

Using Basic Functions in MySQL - I

Session 12



Objectives

- ◆ *Use the Aggregate functions in MySQL*
- ◆ *Use the Mathematical functions in MySQL*

For Aptech Center Use Only

- ◆ In MySQL, you can use functions along with the SELECT command in two different ways:

- ◆ **Value to be retrieved:**

- ◆ In this form, the function is used with column names
- ◆ Consider the following query:

```
SELECT E_ID, E_FNAME, LENGTH(E_FNAME) FROM  
EMP_DETAILS WHERE E_ID=101;
```

- ◆ This query returns the E_ID and E_FNAME columns from the EMP_DETAILS table with the length of E_FNAME

◆ Part of a **WHERE** clause:

- ◆ In this form, the function is used in the **WHERE** clause
- ◆ The value specified is compared for each row in the table
- ◆ Consider the following query:

```
SELECT E_ID, D_NAME FROM EMP_SALARY HAVING  
AVG(BASIC_SAL) < 8000;
```

Using Aggregate Functions in MySQL

- ◆ Aggregate functions operate on a group of values and return a single value as the final result
- ◆ Some aggregate functions work with the `GROUP BY` and `HAVING` clause
- ◆ The `GROUP BY` clause is used to group rows having similar values for a specific column into a single row
- ◆ Group functions do not accept `NULL` values
- ◆ The `HAVING` clause defines the result set based on a set of calculations
- ◆ You can use the `GROUP BY` and `HAVING` clause with the `SELECT` statement to retrieve data that satisfies the specified conditions defined in the `HAVING` clause

- ◆ The `AVG` function returns the mean value of the argument

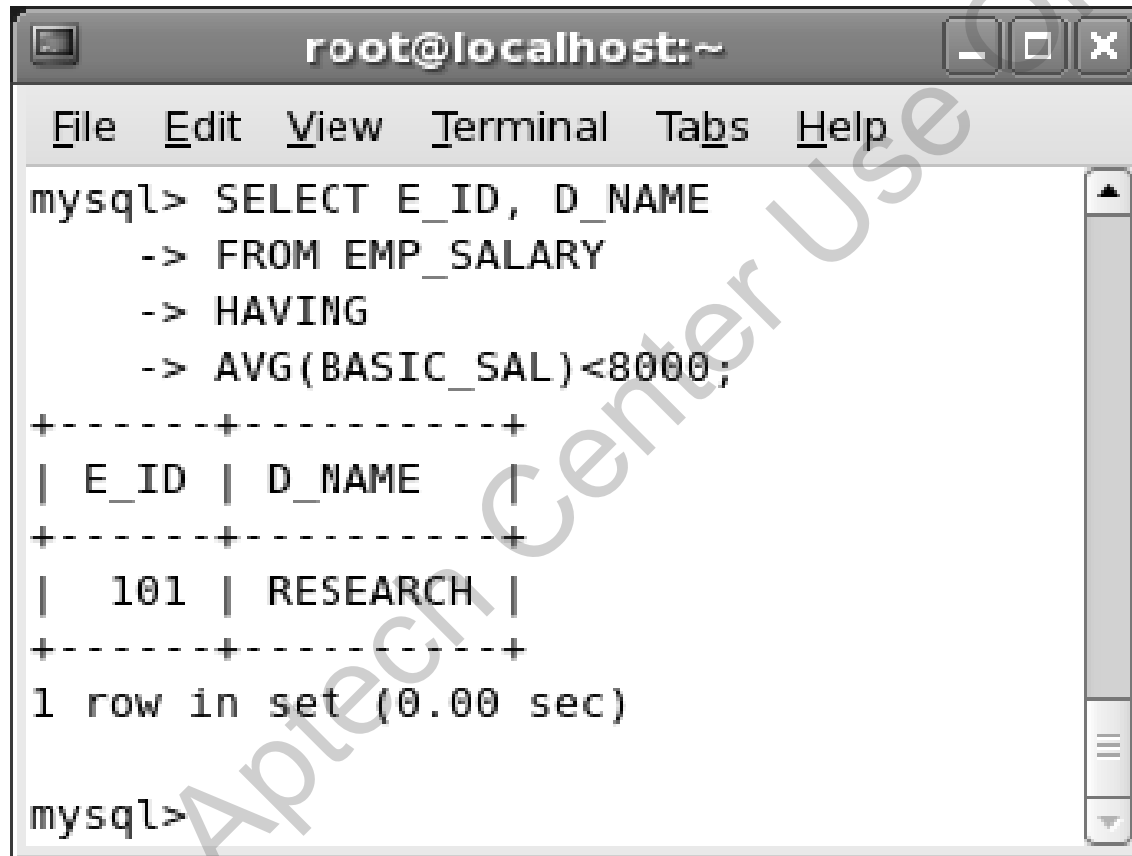
- ◆ The syntax for using this function is:

```
SELECT COLUMN_NAME FROM TABLE_NAME HAVING  
AVG(expression);
```

- ◆ For example, to obtain the `E_ID` and `D_NAME` of the employee whose average basic salary is less than 8000, enter the following command at the command prompt:

```
SELECT E_ID, D_NAME FROM EMP_SALARY HAVING  
AVG(BASIC_SAL)<8000;
```

Figure displays the output of the command



The screenshot shows a terminal window titled 'root@localhost:~'. The terminal contains the following text:

```
mysql> SELECT E_ID, D_NAME
-> FROM EMP_SALARY
-> HAVING
-> AVG(BASIC_SAL)<8000;
+-----+-----+
| E_ID | D_NAME |
+-----+-----+
| 101 | RESEARCH |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

A large, diagonal watermark reading 'For Apteck Center Use Only' is overlaid on the terminal output.

- ◆ The `BITWISE AND` function works with two arguments that have equal length
- ◆ `BITWISE AND` converts the arguments into binary format and compares every bit of the two arguments
- ◆ The syntax for this function is:

```
SELECT BIT1 & BIT2;
```

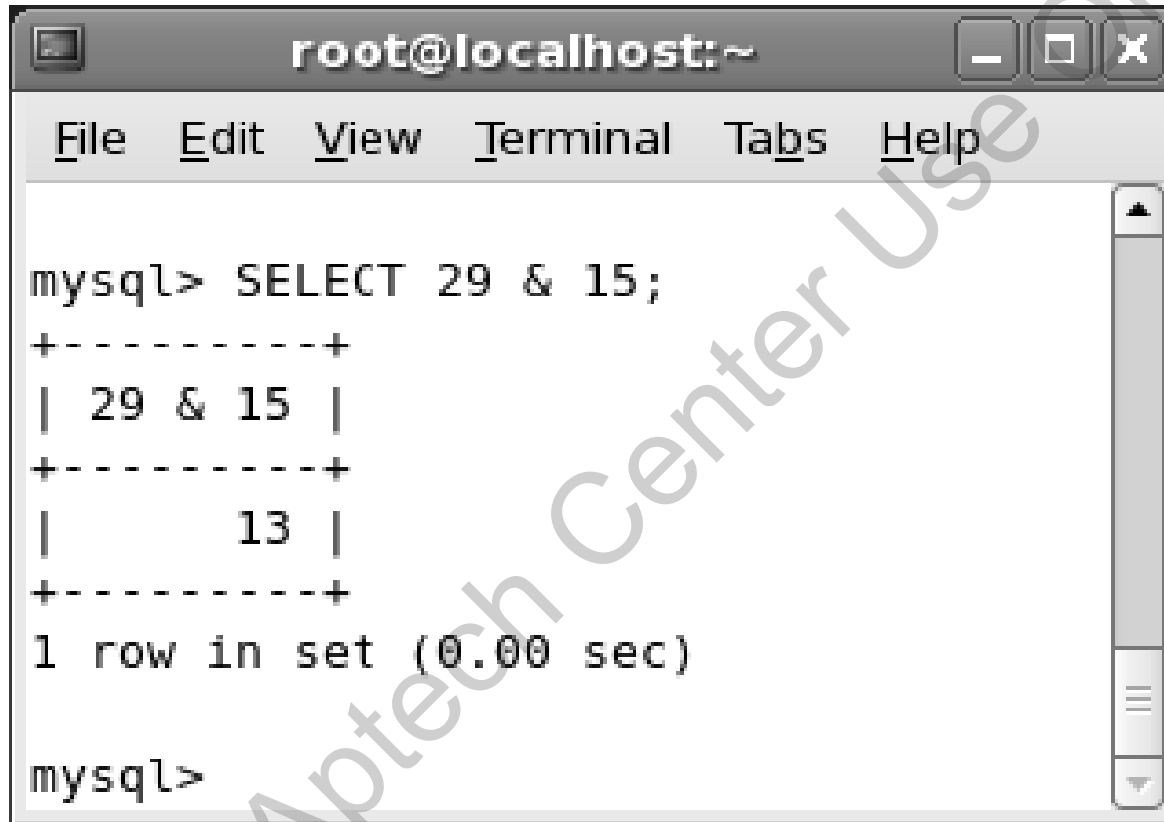
where,

`BIT1, BIT2` – defines the operands

- ◆ To find out the `BITWISE AND` for 29 and 15, enter the following command at the command prompt:

```
SELECT 29 & 15;
```


Figure displays the output of the command



The screenshot shows a terminal window titled 'root@localhost:~'. The terminal displays the MySQL command prompt 'mysql>' followed by the command 'SELECT 29 & 15;'. The output is a table with one row and one column, showing the result of the bitwise AND operation. The table is formatted with a header row and a data row, separated by dashed lines. The header row shows '29 & 15' and the data row shows '13'. Below the table, it says '1 row in set (0.00 sec)'. The prompt 'mysql>' is shown again at the bottom.

```
mysql> SELECT 29 & 15;
+-----+
| 29 & 15 |
+-----+
|      13 |
+-----+
1 row in set (0.00 sec)

mysql>
```

- ◆ The COUNT function returns the total number of non-NULL values of the expression specified as the argument

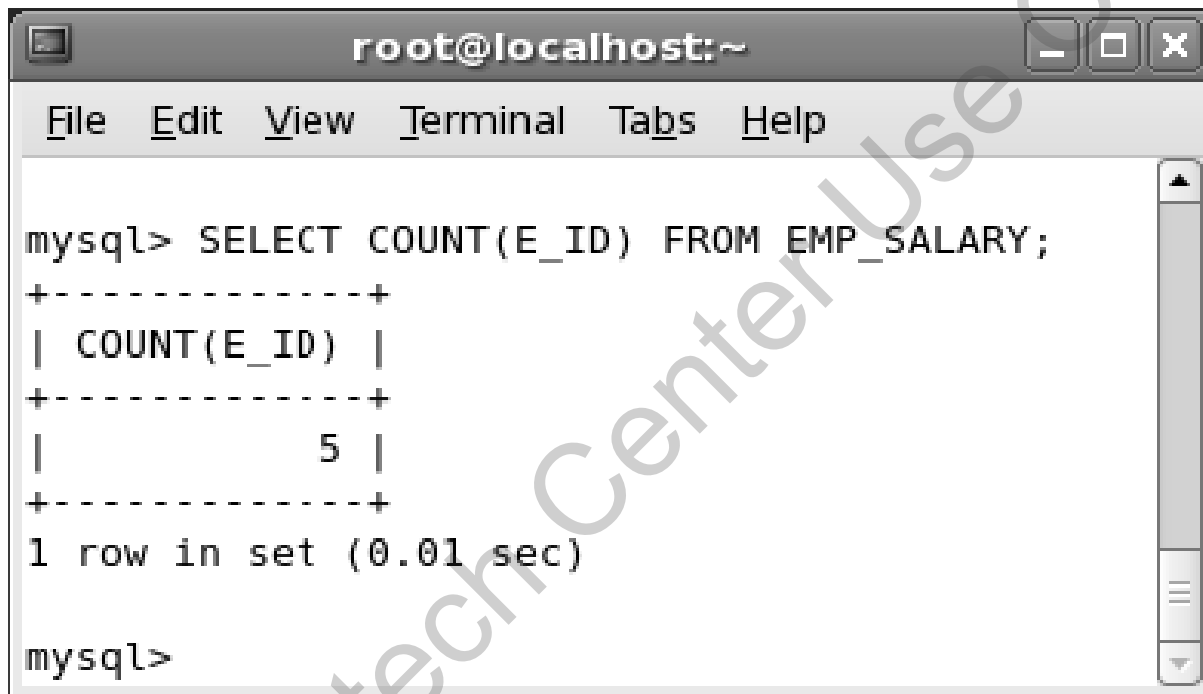
- ◆ The syntax for obtaining the count of any expression is:

```
SELECT COUNT(expression) FROM TABLE_NAME;
```

- ◆ For example, to obtain the count of the E_ID column from the EMP_SALARY table, enter the following command at the command prompt:

```
SELECT COUNT(E_ID) FROM EMP_SALARY;
```

Figure displays the output of the command



The screenshot shows a terminal window titled 'root@localhost:~'. The terminal contains the following text:

```
mysql> SELECT COUNT(E_ID) FROM EMP_SALARY;  
+-----+  
| COUNT(E_ID) |  
+-----+  
|          5 |  
+-----+  
1 row in set (0.01 sec)  
  
mysql>
```

The output displays a single row with the value 5, representing the count of E_IDs in the EMP_SALARY table. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'.

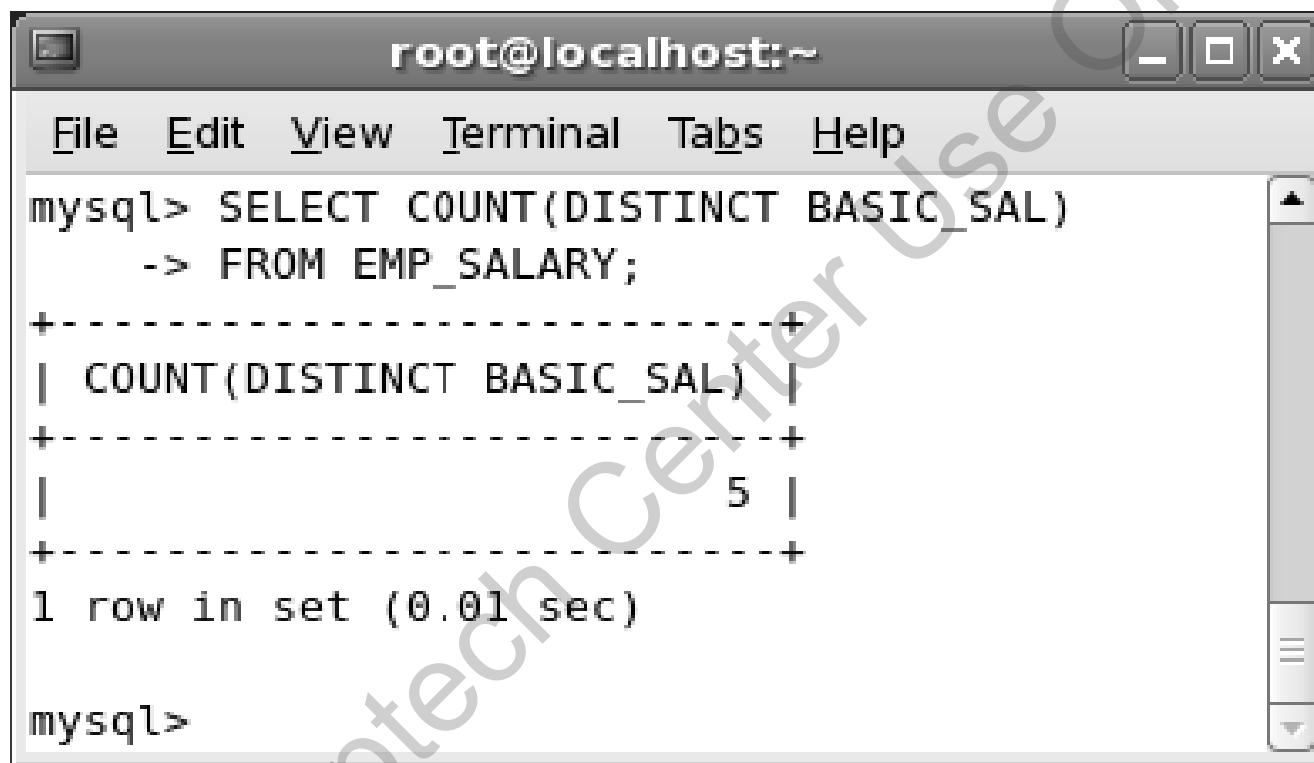
- ◆ The COUNT (DISTINCT) function returns the number of unique values in a column
- ◆ You can retrieve the number of different combinations of unique values in two or more columns of the same table
- ◆ If there are no matching values in any of the columns, then the COUNT (DISTINCT) function returns 0
- ◆ The syntax for using this function is:

```
SELECT COUNT (DISTINCT expression) FROM TABLE_NAME;
```

- ◆ For example, to obtain the number of employees who have different basic salary, enter the following command at the command prompt:

```
SELECT COUNT (DISTINCT BASIC_SAL) FROM EMP_SALARY;
```

Figure displays the output of the command



A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows a MySQL command and its output. The command is 'SELECT COUNT(DISTINCT BASIC_SAL) FROM EMP_SALARY;'. The output is a single row with the value 5. Below the output, it says '1 row in set (0.01 sec)'. The prompt 'mysql>' is visible at the bottom.

```
mysql> SELECT COUNT(DISTINCT BASIC_SAL)
-> FROM EMP_SALARY;
+-----+
| COUNT(DISTINCT BASIC_SAL) |
+-----+
|                    5      |
+-----+
1 row in set (0.01 sec)

mysql>
```

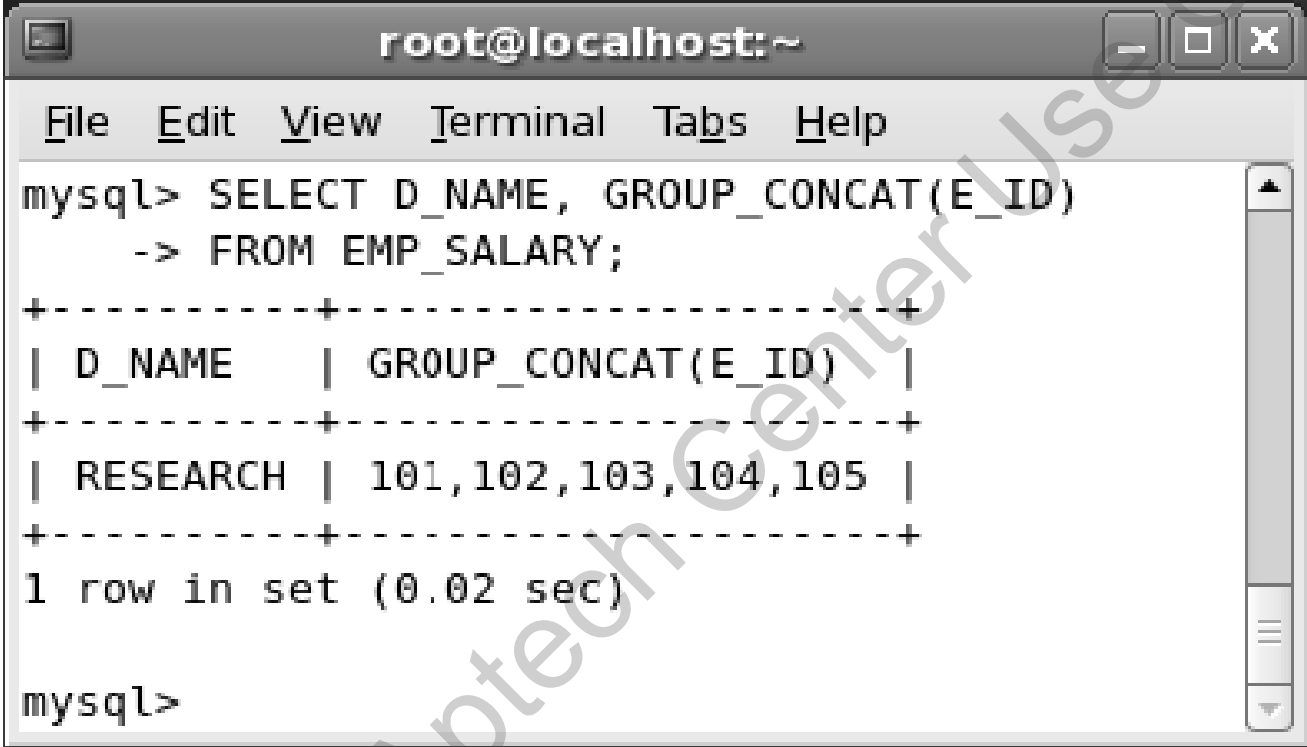
- ◆ The GROUP_CONCAT function joins the unique values of the argument and returns them as a string
- ◆ This function returns NULL if the argument has no non-NULL values
- ◆ To eliminate duplicate values, you can use the DISTINCT clause
- ◆ The syntax for using this function is:

```
SELECT GROUP_CONCAT(expression) FROM TABLE_NAME;
```

- ◆ For example, to display all the E_ID from the EMP_SALARY table that have RESEARCH as the D_NAME, enter the following command at the command prompt:

```
SELECT D_NAME, GROUP_CONCAT(E_ID) FROM EMP_SALARY;
```

Figure displays the output of the command



```
root@localhost:~  
File Edit View Terminal Tabs Help  
mysql> SELECT D_NAME, GROUP_CONCAT(E_ID)  
-> FROM EMP_SALARY;  
+-----+-----+  
| D_NAME | GROUP_CONCAT(E_ID) |  
+-----+-----+  
| RESEARCH | 101,102,103,104,105 |  
+-----+-----+  
1 row in set (0.02 sec)  
  
mysql>
```

The image shows a terminal window titled 'root@localhost:~'. The window contains a MySQL command prompt where the user has entered the command 'SELECT D_NAME, GROUP_CONCAT(E_ID) FROM EMP_SALARY;'. The output is displayed in a table format with two columns: 'D_NAME' and 'GROUP_CONCAT(E_ID)'. The first row shows 'RESEARCH' and '101,102,103,104,105'. Below the table, it says '1 row in set (0.02 sec)'. The prompt 'mysql>' is shown again at the bottom.

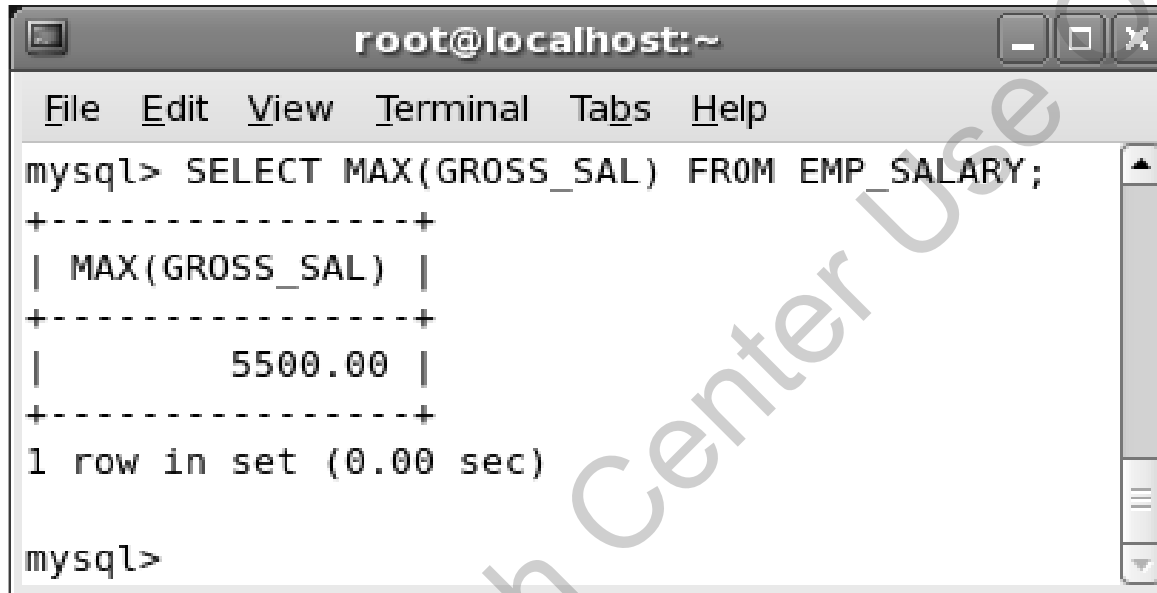
- ◆ The MAX function returns the greatest value of the expression
- ◆ The syntax for obtaining the greatest value is:

```
SELECT COLUMN_NAME FROM TABLE_NAME HAVING  
MAX(expression);
```

- ◆ For example, display the maximum gross salary earned by any of the employee

```
SELECT MAX(GROSS_SAL) FROM EMP_SALARY;
```


Figure displays the output of the command



A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows a MySQL command and its output. The command is 'SELECT MAX(GROSS_SAL) FROM EMP_SALARY;'. The output is a table with one row: 'MAX(GROSS_SAL)' with the value '5500.00'. Below the table, it says '1 row in set (0.00 sec)'. The prompt 'mysql>' is shown at the bottom.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
mysql> SELECT MAX(GROSS_SAL) FROM EMP_SALARY;  
+-----+  
| MAX(GROSS_SAL) |  
+-----+  
|          5500.00 |  
+-----+  
1 row in set (0.00 sec)  
mysql>
```

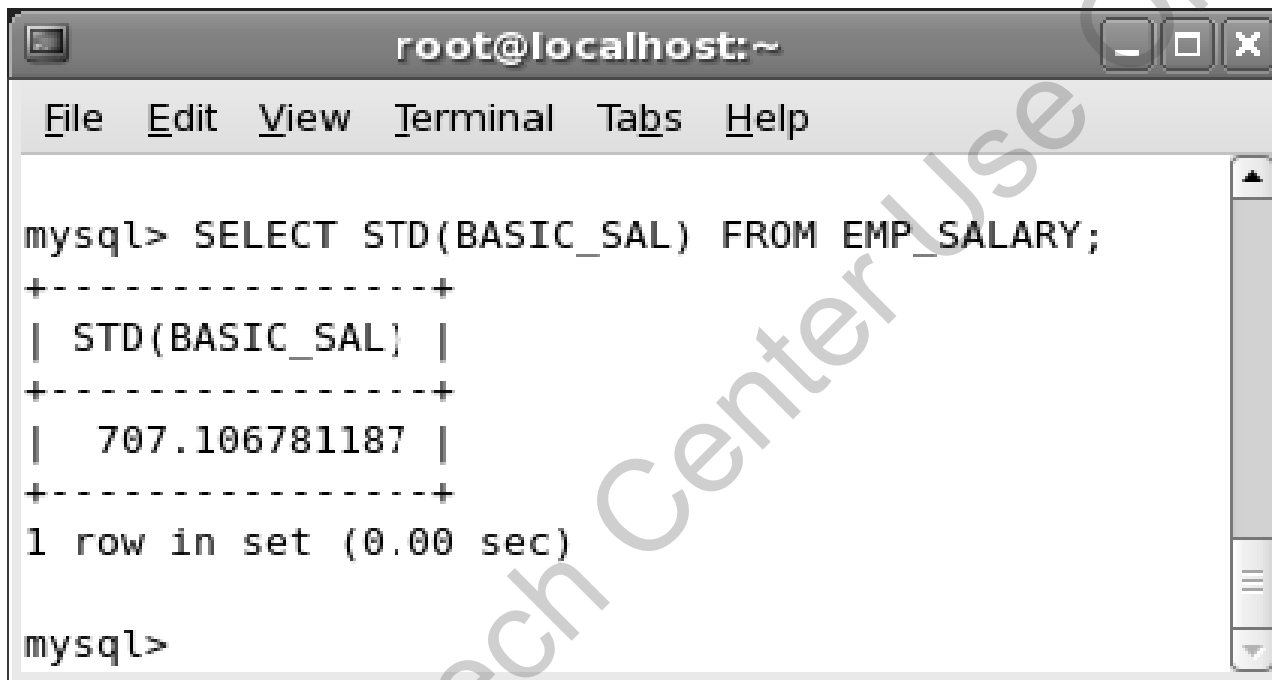
- ◆ The STD function returns the standard deviation of the argument
- ◆ Standard deviation specifies the variation of a population from its mean value
- ◆ The output of this function is NULL if there are no rows satisfying the given query
- ◆ The syntax to use this function is:

```
SELECT STD(expression) FROM TABLE_NAME;
```

- ◆ For example, to obtain the standard deviation of the basic salary of all of the employees, enter the following command at the command prompt:

```
SELECT STD(BASIC_SAL) FROM EMP_SALARY;
```

Figure displays the output of the command



The screenshot shows a MySQL terminal window titled 'root@localhost:~'. The window has a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal content shows the command 'mysql> SELECT STD(BASIC_SAL) FROM EMP_SALARY;' being executed. The output is a single row with the value '707.106781187'. Below the output, it says '1 row in set (0.00 sec)'. The prompt 'mysql>' is visible at the bottom of the terminal.

```
mysql> SELECT STD(BASIC_SAL) FROM EMP_SALARY;
+-----+
| STD(BASIC_SAL) |
+-----+
| 707.106781187 |
+-----+
1 row in set (0.00 sec)

mysql>
```

- ◆ The SUM function adds the values specified in the expression

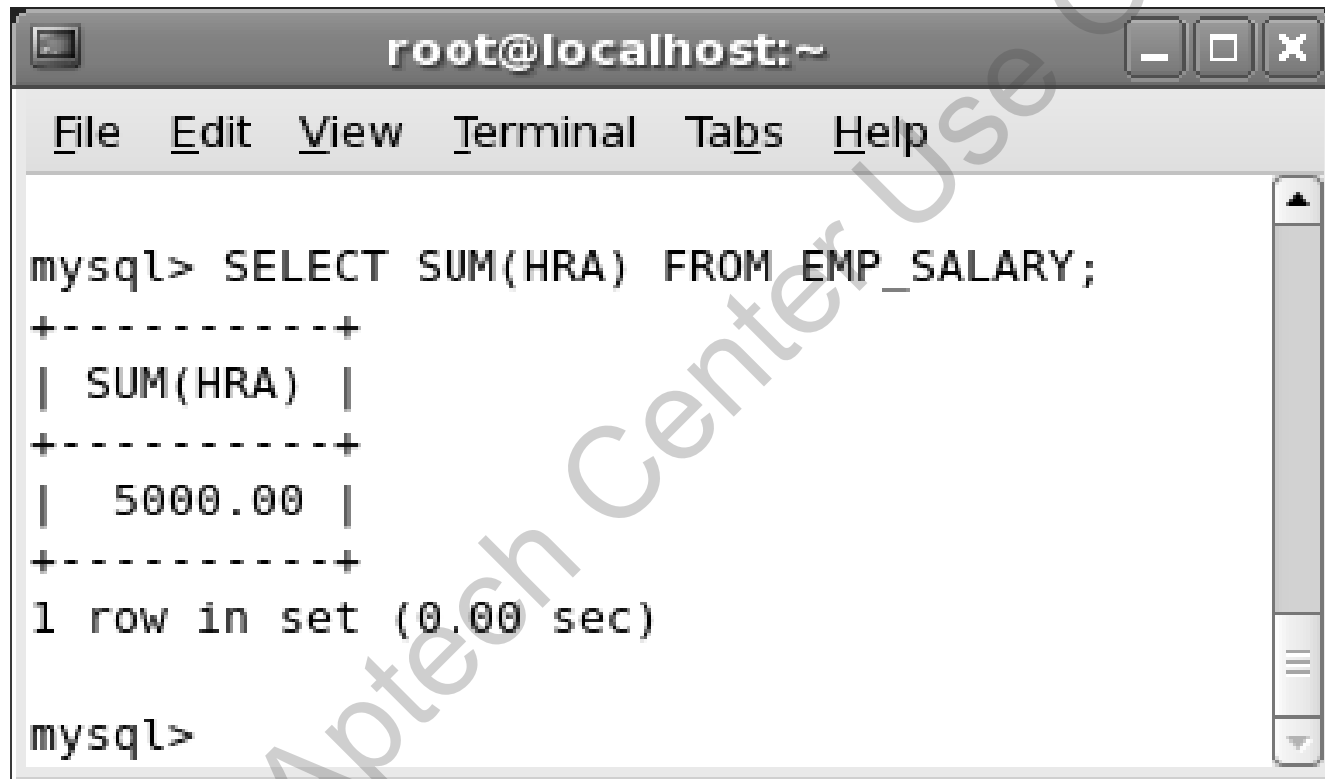
- ◆ The syntax for obtaining the sum is:

```
SELECT SUM(expression) FROM TABLE_NAME;
```

- ◆ For example, to obtain the sum of HRA of all the employees, enter the following command at the command prompt:

```
SELECT SUM(HRA) FROM EMP_SALARY;
```

Figure displays the output of the command



A terminal window titled 'root@localhost:~' with a menu bar containing 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal shows the execution of a MySQL command and its output.

```
mysql> SELECT SUM(HRA) FROM EMP_SALARY;
+-----+
| SUM(HRA) |
+-----+
|  5000.00 |
+-----+
1 row in set (0.00 sec)

mysql>
```

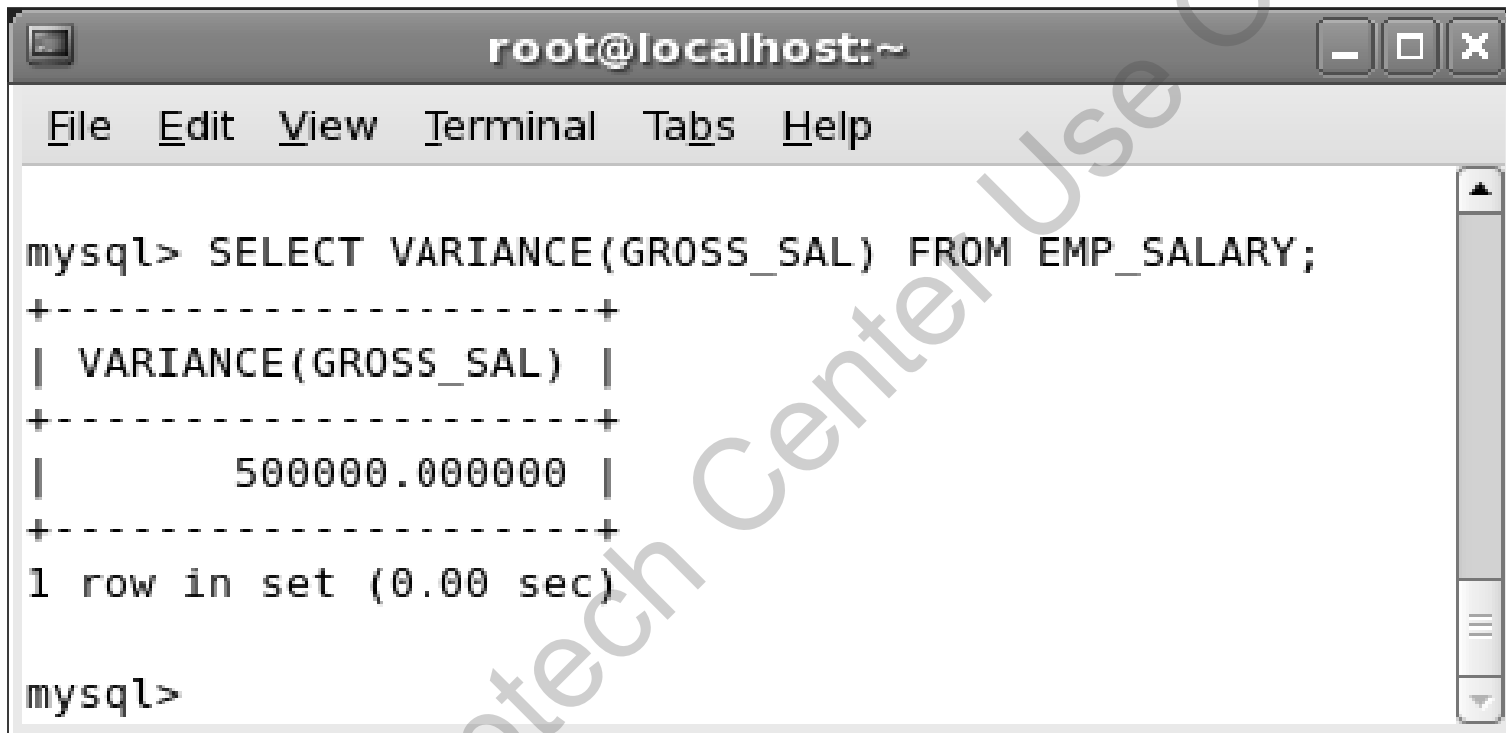
- ◆ The `VARIANCE` function returns the variance of the values of the argument
- ◆ The syntax for obtaining the variance is:

```
SELECT VARIANCE(expression) FROM TABLE_NAME;
```

- ◆ To obtain the variance of gross salary of all the employees, enter the following command at the command prompt:

```
SELECT VARIANCE(GROSS_SAL) FROM EMP_SALARY;
```

Figure displays the output of the command



The screenshot shows a terminal window titled 'root@localhost:~'. The terminal contains the following text:

```
mysql> SELECT VARIANCE(GROSS_SAL) FROM EMP_SALARY;
```

VARIANCE(GROSS_SAL)
500000.000000

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

```
mysql>
```

The output is displayed in a table with one column header 'VARIANCE(GROSS_SAL)' and one row of data '500000.000000'. Below the table, it indicates '1 row in set (0.00 sec)'. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. A large diagonal watermark 'For Aptech Center Use Only' is visible across the image.

Table describes additional AGGREGATE functions supported by MySQL:

Name	Description	Example
BITWISE OR	<p>The BITWISE OR function works with two arguments of equal length. BITWISE OR compares every bit of the two arguments. This function returns 0 only if both the bits being compared equal to 0, else it returns 1. The syntax for this function is:</p> <pre>SELECT BIT1 BIT2;</pre> <p>where,</p> <p>BIT1, BIT2 – specify the values for the operands</p>	<p>To calculate the BITWISE OR for 30 and 12, enter the following command at the command prompt:</p> <pre>SELECT 30 12;</pre> <p>The output of the command is:</p> <p>30</p>
BITWISE XOR	<p>The BITWISE XOR function works with two arguments of equal length. BITWISE XOR compares every bit of the two arguments. This function returns 0 in the output if there are no matching rows in the comparison. The syntax for this function is:</p> <pre>SELECT BIT1 ^ BIT2;</pre> <p>where,</p> <p>BIT1, BIT2 – specifies a value for the operands</p>	<p>To find out the BITWISE XOR for 5 and 2, enter the following command at the command prompt:</p> <pre>SELECT 5 ^ 2;</pre> <p>The output of the command is:</p> <p>7</p>

Name	Description	Example
MIN	<p>The MIN function returns the smallest value of the expression. The syntax for using this function is:</p> <pre>SELECT COLUMN_NAME FROM TABLE_NAME HAVING MIN(expression);</pre>	<p>For example, display the minimum gross salary earned by any of the employee.</p> <pre>SELECT MIN(GROSS_SAL) FROM EMP_SALARY;</pre> <p>The output of this function is:</p> <p>3500.00</p>
STDDEV	<p>The STDDEV function also returns the standard deviation of the values of the argument. The output of this function is NULL if there are no rows satisfying the given query. The syntax for obtaining the standard deviation is:</p> <pre>SELECT STDDEV(expression) FROM TABLE_NAME;</pre>	<p>For example, to obtain the standard deviation of the basic salary of all the employees, enter the following command at the command prompt:</p> <pre>SELECT STTDEV(BASIC_SAL) FROM EMP_SALARY;</pre> <p>The output of the function is:</p> <p>707.106781</p>

Name	Description	Example
STDDEV_POP	<p>The STDDEV_POP function calculates the population standard deviation and returns the square root of the variance. The syntax for this function is:</p> <pre>SELECT STDDEV_POP(expression) FROM TABLE_NAME;</pre>	<p>For example, to obtain the standard deviation of the gross salary of all the employees, enter the following command at the command prompt:</p> <pre>SELECT STDDEV_POP(GROSS_SAL) FROM EMP_SALARY;</pre> <p>The output of the function is:</p> <p>707.106781</p>
STDDEV_SAMP	<p>The STDDEV_SAMP function returns the sample standard deviation of the argument. Sample standard deviation is applicable only to a sample, that is, a part of an entire population. The syntax for using this function is:</p> <pre>SELECT STDDEV_SAMP(expression) FROM TABLE_NAME;</pre>	<p>For example, to obtain the sample standard deviation of the gross salary of all the employees, enter the following command at the command prompt:</p> <pre>SELECT STDDEV_SAMP(GROSS_SAL) FROM EMP_SALARY;</pre> <p>The output of the function is: 790.569415</p>

Name	Description	Example
VAR_POP	<p>The VAR_POP function returns the standard variance of the argument. This function accepts the rows as the argument and defines the number of rows as the denominator. The syntax for using this function is:</p> <pre>SELECT VAR_POP(expression) FROM TABLE_NAME;</pre>	<p>For example, to obtain the variance of gross salary of all the employees, enter the following command at the command prompt:</p> <pre>SELECT VAR_POP(GROSS_SAL) FROM EMP_SALARY;</pre> <p>The output of this function is:</p> <pre>500000.000000</pre>
VAR_SAMP	<p>The VAR_SAMP function returns the sample variance of the argument. The denominator returned by this function is the number of rows minus one. The syntax for using this function is:</p> <pre>SELECT VAR_SAMP(expression) FROM TABLE_NAME;</pre>	<p>For example, to obtain the variance of gross salary of all the employees, enter the following command at the command prompt:</p> <pre>SELECT VAR_SAMP(GROSS_SAL) FROM EMP_SALARY;</pre> <p>The output of this function is:</p> <pre>625000.000000</pre>

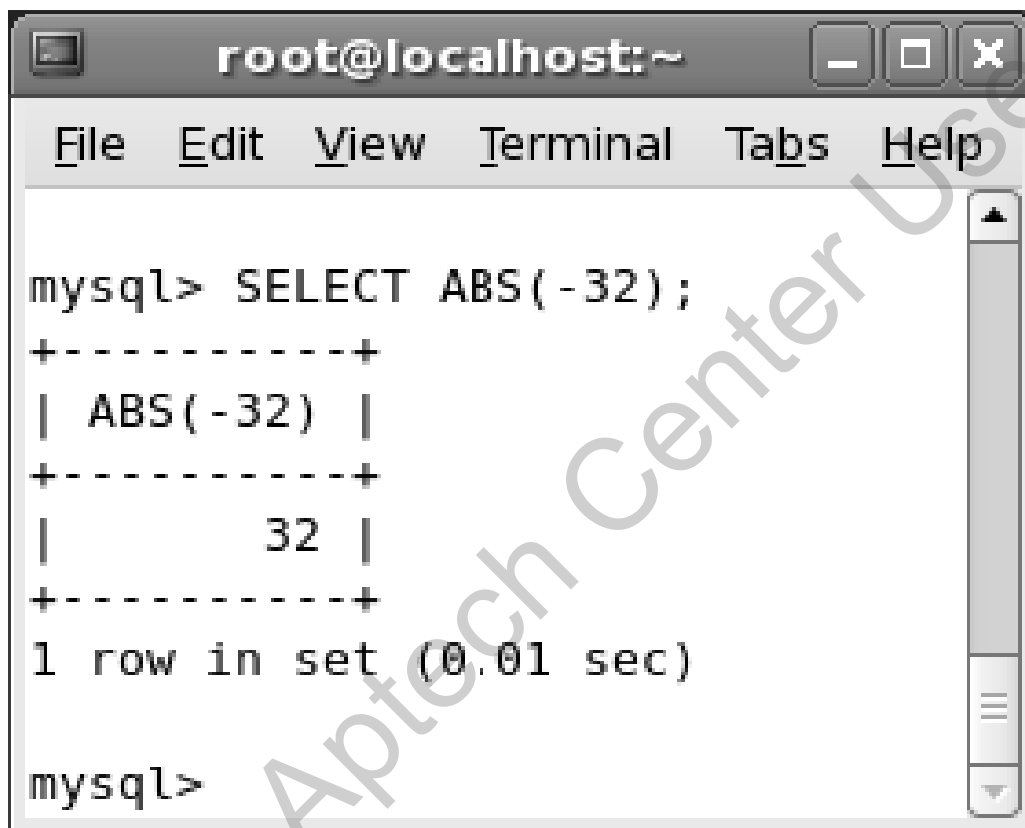
- ◆ The `ABS` function returns the absolute value of the argument
- ◆ In mathematics, the absolute value of a number is its distance from the origin and is therefore, never negative
- ◆ The syntax for using this function is:

```
SELECT ABS (expression);
```

- ◆ For example, to calculate the absolute value of -32, enter the following command at the command prompt:

```
SELECT ABS (-32);
```

Figure displays the output of the command



The screenshot shows a terminal window titled 'root@localhost:~' with a menu bar containing 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal displays the following text:

```
mysql> SELECT ABS(-32);
```

ABS(-32)
32

```
1 row in set (0.01 sec)
```

```
mysql>
```

The output is formatted as a table with a single column header 'ABS(-32)' and one data row containing the value '32'. Below the table, it indicates '1 row in set (0.01 sec)'. The terminal window also features a scrollbar on the right side.

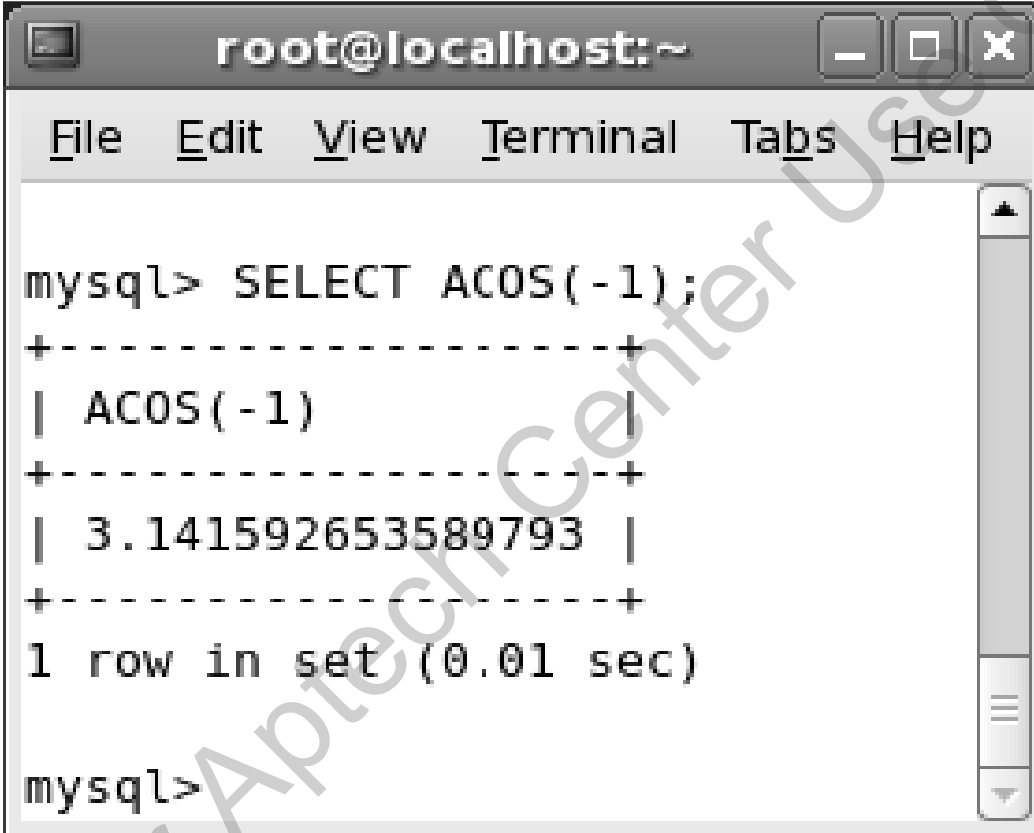
- ◆ The `ACOS` function returns the arc cosine of the specified argument
- ◆ The arc cosine function returns the angle, expressed in radians, of the argument whose cosine is specified
- ◆ The syntax to obtain the arc cosine of a function is:

```
SELECT ACOS (expression) ;
```

- ◆ For example, to calculate the arc cosine value of 1, enter the following command at the command prompt:

```
SELECT ACOS (-1) ;
```

Figure displays the output of the command



The screenshot shows a terminal window titled 'root@localhost:~'. The window has a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal content shows a MySQL prompt 'mysql>' followed by the command 'SELECT ACOS(-1);'. The output is a table with one column 'ACOS(-1)' and one row containing the value '3.141592653589793'. Below the table, it says '1 row in set (0.01 sec)'. The prompt 'mysql>' is shown again at the bottom.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
mysql> SELECT ACOS(-1);  
+-----+  
| ACOS(-1) |  
+-----+  
| 3.141592653589793 |  
+-----+  
1 row in set (0.01 sec)  
mysql>
```


- ◆ The `CEILING` function returns the smallest integer value greater than the argument
- ◆ The syntax for using this function is:

```
SELECT CEILING(X) ;
```

- ◆ To use the `CEILING` function on `5.56`, enter the following command at the command prompt:

```
SELECT CEILING(5.56) ;
```


Figure displays the output of the command



The screenshot shows a terminal window titled 'root@localhost:~' with a menu bar containing 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal displays the following text:

```
mysql> SELECT CEILING(5.56);
+-----+
| CEILING(5.56) |
+-----+
|             6 |
+-----+
1 row in set (0.00 sec)

mysql>
```

The output shows a single row with the value 6, which is the ceiling of 5.56. The terminal window has a scrollbar on the right side.

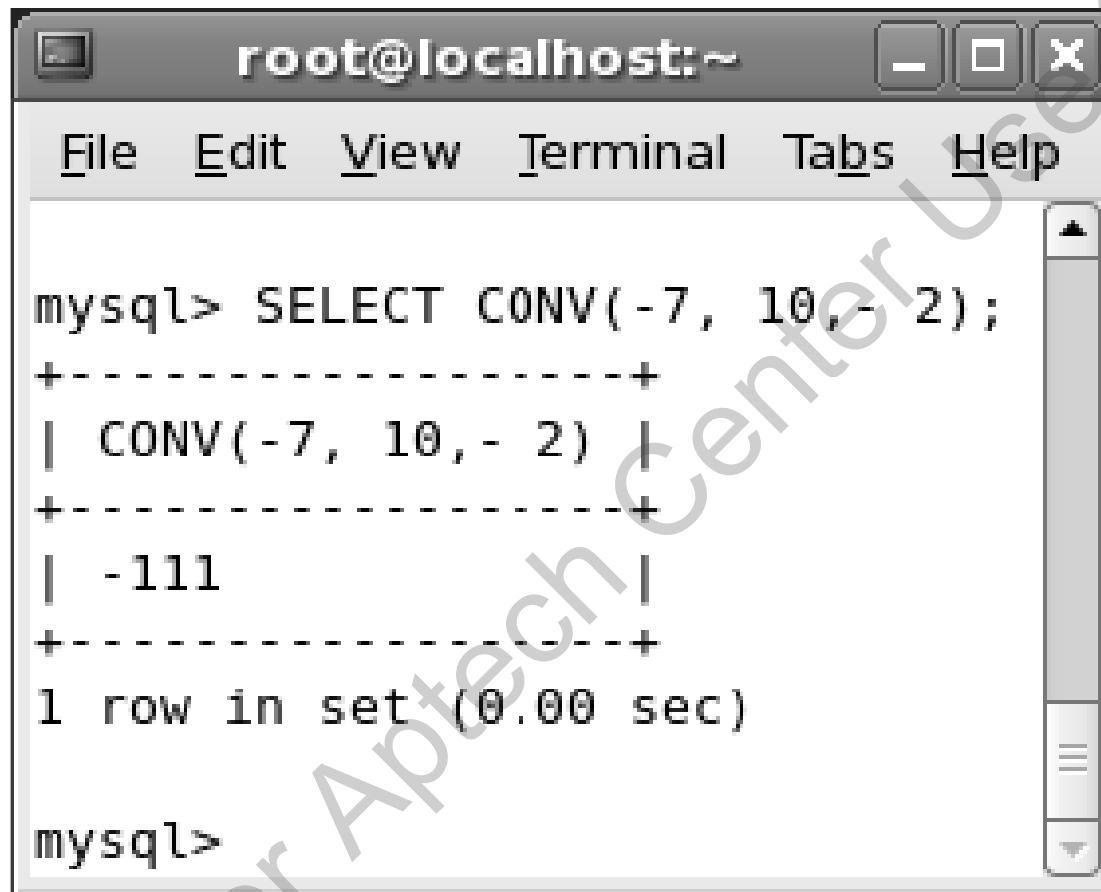
- ◆ The CONV function changes a number from one base to another
- ◆ The output is either generated as a string or as NULL if any argument is NULL
- ◆ The minimum and maximum base values are 2 and 36 respectively
- ◆ You can perform case-insensitive comparisons with this function
- ◆ The syntax to use the CONV function is:

```
SELECT CONV(N, from_base, to_base);
```

- ◆ For example, to convert -7 from base 10 to base -2, enter the following command at the command prompt:

```
SELECT CONV(-7, 10, -2);
```

Figure displays the output of the command



The screenshot shows a terminal window titled 'root@localhost:~'. The terminal displays the execution of the MySQL command 'SELECT CONV(-7, 10, -2);'. The output is a single row with the value '-111'. Below the result, it says '1 row in set (0.00 sec)'. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. A large, diagonal watermark 'For Aptech Center Use Only' is visible across the terminal output.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
mysql> SELECT CONV(-7, 10, -2);  
+-----+  
| CONV(-7, 10, -2) |  
+-----+  
| -111              |  
+-----+  
1 row in set (0.00 sec)  
mysql>
```

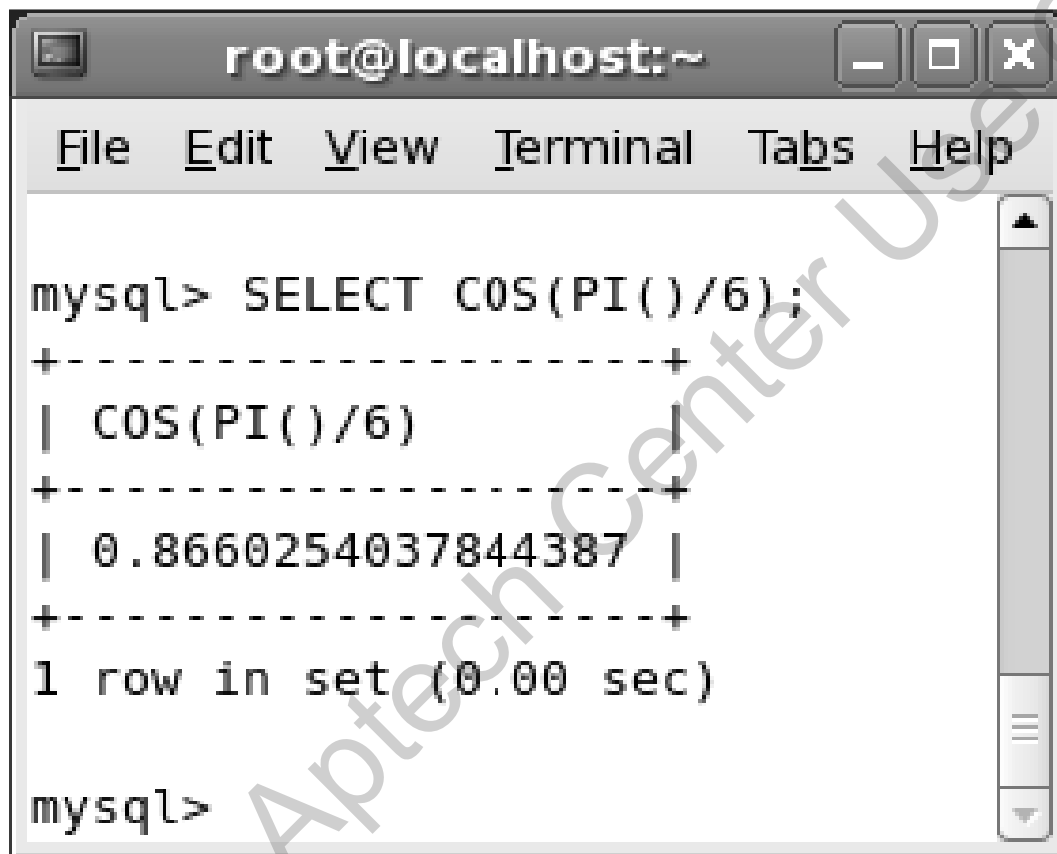
- ◆ The `COS` function returns the cosine of an argument
- ◆ The argument specified in the function should be in radians
- ◆ The syntax for obtaining the cosine of an argument `X` is:

```
SELECT COS (X) ;
```

- ◆ For example, to obtain the cosine value of $\pi/6$, enter the following command at the command prompt:

```
SELECT COS (PI () / 6) ;
```

Figure displays the output of the command



```
root@localhost:~  
File Edit View Terminal Tabs Help  
mysql> SELECT COS(PI()/6);  
+-----+  
| COS(PI()/6) |  
+-----+  
| 0.8660254037844387 |  
+-----+  
1 row in set (0.00 sec)  
mysql>
```

- ◆ The CRC function calculates a cyclic redundancy check value
- ◆ It returns a 32-bit unsigned value
- ◆ It returns a NULL value if a NULL argument is specified. The argument must be a string
- ◆ The function converts the input to a string and computes the cyclic redundancy check value
- ◆ The syntax to calculate the cyclic redundancy value of an argument X is:

```
SELECT CRC32 (X) ;
```

- ◆ For example, to calculate the cyclic redundancy check value for the string JOHN, enter the following command at the command prompt:

```
SELECT CRC32 ( 'JOHN' ) ;
```

Figure displays the output of the command



The screenshot shows a terminal window titled "root@localhost:~". The terminal displays the output of the MySQL command `SELECT CRC32('JOHN');`. The output is formatted as a table with one row and one column, showing the CRC32 value for 'JOHN' as 133104674. Below the table, it indicates "1 row in set (0.03 sec)". The terminal prompt `mysql>` is visible at the bottom.

```
mysql> SELECT CRC32('JOHN');
+-----+
| CRC32('JOHN') |
+-----+
|      133104674 |
+-----+
1 row in set (0.03 sec)

mysql>
```

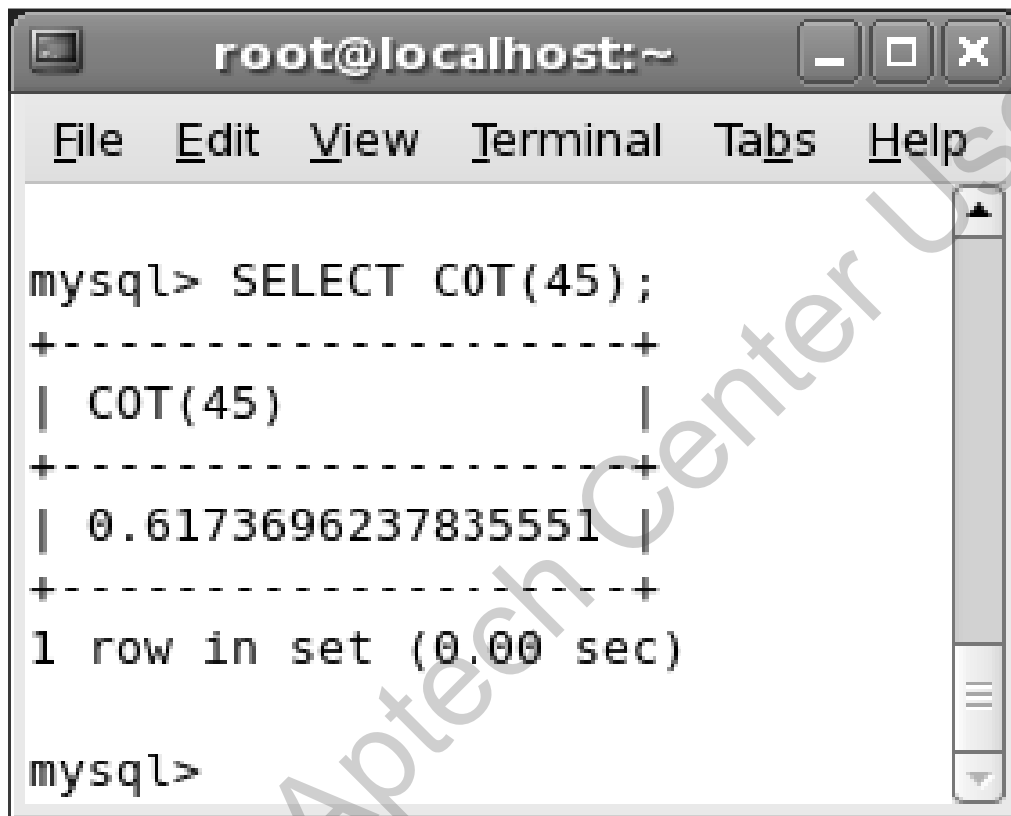
- ◆ The `COT` function returns the cotangent of the argument
- ◆ Cotangent of an argument is the ratio of the adjacent side to the opposite side
- ◆ The syntax to use this function is:

```
SELECT COT(expression);
```

- ◆ To calculate the cotangent of 45, enter the following command at the command prompt:

```
SELECT COT(45);
```


Figure displays the output of the command



The screenshot shows a terminal window titled 'root@localhost:~'. The terminal contains the following text:

```
mysql> SELECT COT(45);  
+-----+  
| COT(45) |  
+-----+  
| 0.6173696237835551 |  
+-----+  
1 row in set (0.00 sec)  
  
mysql>
```

The output displays the result of the COT function for the value 45, which is 0.6173696237835551. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. A vertical scrollbar is visible on the right side of the terminal window.

- ◆ The DEGREES function converts the specified argument from radians to degrees

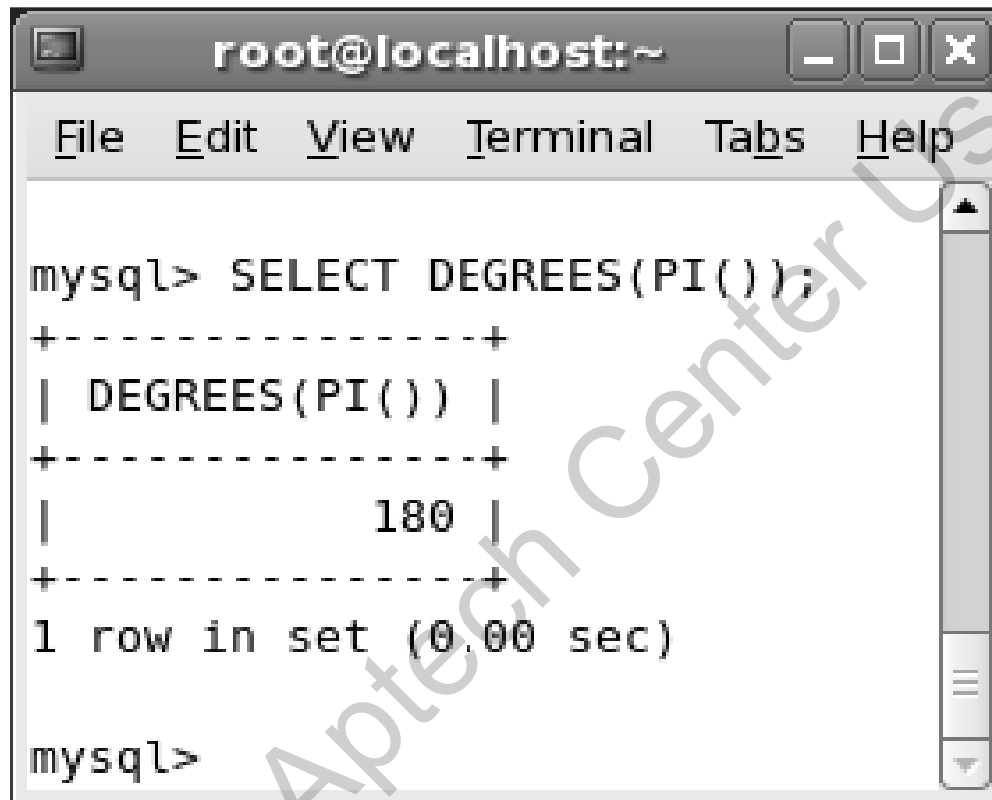
- ◆ The syntax to use this function is:

```
SELECT DEGREES (expression);
```

- ◆ For example, to obtain the value of π in degrees, enter the following command at the command prompt:

```
SELECT DEGREES (PI()) ;
```

Figure displays the output of the command



A terminal window titled 'root@localhost:~' with standard window controls. The terminal shows a MySQL prompt 'mysql>' followed by the command 'SELECT DEGREES(PI());'. The output is a table with one column 'DEGREES(PI())' and one row containing the value '180'. Below the table, it says '1 row in set (0.00 sec)'. The prompt 'mysql>' is shown again at the bottom.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
mysql> SELECT DEGREES(PI());  
+-----+  
| DEGREES(PI()) |  
+-----+  
|           180 |  
+-----+  
1 row in set (0.00 sec)  
mysql>
```

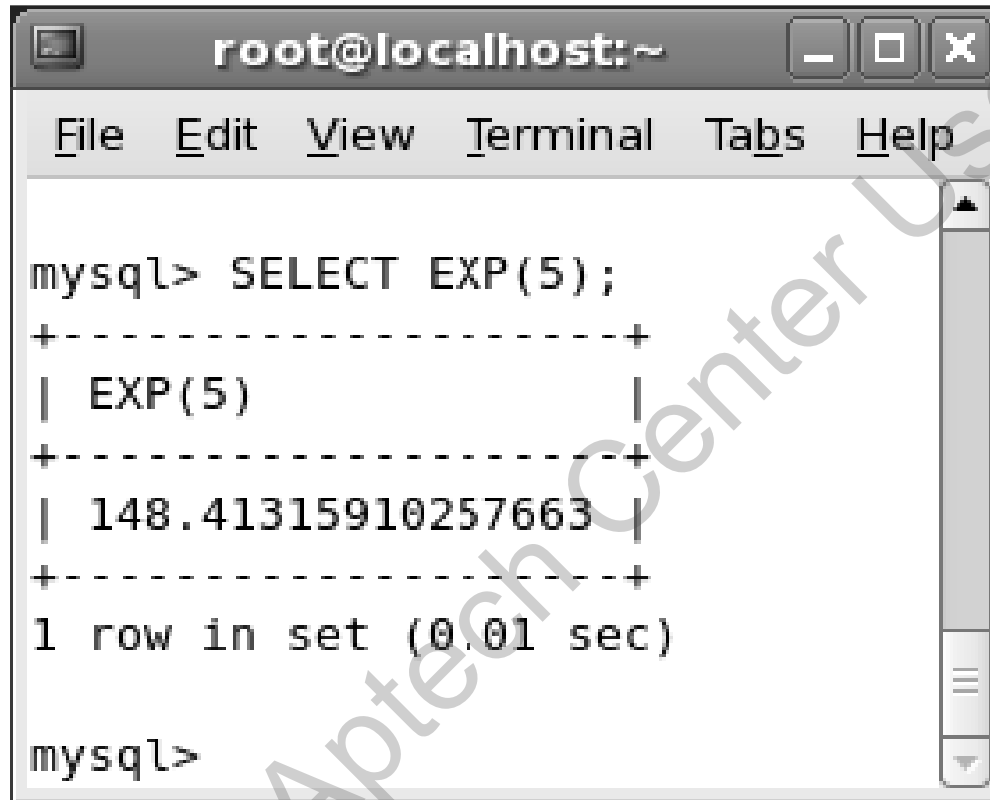
- ◆ The `EXP` function returns the exponential value of the given argument
- ◆ The syntax for obtaining the exponential value of an argument is:

```
SELECT EXP (X) ;
```

- ◆ To calculate the result of e raised to 5, enter the following command at the command prompt:

```
SELECT EXP (5) ;
```

Figure displays the output of the command



The screenshot shows a terminal window titled 'root@localhost:~'. The terminal contains the following text:

```
mysql> SELECT EXP(5);
```

EXP(5)
148.41315910257663

```
1 row in set (0.01 sec)
```

```
mysql>
```

The output is displayed in a table format with a header row and one data row. The value 148.41315910257663 is shown in the data row.

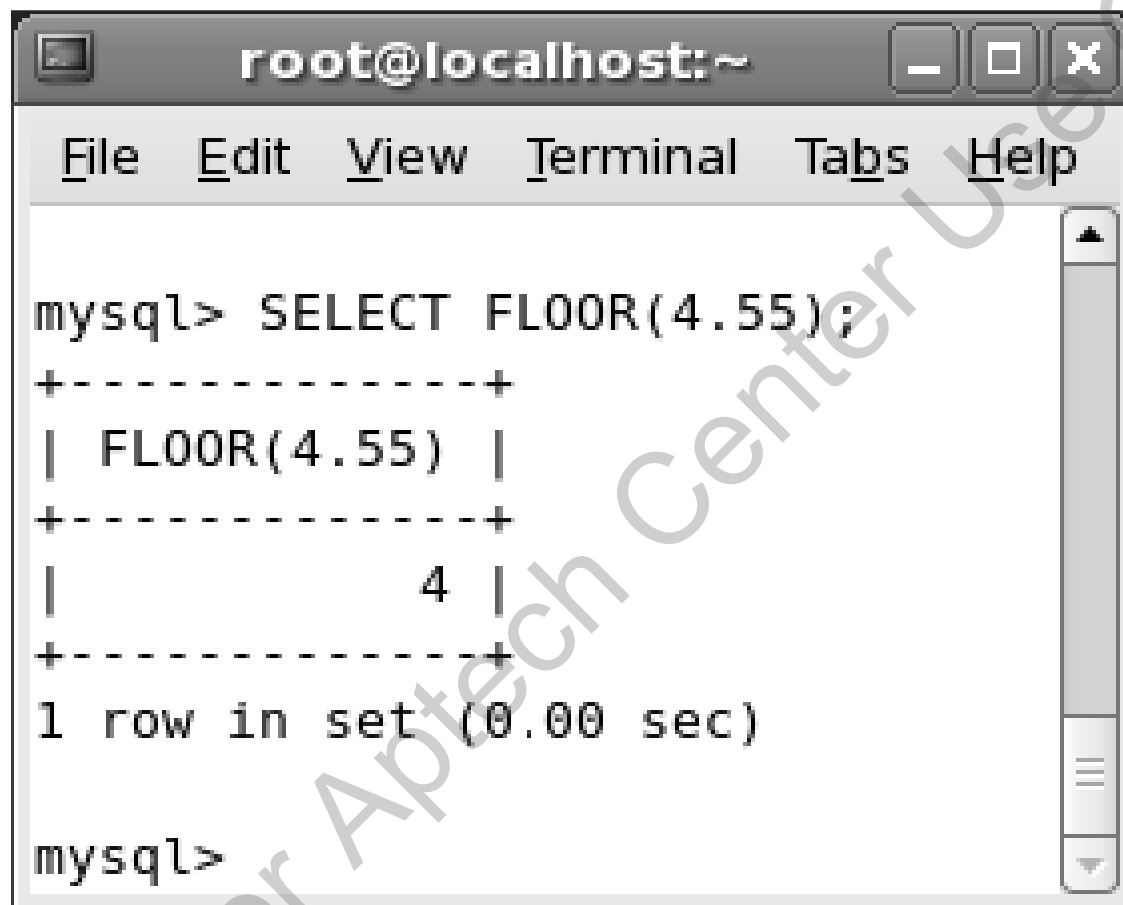
- ◆ The `FLOOR` function returns the largest value less than or equal to the given numeric value
- ◆ The syntax for obtaining the largest value not greater than the argument specified is:

```
SELECT FLOOR(X) ;
```

- ◆ To apply the `FLOOR` function on `4.55`, enter the following command at the command prompt:

```
SELECT FLOOR(4.55) ;
```

Figure displays the output of the command



The screenshot shows a terminal window titled 'root@localhost:~'. The menu bar includes 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal content shows the command 'mysql> SELECT FLOOR(4.55);' followed by a table with one row: 'FLOOR(4.55)' with the value '4'. Below the table, it says '1 row in set (0.00 sec)'. The prompt 'mysql>' is visible at the bottom.

```
mysql> SELECT FLOOR(4.55);
+-----+
| FLOOR(4.55) |
+-----+
|           4 |
+-----+
1 row in set (0.00 sec)

mysql>
```

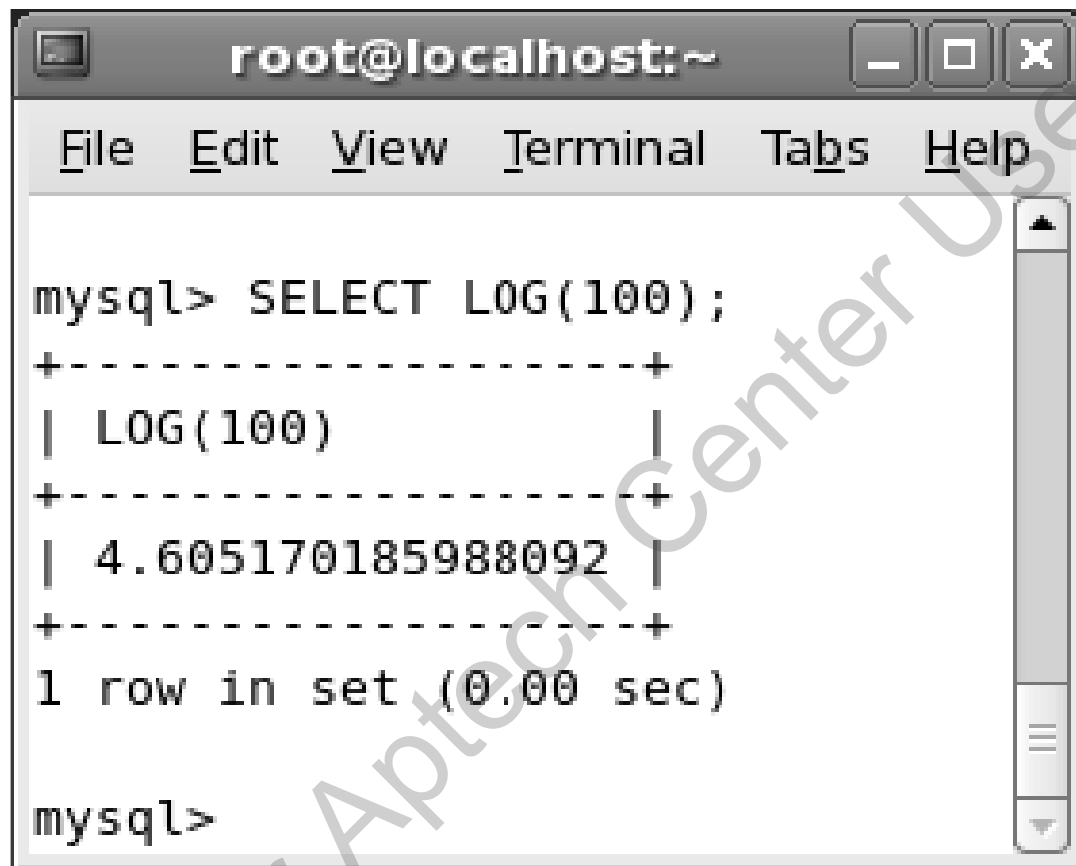
- ◆ The LOG function returns the natural logarithm of the argument
- ◆ The syntax to calculate the logarithm value for a single argument is:

```
SELECT LOG (X) ;
```

- ◆ For example, to calculate the logarithm of 100, enter the following command at the command prompt:

```
SELECT LOG (100) ;
```


Figure displays the output of the command



A terminal window titled 'root@localhost:~' with standard window controls. The menu bar includes 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal content shows a MySQL command and its output:

```
mysql> SELECT LOG(100);
+-----+
| LOG(100) |
+-----+
| 4.605170185988092 |
+-----+
1 row in set (0.00 sec)

mysql>
```

The output is formatted as a table with a single column 'LOG(100)' and one row containing the value '4.605170185988092'. The terminal also indicates '1 row in set (0.00 sec)'.

- ◆ The LOG function allows you to specify two values as arguments
- ◆ Consider an example, where two arguments X and Y are specified in the LOG function
- ◆ This function returns the logarithm of X for an arbitrary base Y
- ◆ The syntax for specifying two arguments in LOG function is:
`LOG (X, Y) ;`

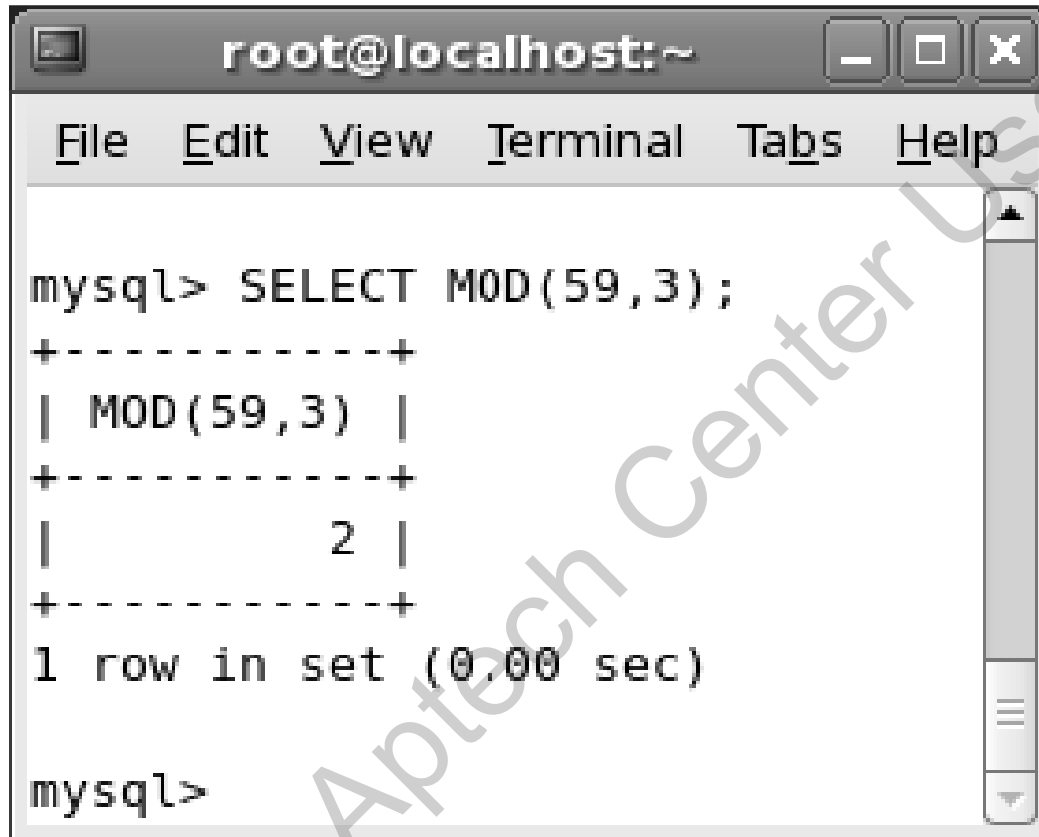
- ◆ The MOD function operates on two numeric arguments
- ◆ The output of the function is the remainder after dividing the number by the divisor
- ◆ The syntax for obtaining the remainder is:

```
SELECT MOD (X, Y) ;
```

- ◆ This function returns the remainder of X divided by Y
- ◆ For example, to obtain the remainder of 59 divided by 3, enter the following command at the command prompt:

```
SELECT MOD (59, 3) ;
```

Figure displays the output of the command



The screenshot shows a terminal window titled 'root@localhost:~'. The terminal contains the following text:

```
mysql> SELECT MOD(59,3);
+-----+
| MOD(59,3) |
+-----+
|          2 |
+-----+
1 row in set (0.00 sec)

mysql>
```

The output displays a single row with the value 2, which is the remainder of 59 divided by 3. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. A large, diagonal watermark 'For Aptech Center Use Only' is visible across the terminal output.

- ◆ The OCT function returns the octal value of a number
- ◆ Octal is a numbering system that has a base of 8 and each digit in the number is represented using only the numerals 0–7

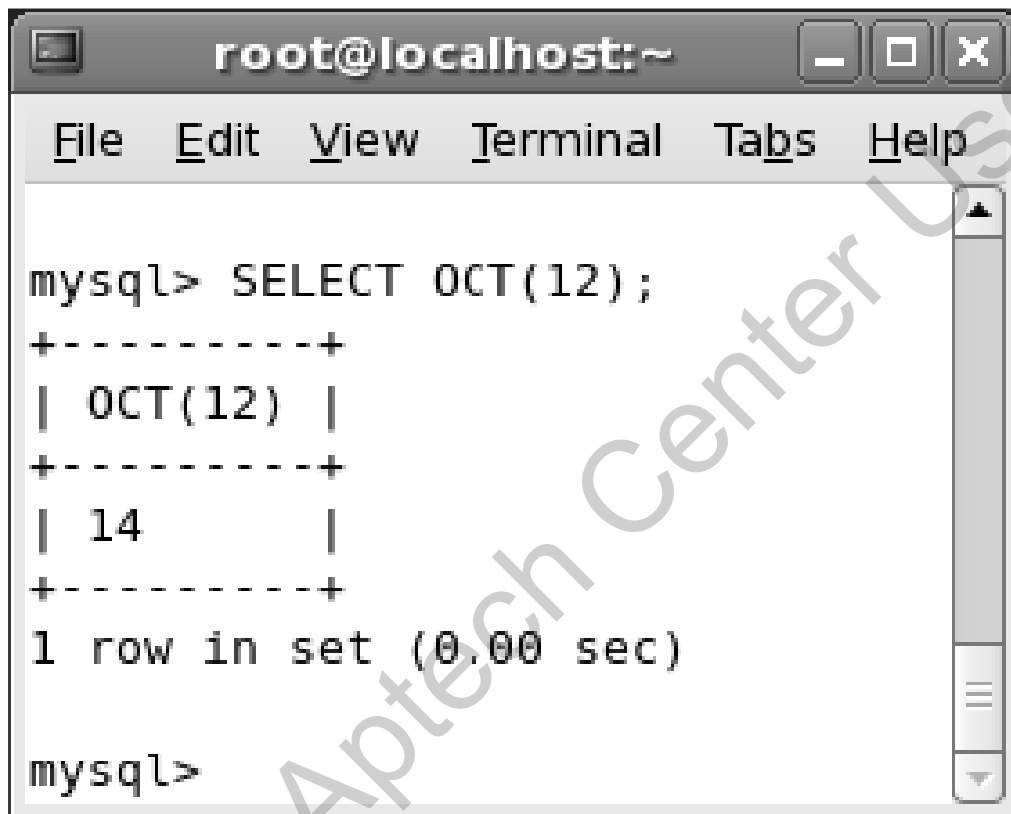
- ◆ The syntax for using this function is:

```
SELECT OCT(expression) ;
```

- ◆ For example, to obtain the octal value of 12, enter the following command at the command prompt:

```
SELECT OCT(12) ;
```

Figure displays the output of the command



The screenshot shows a terminal window titled 'root@localhost:~'. The terminal contains the following text:

```
mysql> SELECT OCT(12);
+-----+
| OCT(12) |
+-----+
| 14      |
+-----+
1 row in set (0.00 sec)

mysql>
```

The output displays the result of the `OCT(12)` function, which is 14. The result is presented in a table format with a header row and a data row, separated by plus signs and dashes. The terminal window also shows a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'.

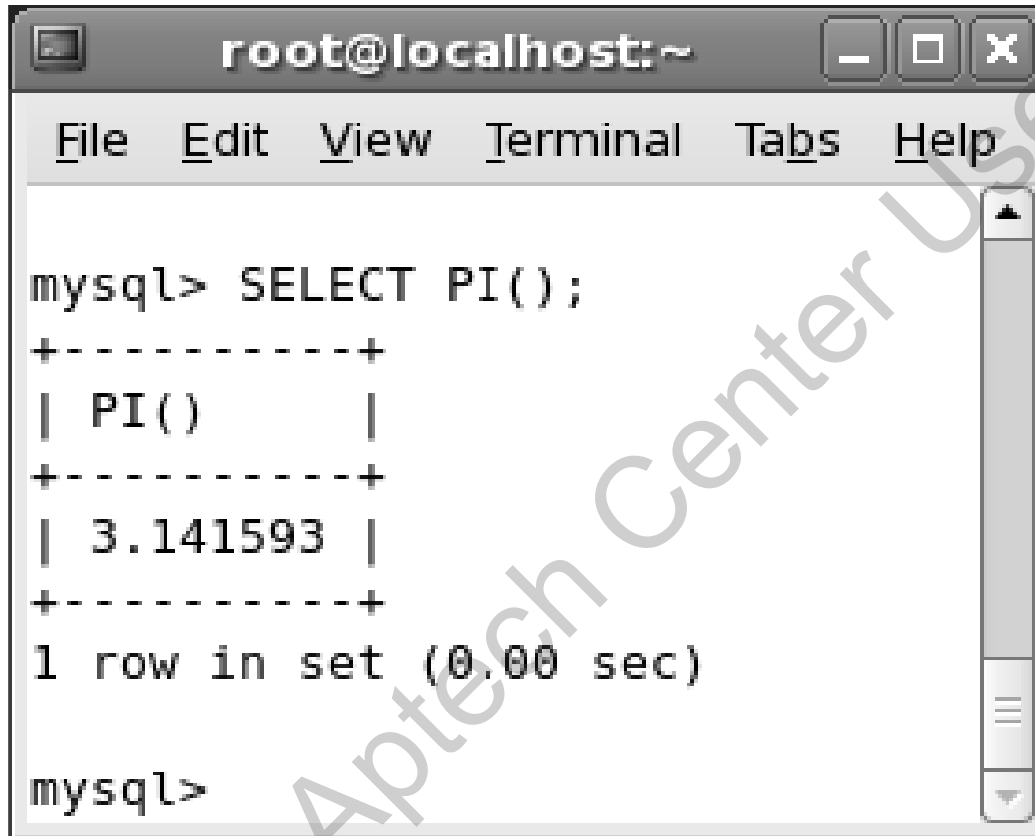
- ◆ The `PI` function returns the value of the mathematical constant, π
- ◆ The syntax for using this function is:

```
SELECT PI ( ) ;
```

- ◆ To view the output of this function, enter the following command at the command prompt:

```
SELECT PI ( ) ;
```

Figure displays the output of the command



The screenshot shows a terminal window titled 'root@localhost:~'. The window has a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal content shows the command 'mysql> SELECT PI();' followed by a table with one row containing the value '3.141593'. Below the table, it says '1 row in set (0.00 sec)'. The prompt 'mysql>' is visible at the bottom.

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

mysql>
```

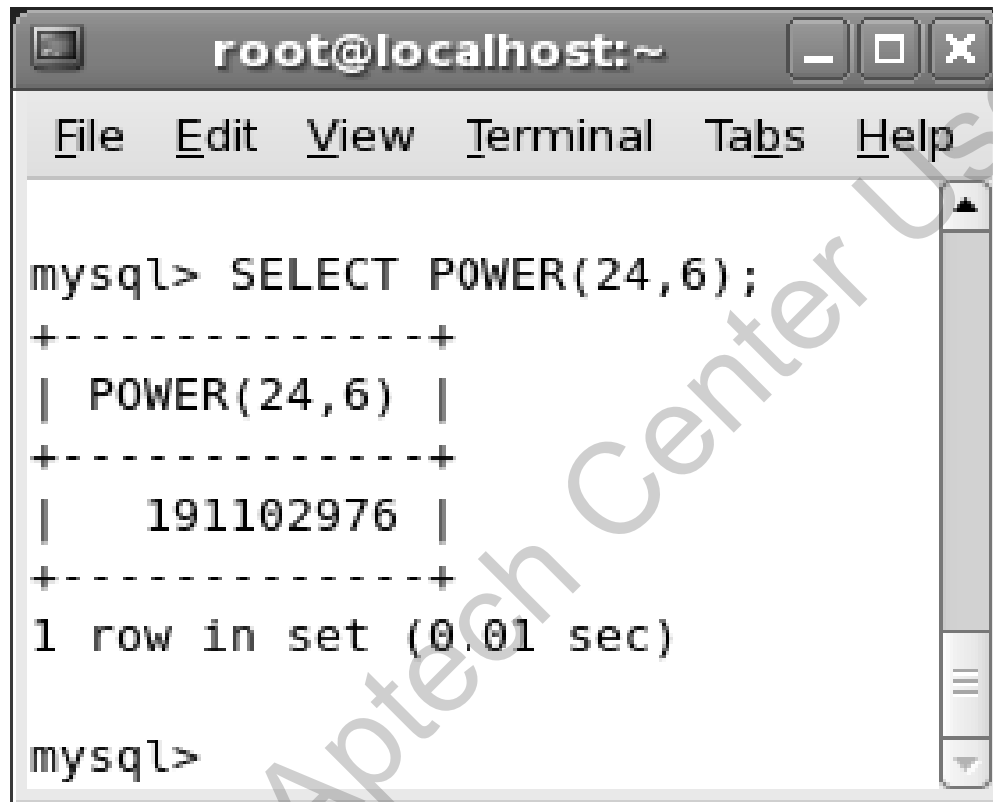

- ◆ The `POWER` function allows you to enter two arguments
- ◆ This function raises the first argument to the power of the second one
- ◆ The syntax for using this function is:

```
SELECT POWER (X, Y) ;
```

- ◆ This command returns the value of `X` raised to the power of `Y`
- ◆ For example, to calculate the result of 24 raised to 6, enter the following command at the command prompt:

```
SELECT POWER (24, 6) ;
```

Figure displays the output of the command



The screenshot shows a terminal window titled 'root@localhost:~'. The terminal contains the following text:

```
mysql> SELECT POWER(24,6);
+-----+
| POWER(24,6) |
+-----+
|    191102976    |
+-----+
1 row in set (0.01 sec)

mysql>
```

- ◆ The alternative syntax for the POWER function is:

```
SELECT POW (X, Y) ;
```

- ◆ The RADIANS function converts the specified argument from degrees to radians

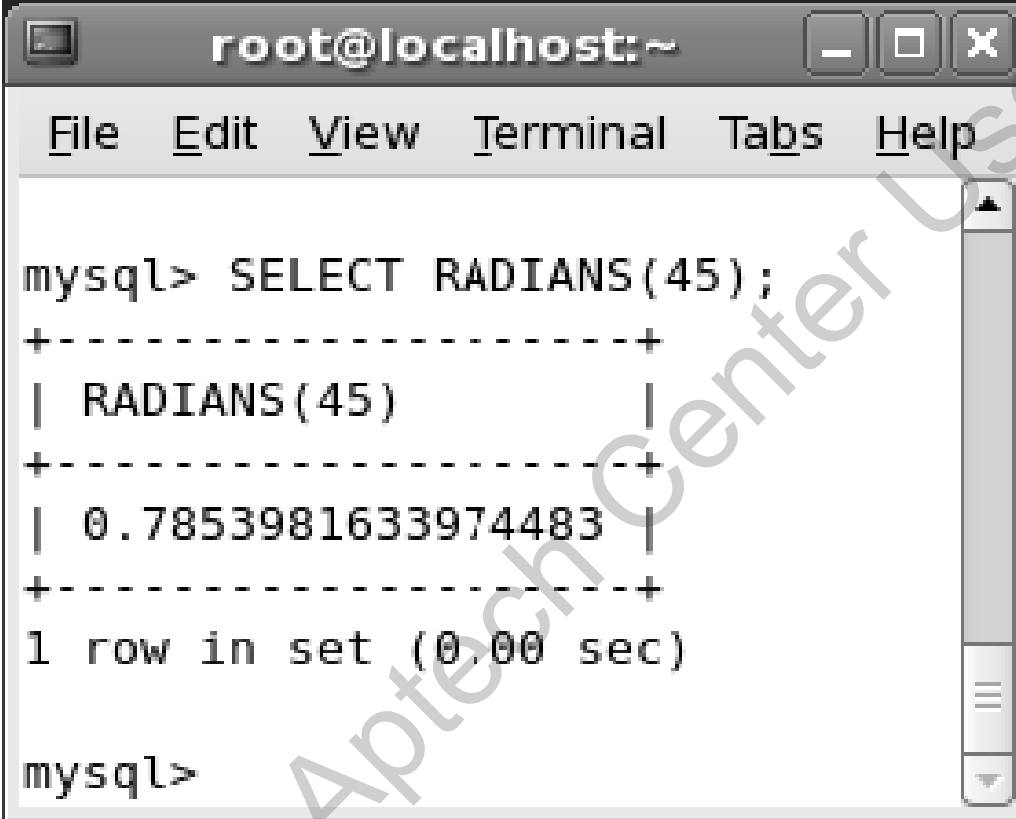
- ◆ The syntax to use this function is:

```
SELECT RADIANS (expression) ;
```

- ◆ For example, to calculate the value of 45 in radians, enter the following command at the command prompt:

```
SELECT RADIANS (45) ;
```

Figure displays the output of the command



The screenshot shows a terminal window titled 'root@localhost:~'. The terminal contains the following text:

```
mysql> SELECT RADIANS(45);
+-----+
| RADIANS(45) |
+-----+
| 0.7853981633974483 |
+-----+
1 row in set (0.00 sec)

mysql>
```

The output is formatted as a table with a single column named 'RADIANS(45)' and one row containing the value '0.7853981633974483'. The terminal also shows the command execution time as '0.00 sec'.

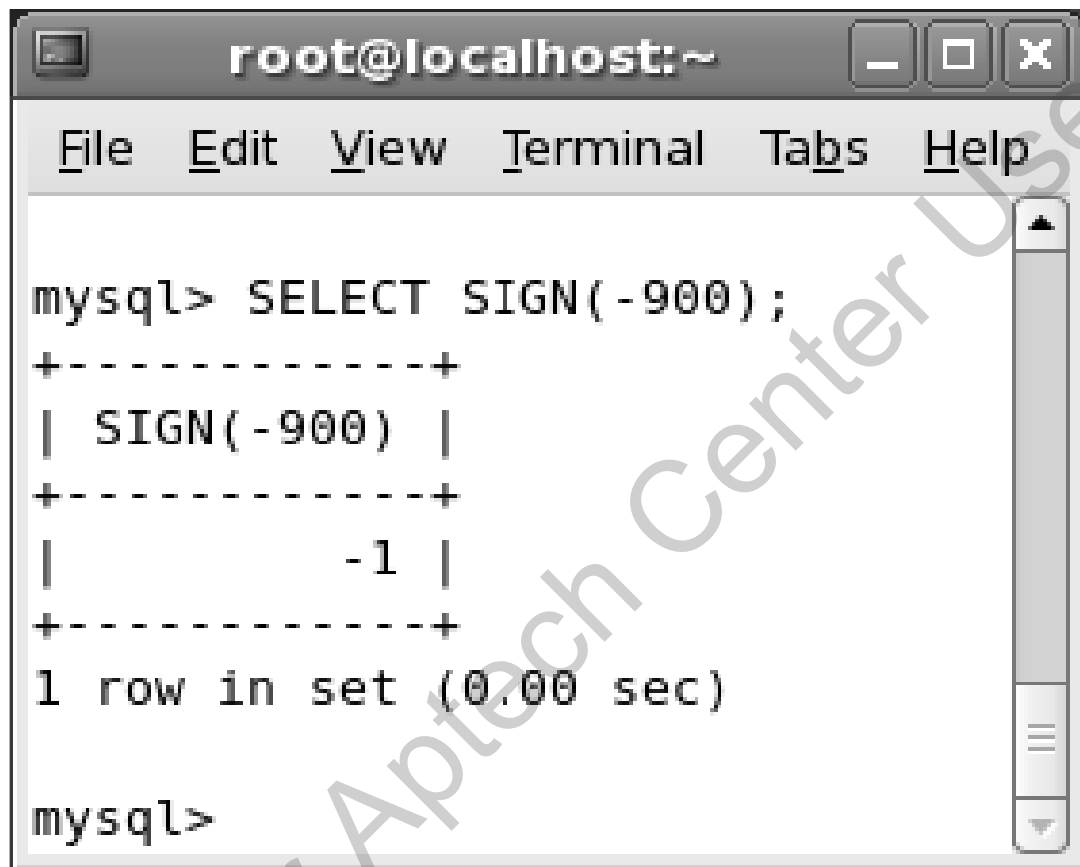
- ◆ The `SIGN` function specifies if the specified argument is negative or non-negative
- ◆ It returns a `-1`, `0`, or `1` depending upon the argument, whether it is positive, zero, or negative respectively
- ◆ The syntax for obtaining the sign of an argument is:

```
SELECT SIGN (X) ;
```

- ◆ For example, to calculate the sign of `-900`, enter the following command at the command prompt:

```
SELECT SIGN (-900) ;
```

Figure displays the output of the command



The screenshot shows a terminal window titled 'root@localhost:~'. The terminal displays the command 'mysql> SELECT SIGN(-900);' and its output. The output is a table with one column, 'SIGN(-900)', containing the value '-1'. Below the table, it says '1 row in set (0.00 sec)'. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. A large, diagonal watermark 'For Apteck Center Use Only' is visible across the terminal content.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
mysql> SELECT SIGN(-900);  
+-----+  
| SIGN(-900) |  
+-----+  
|          -1 |  
+-----+  
1 row in set (0.00 sec)  
mysql>
```

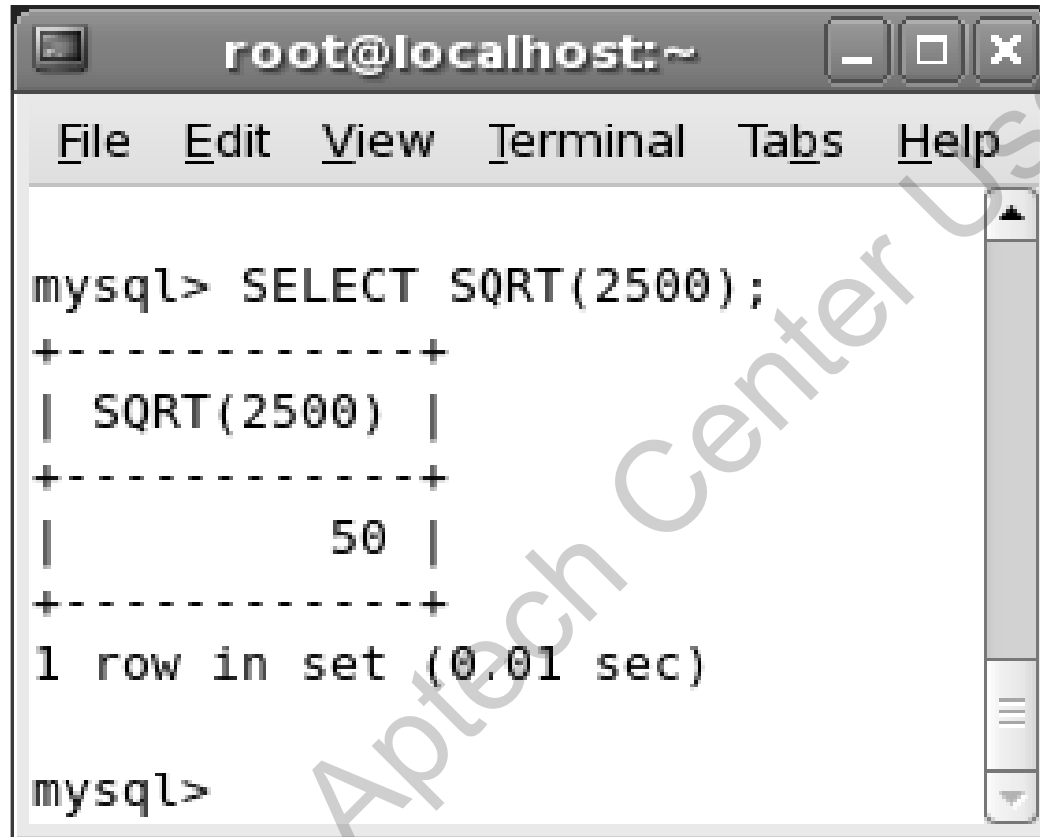
- ◆ The SQRT function returns the square root of the argument
- ◆ The function returns the square root as a non-negative integer
- ◆ The syntax to calculate the square root of an argument is:

```
SELECT SQRT (X) ;
```

- ◆ To calculate the square root of 2500, enter the following command at the command prompt:

```
SELECT SQRT (2500) ;
```

Figure displays the output of the command



The screenshot shows a terminal window titled 'root@localhost:~'. The menu bar includes 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The command 'mysql> SELECT SQRT(2500);' has been entered. The output is displayed in a table format with a single row containing the value '50'. Below the table, it says '1 row in set (0.01 sec)'. The prompt 'mysql>' is visible at the bottom of the terminal.

```
mysql> SELECT SQRT(2500);
+-----+
| SQRT(2500) |
+-----+
|          50 |
+-----+
1 row in set (0.01 sec)

mysql>
```


- ◆ The TRUNCATE function trims the given argument to the specified number of decimal places.

- ◆ The syntax to use this function is:

```
SELECT TRUNCATE (x, d) ;
```

where,

x – specifies the value for the argument

d – specifies the number of decimal places

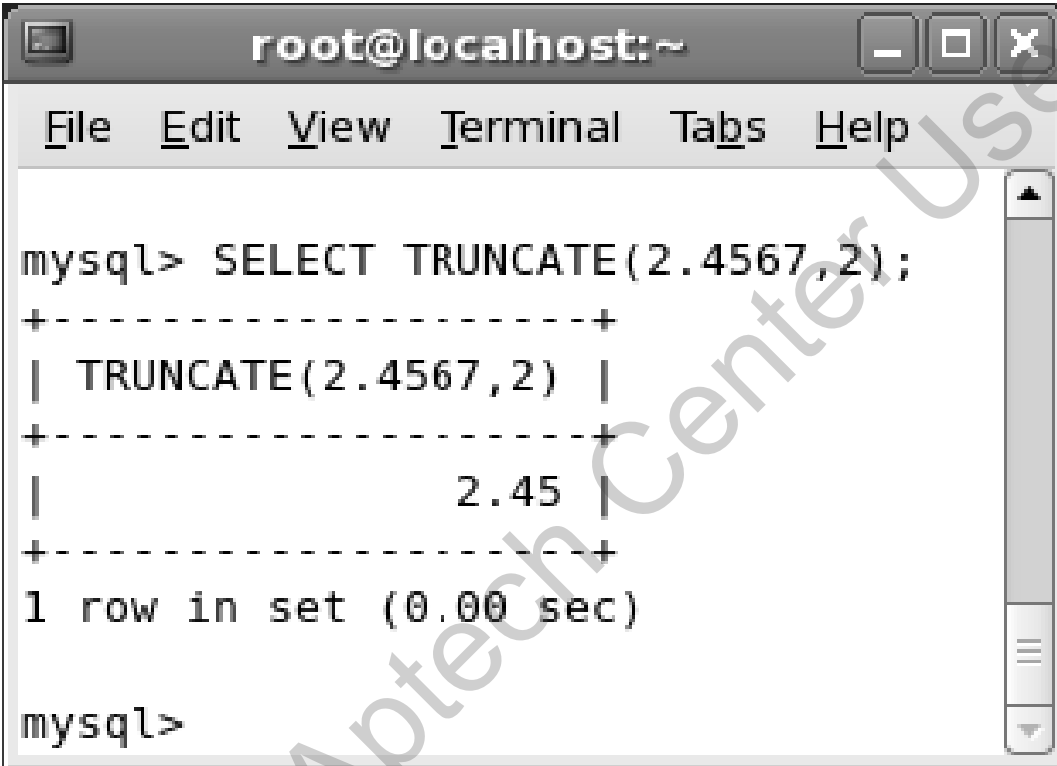
- ◆ If $d = 0$, then the number has no decimal part
- ◆ If d is negative, then d digits to the left of the decimal point are made zero

- ◆ To truncate 2.4567 to two decimal places, enter the following command at the command prompt

```
SELECT TRUNCATE (2.4567, 2) ;
```

For Apteck Center Use Only

Figure displays the output of the command



The screenshot shows a terminal window titled 'root@localhost:~'. The terminal contains the following text:

```
mysql> SELECT TRUNCATE(2.4567,2);
+-----+
| TRUNCATE(2.4567,2) |
+-----+
|                2.45 |
+-----+
1 row in set (0.00 sec)

mysql>
```

The output displays a single row with the value 2.45, which is the result of truncating 2.4567 to two decimal places. The terminal window has a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. A vertical scrollbar is visible on the right side of the terminal window.

Table describes additional Mathematical functions supported by MySQL:

Name	Description	Example
ASIN	<p>The ASIN function returns the arc sine of the specified argument. The arc sine function returns the angle, expressed in radians, of the argument whose sine is specified. The syntax to obtain the arc sine of a function is:</p> <pre>SELECT ASIN(expression);</pre>	<p>For example, to obtain the arcsine value of 1, enter the following command at the command prompt:</p> <pre>SELECT ASIN(-1);</pre> <p>The output of this function is: -1.5707963267948966</p>
LOG10	<p>The LOG10 function returns the base-10 logarithm of the argument. The syntax to use this function is:</p> <pre>SELECT LOG10(expression);</pre>	<p>For example, to obtain the logarithm of 100 to the base 10, enter the following command at the command prompt:</p> <pre>SELECT LOG10(100);</pre> <p>The output of this function is:2</p>

Name	Description	Example
LOG2	<p>The LOG2 function returns the base-2 logarithm of the argument. The syntax to use this function is:</p> <pre>SELECT LOG2 (expression);</pre>	<p>For example, to obtain the logarithm of 16 to the base 2, enter the following command at the command prompt</p> <pre>SELECT LOG2 (16);</pre> <p>The output of this function is:</p> <p>4</p>
SIN	<p>The SIN function returns the sine value of the argument. You should specify the argument in radians. The syntax to use this function is:</p> <pre>SELECT SIN (expression);</pre>	<p>For example, to calculate the sin value of $\pi/2$, enter the following command at the command prompt:</p> <pre>SELECT SIN (PI () / 2);</pre> <p>The output of this function is:</p> <p>1</p>

Name	Description	Example
TAN	<p>The TAN function returns the tangent value of the argument passed. You must specify the argument in radians. The syntax to use this function is:</p> <pre>SELECT TAN(expression);</pre>	<p>For example, to obtain the tan value of $\pi/4$, enter the following command at the command prompt:</p> <pre>SELECT TAN(PI()/4);</pre> <p>The output of this function is:</p> <pre>0.9999999999999999</pre>

- ◆ In SQL queries, functions can be used in place of the column name and in the `WHERE` clause
- ◆ The `GROUP BY` clause can be used with functions to group the data on a specific constraint
- ◆ `GROUP BY` functions are used to obtain the average value of the argument, count the number of values in a given column, calculate the variance and standard deviation of a given set of values, and calculate the maximum or minimum value from a given set of numbers
- ◆ Examples of functions that can be used with the `GROUP BY` clause are `AVG`, `COUNT`, `COUNT (DISTINCT)`, and `SUM`

- ◆ Mathematical functions are used to operate on numbers. They obtain the trigonometric values such as `SIN`, `COS`, `TAN`, `ATAN`, and `ACOS`. In addition, you can use these functions to obtain the square root, logarithmic value, exponential value of a number, and the modulus value
- ◆ Mathematical functions return a `NULL` value if an error occurs while processing. Some examples of mathematical functions are `ABS`, `ACOS`, `ASIN`, `EXP`, `LOG`, and `SQRT`