Ex.no. : 04
Date :

Sub Quires and Join Operations

Aim :

To implement and execute simple, nested, sub & join operation queries in mysql database.

Simple Queries

The SQL SELECT DISTINCT Statement

The SELECT DISTINCT statement is used to return only distinct (different) values. Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.

Syntax

SELECT DISTINCT *column1, column2, ...*FROM *table_name*;

Example

SELECT DISTINCT STU_DEPT FROM student1;

STU_DEPT
CSE
EEE
MECH

The SQL AND, OR and NOT Operators

The WHERE clause can be combined with AND, OR, and NOT operators.

The AND and OR operators are used to filter records based on more than one condition:

- The AND operator displays a record if all the conditions separated by AND are TRUE.
- The OR operator displays a record if any of the conditions separated by OR is TRUE.

The NOT operator displays a record if the condition(s) is NOT TRUE

AND Syntax

SELECT *column1, column2, ...*
FROM *table_name*
WHERE *condition1* AND *condition2* AND *condition3 ..*

Example

SELECT * FROM student1 WHERE stud_id=101 AND stud_dept='mech';

STUD_NAME	STUD_ID	STUD_DEPT	STUD_ROLLNO
Ram	101	MECH	104

OR Syntax

SELECT *column1, column2, ...*
FROM *table_name*
WHERE *condition1* OR *condition2* OR *condition3 ...*;

Example
SELECT * FROM student1 WHERE stud_id=101 OR stud_dept='EEE';

STUD_NAME	STUD_ID	STUD_DEPT	STUD_ROLLNO
Ram	101	MECH	104

NOT Syntax
SELECT *column1, column2, ...*
FROM *table_name*
WHERE NOT *condition*;

Example
SELECT * FROM student WHERE NOT stud_id=101;

STUD_NAME	STUD_ID	STUD_DEPT	STUD_ROLLNO
Vicky	102	EEE	105
David	104	EEE	103

The SQL ORDER BY Keyword
The ORDER BY keyword is used to sort the result-set in ascending or descending order.
The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

ORDER BY Syntax
SELECT *column1, column2, ...*
FROM *table_name*
ORDER BY *column1, column2, ...* ASC|DESC;

Example
SELECT * FROM student1 ORDER BY STUD_ID ;

STUD_NAME	STUD_ID	STUD_DEPT	STUD_ROLLNO
Ram	101	MECH	104
Vicky	102	EEE	105
Saddiq	103	CSE	101
David	104	EEE	103

SELECT * FROM student1 ORDER BY STUD_ID DESC;

STUD_NAME	STUD_ID	STUD_DEPT	STUD_ROLLNO
David	104	EEE	103
Saddiq	103	CSE	101
Vicky	102	EEE	105
Ram	101	MECH	104

Subqueries

A MySQL sub query is a query nested within another query such as SELECT, INSERT, UPDATE or DELETE. In addition, a MySQL sub query can be nested inside another sub query.

A MySQL sub query is called an inner query while the query that contains the sub query is called an outer query. A sub query can be used anywhere that expression is used and must be closed in parentheses.

Example SubQueries

1. SELECT lastName, firstName FROM employees WHERE officeCode IN (SELECT officeCode FROM offices WHERE country = 'USA');

In this example:

- The sub query returns all office codes of the offices located in the USA.
- The outer query selects the last name and first name of employees who work in the offices whose office codes are in the result set returned by the sub query.

2. Select max(sid) from classa where sid <(select max(sid) from classa)

SQL Joins

Here are the different types of the Joins in SQL:

(INNER) JOIN **:** Returns records that have matching values in both tables
LEFT (OUTER) JOIN **:** Return all records from the left table, and the matched records from the right table
RIGHT (OUTER) JOIN **:** Return all records from the right table, and the matched records from the left table
FULL (OUTER) JOIN **:** Return all records when there is a match in either left or right table

Note : To perform join operation we need two different tables.

Sql> select * from student;

Rollno	sname	mark1	mark2
101	kareem	95	90
102	kaasim	92	97
103	ram	85	95
104	sai	93	91

Sql > select * from sports

Rollno	sname	sdept	game
101	kareem	CSE	cricket
104	sai	ECE	football
105	ravi	IT	cricket
107	fizal	CSE	chess

Inner Join
The INNER JOIN keyword selects records that have matching values in both tables.

Syntax
SELECT *column_name(s)*
FROM *table1*
INNER JOIN *table2*
ON *table1.column_name = table2.column_name*;

Example
SELECT student.name, student.mark1, sports.game FROM student INNER JOIN sports ON student.rollno=sports.rollno;

sname	mark1	game
kareem	95	cricket
sai	93	football

Left Join

The LEFT JOIN keyword returns all records from the left table (table1), and the matched records from the right table (table2). The result is NULL from the right side, if there is no match.

Syntax
SELECT *column_name(s)*
FROM *table1*
LEFT JOIN *table2*
ON *table1.column_name = table2.column_name*;

Example
SELECT student.name, student.mark1, sports.game FROM student LEFT JOIN sports ON student.rollno=sports.rollno;

sname	mark1	game
kareem	95	cricket
kaasim	92	Null
ram	85	Null
sai	93	football

RIGHT JOIN Keyword
The RIGHT JOIN keyword returns all records from the right table (table2), and the matched records from the left table (table1). The result is NULL from the left side, when there is no match.

Syntax
SELECT *column_name(s)*
FROM *table1*
RIGHT JOIN *table2*

ON *table1.column_name = table2.column_name*;

Example
SELECT student.name, student.mark1, sports.game FROM student RIGHT JOIN sports ON student.rollno=sports.rollno;

sname	mark1	game
kareem	95	cricket
sai	93	football
Null	Null	cricket
Null	Null	chess

FULL OUTER JOIN Keyword
The FULL OUTER JOIN keyword return all records when there is a match in either left (table1) or right (table2) table records.
Note: FULL OUTER JOIN or FULL JOIN is not directly performed in sql so we can achive it by union operation of left join and right join.

Syntax
SELECT *column_name(s)*
FROM *table1*
FULL OUTER JOIN *table2*
ON *table1.column_name = table2.column_name*;
If the above syntax is not working then we can go with union operation.

SELECT *column_name(s)*
FROM *table1*
LEFT JOIN *table2*
ON *table1.column_name = table2.column_name*;
Union
SELECT *column_name(s)*
FROM *table1*
RIGHT JOIN *table2*
ON *table1.column_name = table2.column_name*;

Example
SELECT student.name, student.mark1, sports.game FROM student LEFT JOIN sports ON student.rollno=sports.rollno
Union
SELECT student.name, student.mark1, sports.game FROM student RIGHT JOIN sports ON student.rollno=sports.rollno

Result
Thus the SQL sub queries, nested queries and various join operation queries are written and executed successfully.