

NAME: Atharva Chavan
TE-A IT
ROLL NO: T1851010
PRN: 71901316L

Group B: SQL & PL/SQL

Assignment No:6

Aim: Execute the aggregate functions like count, sum, avg etc. on the suitable database. Make use of built in functions according to the need of the database chosen. Retrieve the data from the database based on time and date functions like now (), date (), day (), time () etc. Use group by and having clauses.

Objective:

- To understand and implement various types of function in MySQL.
- To learn the concept of group functions

NUMBER FUNCTION:

Abs(n) :Select abs(-15) from dual;

Exp(n): Select exp(4) from dual;

Power(m,n): Select power(4,2) from dual;

Mod(m,n): Select mod(10,3) from dual;

Round(m,n): Select round(100.256,2) from dual;

Trunc(m,n): ;Select trunc(100.256,2) from dual;

Sqrt(m,n);Select sqrt(16) from dual;

Aggregate Functions:

1. Count: COUNT following by a column name returns the count of tuple in that column. If DISTINCT keyword is used then it will return only the count of unique tuple in the column. Otherwise, it will return count of all the tuples (including duplicates) count (*) indicates all the tuples of the column.

Syntax: COUNT (Column name)

Example: SELECT COUNT (Sal) FROM emp;

2. SUM: SUM followed by a column name returns the sum of all the values in that column. **Syntax:** SUM (Columnname)

Example: SELECT SUM (Sal) From emp;

3. AVG: AVG followed by a column name returns the average value of that column values. **Syntax:** AVG (n1,n2...)

Example: Select AVG (10, 15, 30) FROM DUAL;

4. MAX: MAX followed by a column name returns the maximum value of that column. *Syntax:* MAX (Columnname)

Example: SELECT MAX (Sal) FROM emp;

mysql> select deptno, max(sal) from emp group by deptno;

DEPTNO	MAX (SAL)
--------	-----------

----	-----
------	-------

10	5000
----	------

20	3000
----	------

30	2850
----	------

mysql> select deptno, max (sal) from emp group by deptno having max(sal)<3000;

DEPTNO	MAX(SAL)
--------	----------

----	-----
------	-------

30	2850
----	------

5. MIN: MIN followed by column name returns the minimum value of that column.

Syntax: MIN (Column name)

Example: SELECT MIN (Sal) FROM emp;

mysql> select deptno,min(sal) from emp group by deptno having min(sal)>1000;

DATE FUNCTIONS:

CURDATE()

Returns the current date as a value in 'YYYY-MM-DD' or YYYYMMDD format, depending on whether the function is used in a string or numeric context.

```
mysql> SELECT CURDATE();
```

```
-----
```

```
-----
```

```
-----
```

```
---+
```

```
+
```

```
|
```

```
CURDATE()
```

```
|
```

```
-----
```

```
-----
```

```
-----
```

```
-----
-----
-----
-----+
```

1 row in set (0.00 sec)

DAY(date)

DAY() is a synonym for DAYOFMONTH().

DAYNAME(date)

Returns the name of the weekday for date.

```
mysql> SELECT DAYNAME('1998-02-05');
```

```
-----
-----
-----
-----
```

```
-+
```

```
+
```

```
| DAYNAME('1998-02-05') |
```

```
-----
-----
-----
-----
```

```
-+
```

```
+
```

```
| Thursday |
```

```
-----
-----
-----
-----
```

```
-+
```

```
+
```

1 row in set (0.00 sec)

HOUR(time)

Returns the hour for the time. The range of the return value is 0 to 23 for time-of-day values. However, the range of TIME values actually is much larger, so HOUR can return values greater than 23.

```
mysql> SELECT HOUR('10:05:03');
```

```

-----
-----
-----
-----+
|          HOUR('10:05:03')          |
-----
-----
-----
-----+
|          10          |
-----
-----+

```

1 row in set (0.00 sec)

LAST_DAY(date)

Takes a date or datetime value and returns the corresponding value for the last day of the month. Returns NULL if the argument is invalid.

```

mysql> SELECT LAST_DAY('2003-02-05');

-----
-----
-----
-----
-----+
|          LAST_DAY('2003-02-05')          |
-----
-----
-----
-----+
|          2003-02-28          |
-----
-----
-----
-----+

```

1 row in set (0.00 sec)

MINUTE(time)

Returns the minute for time, in the range 0 to 59.

```
mysql> SELECT MINUTE('98-02-03 10:05:03');
```

```
-----
-----
-----
-----
-----
--+                                     +
|      MINUTE('98-02-03 10:05:03')      |
-----
-----
-----
-----
-----
--+                                     +
|              5              |
-----
-----
-----
-----
-----
--+                                     +
```

1 row in set (0.00 sec)

MONTH(date)

Returns the month for date, in the range 0 to 12.

```
mysql> SELECT MONTH('1998-02-03')
```

```
-----
-----
-----
-----+                                     +
|      MONTH('1998-02-03')      |
-----
-----
-----
-----+                                     +
|              2              |
-----
-----
-----
-----+                                     +
```

1 row in set (0.00 sec)

MONTHNAME(date)

Returns the full name of the month for date.

```
mysql> SELECT MONTHNAME('1998-02-05');
```

```
-----+
-----+
-----+
-----+
      +
|      MONTHNAME('1998-02-05')      |
-----+
-----+
-----+
-----+
      +
|      February      |
-----+
-----+
-----+
-----+
      +
```

1 row in set (0.00 sec)

NOW()

Returns the current date and time as a value in 'YYYY-MM-DD HH:MM:SS' or YYYYMMDDHHMMSS format, depending on whether the function is used in a string or numeric context. The value is expressed in the current time zone.

```
mysql> SELECT NOW();
```

```
-----+
-----+
-----+
-----+
      +
|      NOW()      |
-----+
-----+
-----+
-----+
      +
```

	1997-12-15 23:50:26	

-+		+

1 row in set (0.00 sec)

GROUP BY: This query is used to group to all the records in a relation together for each and every value of a specific key(s) and then display them for a selected set of fields the relation.

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 : The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions. The HAVING clause must follow the GROUP BY clause in a query and must also precede the ORDER BY clause if used.

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 empno,SUM(SALARY) FROM emp,dept

WHERE emp.deptno =20 GROUP BY empno;

ORDER BY: This query is used to display a selected set of fields from a relation in an ordered manner base on some field.

Syntax: SELECT <set of fields> FROM<relation_name>

ORDER BY<field_name>;

Example: SQL> SELECT empno, ename, job FROM emp ORDER BY job;

LAB PRACTICE ASSIGNMENT:

Consider the following table structure for this assignment:

CUSTOMER(Cust_id, C_name, City)

BRANCH(Branch_id, bname, City)

DEPOSIT(Acc_no , Cust_id, Amount, Branch_id, Open_date)

BORROW(Loan_no, Cust_id, Branch_id, Amount)

Perform the following queries on the above table:

- 1) List totalloan.
- 2) List totaldeposit.
- 3) List maximum deposit of customers living inMumbai.
- 4) Count total number of branchcities.
- 5) List branch_id and branch wisedeposit.
- 6) List the branches having sum of deposit more than4000.
- 7) List the names of customers having minimumdeposit.
- 8) Count the number of depositors living in'nagpur'.
- 9) Find the maximum deposit of the Akurdibranch.
- 10) Find out number of customers living inPune.
- 11) Find out the customers who are not living in Pune orMumbai.
- 12) List out Cust_id and C_name in descending order of theirC_name.
- 13) Display the number of depositors in branchwise.
- 14) Find out the branch which has notborrowers.
- 15) How many customers have opened deposit after'01-01-2016'

Conclusion:-

In this assignment, we have learned and executed Aggregate and date functions of MYSQL.

Code & Output: -

```
ATHARVA@BRAINMETRON:~$ sudomysql -u root #  
Swipe your finger across the fingerprint reader
```

```
mysql> use ATHARVA
```

```
Database changed
```

```
mysql> select * from employee;
```

```
+-----+-----+-----+-----+  
| EID | Ename   | Salary | Designation |  
+-----+-----+-----+-----+  
| 1001 | ATHARVA | 40000 | Developer   |  
| 1002 | Devansh | 40000 | Developer   |  
| 1003 | Monika  | 60000 | HR          |  
| 1005 | Jaya    | 40000 | Manager     |  
| 1007 | abuzz   | 20000 | Peon        |  
| 1008 | Shreyash | 500   | Worker      |  
| 1006 | Qwerty  | 500   | Worker      |  
| 1009 | Shivk   | 50000 | Manager     |  
| 1010 | aniket  | 500   | Peon        |  
+-----+-----+-----+-----+  
9 rows in set (0.00 sec)
```

```
mysql> select Designation from employee group by Designation having count(*)<1;  
Empty set (0.00 sec)
```

```
mysql> select Designation from employee group by Designation having count(*)<2;
```

```
+-----+  
| Designation |  
+-----+  
| HR          |  
+-----+  
1 row in set (0.00 sec)
```

```
mysql> select Designation from employee group by Designation,Salary having count(*)<2;
```

```
+-----+  
| Designation |  
+-----+  
| HR          |  
| Manager     |  
| Manager     |
```

Peon
Peon

5 rows in set (0.00 sec)

```
mysql> select Designation,Salary from employee group by Designation,Salary having count(*)<2;
```

Designation	Salary
HR	60000
Manager	40000
Manager	50000
Peon	500
Peon	20000

5 rows in set (0.00 sec)

```
mysql> select Ename from employee where Designation in ( select Designation from employee group by Designation having count(*)<2);
```

Ename
Monika

1 row in set (0.00 sec)

```
mysql> select * from employee where Designation in ( select Designation from employee group by Designation having count(*)<2);
```

EID	Ename	Salary	Designation
1003	Monika	60000	HR

1 row in set (0.00 sec)

```
mysql> select Ename from employee where Designation in ( select Designation from employee group by Designation having count(*)<2);
```

Ename
Monika

1 row in set (0.00 sec)

```
mysql> select Ename from employee where Designation not in ( select Designation from employee group by Designation having count(*)<2);
```

Ename

```
| ATHARVA |  
| Devansh |  
| Jaya    |  
| abuzz   |  
| Shreyash|  
| Qwerty  |  
| Shivk   |  
| aniket  |
```

```
+-----+
```

8 rows in set (0.00 sec)

```
mysql> alter table employee add constraint employee_pk primary key(EID);
```

Query OK, 0 rows affected (0.85 sec)

Records: 0 Duplicates: 0 Warnings: 0

```
mysql> create table project( Pro_ID int, EID int not null, location varchar(20), foreign key(EID)  
references employee(EID));
```

Query OK, 0 rows affected (0.45 sec)

```
mysql> alter table employee add address varchar(20);
```

Query OK, 0 rows affected (0.83 sec)

Records: 0 Duplicates: 0 Warnings: 0

```
mysql> select * from employee;
```

```
+-----+-----+-----+-----+-----+  
| EID | Ename  | Salary | Designation | address |  
+-----+-----+-----+-----+-----+  
| 1001 | ATHARVA | 40000 | Developer  | NULL    |  
| 1002 | Devansh | 40000 | Developer  | NULL    |  
| 1003 | Monika  | 60000 | HR         | NULL    |  
| 1005 | Jaya    | 40000 | Manager    | NULL    |  
| 1006 | Qwerty  | 500   | Worker     | NULL    |  
| 1007 | abuzz   | 20000 | Peon       | NULL    |  
| 1008 | Shreyash | 500   | Worker     | NULL    |  
| 1009 | Shivk   | 50000 | Manager    | NULL    |  
| 1010 | aniket  | 500   | Peon       | NULL    |  
+-----+-----+-----+-----+-----+
```

9 rows in set (0.00 sec)

```
mysql> update employee set address='Pune' where EID=1001;
```

Query OK, 1 row affected (0.12 sec)

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> update employee set address='Pune' where EID=1002;
```

Query OK, 1 row affected (0.06 sec)

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> update employee set address='Pune' where EID=1003;
```

Query OK, 1 row affected (0.06 sec)

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> update employee set address='Nagpur' where EID=1004;  
Query OK, 0 rows affected (0.00 sec)
```

Rows matched: 0 Changed: 0 Warnings: 0

```
mysql> update employee set address='Nagpur' where EID=1005;  
Query OK, 1 row affected (0.06 sec)
```

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> update employee set address='Nagpur' where EID=1006;  
Query OK, 1 row affected (0.10 sec)
```

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> update employee set address='Nagpur' where EID=1007;  
Query OK, 1 row affected (0.07 sec)
```

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> update employee set address='Agra' where EID=1008;  
Query OK, 1 row affected (0.06 sec)
```

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> update employee set address='Agra' where EID=1009;  
Query OK, 1 row affected (0.10 sec)
```

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> update employee set address='Agra' where EID=1010;  
Query OK, 1 row affected (0.09 sec)
```

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> select * from employee;
```

EID	Ename	Salary	Designation	address
1001	ATHARVA	40000	Developer	Pune
1002	Devansh	40000	Developer	Pune
1003	Monika	60000	HR	Pune
1005	Jaya	40000	Manager	Nagpur
1006	Qwerty	500	Worker	Nagpur
1007	abuzz	20000	Peon	Nagpur
1008	Shreyash	500	Worker	Agra
1009	Shivk	50000	Manager	Agra
1010	aniket	500	Peon	Agra

9 rows in set (0.00 sec)

```
mysql> insert into employee values (1004,'Sushil',2500000,'CEO','Nagpur');  
Query OK, 1 row affected (0.07 sec)
```

```
mysql> select * from employee;
```

```

+-----+-----+-----+-----+-----+
| EID | Ename | Salary | Designation | address |
+-----+-----+-----+-----+-----+
| 1001 | ATHARVA | 40000 | Developer | Pune |
| 1002 | Devansh | 40000 | Developer | Pune |
| 1003 | Monika | 60000 | HR | Pune |
| 1004 | Sushil | 2500000 | CEO | Nagpur |
| 1005 | Jaya | 40000 | Manager | Nagpur |
| 1006 | Qwerty | 500 | Worker | Nagpur |
| 1007 | abuzz | 20000 | Peon | Nagpur |
| 1008 | Shreyash | 500 | Worker | Agra |
| 1009 | Shivk | 50000 | Manager | Agra |
| 1010 | aniket | 500 | Peon | Agra |
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

```

mysql> describe project;

```

+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Pro_ID | int(11) | YES | | NULL | |
| EID | int(11) | NO | MUL | NULL | |
| location | varchar(20) | YES | | NULL | |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

```

mysql> alter table project add Pname varchar(20);
Query OK, 0 rows affected (0.68 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> describe project;

```

+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Pro_ID | int(11) | YES | | NULL | |
| EID | int(11) | NO | MUL | NULL | |
| location | varchar(20) | YES | | NULL | |
| Pname | varchar(20) | YES | | NULL | |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

mysql> insert into project values (3004,1001,'Nagpur','Pied PiPer');
Query OK, 1 row affected (0.06 sec)

mysql> insert into project values (3005,1002,'Nagpur','Hooli');
Query OK, 1 row affected (0.07 sec)

mysql> insert into project values (3006,1003,'Pune','Slice line');
Query OK, 1 row affected (0.07 sec)

```
mysql> insert into project values (3006,1004,'Nagpur','Bachmanity');
Query OK, 1 row affected (0.06 sec)
```

```
mysql> insert into project values (3007,1005,'Nagpur','Eklow');
Query OK, 1 row affected (0.06 sec)
```

```
mysql> insert into project values (3008,1006,'Pune','Bream Hall');
Query OK, 1 row affected (0.07 sec)
```

```
mysql> select * from project;
```

```
+-----+-----+-----+-----+
| Pro_ID | EID  | location | Pname   |
+-----+-----+-----+-----+
| 3004   | 1001 | Nagpur   | Pied PiPer |
| 3005   | 1002 | Nagpur   | Hooli      |
| 3006   | 1003 | Pune     | Slice line |
| 3006   | 1004 | Nagpur   | Bachmanity |
| 3007   | 1005 | Nagpur   | Eklow      |
| 3008   | 1006 | Pune     | Bream Hall |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

```
mysql> select * from employee where address in ( Nagpur, Pune);
ERROR 1054 (42S22): Unknown column 'Nagpur' in 'where clause'
mysql> select * from employee where address in ( 'Nagpur','Pune');
```

```
+-----+-----+-----+-----+-----+
| EID  | Ename  | Salary | Designation | address |
+-----+-----+-----+-----+-----+
| 1001 | ATHARVA | 40000 | Developer   | Pune    |
| 1002 | Devansh | 40000 | Developer   | Pune    |
| 1003 | Monika  | 60000 | HR          | Pune    |
| 1004 | Sushil  | 2500000 | CEO        | Nagpur  |
| 1005 | Jaya    | 40000 | Manager     | Nagpur  |
| 1006 | Qwerty  | 500    | Worker      | Nagpur  |
| 1007 | abuzz   | 20000 | Peon        | Nagpur  |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

```
ATHARVA@BRAINMETRON:~$ sudomysql -u root #
Swipe your finger across the fingerprint reader
```

```
mysql> use ATHARVA;
Database changed
```

```
mysql> select * from employee;
```

EID	Ename	Salary	Designation	address
1001	ATHARVA	40000	Developer	Pune
1002	Devansh	40000	Developer	Pune
1003	Monika	60000	HR	Pune
1004	Sushil	2500000	CEO	Nagpur
1005	Jaya	40000	Manager	Nagpur
1006	Qwerty	500	Worker	Nagpur
1007	abuzz	20000	Peon	Nagpur
1008	Shreyash	500	Worker	Agra
1009	Shivk	50000	Manager	Agra
1010	aniket	500	Peon	Agra

10 rows in set (0.00 sec)

```
mysql> select max(salary) from employee;
```

max(salary)
2500000

1 row in set (0.00 sec)

```
mysql> select min(salary) from employee;
```

min(salary)
500

1 row in set (0.01 sec)

```
mysql> select count(salary) from employee;
```

count(salary)
10

1 row in set (0.00 sec)

```
mysql> select sum(salary) from employee;
```

sum(salary)
2751500

1 row in set (0.00 sec)


```
mysql> select avg(salary) from employee;
```

```
+-----+  
| avg(salary) |  
+-----+  
| 275150.0000 |
```

```
+-----+  
1 row in set (0.00 sec)
```

```
mysql> select curdate();
```

```
+-----+  
| curdate() |  
+-----+  
| 2019-07-09 |
```

```
+-----+  
1 row in set (0.00 sec)
```

```
mysql> select curtime();
```

```
+-----+  
| curtime() |  
+-----+  
| 13:34:06 |
```

```
+-----+  
1 row in set (0.00 sec)
```

```
mysql> select date('2003-12-23 12:40:30');
```

```
+-----+  
| date('2003-12-23 12:40:30') |  
+-----+  
| 2003-12-23 |
```

```
+-----+  
1 row in set (0.00 sec)
```

```
mysql> select time('2003-12-23 12:40:30');
```

```
+-----+  
| time('2003-12-23 12:40:30') |  
+-----+  
| 12:40:30 |
```

```
+-----+  
1 row in set (0.00 sec)
```

```
mysql> select dayname('2003-12-23');
```

```
+-----+  
| dayname('2003-12-23') |  
+-----+  
| Tuesday |
```

```
+-----+  
1 row in set (0.00 sec)
```

```
mysql> select hour('12:40:30');
```

```
+-----+
```

```
| hour('12:40:30') |
+-----+
|          12 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select min('12:40:30');
+-----+
| min('12:40:30') |
+-----+
| 12:40:30      |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select second('12:40:30');
```

```
+-----+
| second('12:40:30') |
+-----+
|          30 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select last_day('2003-12-23');
+-----+
| last_day('2003-12-23') |
+-----+
| 2003-12-31      |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select monthname('2003-12-23');
+-----+
| monthname('2003-12-23') |
+-----+
| December          |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select now();
+-----+
| now()          |
+-----+
| 2019-07-09 13:40:56 |
+-----+
1 row in set (0.00 sec)
```