jinyPHP 2권 - 26. 내장함수

**26. 내장함수**

====================

MYSQL은 다양한 데이터 가공을 위해서 몇가지 내부 함수를 제공합니다. 내부함수는 주로 Select 키워드와 같이 사용을 하지만, 몇 몇 함수들은 테이블 속성값이나 검색 조건값등으로도 사용을 할 수 있습니다.

MYSQL에서는 사용자함수는 stored function (저장함수)라고 부르기도 합니다. 내장함수는 연산과 관련된 산술함수, 문자열 함수, 날짜 시간관련 함수등이 있습니다.

**26.1 산술함수**

====================

산술함수는 컬럼의 데이터의 합계, 평균 등 값을 연산하는 유형의 기능을 하는 함수들 입니다.

데이터베이스의 값을 조회하여 PHP 프로그램에서 연산처리를 할 수도 있습니다. 간단한 유형의 산술기능은 SQL 내장함수를 통하여 데이터베이스 서버상에서 빠르게 처리할 수 있습니다.

**26.1.1 AVG()**

====================

숫자형 컬럼데이터의 경우 선택된 컬럼필드의 평균값을 구할 수 있습니다. 평균값은 전체의 값을 합산한 후에 총 개수로 나눈 값 입니다.

AVG() 문법

|  |
| --- |
| SELECT AVG(컬럼명) FROM 테이블명  WHERE 조건; |

테이블의 컬럼 전체, 또는 조건을 정하여 추출된 데이터를 기반으로 평균을 계산할 수 도 있습니다. 또한 as 키워드를 통하여 별칭을 선언하여 평균값을 접근할 수 있습니다.

SELECT AVG(Price) FROM Products;

|  |
| --- |
| 테이블의 데이터를 미리 확인합니다.  mysql> select \* from products;  +----+--------+-------+------+-------+  | Id | name | price | vat | stock |  +----+--------+-------+------+-------+  | 1 | S-827 | 20000 | 2000 | 5 |  | 2 | S-532D | 36000 | 3600 | 7 |  | 3 | R-530 | 25000 | 2500 | 10 |  +----+--------+-------+------+-------+  3 rows in set (0.00 sec)  제품 테이블의 가격의 평균값을 계산합니다.  mysql> select avg(price) from products;  +------------+  | avg(price) |  +------------+  | 27000.0000 |  +------------+  1 row in set (0.01 sec) |

**26.1.2 COUNT()**

====================

Select 키워드를 통하여 데이터를 조회할 경우에 조회된 데이터의 개수를 확인할 수 있습니다.

콘솔상에서 명령을 실행할때 출력 데이터의 갯수가 화면에 표시가 됩니다. 하지만, PHP 프로그램을 통하여 조회된 데이터의 갯수를 파악을 하기 위해서는 데이터의 갯수를 파악할 수 있는 함수를 사용해야 합니다.

**쿼리문법)**

|  |
| --- |
| SELECT count(컬럼명) from 테이블명; |

select count(\*) from products;

|  |
| --- |
| 테이블의 데이터를 미리 확인합니다.  mysql> select \* from products;  +----+--------+-------+------+-------+  | Id | name | price | vat | stock |  +----+--------+-------+------+-------+  | 1 | S-827 | 20000 | 2000 | 5 |  | 2 | S-532D | 36000 | 3600 | 7 |  +----+--------+-------+------+-------+  2 rows in set (0.00 sec)  mysql> select count(\*) from products;  +----------+  | count(\*) |  +----------+  | 2 |  +----------+  1 row in set (0.00 sec) |

count() 함수의 인자로는 컬럼명을 입력합니다. 하지만 전체 컬럼을 의미하는 와일드 기호를 사용할 수도 있습니다. 아스테리크 기호 \* 는 전체 컬럼을 선택하는 기호입니다.

**26.1.3 MAX()**

====================

테이블 선택 조건 상에서 컬럼필드의 최대값을 구합니다. 최대값은 조건상에서 존재하는 숫자형 데이터값의 최대값을 출력합니다.

MAX() 문법

|  |
| --- |
| SELECT MAX(컬럼명) FROM 테이블명  WHERE 조건; |

콘솔 실습)

|  |
| --- |
| mysql> select \* from products;  +----+--------+-------+------+-------+  | Id | name | price | vat | stock |  +----+--------+-------+------+-------+  | 1 | S-827 | 20000 | 2000 | 5 |  | 2 | S-532D | 36000 | 3600 | 7 |  | 3 | R-530 | 25000 | 2500 | 10 |  +----+--------+-------+------+-------+  3 rows in set (0.00 sec)  mysql> select max(price) from products;  +------------+  | max(price) |  +------------+  | 36000 |  +------------+  1 row in set (0.00 sec) |

**26.1.4 MIN()**

====================

테이블 선택 조건 상에서 컬럼필드의 최소값을 구합니다. 최소값은 조건상에서 존재하는 숫자형 데이터값의 최소값을 출력합니다.

MIN() 문법

|  |
| --- |
| SELECT MIN(컬럼명) FROM 테이블명  WHERE 조건; |

콘솔 실습)

|  |
| --- |
| mysql> select min(price) from products;  +------------+  | min(price) |  +------------+  | 20000 |  +------------+  1 row in set (0.00 sec) |

**26.1.5 ROUND()**

====================

실수형 숫자값의 데이터의 경우 소수점 처리 반올림을 합니다. 반올림의 자리수는 생략하거나 소수점이하 자리수를 지정할 수 있습니다.

**함수)**

|  |
| --- |
| round(숫자, 자리수); |

Round 함수는 2개의 매개변수 인자를 가지고 있습니다. 첫번째는 데이터 값이고 두번째는 반올림 소수점 자리수 입니다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT ROUND(-1.23), ROUND(-1.23,1);  +--------------+----------------+  | ROUND(-1.23) | ROUND(-1.23,1) |  +--------------+----------------+  | -1 | -1.2 |  +--------------+----------------+  1 row in set (0.00 sec) |

**26.1.6 SUM()**

====================

합계를 구합니다. 컬럼 필드가 숫자타입 일 때 전체 합계를 구합니다.

SUM() 문법

|  |
| --- |
| SELECT SUM(column\_name) FROM table\_name WHERE condition; |

콘솔 실습)

|  |
| --- |
| mysql> select sum(price) from products;  +------------+  | sum(price) |  +------------+  | 81000 |  +------------+  1 row in set (0.00 sec) |

**26.1.7 ABS(숫자)**

====================

입력값의 절대값 출력을 출력합니다. 절대값은 부호에 상관없이 0으로 부터의 위치크기 값을 의미합니다.

콘솔 실습)

|  |
| --- |
| mysql> select abs(2), abs(-2);  +--------+---------+  | abs(2) | abs(-2) |  +--------+---------+  | 2 | 2 |  +--------+---------+  1 row in set (0.00 sec) |

**26.1.8 CEILING(숫자)**

====================

입력한 값보다 큰 정수 중에서 가장 작은 수를 출력합니다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT CEILING(1.23), CEIL(-1.23);  +---------------+-------------+  | CEILING(1.23) | CEIL(-1.23) |  +---------------+-------------+  | 2 | -1 |  +---------------+-------------+  1 row in set (0.00 sec) |

변환된 데이터는 BIGINT 형으로 반환됩니다.

**26.1.9 FLOOR(숫자)**

====================

FLOOR()는 입력 값보다 작은 정수 중 가장 큰 수를 반환합니다.. 실수를 무조건 버림 합니다. (음수일 경우는 제외)

|  |
| --- |
| mysql> SELECT FLOOR(1.23), FLOOR(-1.23);  +-------------+--------------+  | FLOOR(1.23) | FLOOR(-1.23) |  +-------------+--------------+  | 1 | -2 |  +-------------+--------------+  1 row in set (0.00 sec) |

**26.1.10 TRUNCATE(숫자,자리수)**

====================

숫자를 소수점 이하 자릿수에서 버림. 숫자 *X*를 *D* 자릿수 뒤의 숫자를 없앤 후 리턴한다. 만일 *D*가 0 이면, 리턴되는 숫자는 소수점 또는 소수 부분이 없게 된다. *D*는 음수로 표시할 수도 있는데, 이렇게 표시하면 숫자 X의 D 자리로부터 왼쪽 부분이 0으로 표시가 된다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT TRUNCATE(1.223,1), TRUNCATE(1.999,1), TRUNCATE(1.999,0);  +-------------------+-------------------+-------------------+  | TRUNCATE(1.223,1) | TRUNCATE(1.999,1) | TRUNCATE(1.999,0) |  +-------------------+-------------------+-------------------+  | 1.2 | 1.9 | 1 |  +-------------------+-------------------+-------------------+  1 row in set (0.00 sec)  mysql> SELECT TRUNCATE(-1.999,1), TRUNCATE(122,-2), TRUNCATE(10.28\*100,0);  +--------------------+------------------+-----------------------+  | TRUNCATE(-1.999,1) | TRUNCATE(122,-2) | TRUNCATE(10.28\*100,0) |  +--------------------+------------------+-----------------------+  | -1.9 | 100 | 1028 |  +--------------------+------------------+-----------------------+  1 row in set (0.00 sec) |

**26.1.11 GREATEST(숫자1,숫자2,숫자3...)**

====================

입력된 여러개의 수 중에서 제일 큰 수 반환합니다.

콘솔 실습)

|  |
| --- |
| mysql> select GREATEST(1,2,3,4,5,6,7);  +-------------------------+  | GREATEST(1,2,3,4,5,6,7) |  +-------------------------+  | 7 |  +-------------------------+  1 row in set (0.00 sec) |

**26.1.12 LEAST(숫자1,숫자2,숫자3...)**

====================

입력된 여러개의 수 중에서 제일 작은 수 를 반환합니다.

콘솔 실습)

|  |
| --- |
| mysql> select LEAST(1,2,3,4,5,6,7);  +----------------------+  | LEAST(1,2,3,4,5,6,7) |  +----------------------+  | 1 |  +----------------------+  1 row in set (0.00 sec) |

**26.2 문자열 내부함수**

====================

**26.2.1 CONCAT()**

====================

문자열을 결합할 수 있습니다. 결합하고자 하는 컬럼필드 또는 문자들은 콤마(,)로 구분합니다.

콘솔 출력화면)

|  |
| --- |
| mysql> select \* from members;  +----+----------+-----------+-------------+-------+---------+  | Id | LastName | FirstName | Address | City | Country |  +----+----------+-----------+-------------+-------+---------+  | 1 | hojin | lee | shinchon | seoul | korea |  | 2 | jiny | NULL | NULL | NULL | korea |  | 3 | james | kim | youngdun-po | seoul | korea |  | 4 | eric | jung | seocho-gu | seoul | korea |  +----+----------+-----------+-------------+-------+---------+  4 rows in set (0.00 sec)  mysql> select concat(FirstName, ' ', LastName, '님') as customer from members;  +--------------+  | customer |  +--------------+  | lee hojin님 |  | NULL |  | kim james님 |  | jung eric님 |  +--------------+  4 rows in set (0.00 sec) |

그냥 문자를 넣는 경우에는 작은따옴표(‘)로 감싸서 넣습니다.

**26.2.2 LEFT() / RIGHT()**

====================

컬럼의 데이터를 왼쯕/오른쪽 부터 지정한 숫자만큼만 출력합니다.

콘솔 출력화면)

|  |
| --- |
| // 왼쪽부터 2글자만 출력합니다.  mysql> select left(country,2) from members;  +-----------------+  | left(country,2) |  +-----------------+  | ko |  | ko |  | ko |  | ko |  +-----------------+  4 rows in set (0.00 sec)  // 오른쪽부터 3글자만 출력합니다.  mysql> select right(country,3) from members;  +------------------+  | right(country,3) |  +------------------+  | rea |  | rea |  | rea |  | rea |  +------------------+  4 rows in set (0.00 sec) |

**26.2.3 LENGTH()**

====================

문자열의 길이를 측정합니다.

콘솔 출력화면)

|  |
| --- |
| mysql> select name, length(name) from products;  +--------+--------------+  | name | length(name) |  +--------+--------------+  | S-827 | 5 |  | S-532D | 6 |  | R-530 | 5 |  +--------+--------------+  3 rows in set (0.00 sec) |

**26.2.4 공백제거 함수**

====================

LTRIM()은 좌측의 공백을 제거합니다. RTRIM()은 우측의 공백을 제거합니다.

콘솔 출력화면)

|  |
| --- |
| mysql> select ltrim(' jiny ');  +-----------------+  | ltrim(' jiny ') |  +-----------------+  | jiny |  +-----------------+  1 row in set (0.00 sec)  mysql> select rtrim(' jiny ');  +-----------------+  | rtrim(' jiny ') |  +-----------------+  | jiny |  +-----------------+  1 row in set (0.00 sec) |

**26.2.5 SUBSTRING() / MID()**

====================

컬럼의 데이터의 일부분만 출력합니다.

콘솔 출력화면)

|  |
| --- |
| mysql> select substring(Country, 2,2) from members;  +-------------------------+  | substring(Country, 2,2) |  +-------------------------+  | or |  | or |  | or |  | or |  +-------------------------+  4 rows in set (0.00 sec) |

**26.2.6 REPLACE()**

====================

문자열의 일부분을 치환할 수 있습니다.

콘솔 출력화면)

|  |
| --- |
| mysql> select \* from products;  +----+--------+-------+------+-------+  | Id | name | price | vat | stock |  +----+--------+-------+------+-------+  | 1 | S-827 | 20000 | 2000 | 5 |  | 2 | S-532D | 36000 | 3600 | 7 |  | 3 | R-530 | 25000 | 2500 | 10 |  +----+--------+-------+------+-------+  3 rows in set (0.00 sec)  mysql> select replace(name, 'S-', 'shiny S-') from products;  +---------------------------------+  | replace(name, 'S-', 'shiny S-') |  +---------------------------------+  | shiny S-827 |  | shiny S-532D |  | R-530 |  +---------------------------------+  3 rows in set (0.00 sec) |

**26.2.7 UPPER() / UCASE()**

====================

입력된 문자열을 대문자로 전환 합니다.

콘솔 출력화면)

|  |
| --- |
| mysql> select UCASE('abcde'), UPPER('abcde');  +----------------+----------------+  | UCASE('abcde') | UPPER('abcde') |  +----------------+----------------+  | ABCDE | ABCDE |  +----------------+----------------+  1 row in set (0.01 sec) |

**26.2.8 LCASE()**

====================

입력된 문자열을 소문자로 전환합니다.

콘솔 출력화면)

|  |
| --- |
| mysql> select name, LCASE(name) from products;  +--------+-------------+  | name | LCASE(name) |  +--------+-------------+  | S-827 | s-827 |  | S-532D | s-532d |  | R-530 | r-530 |  +--------+-------------+  3 rows in set (0.01 sec)  mysql> |

**26.2.9 REPEAT(문자, 반복횟수)**

====================

입력된 문자열을 지정 횟수만큼 반복 출력합니다.

콘솔 출력화면)

|  |
| --- |
| mysql> select \* from products;  +----+--------+-------+------+-------+  | Id | name | price | vat | stock |  +----+--------+-------+------+-------+  | 1 | S-827 | 20000 | 2000 | 5 |  | 2 | S-532D | 36000 | 3600 | 7 |  +----+--------+-------+------+-------+  2 rows in set (0.00 sec)  mysql> select repeat('\*',stock) from products;  +-------------------+  | repeat('\*',stock) |  +-------------------+  | \*\*\*\*\* |  | \*\*\*\*\*\*\* |  +-------------------+  2 rows in set (0.00 sec) |

재고 숫자만큼 \* 를 출력합니다.

**26.2.10 REVERSE()**

====================

입력된 문자열을 꺼꾸로 뒤집습니다.

콘솔 출력화면)

|  |
| --- |
| mysql> select reverse(Country) from members;  +------------------+  | reverse(Country) |  +------------------+  | aerok |  | aerok |  | aerok |  | aerok |  +------------------+  4 rows in set (0.00 sec) |

**26.2.11 ASCII(문자)**

====================

입력된 단일 문자의 아스키 코드값 리턴.

콘솔 실습)

|  |
| --- |
| mysql> select ascii('a');  +------------+  | ascii('a') |  +------------+  | 97 |  +------------+  1 row in set (0.00 sec) |

**26.2.12 INSTR('문자열','찾는문자열')**

====================

입력된 문자열 중 찾는 문자열의 위치값을 출력 합니다.

콘솔 실습)

|  |
| --- |
| mysql> select email, instr(email,'@') from members;  +-------------------+------------------+  | email | instr(email,'@') |  +-------------------+------------------+  | hojin@jinyphp.com | 6 |  | jiny@jinyphp.com | 5 |  | james@jinyphp.com | 6 |  | NULL | NULL |  | NULL | NULL |  | NULL | NULL |  | NULL | NULL |  +-------------------+------------------+  7 rows in set (0.00 sec)  mysql> |

**26.2.13 IF(논리식,참일 때 값,거짓일 때 값)**

====================

논리식이 참이면 참일 때 값을 출력하고 논리식이 거짓이면 거짓일 때 출력한다.

콘솔 실습)

|  |
| --- |
| mysql> select \* from products;  +----+--------+-------+------+-------+  | Id | name | price | vat | stock |  +----+--------+-------+------+-------+  | 1 | S-827 | 20000 | 2000 | 5 |  | 2 | S-532D | 36000 | 3600 | 7 |  | 3 | R-530 | 25000 | 2500 | 10 |  +----+--------+-------+------+-------+  3 rows in set (0.00 sec)  mysql> select stock, if(stock>5,'safeStock','Please order this') from products;  +-------+---------------------------------------------+  | stock | if(stock>5,'safeStock','Please order this') |  +-------+---------------------------------------------+  | 5 | Please order this |  | 7 | safeStock |  | 10 | safeStock |  +-------+---------------------------------------------+  3 rows in set (0.00 sec)  mysql> |

**26.2.14 IFNULL(값1,값2)**

====================

값1이 NULL 이면 값2로 대치하고 그렇지 않으면 값1을 출력

콘솔 실습)

|  |
| --- |
| mysql> select \* from members;  +-----+----------+-----------+-------------+----------+-------+---------+-----------+-------------------+  | Id | LastName | FirstName | Address | address2 | City | Country | manager | email |  +-----+----------+-----------+-------------+----------+-------+---------+-----------+-------------------+  | 1 | hojin | lee | shinchon | NULL | seoul | Korea | infohojin | hojin@jinyphp.com |  | 2 | lee | hojin | NULL | NULL | | Korea | infohojin | jiny@jinyphp.com |  | 3 | james | kim | youngdun-po | NULL | seoul | Korea | info | james@jinyphp.com |  | 5 | 1234 | jiny | NULL | NULL | | Korea | NULL | NULL |  | 6 | 1234 | jiny | NULL | NULL | | Korea | NULL | NULL |  | 7 | 123400 | jiny | NULL | NULL | | Korea | NULL | NULL |  | 100 | dfasdf | NULL | NULL | NULL | | NULL | NULL | NULL |  +-----+----------+-----------+-------------+----------+-------+---------+-----------+-------------------+  7 rows in set (0.00 sec)  mysql> select email, ifnull(email,'empty') from members;  +-------------------+-----------------------+  | email | ifnull(email,'empty') |  +-------------------+-----------------------+  | hojin@jinyphp.com | hojin@jinyphp.com |  | jiny@jinyphp.com | jiny@jinyphp.com |  | james@jinyphp.com | james@jinyphp.com |  | NULL | empty |  | NULL | empty |  | NULL | empty |  | NULL | empty |  +-------------------+-----------------------+  7 rows in set (0.01 sec) |

**26.3 SQL 날짜 함수**

====================

MYSQL은 시간과 날짜는 “proleptic Gregorian calendar” 그레고리력 을 사용합니다. 또한 다양한 시간관련 처리를 위한 내장함수를 제공합니다.

**26.3.1 NOW() , SYSDATE(), CURRENT\_TIMESTAMP()**

====================

현재의 날짜와 시간을 출력합니다.

콘솔 출력화면)

|  |
| --- |
| mysql> select now(), sysdate(), current\_timestamp();  +---------------------+---------------------+---------------------+  | now() | sysdate() | current\_timestamp() |  +---------------------+---------------------+---------------------+  | 2017-06-11 18:11:37 | 2017-06-11 18:11:37 | 2017-06-11 18:11:37 |  +---------------------+---------------------+---------------------+  1 row in set (0.01 sec) |

**26.3.2 CURDATE()**

====================

현재의 일자를 반환합니다.

콘솔 출력화면)

|  |
| --- |
| mysql> select curdate(), CURRENT\_DATE();  +------------+----------------+  | curdate() | CURRENT\_DATE() |  +------------+----------------+  | 2017-06-11 | 2017-06-11 |  +------------+----------------+  1 row in set (0.00 sec) |

**26.3.3 CURTIME()**

====================

현재의 시간을 반합니다.

콘솔 출력화면)

|  |
| --- |
| mysql> select curtime(), CURRENT\_TIME();  +-----------+----------------+  | curtime() | CURRENT\_TIME() |  +-----------+----------------+  | 18:10:28 | 18:10:28 |  +-----------+----------------+  1 row in set (0.00 sec) |

**26.3.4 DATE\_ADD(**날짜,INTERVAL 기준값**)**

====================

날짜에서 기준값 만큼 더한다.

콘솔 출력화면)

|  |
| --- |
| mysql> SELECT DATE\_ADD('2017-01-02', INTERVAL 42 DAY);  +-----------------------------------------+  | DATE\_ADD('2017-01-02', INTERVAL 42 DAY) |  +-----------------------------------------+  | 2017-02-13 |  +-----------------------------------------+  1 row in set (0.00 sec)  mysql> SELECT DATE\_ADD('2017-06-09', INTERVAL 1 MONTH);  +------------------------------------------+  | DATE\_ADD('2017-06-09', INTERVAL 1 MONTH) |  +------------------------------------------+  | 2017-07-09 |  +------------------------------------------+  1 row in set (0.00 sec)  mysql> SELECT DATE\_ADD('2017-06-09', INTERVAL 1 YEAR);  +-----------------------------------------+  | DATE\_ADD('2017-06-09', INTERVAL 1 YEAR) |  +-----------------------------------------+  | 2018-06-09 |  +-----------------------------------------+  1 row in set (0.00 sec) |

※ 기준값 : YEAR, MONTH, DAY, HOUR, MINUTE, SECOND

**26.3.5 DATE\_SUB(날짜,INTERVAL 기준값)**

====================

날짜에서 기준값 만큼 뺀다

콘솔 출력화면)

|  |
| --- |
| mysql> SELECT DATE\_SUB('2017-01-02', INTERVAL 42 DAY);  +-----------------------------------------+  | DATE\_SUB('2017-01-02', INTERVAL 42 DAY) |  +-----------------------------------------+  | 2016-11-21 |  +-----------------------------------------+  1 row in set (0.00 sec)  mysql> SELECT DATE\_SUB('2017-06-09', INTERVAL 1 MONTH);  +------------------------------------------+  | DATE\_SUB('2017-06-09', INTERVAL 1 MONTH) |  +------------------------------------------+  | 2017-05-09 |  +------------------------------------------+  1 row in set (0.00 sec)  mysql> SELECT DATE\_SUB('2017-06-09', INTERVAL 1 YEAR);  +-----------------------------------------+  | DATE\_SUB('2017-06-09', INTERVAL 1 YEAR) |  +-----------------------------------------+  | 2016-06-09 |  +-----------------------------------------+  1 row in set (0.00 sec) |

※ 기준값 : YEAR, MONTH, DAY, HOUR, MINUTE, SECOND

**26.3.6 YEAR(날짜), MONTH(날짜), DAY(날짜)**

====================

입력된 날짜에서 각각 년도, 월, 일 을 찾아서 리턴합니다.

콘솔 실습)

|  |
| --- |
| mysql> select year('2017-06-09'), month('2017-06-09'), day('2017-06-09');  +--------------------+---------------------+-------------------+  | year('2017-06-09') | month('2017-06-09') | day('2017-06-09') |  +--------------------+---------------------+-------------------+  | 2017 | 6 | 9 |  +--------------------+---------------------+-------------------+  1 row in set (0.00 sec) |

**26.3.7 HOUR(), MINUTE(), SECOND()**

====================

시간, 분, 초 를 반환합니다.

콘솔 실습)

|  |
| --- |
| mysql> select hour(now()), minute(now()), second(now());  +-------------+---------------+---------------+  | hour(now()) | minute(now()) | second(now()) |  +-------------+---------------+---------------+  | 17 | 15 | 46 |  +-------------+---------------+---------------+  1 row in set (0.00 sec) |

**26.3.8 MONTHNAME(날짜), DAYNAME(날짜)**

====================

날짜의 월을 영어로 출력. 날짜의 요일일 영어로 출력.

콘솔 실습)

|  |
| --- |
| mysql> select MONTHNAME('2017-06-09'), DAYNAME('2017-06-09');  +-------------------------+-----------------------+  | MONTHNAME('2017-06-09') | DAYNAME('2017-06-09') |  +-------------------------+-----------------------+  | June | Friday |  +-------------------------+-----------------------+  1 row in set (0.00 sec) |

**26.3.9 DAYOFMONTH(날짜),**

====================

날짜의 월별 일자 출력.

콘솔 실습)

|  |
| --- |
| mysql> select DAYOFMONTH('2017-06-09');  +--------------------------+  | DAYOFMONTH('2017-06-09') |  +--------------------------+  | 9 |  +--------------------------+  1 row in set (0.00 sec) |

**26.3.10 DAYOFWEEK(날짜)**

====================

날짜의 주별 일자 출력(월요일(0),화요일(1)...일요일(6))

콘솔 실습)

|  |
| --- |
| ysql> select DAYOFWEEK('2017-06-09');  +-------------------------+  | DAYOFWEEK('2017-06-09') |  +-------------------------+  | 6 |  +-------------------------+  1 row in set (0.00 sec) |

**26.3.11 WEEKDAY(날짜)**

====================

날짜의 주별 일자 출력(월요일(0),화요일(1)...일요일(6))

콘솔 실습)

|  |
| --- |
| mysql> select WEEKDAY('2017-06-09');  +-----------------------+  | WEEKDAY('2017-06-09') |  +-----------------------+  | 4 |  +-----------------------+  1 row in set (0.00 sec) |

**26.3.12 QUARTER(날짜)**

====================

입력한 날짜의 분기를 출력합니다.

콘솔 실습)

|  |
| --- |
| mysql> select quarter('2017-07-09');  +-----------------------+  | quarter('2017-07-09') |  +-----------------------+  | 3 |  +-----------------------+  1 row in set (0.00 sec) |

**26.3.13 DAYOFYEAR(날짜)**

====================

일년을 기준으로 한 날짜까지의 날 수.

콘솔 실습)

|  |
| --- |
| mysql> select DAYOFYEAR('2017-06-09');  +-------------------------+  | DAYOFYEAR('2017-06-09') |  +-------------------------+  | 160 |  +-------------------------+  1 row in set (0.00 sec) |

**26.3.14 WEEK(날짜)**

====================

일년 중 몇 번째 주.

콘솔 실습)

|  |
| --- |
| mysql> select WEEK('2017-06-09');  +--------------------+  | WEEK('2017-06-09') |  +--------------------+  | 23 |  +--------------------+  1 row in set (0.00 sec) |

**26.3.15 FROM\_DAYS(날 수)**

====================

0000년 00월 00일부터 날 수 만큼 경과한 날의 날짜 출력.

콘솔 실습)

|  |
| --- |
| mysql> select from\_days(736854);  +-------------------+  | from\_days(736854) |  +-------------------+  | 2017-06-09 |  +-------------------+  1 row in set (0.00 sec) |

**26.3.16 TO\_DAYS(날짜)**

====================

0000 년 00 월 00일 부터 날짜까지의 일자 수 출력.

콘솔 실습)

|  |
| --- |
| mysql> select to\_days('2017-06-09');  +-----------------------+  | to\_days('2017-06-09') |  +-----------------------+  | 736854 |  +-----------------------+  1 row in set (0.00 sec) |

**26.3.17 DATE\_FORMAT(날짜,'형식')**

====================

날짜를 형식에 맞게 출력 합니다.

콘솔 실습)

|  |
| --- |
| mysql> select date\_format(now(),'%Y %M %D %W');  +----------------------------------+  | date\_format(now(),'%Y %M %D %W') |  +----------------------------------+  | 2017 June 9th Friday |  +----------------------------------+  1 row in set (0.00 sec) |

**26.3.18 TIME\_FORMAT(날짜,'형식')**

====================

시간과 형식을 문자열로 반환을 합니다.

콘솔 실습)

|  |
| --- |
| mysql> select time\_format(now(),'%p %k %i');  +-------------------------------+  | time\_format(now(),'%p %k %i') |  +-------------------------------+  | PM 17 05 |  +-------------------------------+  1 row in set (0.00 sec) |

**26.3.19 TIME\_TO\_SEC(시간)**

====================

입력한 시간의 0시부터 초를 반환합니다.

콘솔 실습)

|  |
| --- |
| mysql> select time\_to\_sec(curtime());  +------------------------+  | time\_to\_sec(curtime()) |  +------------------------+  | 61734 |  +------------------------+  1 row in set (0.00 sec) |

**26.3.20 SEC\_TO\_TIME(N)**

====================

0시부터 N초만큼 지난 시간을 반환합니다.

콘솔 실습)

|  |
| --- |
| mysql> select sec\_to\_time(1000);  +-------------------+  | sec\_to\_time(1000) |  +-------------------+  | 00:16:40 |  +-------------------+  1 row in set (0.00 sec) |

**24.4 수학 함수**

====================

**24.4.1 PI()**

====================

원주율을 계산하는 함수 입니다.

콘솔 출력화면)

|  |
| --- |
| mysql> select pi(), 2\*pi();  +----------+----------+  | pi() | 2\*pi() |  +----------+----------+  | 3.141593 | 6.283185 |  +----------+----------+  1 row in set (0.00 sec) |

**24.4.2 POW(X,Y) or POWER(X,Y)**

====================

입력된 X의 Y승 을 계산합니다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT POW(2,2), POW(2,-2);  +----------+-----------+  | POW(2,2) | POW(2,-2) |  +----------+-----------+  | 4 | 0.25 |  +----------+-----------+  1 row in set (0.00 sec) |

**24.4.3 MOD (분자, 분모)**

====================

연산자 %와 같이 입력된 분자를 분모로 나눈 나머지를 구합니다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT MOD(234, 10), 253 % 7, MOD(29,9), 29 MOD 9;  +--------------+---------+-----------+----------+  | MOD(234, 10) | 253 % 7 | MOD(29,9) | 29 MOD 9 |  +--------------+---------+-----------+----------+  | 4 | 1 | 2 | 2 |  +--------------+---------+-----------+----------+  1 row in set (0.00 sec) |

**24.4.4 SQRT(*X*)**

====================

음수가 아닌 입력된 값의 제곱루트 값을 출력합니다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT SQRT(4), SQRT(20), SQRT(-16);  +---------+------------------+-----------+  | SQRT(4) | SQRT(20) | SQRT(-16) |  +---------+------------------+-----------+  | 2 | 4.47213595499958 | NULL |  +---------+------------------+-----------+  1 row in set (0.00 sec) |

**24.4.5 TAN(*X*)**

====================

라디안으로 주어진 x의 탄젠트 값을 반환합니다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT TAN(PI()), TAN(PI()+1);  +-------------------------+--------------------+  | TAN(PI()) | TAN(PI()+1) |  +-------------------------+--------------------+  | -1.2246467991473532e-16 | 1.5574077246549018 |  +-------------------------+--------------------+  1 row in set (0.01 sec) |

**24.4.6 SIN(*X*)**

====================

라디안으로 주어진 *X*의 사인(sin) 값을 반환합니다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT SIN(PI()), ROUND(SIN(PI()));  +------------------------+------------------+  | SIN(PI()) | ROUND(SIN(PI())) |  +------------------------+------------------+  | 1.2246467991473532e-16 | 0 |  +------------------------+------------------+  1 row in set (0.00 sec) |

**24.4.7 SIGN(*X*)**

====================

인수의 부호를 -1, 0, 또는 1로 반환합니다. 리턴 되는 숫자는 각각 음수, 0, 양수를 표시합니다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT SIGN(-32), SIGN(0), SIGN(234);  +-----------+---------+-----------+  | SIGN(-32) | SIGN(0) | SIGN(234) |  +-----------+---------+-----------+  | -1 | 0 | 1 |  +-----------+---------+-----------+  1 row in set (0.01 sec) |

**24.4.8 COT(*X*)**

====================

*입력된 X* 의 코 탄젠트 값을 리턴한다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT COT(12);  +---------------------+  | COT(12) |  +---------------------+  | -1.5726734063976893 |  +---------------------+  1 row in set (0.00 sec) |

**24.4.9 COS(*X*)**

====================

*입력된 X* 의 코사인 값을 리턴 하는데, 여기에서 *X* 는 라디안(radian)으로 주어진다.

콘솔 실습)

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

**24.4.10 ATAN(*X*)**

====================

*X*의 아크 탄젠트(arc tangent) 값을 반환 합니다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT ATAN(2), ATAN(-2);  +--------------------+---------------------+  | ATAN(2) | ATAN(-2) |  +--------------------+---------------------+  | 1.1071487177940904 | -1.1071487177940904 |  +--------------------+---------------------+  1 row in set (0.00 sec) |

**24.4.11 ATAN(*Y*,*X*), ATAN2(*Y*,*X*)**

====================

두 변수 *X* 와 *Y*의 아크 탄젠트 값을 반환합니다. 이것은 *Y* / *X*의 아크 탄젠트 값을 계산하는 방식과 유사하지만, 두 인수의 부호는 결과에 대한 쿼드런트 (quadrant)를 알아내기 위해 사용된다는 점은 다릅니다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT ATAN(-2,2), ATAN2(PI(),0);  +---------------------+--------------------+  | ATAN(-2,2) | ATAN2(PI(),0) |  +---------------------+--------------------+  | -0.7853981633974483 | 1.5707963267948966 |  +---------------------+--------------------+  1 row in set (0.01 sec) |

**24.4.12 ACOS(*X*)**

====================

*X*의 아크 코사인(arc cosine) 값을 반환합니다. 코사인이 *X*인 값을 리턴한다. 만일 *X* 가 -1 에서 1 사이의 범위에 있지 않으면, NULL을 반환합니다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT ACOS(1), ACOS(1.0001), ACOS(0);  +---------+--------------+--------------------+  | ACOS(1) | ACOS(1.0001) | ACOS(0) |  +---------+--------------+--------------------+  | 0 | NULL | 1.5707963267948966 |  +---------+--------------+--------------------+  1 row in set (0.01 sec) |

**24.4.13 ASIN(*X*)**

====================

*X* 의 아크 사인 값(arc sine)을 반환합니다. 사인 값이 *X*인 값. 만일 *X* 가 -1 에서 1 사이의 범위에 있지 않으면, NULL을 반환 합니다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT ASIN(0.2);  +--------------------+  | ASIN(0.2) |  +--------------------+  | 0.2013579207903308 |  +--------------------+  1 row in set (0.00 sec)  mysql> SELECT ASIN('foo');  +-------------+  | ASIN('foo') |  +-------------+  | 0 |  +-------------+  1 row in set, 1 warning (0.00 sec)  mysql> SHOW WARNINGS;  +---------+------+-----------------------------------------+  | Level | Code | Message |  +---------+------+-----------------------------------------+  | Warning | 1292 | Truncated incorrect DOUBLE value: 'foo' |  +---------+------+-----------------------------------------+  1 row in set (0.00 sec) |

**24.4.14 RADIANS(*X*)**

====================

각도를 라디안으로 변경한 인수 *X*를 리턴합니다. ( 라디안은 180 도와 같습니다)

콘솔 실습)

|  |
| --- |
| mysql> SELECT RADIANS(90);  +--------------------+  | RADIANS(90) |  +--------------------+  | 1.5707963267948966 |  +--------------------+  1 row in set (0.00 sec) |

**24.4.15 LOG10(*X*)**

====================

*X*의 베이스-10 로그를 반환합니다. LOG10(*X*)는 LOG(10,*X*)와 같습니다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT LOG10(2), LOG10(100), LOG10(-100);  +--------------------+------------+-------------+  | LOG10(2) | LOG10(100) | LOG10(-100) |  +--------------------+------------+-------------+  | 0.3010299956639812 | 2 | NULL |  +--------------------+------------+-------------+  1 row in set (0.00 sec) |

**24.4.16 LOG2(*X*)**

====================

*X*의 베이스-e 로그를 반환합니다. LOG2()는 스토리지용으로 필요한 비트의 수가 얼마나 되는지를 알아 보는데 유용합니다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT LOG2(65536), LOG2(-100);  +-------------+------------+  | LOG2(65536) | LOG2(-100) |  +-------------+------------+  | 16 | NULL |  +-------------+------------+  1 row in set (0.00 sec) |

**24.4.17 LOG(*X*), LOG(*B*,*X*)**

====================

X의 자연 로그를 반환합니다. 매개변수 인자로 2개를 전달하면 부정 베이스 (arbitrary base) B 에 대한 X의 자연 로그를 반환 합니다. LOG(B,X)는 LOG(X) / LOG(B)와 같습니다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT LOG(2), LOG(-2), LOG(2,65536), LOG(10,100);  +--------------------+---------+--------------+-------------+  | LOG(2) | LOG(-2) | LOG(2,65536) | LOG(10,100) |  +--------------------+---------+--------------+-------------+  | 0.6931471805599453 | NULL | 16 | 2 |  +--------------------+---------+--------------+-------------+  1 row in set (0.00 sec) |

**24.4.18 LN(*X*)**

====================

X의 자연 로그를 반환합니다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT LN(2), LN(-2);  +--------------------+--------+  | LN(2) | LN(-2) |  +--------------------+--------+  | 0.6931471805599453 | NULL |  +--------------------+--------+  1 row in set (0.01 sec) |

**24.4.19 FORMAT(*X*,*D*)**

====================

숫자 X 의 형태를 '#,###,###.##'로 변경 시키고, D 자릿수 위치에서 삭제 한 다음에, 그 결과를 스트링으로 반환합니다.

콘솔 실습)

|  |
| --- |
| mysql> select format(123456789,'#,###,###.###');  +-----------------------------------+  | format(123456789,'#,###,###.###') |  +-----------------------------------+  | 123,456,789 |  +-----------------------------------+  1 row in set, 1 warning (0.00 sec) |

**24.4.20 EXP(*X*)**

====================

입력된 X 의 제곱에 대한 e (자연 로그 근) 값을 반환합니다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT EXP(2), EXP(-2), EXP(0);  +------------------+--------------------+--------+  | EXP(2) | EXP(-2) | EXP(0) |  +------------------+--------------------+--------+  | 7.38905609893065 | 0.1353352832366127 | 1 |  +------------------+--------------------+--------+  1 row in set (0.01 sec) |

**24.4.21 DEGREES(*X*)**

====================

입력된 인수 *X*를 반환하는데, 레디안에서 차수 (degree)로 변경된 값을 출력합니다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT DEGREES(PI()), DEGREES(PI() / 2);  +---------------+-------------------+  | DEGREES(PI()) | DEGREES(PI() / 2) |  +---------------+-------------------+  | 180 | 90 |  +---------------+-------------------+  1 row in set (0.01 sec) |

**24.4.22 CRC32(*expr*)**

====================

사이클릭 리던던시 체크 값 (cyclic redundancy check value)을 계산후에 32-비트 부호화되지 않은 값을 반환한다. 인수가 NULL이면 NULL 을 리턴한다. 인수는 스트링 값이 되어야 한다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT CRC32('MySQL'), CRC32('mysql');  +----------------+----------------+  | CRC32('MySQL') | CRC32('mysql') |  +----------------+----------------+  | 3259397556 | 2501908538 |  +----------------+----------------+  1 row in set (0.02 sec) |

**24.4.23 RAND(), RAND(*N*)**

====================

RAND 함수는 0과 1 사이에 있는 임의 부동 소수점 하나를 반환합니다. 만일, 인수로 정수N을 입력하면 이값을 기본값으로하여 반복 시퀀스를 출력합니다.

콘솔 실습)

|  |
| --- |
| mysql> SELECT RAND();  +---------------------+  | RAND() |  +---------------------+  | 0.36988361773070283 |  +---------------------+  1 row in set (0.00 sec)  mysql> SELECT RAND(20);  +---------------------+  | RAND(20) |  +---------------------+  | 0.15888261251047497 |  +---------------------+  1 row in set (0.00 sec)  mysql> SELECT RAND(20);  +---------------------+  | RAND(20) |  +---------------------+  | 0.15888261251047497 |  +---------------------+  1 row in set (0.00 sec) |

**26.5 기타**

====================

MYSQL은 쿼리 명령을 통하여 MYSQL의 상태나 정보를 확인할 수 있는 내부함수들을 몇가지 제공하고 있습니다. 유용한 함수들 몇가지만 본 지면에서 소개하도록 하겠습니다.

**26.5.1 VERSION()**

====================

현재 MYSQL의 버전을 출력합니다. 만일 자신의 응용프로그램이 MYSQL의 버젼을 체크하여야 한다면 이 함수를 사용하여 처리하면 좋을듯 합니다.

콘솔 실습)

|  |
| --- |
| mysql> select version();  +-----------+  | version() |  +-----------+  | 5.6.35 |  +-----------+  1 row in set (0.00 sec) |

**26.5.2 DATABASE()**

====================

MYSQL은 수많은 데이터베이스(폴더)명을 관리합니다. 현재 작업 선택되어 있는 데이터베이스명 출력합니다.

콘솔 실습)

|  |
| --- |
| mysql> select database();  +------------+  | database() |  +------------+  | jiny |  +------------+  1 row in set (0.00 sec) |

**26.5.3 USER()**

====================

현재 접속되어진 user 정보를 출력합니다.

콘솔 실습)

|  |
| --- |
| mysql> select user();  +----------------+  | user() |  +----------------+  | root@localhost |  +----------------+  1 row in set (0.00 sec) |

**26.5.4 CHARSET(문자열)**

====================

입력 매개값(문자열)의 문자코드셋 정보를 출력합니다.

콘솔 실습)

|  |
| --- |
| mysql> select charset('jiny');  +-----------------+  | charset('jiny') |  +-----------------+  | utf8 |  +-----------------+  1 row in set (0.00 sec) |