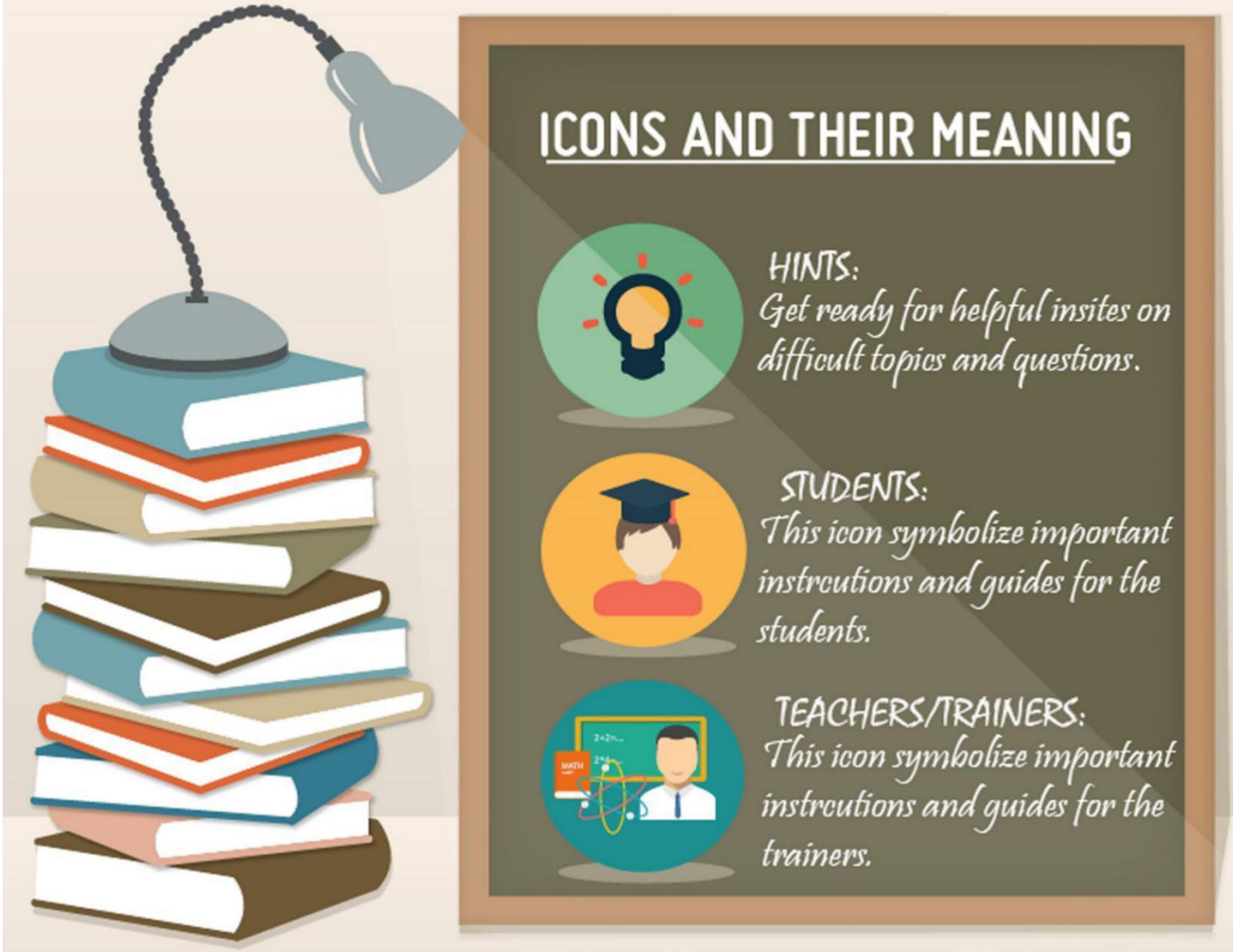
MySQL

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.

MySQL CROSS JOIN

Objective: After completing this lesson you will be able to : * Gain an understanding of MySQL Cross Join and Self-Join	**Materials Required:** 1. Computer 2. Internet access
Theory Duration: 90 minutes	**Practical Duration:** 0 minute
Total Duration: 90 minutes	

Chapter 31

MySQL CROSS JOIN

MySQL CROSS JOIN combines every possibility of multiple tables and returns a result with every row from all tables involved. The CROSS JOIN or CARTESIAN JOIN produces the CARTESIAN product of all tables. It can be referred to as the rows in a table multiplied by rows in a second table.

Each row returned is a combination of rows from both the tables.

MySQL CROSS JOIN Syntax

The CROSS JOIN keyword is used with the SELECT statement, and is written after mentioning the FROM clause. The syntax below retrieves records from the two joined tables -

```
SELECT column-lists
FROM table1
CROSS JOIN table2;
```

In the syntax, the name of the column or the files to be returned is 'column-lists'. The records are being fetched from table1 and table2.

MySQL CROSS JOIN Example

Joining two tables with the MySQL CROSS JOIN clause -

Two tables named customers and contacts are created for this example.

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: contacts

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

Query for fetching all table records -

```
SELECT *
FROM customers

CROSS JOIN contacts;
```

Output - A table with all the columns of table1 and table2 is created.

After CROSS JOIN, the table has n number of rows, where n is the value obtained by multiplying the rows of table1 and table2.

Ambiguous Columns Issue in MySQL CROSS JOIN

A MySQL user may need to fetch selected records from multiple tables, which can have similar column names. If that happens an error can be thrown by MySQL CROSS JOIN - error: the column name is ambiguous. The error is shown as the column exists across both tables.

The issue can be solved by mentioning the table name before a column name. Query -

```
SELECT customer.customer_id, customer.cust_name, customer.income, orders.order_id,
orders.price
FROM customer
CROSS JOIN orders;
```

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

LEFT JOIN with WHERE Clause

The MySQL WHERE clause returns a filter result from a table. Example -

```
SELECT customers.customer_id, customers.cust_name, customers.income, orders.order_id,
orders.price
FROM customers
CROSS JOIN orders
USING(customer_id) WHERE price>2000 AND price<4800;
```

Output - A table with customer_id, cust_name, income, order_id and price columns and their respective data. Only records with a price range of 2000-4800 are showcased.

MySQL CROSS JOIN Multiple Tables

Two tables named customers and orders are created earlier in the chapter. The contacts table is created.

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

The use of CROSS JOIN with LEFT JOIN is shown with the three tables. The statement below is joining the orders, customers, and contacts tables. The statement first CROSS joins the Orders and Contacts tables. The LEFT JOIN is executed based on the mentioned condition.

```
SELECT * FROM customer


LEFT JOIN(orders CROSS JOIN contacts)
```

```
ON customer.customer_id=contact_id
ORDER BY income;
```

Output - A table with all the columns of the three tables is outputted. See example image -

customer_id	cust_name	occupation	income	qualification	order_id	date	customer_id	price	contact_id	cellphone	homephone
4	Shane Trump	Business...	10000	MBA	1001	2020-03-20	2	3000	4	7655654336	9934345363
4	Shane Trump	Business...	10000	MBA	1002	2020-02-15	4	2500	4	7655654336	9934345363
4	Shane Trump	Business...	10000	MBA	1003	2020-01-31	5	5000	4	7655654336	9934345363
4	Shane Trump	Business...	10000	MBA	1004	2020-03-10	2	1500	4	7655654336	9934345363
4	Shane Trump	Business...	10000	MBA	1005	2020-02-20	1	4500	4	7655654336	9934345363
1	John Miller	Developer	20000	Btech	1001	2020-03-20	2	3000	1	6546645978	4565242557
1	John Miller	Developer	20000	Btech	1002	2020-02-15	4	2500	1	6546645978	4565242557
1	John Miller	Developer	20000	Btech	1003	2020-01-31	5	5000	1	6546645978	4565242557
1	John Miller	Developer	20000	Btech	1004	2020-03-10	2	1500	1	6546645978	4565242557
1	John Miller	Developer	20000	Btech	1005	2020-02-20	1	4500	1	6546645978	4565242557
6	Rincky Pon...	Cricketer	200000	Btech	1001	2020-03-20	2	3000	6	NULL	9086053684
6	Rincky Pon...	Cricketer	200000	Btech	1002	2020-02-15	4	2500	6	NULL	9086053684

MySQL SELF JOIN

A SELF JOIN is used for joining a table with itself. It is used for combining data with other data within the same table.

Table aliases can be used for performing Self Join. Aliases prevent users from utilizing the same table name again with a single statement. An error is thrown when the same table name is used twice or more in a single query without table aliases.

Table aliases enable users to use a temporary name for a table that will be used in a query. See the example below -

The 'student' table has to be used twice within the single query. Aliasing the student table -

```
Select … FROM student AS S1
INNER JOIN student AS S2;
```

SELF JOIN Syntax

The SELF JOIN syntax is the same as the syntax used for joining two separate tables. The table aliases names are used for tables since the table names are identical. SELF JOIN syntax -

```
SELECT s1.col_name, s2.col_name...
FROM table1 s1, table1 s2
WHERE s1.common_col_name = s2.common_col_name;
```

SELF JOIN Example

For this example, a table named student is used -

student_id	name	course_id	duration
1	Adam	1	3
2	Peter	2	4
1	Adam	2	4
3	Brian	3	2
2	Shane	3	5

The user attempts to get the student_id and name from a table where the student_id is equal but course_id is not. The query is executed -

```
SELECT  s1.student_id, s1.name
FROM student AS s1, student s2
WHERE s1.student_id=s2.student_id
AND s1.course_id<>s2.course_id;
```

Output - A table with the student_id and name. Example image -

student_id	name
1	Adam
2	Shane
1	Adam
2	Peter

SELF JOIN using INNER JOIN clause

The example below shows how Inner Join and Self Join can be used together. The query returns the name and student id if both the tables have equal student_id, but the course_id are not equal.

```
SELECT  s1.student_id, s1.name
FROM student s1
INNER JOIN student s2
ON s1.student_id=s2.student_id
AND s1.course_id<>s2.course_id
GROUP BY student_id;
```

Output - A table with the names of student_id 1 and 2.

SELF JOIN using LEFT JOIN clause

The example shows how LEFT Join is used with Self Join. The query below exhibits the student name as monitor and city when both tables have equal student_id values.

```
SELECT (CONCAT(s1.stud_lname, ' ', s2.stud_fname)) AS 'Monitor', s1.city
FROM students s1
LEFT JOIN students s2 ON s1.student_id=s2.student_id
ORDER BY s1.city DESC;
```

Output - A table with the monitor and city columns and data for each.

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

MCQ

1. Each row returned by MySQL CROSS JOIN is a combination of _____.

a) columns from both tables

b) rows from both tables

c) both tables

d) none of the mentioned

2. What clause has to be used before the Cross Join Keyword?

a) WHERE

b) FROM

c) HAVING

d) none of the mentioned

3. After cross join the resulting table has n number of rows, where n is obtained by _____.

a) adding table1 and table2 rows

b) multiplying table1 and table2 rows

c) dividing the sum of table1 and table2 rows by 2

d) none of the mentioned

4. MySQL JOIN can throw an error that column name is ambiguous when _____.

a) column exists in none of the tables

b) both tables have similar column names

c) column names cannot be found

d) none of the mentioned

5. During SELF JOIN an error is thrown when a table name is used multiple times in _____ without aliases.

a) a single query

b) two different queries

c) two identical queries

d) none of the mentioned