Government Engineering College, Thrissur

CS333– Application Software Development Lab

Documentation -

Exp 8 – Built-in Functions

Exp 9 – Aggregate Functions

Date of Submission
30 September 2020

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

Experiment 8

AIM

Implementation of Built in functions in RDBMS

Description

Implement built-in functions in MySQL to get valuable insight from the previously created database. Built-in functions include String functions, Date functions, Numeric functions, and some advanced functions. (You can select 4 relevant functions from each category).

Output / Screenshots

String Functions

1. Select the consultants table and output string lenth of presentAddress and presentAddress

CHAR_LENGTH

```
mysql> mysql> SELECT CHAR_LENGTH(presentAddress), presentAddress FROM Consultants;
+----------------------------+-----------------------------------------------+
| CHAR_LENGTH(presentAddress) | presentAddress                                |
+----------------------------+-----------------------------------------------+
|                         36 | 405 Fulton Court Attleboro, MA 02703          |
|                         40 | 47 Glenridge St.
Huntersville, NC 28078         |
|                         33 | 945 Mill Street
Parlin, NJ 08859               |
|                         38 | 613 East Clay Lane
Palatine, IL 60067           |
|                         38 | 479 Thorne St.
Saint Joseph, MI 49085          |
|                         36 | 128 Hillside St.
Bismarck, ND 58501             |
|                         39 | 7990 Lakewood Dr.
Plainfield, NJ 07060           |
|                         38 | 7555 NW. Gulf Drive
Jamaica, NY 11432            |
|                         48 | 478 South Crescent Street
Bergenfield, NJ 07621 |
+----------------------------+-----------------------------------------------+
9 rows in set (0.00 sec)
```

2. Select Consultants and print the name in the format:- Name (DESIGNATION)

CONCAT, SPACE, UPPER

```
mysql> SELECT CONCAT(Consultants.name, SPACE(1), "(", UPPER(Designations.Designation), ")")  FROM Consultants JOIN Designations ON Designations.ID = C
onsultants.designation;
+-----------------------------------------------------------------------------+
| CONCAT(Consultants.name, SPACE(1), "(", UPPER(Designations.Designation), ")") |
+-----------------------------------------------------------------------------+
| Petey Cruiser (CEO)                                                         |
| Anna Sthesia (ACCOUNTS MANAGER)                                             |
| Paul Molive (DESIGN MANAGER)                                                |
| Anna Mull (PROJECT MANAGER)                                                 |
| Gail Forcewind (MARKETING MANAGER)                                          |
| Paige Turner (CFO)                                                          |
| Walter Melon (ANDROID DEVELOPER)                                            |
| Nick R. Bocker (DATA ANALYST)                                               |
| Barb Ackue (WEB DEVELOPER)                                                  |
+-----------------------------------------------------------------------------+
9 rows in set (0.00 sec)
```

Date Functions

1. Display the current date in console

CURDATE

```
mysql> SELECT CURDATE();
+------------+
| CURDATE()  |
+------------+
| 2020-09-28 |
+------------+
1 row in set (0.00 sec)
```

2. Select Project table and display the title, project start date. Date must be in format Day Date. Month must be displayed in name format

DATE_FORMAT

```
mysql> SELECT title, DATE_FORMAT(Start_Date, "%a %D %b %Y") FROM Projects;
+------------+-----------------------------------------+
| title      | DATE_FORMAT(Start_Date, "%a %D %b %Y") |
+------------+-----------------------------------------+
| Sirius     | Sat 12th Sep 2020                       |
| Open PPM   | Sat 12th Sep 2020                       |
| Sirius     | Sat 12th Sep 2020                       |
| Pide Piper | Tue 22nd Sep 2020                       |
| Pide Piper | Tue 22nd Sep 2020                       |
| Pide Piper | Tue 22nd Sep 2020                       |
+------------+-----------------------------------------+
6 rows in set (0.00 sec)
```

3. Display the number of days between start date and finish date for each project

DATEDIFF

```
mysql> SELECT title, DATEDIFF(Finish_Date, Start_Date) FROM Projects;
+-------------+-------------------------------------+
| title       | DATEDIFF(Finish_Date, Start_Date) |
+-------------+-------------------------------------+
| Sirius      |                                 168 |
| Open PPM    |                                 216 |
| Sirius      |                                   0 |
| Pide Piper  |                                   0 |
| Pide Piper  |                                   0 |
| Pide Piper  |                                   0 |
+-------------+-------------------------------------+
6 rows in set (0.00 sec)
```

4. Display the date which is 10 days before finish date in specific format

DATE_SUB

```
mysql> SELECT title, DATE_FORMAT(DATE_SUB(Finish_Date, INTERVAL 10 DAY), "%a %D, %b %Y") FROM Projects;
+------------+-------------------------------------------------------------------------+
| title      | DATE_FORMAT(DATE_SUB(Finish_Date, INTERVAL 10 DAY), "%a %D, %b %Y") |
+------------+-------------------------------------------------------------------------+
| Sirius     | Wed 17th, Feb 2021                                                      |
| Open PPM   | Tue 6th, Apr 2021                                                       |
| Sirius     | Wed 2nd, Sep 2020                                                       |
| Pide Piper | Sat 12th, Sep 2020                                                      |
| Pide Piper | Sat 12th, Sep 2020                                                      |
| Pide Piper | Sat 12th, Sep 2020                                                      |
+------------+-------------------------------------------------------------------------+
```

Numeric Functions

1. Print Value of PI

PI

```
mysql> SELECT PI();
+----------+
| PI()     |
+----------+
| 3.141593 |
+----------+
1 row in set (0.00 sec)
```

2. Select Project Transactions table and display the project name and entries of percentage completed of each resource. Percentage completed must be rounded of to one decimal

ROUND

```
mysql> SELECT Projects.title, ROUND(Project_Transaction.Percentage_Completed, 1) FROM Projects JOIN Project_Transaction ON Projects.ID = Project_Trans
action.ProjectID;
+----------+-----------------------------------------------------+
| title    | ROUND(Project_Transaction.Percentage_Completed, 1) |
+----------+-----------------------------------------------------+
| Open PPM |                                                 0.2 |
| Open PPM |                                                 0.0 |
| Open PPM |                                                 0.1 |
| Open PPM |                                                 1.0 |
| Open PPM |                                                 5.0 |
| Open PPM |                                                 0.0 |
| Open PPM |                                                 0.0 |
| Open PPM |                                                10.0 |
| Sirius   |                                                30.0 |
| Sirius   |                                                20.0 |
+----------+-----------------------------------------------------+
10 rows in set (0.00 sec)
```

3. Select Project Transactions table and display the project name and entries of percentage completed of each resource. Percentage completed must be to nearest integer (not decimal)

FLOOR

```
mysql> SELECT Projects.title, FLOOR(Project_Transaction.Percentage_Completed) FROM Projects JOIN Project_Transaction ON Projects.ID = Project_Transact
ion.ProjectID;
+----------+-------------------------------------------------+
| title    | FLOOR(Project_Transaction.Percentage_Completed) |
+----------+-------------------------------------------------+
| Open PPM |                                               0 |
| Open PPM |                                               0 |
| Open PPM |                                               0 |
| Open PPM |                                               1 |
| Open PPM |                                               5 |
| Open PPM |                                               0 |
| Open PPM |                                               0 |
| Open PPM |                                              10 |
| Sirius   |                                              30 |
| Sirius   |                                              20 |
+----------+-------------------------------------------------+
10 rows in set (0.00 sec)
```

4. Display the minimum and maximum in project transaction

MIN, MAX

```
mysql> SELECT MIN(Project_Transaction.Percentage_Completed), MAX(Project_Transaction.Percentage_Completed) FROM Project_Transaction;
+-----------------------------------------------+-----------------------------------------------+
| MIN(Project_Transaction.Percentage_Completed) | MAX(Project_Transaction.Percentage_Completed) |
+-----------------------------------------------+-----------------------------------------------+
|                                          0.01 |                                         30.00 |
+-----------------------------------------------+-----------------------------------------------+
1 row in set (0.00 sec)
```

Advanced Functions

1. Display the status in text format corresponding to the integer format for each projects

CASE

```
mysql> SELECT title, status,  CASE  WHEN status = 0 THEN "Under Waiting"  WHEN status = 1 THEN "Under Development" END AS Status_Text FROM Projects;
+-----------+--------+------------------+
| title     | status | Status_Text      |
+-----------+--------+------------------+
| Sirius    |      0 | Under Waiting    |
| Open PPM  |      0 | Under Waiting    |
| Sirius    |      0 | Under Waiting    |
| Pide Piper |      1 | Under Development |
| Pide Piper |      1 | Under Development |
| Pide Piper |      1 | Under Development |
+-----------+--------+------------------+
6 rows in set (0.00 sec)
```

2. Display the user_status in text format. Use IF to convert to text

IF

```
mysql> SELECT username, active, IF(active=1, "ACTIVE", "DEACTIVATED") AS User_Status FROM Users
+------------+--------+-------------+
| username   | active | User_Status |
+------------+--------+-------------+
| dkowsikpai |      1 | ACTIVE      |
| pide piper |      1 | ACTIVE      |
+------------+--------+-------------+
2 rows in set (0.00 sec)
```

3. Display the details of the current connection

DATABASE, CURRENT_USER, CONNECTION_ID, VERSION

```
mysql> SELECT DATABASE(), CURRENT_USER(), CONNECTION_ID(), VERSION();
+------------+----------------+-----------------+-------------------------+
| DATABASE() | CURRENT_USER() | CONNECTION_ID() | VERSION()               |
+------------+----------------+-----------------+-------------------------+
| open_ppm   | root@localhost |               4 | 5.7.31-0ubuntu0.18.04.1 |
+------------+----------------+-----------------+-------------------------+
1 row in set (0.00 sec)
```

Experiment 9

AIM

Implementation of various aggregate functions in SQL

Description

Implement aggregate functions in MySQL to get valuable insight from the previously created database. Aggregate functions include Average, count, summation, maximum, minimum, standard deviation and variance

Output / Screenshots

1. AVG 2. COUNT

3. SUM

4. STD

5. VARIANCE

```
mysql> SELECT
    ->     Projects.title,
    ->     COUNT(Project_Transaction.Effort_Spent) AS "No of Transactions",
    ->     AVG(Project_Transaction.Effort_Spent) AS "Average",
    ->     SUM(Project_Transaction.Effort_Spent) AS "Sum",
    ->     STD(Project_Transaction.Effort_Spent) AS "Standard Deviation",
    ->     VARIANCE(Project_Transaction.Effort_Spent) AS "Variance"
    -> FROM
    ->     Projects JOIN Project_Transaction
    -> ON Projects.ID = Project_Transaction.ProjectID
    -> GROUP BY Projects.title;
+----------+--------------------+-----------+--------+--------------------+-----------+
| title    | No of Transactions | Average   | Sum    | Standard Deviation | Variance  |
+----------+--------------------+-----------+--------+--------------------+-----------+
| Open PPM |                  8 | 14.275000 | 114.20 |  8.424925815697133 | 70.979375 |
| Sirius   |                  2 | 18.000000 |  36.00 |                  8 |        64 |
+----------+--------------------+-----------+--------+--------------------+-----------+
2 rows in set (0.01 sec)
```

6. MIN, MAX

```
mysql> SELECT MIN(Project_Transaction.Percentage_Completed), MAX(Project_Transaction.Percentage_Completed) FROM Project_Transaction;
+-----------------------------------------------+-----------------------------------------------+
| MIN(Project_Transaction.Percentage_Completed) | MAX(Project_Transaction.Percentage_Completed) |
+-----------------------------------------------+-----------------------------------------------+
|                                          0.01 |                                         30.00 |
+-----------------------------------------------+-----------------------------------------------+
1 row in set (0.00 sec)
```