Practical MySQL

Session 4

Joins in MySQL

Session Overview

- Define JOINS
- Explain the use of JOINS
- Identify different types of JOINS
- Explain the use of SET operators
- Outline the differences between JOINS and UNIONS

Introduction to JOINS

MySQL JOIN is a feature used to combine and retrieve data from more than one table.

The JOIN clause retrieves the data by joining two tables and creating a relationship between them.

A JOIN operation returns the values (rows) whose data matches both tables in that column(s).

How to Use Joins?

CustomerID	FirstName	LastName	Country	Age	Gender
214431	Jack	Swanson	Germany	34	M
956437	Joe	Voight	France	22	F
111278	Christian	Gale	Italy	19	M
897867	Karl	Davis	Switzerland	22	M
789023	Berry	Chase	Netherlands	24	F

Customer Table

OrderID	CustomerID	Status	Amount
567834	214431	Paid	10000
113425	956437	Unpaid	6780
564721	564721	Unpaid	12000
999982	789023	Paid	4500
453110	897867	Paid	2300

Orders Table

JOIN Clause

Example for JOIN Clause

```
SELECT *
FROM Customer JOIN Orders
ON Customer.CustomerID = Orders.CustomerID;
```

Output

```
mysql> SELECT *
    -> FROM Customer JOIN Orders
    -> ON Customer.CustomerID = Orders.CustomerID;
+------------+-----------+----------+-------------+------+--------+---------+------------+--------+--------+
| CustomerID | FirstName | LastName | Country     | Age  | Gender | OrderID | CustomerID | Status | amount |
+------------+-----------+----------+-------------+------+--------+---------+------------+--------+--------+
|     214431 | Jack      | Swanson  | Germany     |   34 | M      |  567834 | 214431     | Paid   |  10000 |
|     956437 | Joe       | Voight   | France      |   22 | F      |  113425 | 956437     | Unpaid |   6780 |
|     789023 | Berry     | Chase    | Netherlands |   24 | F      |  999982 | 789023     | Paid   |   4500 |
|     897867 | Karl      | Davis    | Switzerland |   22 | M      |  453110 | 897867     | Paid   |   2300 |
+------------+-----------+----------+-------------+------+--------+---------+------------+--------+--------+
4 rows in set (0.15 sec)
```

Why to Use JOINS?

JOINS are used to:

- Filter data from relational tables
- Combine data from two or more tables and filter it based on the conditions specified by the user
- Reduces duplicate records in the combinational result

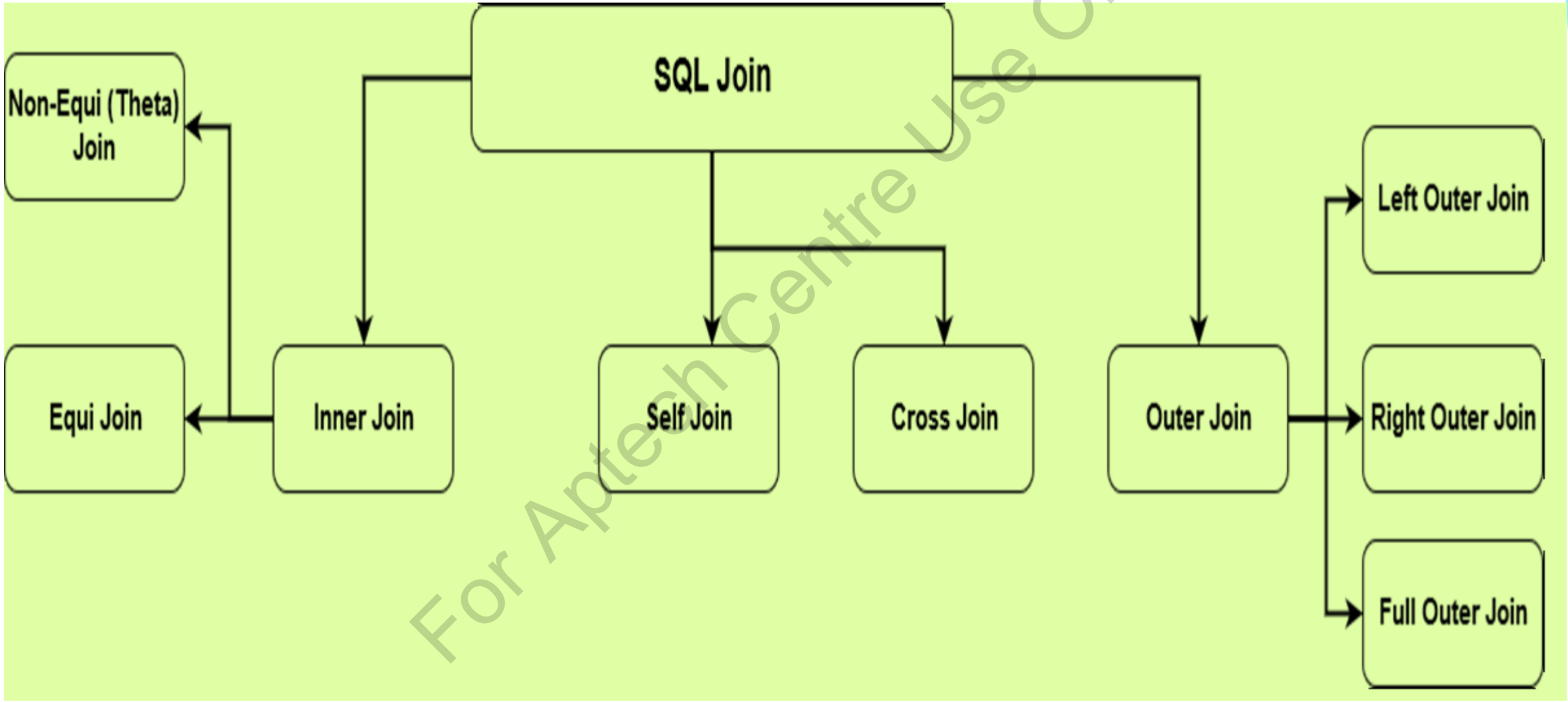
JOINS with Subquery

A subquery can also be used with JOIN operation to increase its usage. Following query will display **Name**, **Age**, and **OrderID** in a Derived Table:

```
SELECT * FROM
(SELECT FirstName, LastName, Age, OrderID
FROM Customer JOIN Orders
ON Customer.CustomerID = Orders.CustomerID) AS Order_Details
```

```
mysql> SELECT * FROM
    -> (SELECT FirstName, LastName, Age, OrderID
    -> FROM Customer JOIN Orders
    -> ON Customer.CustomerID = Orders.CustomerID)
    -> AS Order_Details
    -> ;
+-----------+----------+------+---------+
| FirstName | LastName | Age  | OrderID |
+-----------+----------+------+---------+
| Jack      | Swanson  |   34 |  567834 |
| Joe       | Voight   |   22 |  113425 |
| Berry     | Chase    |   24 |  999982 |
| Karl      | Davis    |   22 |  453110 |
+-----------+----------+------+---------+
4 rows in set (0.00 sec)
```

Types of JOINS [1-3]



Types of JOINS [2-3]

INNER JOIN	It displays rows of both tables. It is mandatory to have a common column between the two tables for the results to be non-empty.
LEFT JOIN	Also known as Left Outer Join, it returns all rows of the left table and the matching rows of the right table.
RIGHT JOIN	Also known as Right Outer Join, it returns all rows of the right table and the matching rows of the left table.
CROSS JOIN	It returns a combined result of rows from the first table with that on the second table, irrespective of any matching rows
SELF JOIN	It is a join of a table with itself. It is used to find query results within itself.

Types of JOINS [3-3]

DELETE JOIN	It is used when the user wants to delete rows from more than one table.
UPDATE JOIN	It modifies/sets values of rows of more than one table using join.
Equal JOIN	It is performed on tables with matching row values only. It requires an equality operator that checks if the row values of one table match with row values of the other table.
NATURAL JOIN	It returns a Cartesian product of the two tables. It is used to remove the duplicate values and display a single column.

Joining Three Tables [1-2]

StudentID	Student_Name
1001	Mary
1002	Jane

CourseID	Course_ Name	StudentID
5001	PHP	1004
5002	JAVA	1001

TeacherID	Teacher_N ame	CourseID
3001	Jill	5003
3002	Greul	5002

Syntax to join the three tables using INNER JOIN:

```
SELECT table1.column1_name, table1.column2_name,...,
table2.column1_name, table2.column2_name,...,
table3.column1_name, table3.column2_name,...,
FROM table1
INNER JOIN table2 ON table1.table1_id = table2.table1_id
INNER JOIN table3 ON table2.table2_id = table3.table2_id;
```

Joining Three Tables [2-2]

Following is the example to display all the rows of all three tables using INNER JOIN:

```
SELECT Student.Student_Name, Course.Course_Name,
Teacher.Teacher_Name
FROM Student
INNER JOIN Course ON Student.StudentID = Course.StudentID
INNER JOIN Teacher ON Course
```

```
mysql> SELECT Student.Student_Name, Course.Course_Name,
    -> Teacher.Teacher_Name
    -> FROM Student
    -> INNER JOIN Course ON Student.StudentID = Course.StudentID
    -> INNER JOIN Teacher ON Course.CourseID = Teacher.CourseID;
+--------------+-------------+--------------+
| Student_Name | Course_Name | Teacher_Name |
+--------------+-------------+--------------+
| Mary         | JAVA        | Greul        |
| Mary         | ASP.NET     | Jill         |
| Jane         | RUBY        | Polo         |
| Jack         | PHP         | Jemkins      |
+--------------+-------------+--------------+
4 rows in set (0.08 sec)
```

Set Operators

Set operators join more than two statements and return results.

Union	Union All	Intersect	Minus
• Combines the result of `SELECT` statements used in the query without returning duplicate values	• Executes the queries and returns all the rows. It allows duplicate values	• Combines results of two or more `SELECT` statements and returns only those rows from the first table which matches the rows of the second table	• Returns the rows that are not present in the second statement

Summary

- A JOIN clause is used to retrieve combined data sets from more than one table.
- JOINS are more efficient than complex queries and subqueries.
- INNER JOIN and OUTER JOIN are the most used JOINS in MySQL.
- Other types of JOINS include CROSS JOIN, EQUIJOIN, and NATURAL JOIN.
- DELETE JOIN and UPDATE JOIN are used for deleting and updating records from multiple tables respectively.
- SET Operators are used with queries to compare records and form a single set of records.
- The most common types of SET Operators are Union SET Operator, Union All SET Operator, Intersect SET Operator, and Minus SET Operator.
- UNION clause adds two tables and returns the result.
- UNION ALL includes duplicate records whereas, UNION filters out distinct records from the tables.
- INTERSECT provides a result with all distinct records from both queries.
- When EXCEPT is used in queries, it displays the results of the first of the first query after removing the results of the second query.