```
        -> 3.141593
mysql> SELECT PI()+0.000000000000000000;
        -> 3.141592653589793116
```

- `POW(X,Y)`

  Returns the value of *X* raised to the power of *Y*.

  ```
  mysql> SELECT POW(2,2);
          -> 4
  mysql> SELECT POW(2,-2);
          -> 0.25
  ```

- `POWER(X,Y)`

  This is a synonym for `POW()`.

- `RADIANS(X)`

  Returns the argument *X*, converted from degrees to radians. (Note that π radians equals 180 degrees.)

  ```
  mysql> SELECT RADIANS(90);
          -> 1.5707963267949
  ```

- `RAND([N])`

  Returns a random floating-point value *v* in the range `0` <= *v* < `1.0`. To obtain a random integer *R* in the range *i* <= *R* < *j*, use the expression `FLOOR(i + RAND() * (j − i))`. For example, to obtain a random integer in the range the range `7` <= *R* < `12`, use the following statement:

  ```
  SELECT FLOOR(7 + (RAND() * 5));
  ```

  If an integer argument *N* is specified, it is used as the seed value:

  - With a constant initializer argument, the seed is initialized once when the statement is prepared, prior to execution.

  - With a nonconstant initializer argument (such as a column name), the seed is initialized with the value for each invocation of `RAND()`.

  One implication of this behavior is that for equal argument values, `RAND(N)` returns the same value each time, and thus produces a repeatable sequence of column values. In the following example, the sequence of values produced by `RAND(3)` is the same both places it occurs.

  ```

  mysql> CREATE TABLE t (i INT);
  Query OK, 0 rows affected (0.42 sec)

  mysql> INSERT INTO t VALUES(1),(2),(3);
  Query OK, 3 rows affected (0.00 sec)
  Records: 3  Duplicates: 0  Warnings: 0

  mysql> SELECT i, RAND() FROM t;
  +------+------------------+
  | i    | RAND()           |
  +------+------------------+
  |    1 | 0.61914388706828 |
  |    2 | 0.93845168309142 |
  |    3 | 0.83482678498591 |
  +------+------------------+
  3 rows in set (0.00 sec)

  mysql> SELECT i, RAND(3) FROM t;
  ```