Government Engineering College, Thrissur

CS333– Application Software Development Lab

Documentation -

Exp 10 – Order By, Group By & Having
Exp 11 – Set operators, Nested queries & Join queries

Date of Submission
9 October 2020

Submitted By
Kowsik Nandagopan D
Roll No 31
TCR18CS031
GECT CSE S5

Experiment 10

AIM

Implementation of Order By, Group By& Having clause.

Description

Implement Order By, Group By& Having clause in MySQL to get valuable insight from the previously created database. Functions available in MySQL are - ORDER BY, GROUP BY & HAVING

Output / Screenshots

Full Table

```
mysql> SELECT * FROM Projects_Efforts
    -> ;
+----+----------+-----+----------------+--------------+
| ID | Title    | CID | name           | Effort_Spent |
+----+----------+-----+----------------+--------------+
|  2 | Open PPM |   1 | Petey Cruiser  |        22.00 |
|  2 | Open PPM |   2 | Anna Sthesia   |        10.00 |
|  2 | Open PPM |   3 | Paul Molive    |        10.00 |
|  2 | Open PPM |   4 | Anna Mull      |        15.00 |
|  2 | Open PPM |   5 | Gail Forcewind |        25.20 |
|  2 | Open PPM |   6 | Paige Turner   |         5.00 |
|  2 | Open PPM |   7 | Walter Melon   |         2.00 |
|  2 | Open PPM |   8 | Nick R. Bocker |        25.00 |
|  1 | Sirius   |   9 | Barb Ackue     |        26.00 |
|  1 | Sirius   |   3 | Paul Molive    |        10.00 |
+----+----------+-----+----------------+--------------+
```

1. GROUP BY - Group By Title Form the Project Effort Views

```
mysql> SELECT Title, SUM(Effort_Spent) FROM Projects_Efforts GROUP BY (Title);
+----------+-------------------+
| Title    | SUM(Effort_Spent) |
+----------+-------------------+
| Open PPM |            114.20 |
| Sirius   |             36.00 |
+----------+-------------------+
2 rows in set (0.00 sec)
```

2. HAVING - Group By Title from the Project Effort Views and select the one having the Sum of Efforts greater than 100

```
mysql> SELECT Title, SUM(Effort_Spent) as Sum FROM Projects_Efforts GROUP BY (Title) HAVING Sum > 100;
+----------+--------+
| Title    | Sum    |
+----------+--------+
| Open PPM | 114.20 |
+----------+--------+
1 row in set (0.00 sec)
```

3. ORDER BY - Order the Projects Efforts View in Decreasing Order

```
mysql> SELECT Title, Name, Effort_Spent FROM Projects_Efforts ORDER BY Effort_Spent DESC;
+----------+----------------+--------------+
| Title    | name           | Effort_Spent |
+----------+----------------+--------------+
| Sirius   | Barb Ackue     |        26.00 |
| Open PPM | Gail Forcewind |        25.20 |
| Open PPM | Nick R. Bocker |        25.00 |
| Open PPM | Petey Cruiser  |        22.00 |
| Open PPM | Anna Mull      |        15.00 |
| Open PPM | Paul Molive    |        10.00 |
| Sirius   | Paul Molive    |        10.00 |
| Open PPM | Anna Sthesia   |        10.00 |
| Open PPM | Paige Turner   |         5.00 |
| Open PPM | Walter Melon   |         2.00 |
+----------+----------------+--------------+
10 rows in set (0.00 sec)
```

Experiment 11

AIM

Implementation of set operators, nested queries and Join queries

Description

Implement set operators, nested queries and Join queries in MySQL to get valuable insight from the previously created database.

- For union there is UNION operator.
- For Intersection and minus (set difference) may not be available in some MySQL versions so we can use IN or NOT IN respectively.
- For Cross Product there is a function called CROSS JOIN.
- Also for nested queries it is select-from where inside another query.

Output / Screenshots

1. JOIN - Projects Efforts View and Join with the Consultants table and display the phone number corresponding to their efforts

```
mysql> SELECT Consultants.Name, Projects_Efforts.Effort_Spent, Projects_Efforts.Effort_Spent, Consultants.phone FROM Projects_Efforts JOIN Consultants
 ON Projects_Efforts.CID = Consultants.ID;
+---------------+-------------+-------------+-------------+
| Name          | Effort_Spent | Effort_Spent | phone       |
+---------------+-------------+-------------+-------------+
| Petey Cruiser |        22.00 |        22.00 | 410763981007 |
| Anna Sthesia  |        10.00 |        10.00 | 10105525611  |
| Paul Molive   |        10.00 |        10.00 | 6011377914   |
| Anna Mull     |        15.00 |        15.00 | 7901584026   |
| Gail Forcewind |        25.20 |        25.20 | 1594941628   |
| Paige Turner  |         5.00 |         5.00 | 0148042499   |
| Walter Melon  |         2.00 |         2.00 | 1171089059   |
| Nick R. Bocker |        25.00 |        25.00 | 5779877107   |
| Barb Ackue    |        26.00 |        26.00 | 9646946767   |
| Paul Molive   |        10.00 |        10.00 | 6011377914   |
+---------------+-------------+-------------+-------------+
10 rows in set (0.00 sec)
```

2. Nested Query - Use the nested query and select the CEO From the Project Effots Table and display his Effort

```
mysql> SELECT Consultants.Name, Projects_Efforts.Effort_Spent, Projects_Efforts.Effort_Spent, Consultants.phone FROM Projects_Efforts JOIN Consultants
 ON Projects_Efforts.CID = Consultants.ID WHERE Consultants.Name IN (SELECT Name FROM Consultant_Designation WHERE Designation = "CEO" );
+---------------+-------------+-------------+-------------+
| Name          | Effort_Spent | Effort_Spent | phone       |
+---------------+-------------+-------------+-------------+
| Petey Cruiser |        22.00 |        22.00 | 410763981007 |
+---------------+-------------+-------------+-------------+
1 row in set (0.01 sec)
```

3. UNION - Use the nested query and select the CEO and Marketing Manager From the Project Efforts Table and display his Effort

```
mysql> SELECT Consultants.Name, Projects_Efforts.Effort_Spent, Projects_Efforts.Effort_Spent, Consultants.phone FROM Projects_Efforts JOIN Consultants
 ON Projects_Efforts.CID = Consultants.ID WHERE Consultants.Name IN (SELECT Name FROM Consultant_Designation WHERE Designation = "CEO" )
    -> UNION
    -> SELECT Consultants.Name, Projects_Efforts.Effort_Spent, Projects_Efforts.Effort_Spent, Consultants.phone FROM Projects_Efforts JOIN Consultants
 ON Projects_Efforts.CID = Consultants.ID WHERE Consultants.Name IN (SELECT Name FROM Consultant_Designation WHERE Designation = "Marketing Manager" )

    -> ;
+---------------+-------------+-------------+-------------+
| Name          | Effort_Spent | Effort_Spent | phone       |
+---------------+-------------+-------------+-------------+
| Petey Cruiser |        22.00 |        22.00 | 410763981007 |
| Gail Forcewind |        25.20 |        25.20 | 1594941628  |
+---------------+-------------+-------------+-------------+
2 rows in set (0.01 sec)
```

4. Set Difference - Select the name of those working in Open PPM Project not in Sirius. Use Projects_Efforts View

```
mysql> SELECT Name FROM Projects_Efforts WHERE Title = "Open PPM" AND Name IN (SELECT Name FROM Projects_Efforts WHERE Title = "Sirius");
+-------------+
| name        |
+-------------+
| Paul Molive |
+-------------+
1 row in set (0.00 sec)
```

5. Intersection - Select the consultants working on both Open PPM and Sirius. Form Projects_Efforts View

```
mysql> SELECT Name FROM Projects_Efforts WHERE Title = "Open PPM" AND Name NOT IN (SELECT Name FROM Projects_Efforts WHERE Title = "Sirius");
+---------------+
| name          |
+---------------+
| Petey Cruiser |
| Anna Sthesia  |
| Anna Mull     |
| Gail Forcewind |
| Paige Turner  |
| Walter Melon  |
| Nick R. Bocker |
+---------------+
7 rows in set (0.00 sec)
```

6. CROSS Product - Just See the Cross product on Projects_Effort and Consultants Table

```
mysql> SELECT Consultants.Name, Projects_Efforts.Effort_Spent, Projects_Efforts.Effort_Spent, Consultants.phone FROM Projects_Efforts CROSS JOIN Consu
ltants;
+---------------+-------------+-------------+-------------+
| Name          | Effort_Spent | Effort_Spent | phone       |
+---------------+-------------+-------------+-------------+
| Petey Cruiser |        22.00 |        22.00 | 410763981007 |
| Petey Cruiser |        10.00 |        10.00 | 410763981007 |
| Petey Cruiser |        10.00 |        10.00 | 410763981007 |
| Petey Cruiser |        10.00 |        10.00 | 410763981007 |
| Petey Cruiser |        15.00 |        15.00 | 410763981007 |
| Petey Cruiser |        25.20 |        25.20 | 410763981007 |
| Petey Cruiser |         5.00 |         5.00 | 410763981007 |
| Petey Cruiser |         2.00 |         2.00 | 410763981007 |
| Petey Cruiser |        25.00 |        25.00 | 410763981007 |
| Petey Cruiser |        26.00 |        26.00 | 410763981007 |
| Anna Sthesia  |        22.00 |        22.00 | 10105525611  |
| Anna Sthesia  |        10.00 |        10.00 | 10105525611  |
| Anna Sthesia  |        10.00 |        10.00 | 10105525611  |
| Anna Sthesia  |        10.00 |        10.00 | 10105525611  |
| Anna Sthesia  |        15.00 |        15.00 | 10105525611  |
| Anna Sthesia  |        25.20 |        25.20 | 10105525611  |
| Anna Sthesia  |         5.00 |         5.00 | 10105525611  |
| Anna Sthesia  |         2.00 |         2.00 | 10105525611  |
| Anna Sthesia  |        25.00 |        25.00 | 10105525611  |
| Anna Sthesia  |        26.00 |        26.00 | 10105525611  |
| Paul Molive   |        22.00 |        22.00 | 6011377914   |
| Paul Molive   |        10.00 |        10.00 | 6011377914   |
| Paul Molive   |        10.00 |        10.00 | 6011377914   |
| Paul Molive   |        10.00 |        10.00 | 6011377914   |
| Paul Molive   |        15.00 |        15.00 | 6011377914   |
| Paul Molive   |        25.20 |        25.20 | 6011377914   |
| Paul Molive   |         5.00 |         5.00 | 6011377914   |
| Paul Molive   |         2.00 |         2.00 | 6011377914   |
| Paul Molive   |        25.00 |        25.00 | 6011377914   |
| Paul Molive   |        26.00 |        26.00 | 6011377914   |
| Anna Mull     |        22.00 |        22.00 | 7901584026   |
```