Internet and Web Programming

Digital Assignment – 2

Aim – The objective of this lab session is to introduce students to SQL (Structured Query Language) and its practical application in querying databases. Through a series of exercises, students will gain hands-on experience in executing SQL queries to retrieve, manipulate, and analyse data stored in a relational database.

Theory –

Structured Query Language (SQL) serves as the primary interface for interacting with relational databases. SQL queries are used to retrieve, manipulate, and manage data stored in these databases. Understanding SQL is crucial for anyone involved in database management, data analysis, or software development.

SQL queries are divided into several categories, including Data Definition Language (DDL), Data Manipulation Language (DML), Data Control Language (DCL), and Transaction Control Language (TCL). Each category serves a distinct purpose in managing and manipulating database objects and data.

Data Definition Language (DDL):

DDL commands are used to define and modify the structure of database objects such as tables, views, indexes, and constraints. Common DDL commands include CREATE, ALTER, and DROP. For example, the CREATE TABLE command is used to create a new table in the database, specifying its columns, data types, and constraints.

Data Manipulation Language (DML):

DML commands are used to manipulate data within the database. The most common DML commands are SELECT, INSERT, UPDATE, and DELETE. SELECT is used to retrieve data from one or more tables, while INSERT, UPDATE, and DELETE are used to add, modify, and delete data, respectively.

Data Control Language (DCL):

DCL commands are used to control access to data within the database. These commands include GRANT and REVOKE, which grant and revoke privileges on database objects to users or roles.

Transaction Control Language (TCL):

TCL commands are used to manage transactions within the database. Transactions ensure the integrity and consistency of data by grouping multiple SQL statements into a single unit of work. The most common TCL commands are COMMIT, ROLLBACK, and SAVEPOINT, which are used to commit or rollback transactions and set save points within transactions.

SQL queries are written using a specific syntax that adheres to the SQL standards defined by organizations such as the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO). However, different database management systems (DBMS) may implement SQL with slight variations, known as dialects.

In conclusion, SQL queries form the foundation of database operations, allowing users to interact with and manipulate data

efficiently. Understanding SQL is essential for anyone working with relational databases, as it enables them to extract valuable insights and information from data stored in these systems.

Database Screenshots –

```
mysql> create database iwpda;
Query OK, 1 row affected (0.01 sec)
```

```
mysql> use iwpda;
Database changed
mysql> CREATE TABLE Employee (
    ->   Empid INT(4) PRIMARY KEY,
    ->   Name VARCHAR(15) NOT NULL,
    ->   Department VARCHAR(15) NOT NULL CHECK (Department IN ('HR', 'Testing', 'Development', 'Accounts')),
    ->   Grade VARCHAR(3) NOT NULL CHECK (Grade IN ('A', 'B', 'C')),
    ->   BasicPay DECIMAL(10,2),
    ->   Salary DECIMAL(10,2),
    ->   BonusPoints DECIMAL(5,2),
    ->   HireDate DATE
    -> );
Query OK, 0 rows affected, 1 warning (0.06 sec)
```

```
mysql> INSERT INTO Employee (Empid, Name, Department, Grade, BasicPay, Salary, BonusPoints, HireDate)
    -> VALUES (1, 'John Doe', 'HR', 'A', 5000.00, 6000.00, 50.00, '2023-01-15');
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO Employee (Empid, Name, Department, Grade, BasicPay, Salary, BonusPoints, HireDate)
    -> VALUES (2, 'Jane Smith', 'Testing', 'B', 4500.00, 5500.00, 40.00, '2023-02-20');
Query OK, 1 row affected (0.03 sec)

mysql> INSERT INTO Employee (Empid, Name, Department, Grade, BasicPay, Salary, BonusPoints, HireDate)
    -> VALUES (3, 'Alice Johnson', 'Development', 'C', 4000.00, 5000.00, 35.00, '2023-03-10');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Employee (Empid, Name, Department, Grade, BasicPay, Salary, BonusPoints, HireDate)
    -> VALUES (4, 'Michael Brown', 'Accounts', 'A', 5200.00, 6200.00, 55.00, '2023-04-05');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Employee (Empid, Name, Department, Grade, BasicPay, Salary, BonusPoints, HireDate)
    -> VALUES (5, 'Emily Davis', 'HR', 'B', 4800.00, 5800.00, 45.00, '2023-05-12');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Employee (Empid, Name, Department, Grade, BasicPay, Salary, BonusPoints, HireDate)
    -> VALUES (6, 'David Wilson', 'Testing', 'C', 4300.00, 5300.00, 38.00, '2023-06-18');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Employee (Empid, Name, Department, Grade, BasicPay, Salary, BonusPoints, HireDate)
    -> VALUES (7, 'Sarah Taylor', 'Development', 'A', 5100.00, 6100.00, 50.00, '2023-07-22');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Employee (Empid, Name, Department, Grade, BasicPay, Salary, BonusPoints, HireDate)
    -> VALUES (8, 'James Martinez', 'Accounts', 'B', 4900.00, 5900.00, 48.00, '2023-08-30');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Employee (Empid, Name, Department, Grade, BasicPay, Salary, BonusPoints, HireDate)
    -> VALUES (9, 'Olivia Clark', 'HR', 'C', 4200.00, 5200.00, 33.00, '2023-09-10');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Employee (Empid, Name, Department, Grade, BasicPay, Salary, BonusPoints, HireDate)
    -> VALUES (10, 'Benjamin Adams', 'Testing', 'A', 5300.00, 6300.00, 60.00, '2023-10-25');
Query OK, 1 row affected (0.00 sec)
```

Database Data –

```
mysql> select * from employee;
+-------+---------------+-------------+-------+----------+---------+-------------+------------+
| Empid | Name          | Department  | Grade | BasicPay | Salary  | BonusPoints | HireDate   |
+-------+---------------+-------------+-------+----------+---------+-------------+------------+
|     1 | John Doe      | HR          | A     |  5000.00 | 6000.00 |       50.00 | 2023-01-15 |
|     2 | Jane Smith    | Testing     | B     |  4500.00 | 5500.00 |       40.00 | 2023-02-20 |
|     3 | Alice Johnson | Development | C     |  4000.00 | 5000.00 |       35.00 | 2023-03-10 |
|     4 | Michael Brown | Accounts    | A     |  5200.00 | 6200.00 |       55.00 | 2023-04-05 |
|     5 | Emily Davis   | HR          | B     |  4800.00 | 5800.00 |       45.00 | 2023-05-12 |
|     6 | David Wilson  | Testing     | C     |  4300.00 | 5300.00 |       38.00 | 2023-06-18 |
|     7 | Sarah Taylor  | Development | A     |  5100.00 | 6100.00 |       50.00 | 2023-07-22 |
|     8 | James Martinez | Accounts    | B     |  4900.00 | 5900.00 |       48.00 | 2023-08-30 |
|     9 | Olivia Clark  | HR          | C     |  4200.00 | 5200.00 |       33.00 | 2023-09-10 |
|    10 | Benjamin Adams | Testing     | A     |  5300.00 | 6300.00 |       60.00 | 2023-10-25 |
+-------+---------------+-------------+-------+----------+---------+-------------+------------+
10 rows in set (0.00 sec)
```

Code –

```php
<?php

echo "<br>";
echo "<center><h1>IWP DA2</h1></center>";
echo "<center><h2>By - Ojas Aklecha ( 21BCE1109 )</h2></center>";

// Database connection parameters
$servername = "localhost";
$username = "root";
$password = "password";
$dbname = "iwpda";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Execute queries

// i) Display the employee details in Grade is not “B”
$sql = "SELECT * FROM Employee WHERE Grade != 'B'";
$result = $conn->query($sql);

echo "<div class='result'>";
echo "<h3>i) Display the employee details in Grade is not “B”:</h3><p>";
if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        echo "Name: " . $row["Name"] . ", Department: " . $row["Department"] .
", Grade: " . $row["Grade"] . "<br>";
```

```
    }
} else {
    echo "No records found<br>";
}
echo "</p></div>";

// ii) List the number of employees in each department. Only include
department with more than 3 employees. (Use having clause)
$sql = "SELECT Department, COUNT(*) as numEmployees FROM Employee GROUP BY
Department HAVING numEmployees > 3";
$result = $conn->query($sql);

echo "<div class='result'>";
echo "<h3>ii) List the number of employees in each department. Only include
department with more than 3 employees. (Use having clause):</h3><p>";
if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        echo "Department: " . $row["Department"] . ", Number of employees: " .
$row["numEmployees"] . "<br>";
    }
} else {
    echo "No records found<br>";
}
echo "</p></div>";

// iii) Lists the number of employees in each department and group by their
departments (Use Group by)
$sql = "SELECT Department, COUNT(*) as numEmployees FROM Employee GROUP BY
Department";
$result = $conn->query($sql);

echo "<div class='result'>";
echo "<h3>iii) Lists the number of employees in each department and group by
their departments (Use Group by):</h3><p>";
if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        echo "Department: " . $row["Department"] . ", Number of employees: " .
$row["numEmployees"] . "<br>";
    }
} else {
    echo "No records found<br>";
}
echo "</p></div>";

// iv) List the distinct department names
$sql = "SELECT DISTINCT Department FROM Employee";
$result = $conn->query($sql);
```

```php
echo "<div class='result'>";
echo "<h3>iv) List the distinct department names:</h3><p>";
if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        echo $row["Department"] . "<br>";
    }
} else {
    echo "No records found<br>";
}
echo "</p></div>";

// v) How many employees earn salary in the range between 30k and 40k
$sql = "SELECT COUNT(*) as numEmployees FROM Employee WHERE Salary BETWEEN
30000 AND 40000";
$result = $conn->query($sql);
$row = $result->fetch_assoc();

echo "<div class='result'>";
echo "<h3>v) How many employees earn salary in the range between 30k and
40k:</h3><p>";
echo "Number of employees earning salary between 30k and 40k: " .
$row["numEmployees"] . "<br>";
echo "</p></div>";

// vi) Find the rounded value of the bonus points. (Differentiate using CEIL,
FLOOR, TRUNC, ROUND)
$sql = "SELECT BonusPoints, CEIL(BonusPoints) as CEIL, FLOOR(BonusPoints) as
FLOOR, TRUNCATE(BonusPoints, 0) as TRUNC, ROUND(BonusPoints) as ROUND FROM
Employee";
$result = $conn->query($sql);

echo "<div class='result'>";
echo "<h3>vi) Find the rounded value of the bonus points. (Differentiate using
CEIL, FLOOR, TRUNC, ROUND):</h3><p>";
if ($result->num_rows > 0) {
    echo "Original\tCEIL\tFLOOR\tTRUNC\tROUND<br>";
    while ($row = $result->fetch_assoc()) {
        echo $row["BonusPoints"] . "\t" . $row["CEIL"] . "\t" . $row["FLOOR"]
. "\t" . $row["TRUNC"] . "\t" . $row["ROUND"] . "<br>";
    }
} else {
    echo "No records found<br>";
}
echo "</p></div>";

// vii) List the employee details who got the minimum bonus points
$sql = "SELECT * FROM Employee WHERE BonusPoints = (SELECT MIN(BonusPoints)
FROM Employee)";
```

```php
$result = $conn->query($sql);

echo "<div class='result'>";
echo "<h3>vii) List the employee details who got the minimum bonus
points:</h3><p>";
if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        echo "Name: " . $row["Name"] . ", Department: " . $row["Department"] .
", Bonus Points: " . $row["BonusPoints"] . "<br>";
    }
} else {
    echo "No records found<br>";
}
echo "</p></div>";

// viii) Calculate the total salary of all employees
$sql = "SELECT SUM(Salary) as totalSalary FROM Employee";
$result = $conn->query($sql);
$row = $result->fetch_assoc();

echo "<div class='result'>";
echo "<h3>viii) Calculate the total salary of all employees:</h3><p>";
echo "Total salary of all employees: $" . $row["totalSalary"] . "<br>";
echo "</p></div>";

// ix) List the employee details in “Testing” department.
$sql = "SELECT * FROM Employee WHERE Department = 'Testing'";
$result = $conn->query($sql);

echo "<div class='result'>";
echo "<h3>ix) List the employee details in “Testing” department:</h3><p>";
if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        echo "Name: " . $row["Name"] . ", Department: " . $row["Department"] .
", Grade: " . $row["Grade"] . "<br>";
    }
} else {
    echo "No records found<br>";
}
echo "</p></div>";

echo "<br>";
echo "<br>";

?>
```

Output Screenshots –

i) Display the employee details in Grade is not "B":

Name: John Doe, Department: HR, Grade: A
Name: Alice Johnson, Department: Development, Grade: C
Name: Michael Brown, Department: Accounts, Grade: A
Name: David Wilson, Department: Testing, Grade: C
Name: Sarah Taylor, Department: Development, Grade: A
Name: Olivia Clark, Department: HR, Grade: C
Name: Benjamin Adams, Department: Testing, Grade: A

ii) List the number of employees in each department. Only include department with more than 3 employees. (Use having clause):

No records found

iii) Lists the number of employees in each department and group by their departments (Use Group by):

Department: HR, Number of employees: 3
Department: Testing, Number of employees: 3
Department: Development, Number of employees: 2
Department: Accounts, Number of employees: 2

iv) List the distinct department names:

HR
Testing
Development
Accounts

v) How many employees earn salary in the range between 30k and 40k:

Number of employees earning salary between 30k and 40k: 0

vi) Find the rounded value of the bonus points. (Differentiate using CEIL, FLOOR, TRUNC, ROUND):

Original CEIL FLOOR TRUNC ROUND
50.00 50 50 50 50
40.00 40 40 40 40
35.00 35 35 35 35
55.00 55 55 55 55
45.00 45 45 45 45
38.00 38 38 38 38
50.00 50 50 50 50
48.00 48 48 48 48
33.00 33 33 33 33
60.00 60 60 60 60

vii) List the employee details who got the minimum bonus points:

Name: Olivia Clark, Department: HR, Bonus Points: 33.00

viii) Calculate the total salary of all employees:

Total salary of all employees: $57300.00

ix) List the employee details in "Testing" department:

Name: Jane Smith, Department: Testing, Grade: B
Name: David Wilson, Department: Testing, Grade: C
Name: Benjamin Adams, Department: Testing, Grade: A

Conclusion –

Through this lab session, students have gained practical experience in executing various SQL queries to retrieve specific information from a database. By analysing the results obtained from these queries, students have developed a better understanding of SQL syntax and its application in real-world scenarios. This hands-on experience will prove invaluable as they continue to explore the fundamentals of database management and SQL query optimization.