Experiment 6: Implementation of Various Claases in SQL

Aim:

Implementation of various clauses in SQL

Theory

GROUP BY

- **Purpose:** Groups all records in a relation based on specified key(s) and displays selected fields.
- **Syntax:**

```
SELECT <set of fields> FROM <relation_name>
GROUP BY <field_name>;
```

- **Example:**

```
SELECT EMPNO, SUM(SALARY) FROM EMP GROUP BY EMPNO;
```

GROUP BY-HAVING

- **Purpose:** The HAVING clause is used with GROUP BY to filter records based on aggregate functions.
- **Syntax:**

```
SELECT column_name, aggregate_function(column_name)
FROM table_name
WHERE column_name operator value
GROUP BY column_name
HAVING aggregate_function(column_name) operator value;
```

- **Example:**

```
SELECT Employees.LastName, COUNT(Orders.OrderID) AS NumberOfOrders
FROM Orders
```

```
INNER JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID
GROUP BY LastName
HAVING COUNT(Orders.OrderID) > 10;
```

JOIN using GROUP BY

- **Purpose:** Displays fields from two relations by matching a common field and grouping records based on a specified key.
- **Syntax:**

```
SELECT <set of fields (from both relations)>
FROM relation_1, relation_2
WHERE relation_1.field_x = relation_2.field_y
GROUP BY field_z;
```

- **Example:**

```
SELECT empno, SUM(SALARY)
FROM emp, dept
WHERE emp.deptno = dept.deptno
GROUP BY empno;
```

ORDER BY

- **Purpose:** Displays selected fields from a relation in an ordered manner based on a specified field.
- **Syntax:**

```
SELECT <set of fields> FROM <relation_name>
ORDER BY <field_name>;
```

- **Example:**

```
SELECT empno, ename, job
FROM emp
ORDER BY job;
```

JOIN using ORDER BY

- **Purpose:** Displays fields from two relations by matching a common field and ordering records based on a specified field.
- **Syntax:**

  ```
  SELECT <set of fields (from both relations)>
  FROM relation_1, relation_2
  WHERE relation_1.field_x = relation_2.field_y
  ORDER BY field_z;
  ```

- **Example:**

  ```
  SELECT empno, ename, job, dname
  FROM emp, dept
  WHERE emp.deptno = dept.deptno
  ORDER BY job;
  ```

INDEXING

- **Purpose:** Creates an ordered set of pointers to data in a table to improve performance and ensure uniqueness.
- **Syntax:**

  ```
  CREATE INDEX <index_name> ON <table_name> (attrib1, attrib2, ..., attribn);
  ```

- **Example:**

  ```
  CREATE INDEX id1 ON emp(empno, dept_no);
  ```

---

conclusion: hence we successfully

Implemented various clauses in SQL