioned

2. The WHERE clause works similar to _____ condition.

a) else

b) if

c) elseif

d) none of the mentioned

3. What is returned if a given value matches the available field in a MySQL table?

a) column

b) row

c) operator

d) none of the mentioned

4. What does the != operator do?

a) verifying if operands are equal or not

b) verifying if operands exist

c) verifying if the left operand larger than the right operand

d) none of the mentioned

5. Which keyword can be used for making a search case sensitive?

a) VARBINARY

b) BINARY

c) VARCHAR

d) none of the mentioned

6. What SQL command is used with the WHERE clause in the mysql_query() PHP function?

a) DROP

b) SELECT

c) CREATE

d) none of the mentioned

7. Which MySQL operator is used for comparing two expressions?

a) AND

b) &&

c) both a and b

d) none of the mentioned

8. What can be used for modifying the evaluation order of the AND and OR operators?

a) commas

b) parentheses

c) periods

d) none of the mentioned

9. If one operand is non-NULL and another is _____ the OR operator returns 1.

a) zero

b) non-zero

c) both a and b

d) none of the mentioned

10. If both operands are NULL _____ returned

a) two NULLs are

b) one NULL is

c) a non-zero is

d) none of the mentioned