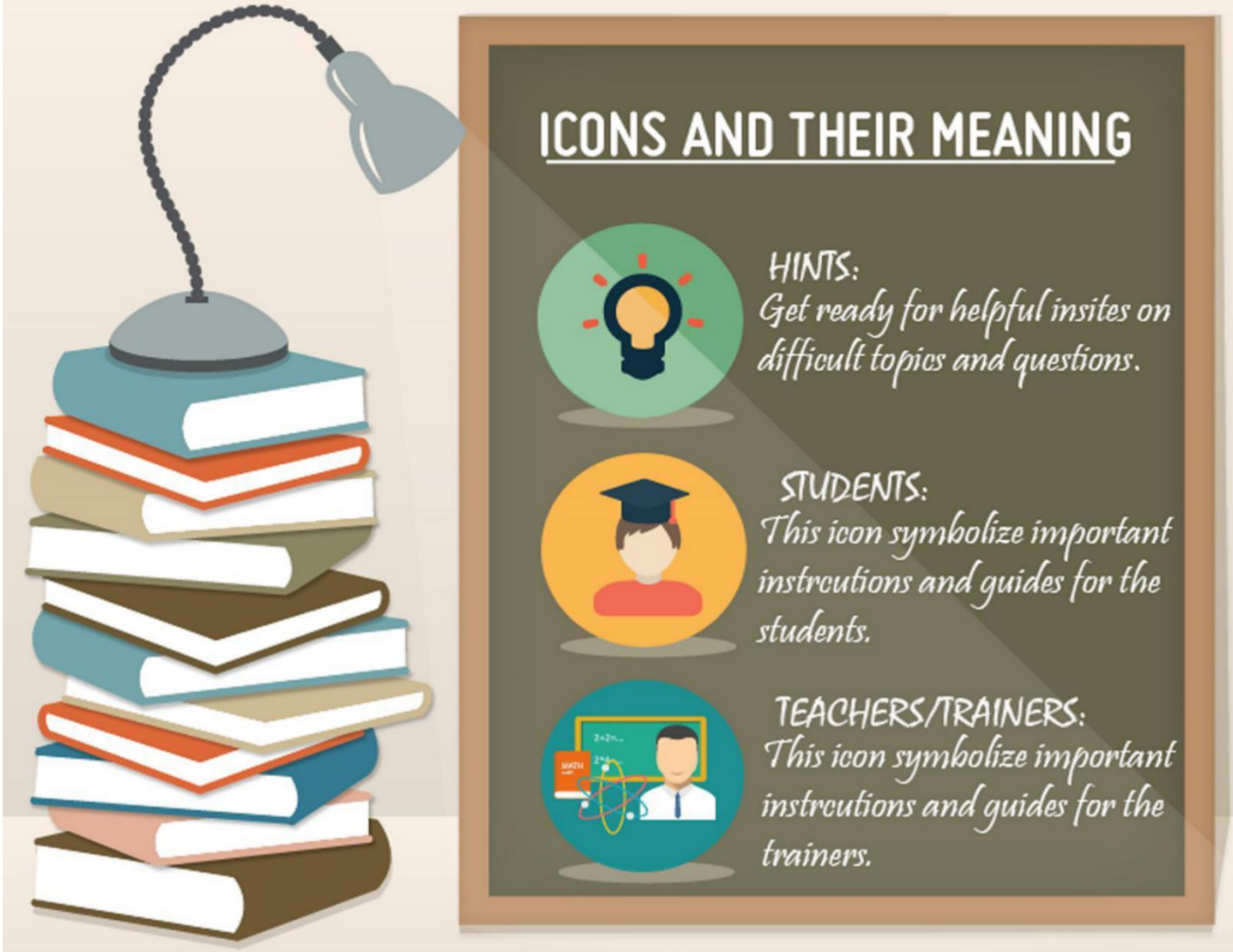
MySQL

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.

Joining Tables

Objective: After completing this lesson you will be able to : * Gain an understanding of MySql Join, INNER JOIN, LEFT JOIN and RIGHT JOIN	**Materials Required:** 1. Computer 2. Internet access
Theory Duration: 120 minutes	**Practical Duration:** 0 minute
Total Duration: 120 minutes	

Chapter 30

MySQL JOINS

MySQL JOINS are used as an accompaniment to a SELECT statement. It can fetch data from multiple tables and is used when there is a need for fetching records from multiple tables.

Three main types of MySQL joins:

* MySQL INNER JOIN (or simple join)
* MySQL LEFT OUTER JOIN (or LEFT JOIN)
* MySQL RIGHT OUTER JOIN (or RIGHT JOIN)

MySQL Inner Join

MySQL Inner Join solely returns results from tables matching a given condition, and hiding other columns and rows. Inner Join is regarded as the default Join. Hence, using the Inner Join keyword may be considered optional.

Syntax -

The Inner JOIN keyword is used along with a SELECT statement, and has to be written following the FROM clause. Example -

```
SELECT columns
FROM table1
INNER JOIN table2 ON condition1
INNER JOIN table3 ON condition2
...;
```

The first step is selecting the column list, followed by specifying the name of the table to be joined with the main table. Then the user has to specify the ON keyword condition. The Join condition outputs matching rows of the two tables as mentioned in the Inner clause.

MySQL Inner Join Example

Creating two tables named 'students' and 'technologies' -

Table - student

student_id	stud_fname	stud_lname	city
1	Devine	Putin	France
2	Michael	Clark	Australiya
3	Ethon	Miller	England
4	Mark	Strauss	America

Table - technologies

student_id	tech_id	inst_name	technology
1	1	Java Training Inst	Java
2	2	Chroma Campus	Angular
3	3	CETPA Infotech	Big Data
4	4	Aptron	IOS

The query below can be used for selecting records from both tables -

```
SELECT students.stud_fname, students.stud_lname, students.city, technologies.technology
FROM students
INNER JOIN technologies
ON students.student_id = technologies.tech_id;
```

Output - A table with the stud_fname, stud_lname, city, and technology with their data.

MySQL Inner Join with Group By Clause

Inner Join is used by the GROUP BY clause. The statement below returns several columns with the Inner Join and GROUP BY clause -

```
SELECT students.student_id, technologies.inst_name, students.city, technologies.technology
FROM students
INNER JOIN technologies
ON students.student_id = technologies.tech_id GROUP BY inst_name;
```

Output - A table with student_id, inst_name, city and technology columns and their data.

MySQL Inner Join with USING clause

Columns can have identical names in two tables. The USING keyword can be utilized for fetching the records. Example query -

```
SELECT student_id, inst_name, city, technology
FROM students
INNER JOIN technologies
USING (student_id);
```

Output - A table with student_id, inst_name, city and technology columns and their data.

Inner Join with WHERE Clause

The WHERE clause is used for returning the filter result. Example -

```
SELECT tech_id, inst_name, city, technology
FROM students
INNER JOIN technologies
USING (student_id) WHERE technology = "Java";
```

Output - A table with the tech-id, inst_name, city and technology columns showing the record where technology is Java.

MySQL Inner Join Multiple Tables

The two tables created earlier exist. A new table named contact is created.

college_id	cellphone	homephone
1	5465465645	4576787687
2	4987246464	5795645568
3	8907654334	8654784126
4	8973454904	9865321475

Statement for joining the three tables -

```
SELECT student_id, inst_name, city, technology, cellphone
FROM students
INNER JOIN technologies USING (student_id)
INNER JOIN contact ORDER BY student_id;
```

Output - A table with the student_id, inst_name, city, technology and cellphone columns with their data.

MySQL Inner Join using Operators

MySQL users can utilize many operators with Inner Join such as less than (<), greater than (>), equal (=), and not equal (!=). Example query -

```
SELECT emp_id, designation, income, qualification
FROM employee
INNER JOIN customer
WHERE income>25000 and   income<78000;
```

Output - Table listing all employees with income between 25000 and 78000. It displays the columns emp_id, designation, income and qualification.

MySQL LEFT JOIN

MySQL Left Join is utilized for querying data across multiple tables. It is similar to the Inner Join clause, and can be utilized with the SELECT statement following the FROM keyword.

The Left Join clause returns all records from the first table (left table), even if no records match in the second (right side) table. NULL is returned if there are no matches in the right side table. The Left Join is also known as the Left Outer Join clause.

MySQL LEFT JOIN Syntax

Left Join clause for joining the two or more tables -

```
SELECT columns
FROM table1
LEFT [OUTER] JOIN table2
ON Join_Condition;
```

Here, table1 is the left-hand table and the right-hand table is table2. The LEFT JOIN clause returns all table1 records and matching records from table 2, depending on the given join condition.

MySQL LEFT JOIN Example

Joining two tables with the LEFT JOIN clause

The two sample tables below are joined by the LEFT JOIN clause -

Table: customers

customer_id	cust_name	occupation	income	qualification
1	John Miller	Developer	20000	Btech
2	Mark Robert	Enginneer	40000	Btech
3	Reyan Watson	Scientists	60000	MSc
4	Shane Trump	Businessman	10000	MBA
5	Adam Obama	Manager	80000	MBA
6	Rincky Ponting	Cricketer	200000	Btech

Table: orders

order_id	date	customer_id	price
1001	2020-03-20	2	3000
1002	2020-02-15	4	2500
1003	2020-01-31	5	5000
1004	2020-03-10	2	1500
1005	2020-02-20	1	4500

To select records from both tables, execute the following query:

```
SELECT customers.customer_id, cust_name, price, date
FROM customers
LEFT JOIN orders ON customers.customer_id = orders.customer_id;
```

Output - A table with the customer_id, cust_name, price and date columns with their respective data.

MySQL LEFT JOIN with USING Clause

The table customers and orders have the same column name, which is customer_id. In that case, MySQL Left Join can also be used with the USING clause to access the records. The following statement returns customer id, customer name, occupation, price, and date using the Left Join clause with the USING keyword.

```
SELECT customer_id, cust_name, occupation, price, date
FROM customers
LEFT JOIN orders USING(customer_id);
```

Output - A table with customer_id, cust_name, occupation, price and date columns and their data.

MySQL LEFT JOIN with Group By Clause

MySQL lets users utilize Left Join with the GROUP BY clause. Using the customers and orders sample tables above, the statement below is made to return customer name, customer id, price, qualification, and date with the combination of the Left Join and GROUP BY clauses.

```
SELECT customers.customer_id, cust_name, qualification, price, date
```

```
FROM customers
LEFT JOIN orders ON customers.customer_id = orders.customer_id
GROUP BY price;
```

Output - A table with customer_id, cust_name, qualification, price and date columns and their data.

LEFT JOIN with WHERE Clause

The WHERE clause returns the table's filter result. Example of Left Join with the WHERE clause -

```
SELECT customer_id, cust_name, occupation, price, date
FROM customers
LEFT JOIN orders
USING(customer_id) WHERE price>3000;
```

Output - A table with customer_id, cust_name, occupation, price and date columns and their data.

MySQL LEFT JOIN Multiple Tables

The customers and orders tables already exist. Now, the user attempts to create a new table named 'contacts' which has its own data.

contact_id	cellphone	homephone
1	6546645978	4565242557
2	8798634532	8652413954
3	8790744345	9874437396
4	7655654336	9934345363

The statement below is written for joining the customers, orders and contacts tables -

```
SELECT customers.customer_id, cust_name, order_id, price, cellphone
FROM customers
LEFT JOIN contacts ON customer_id = contact_id
LEFT JOIN orders ON customers.customer_id = orders.customer_id ORDER BY income;
```

Output - A table with the customer_id, cust_name, order_id, price and cellphone columns with their respective data.

Using LEFT JOIN clause for unmatched records

The LEFT JOIN clause can be used for fetching records without any matching data rows from another table.

In the example below, the LEFT JOIN clause is used for finding a custom without a cellphone number -

```
SELECT customer_id, cust_name, cellphone, homephone
FROM customers
LEFT JOIN contacts ON customer_id = contact_id
WHERE cellphone IS NULL ;
```

Output - A table with the customer_id, cust_name, cellphone and homephone columns. The cellphone column has NULL values as the customer does not have any a cellphone number.

MySQL RIGHT JOIN

MySQL Right Join joins multiple tables to return all right-hand table rows. It only returns the results that comply to the Join condition. Unmatched records are returned as NULL. Right Join is also known as the Right Outer Join.

RIGHT JOIN Syntax

Right Join syntax for joining the tables Table1 and Table2 -

```
SELECT column_list
FROM Table1
RIGHT [OUTER] JOIN Table2
ON join_condition;
```

MySQL RIGHT JOIN Examples

Joining two tables with the RIGHT JOIN clause
The customers and orders tables are used for showing examples -

Table - customers

customer_id	cust_name	occupation	income	qualification
1	John Miller	Developer	20000	Btech
2	Mark Robert	Enginneer	40000	Btech
3	Reyan Watson	Scientists	60000	MSc
4	Shane Trump	Businessman	10000	MBA
5	Adam Obama	Manager	80000	MBA
6	Rincky Ponting	Cricketer	200000	Btech

Table - orders

order_id	date	customer_id	price
1001	2020-03-20	2	3000
1002	2020-02-15	4	2500
1003	2020-01-31	5	5000
1004	2020-03-10	2	1500
1005	2020-02-20	1	4500

The statement below can be used for selecting records from both the customers and orders tables with RIGHT JOIN -

```
SELECT customers.customer_id, cust_name, price, date
FROM customers
RIGHT JOIN orders ON customers.customer_id = orders.customer_id
ORDER BY  customer_id;
```

OR,

```
SELECT customers.customer_id, cust_name, price, date
FROM customers
RIGHT JOIN orders USING(customer_id)
```

```
ORDER BY  customer_id;
```

Output - A table with the customer_id, cust_name, price and date columns, with the data for each column. It is sorted by the cusutomer_id.

RIGHT JOIN with WHERE Clause

The WHERE clause of MySQL filters table results. Using it with the Right Join clause -

```
SELECT * FROM customers
RIGHT JOIN orders USING(customer_id)
WHERE price>2000 AND price<4500;
```

Output - A table with the customer_id, order_id, date, price, cust_name, occupation, income and qualification columns. Only the records where price is between 2000 and 4500 are shown in this table.

MySQL RIGHT JOIN Multiple Tables

The customers and orders tables already exist. Another table named contacts is created with its own data.

contact_id	cellphone	homephone
1	6546645978	4565242557
2	8798634532	8652413954
3	8790744345	9874437396
4	7655654336	9934345363

The statement below joins the three tables -

```
SELECT customers.customer_id, cust_name, order_id, price, cellphone
FROM customers
RIGHT JOIN contacts ON customer_id = contact_id
RIGHT JOIN orders ON customers.customer_id = orders.customer_id ORDER BY order_id;
```

After successful execution of the above query, it will give the following output:

Output - A table with the customer_id, cust_name, order_id, price and cellphone columns, each with its respective data.

Using RIGHT JOIN clause for fetching unmatched records

The Right Join clause can be used for fetching records that do not contain matching rows from a different table.

In this example, the RIGHT JOIN clause is used for finding a customer without a cellphone number -

```
SELECT customer_id, cust_name, cellphone, homephone
FROM customers
RIGHT JOIN contacts ON customer_id = contact_id
WHERE cellphone IS NULL
ORDER BY cellphone;
```

Output - A table with customer_id, cust_name, cellphone and homephone columns, with each having their respective data. The cellphone column has NULL values as the customers do not have cellphones.

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

MCQ

1. MySQL JOINS are used for accompanying _____ statements.

a) SELECT

b) WHERE

c) SORT

d) none of the mentioned

2. Which of these are not MySQL JOIN types?

a) RIGHT OUTER JOIN

b) LEFT JOIN

c) LEFT OUTER JOIN

d) none of the mentioned

3. What clause has to be used before the Inner Join Keyword?

a) WHERE

b) FROM

c) HAVING

d) none of the mentioned

4. Which is the default JOIN in MySQL?

a) RIGHT JOIN

b) LEFT JOIN

c) INNER JOIN

d) none of the mentioned

5. Which of these clauses can MySQL inner JOIN not be used with?

a) GROUP BY

b) USING

c) WHERE

d) none of the mentioned

6. Which of these operators cannot be utilized by MySQL users?

a) >

b) =

c) <

d) none of the mentioned

7. The LEFT JOIN clause returns records from the

a) first table

b) left table

c) both a and b

d) none of the mentioned

8. What is returned if a user is using LEFT JOIN finds no match in the right side table?

a) NULL

b) 0

c) ERROR

d) none of the mentioned

9. MySQL RIGHT JOIN joins multiple tables for returning _____.

a) last right hand row

b) all right hand rows

c) first right hand row

d) none of the mentioned

10. Right Join is also known as the _____.

a) RIGHT OUTER JOIN

b) RIGHT INNER JOIN

c) both a and b

d) none of the mentioned