jinyPHP 2권 - 006. 데이터타입

**06. 데이터타입**

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

테이블에는 각각 데이터를 저장하기 위한 다수의 컬럼들을 같이 선언합니다. 컬럼의 선언 할 때는 컬럼의 데이터 타입도 같이 지정을 하게 됩니다. 데이터의 타입을 같이 지정 하는 이유는 테이블의 데어 저장공간을 효율적으로 사용하기 위해서 입니다. 또한, 빠른 접근과 처리를 위해서 입니다.

테이블의 컬럼에서 데이터 타입을 지정하는 것은 프로그램 언어에서 변수타입을 지정하는 것과 같은 의미 입니다. 데이터의 타입은 입력되는 데이터의 종류에도 제한을 하게 됩니다. 데이터 타입은 입력되는 자료의 신뢰성을 높여줍니다. 따라서 실수로 잘못된 데이터를 입력하는 것을 방지하기도 합니다.

만일, 데이터 타입과 일치하는 유형의 자료를 입력하고자 할 때에는 DB시스템이 쿼리 실행 오류를 발생하기도 합니다.

데이터 타입은 크게 4가지로 구분해 볼 수 있습니다.

* 숫자형
* 문자형
* 날짜형
* 그외

그중 가장 많이 사용하는 형식만 살펴봅니다.

**06.1 숫자형 데이터**

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

숫자형 데이터란 숫자의 값을 저장하는 유형입니다. 컴퓨터는 우리가 수학적으로 알고있는 모든 가능한 수를 다 표현을 할 수 없습니다. 그 이유는 컴퓨터는 모든 데이터를 2진수의 비트 값으로 처리하기 때문입니다.

컴퓨터는 2진법과 데이터 비트수의 한계로 정확한 값을 표현하는 정수형 타입과 근사값을 표현하는 부동소수점 타입으로 구분할 수 있습니다. MYSQL에서는 정수의 데이터 타입을 데이터 비트의 크기별로 5가지로 구분을 합니다.

부동 소수형은 크게 float형과 double형으로 구분됩니다. 부동 소수형 일반점으로 점(.)을 포함하는 실수의 값을 처리합니다. 이때 소수형 앞에 있는 부동이라는 의미는 소수점의 위치가 고정되어 있지 않다는 의미입니다. 표현을 해야 하는 숫자의 크기에 따라서 유효한 범위의 소수점 자리가 변경이 됩니다.

숫자는 처리 보관하는 바이트 크기에 따라서 표현할 수 있는 숫자의 범위가 달라 집니다. 따라서 정확한 유효숫자를 통하여 크기를 결정을 하는 것은 데이터의 저장공간을 효율적으로 사용하는데 매우 중요합니다.

**06.1.1 INT**

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

int 형타입은 가장 기본적인 숫자형 데이터 타입입니다. 정수형 자료의 선언할 때는 컬럼명 뒤에 int(자리수) 형태로 선언하면 됩니다. 정수형 타입은 4바이트의 크기로 -2147483648 ~ 2147483647 사이의 정수값을 저장할 수 있습니다.

**테이블 생성시)**

|  |
| --- |
| CREATE TABLE `tableName` (  컬럼명 **int(11)**,  ….  } |

정수형 타입의 데이터형은 테이블 생성시에 같이 지정하여 선언할 수 있습니다. 또는 기존 테이블에 새로운 정수형 데이터 타입의 컬럼을 추가할 수 있습니다.

**컬럼추가)**

|  |
| --- |
| ALTER TABLE 테이블명ADD 컬럼이름 **int(11)**; |

**06.1.2 TINYINT**

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

tinyint 형은 기본 int형 데이터 타입 보다 작은 정수형 타입입니다. 타입선언 방법은 컬럼명 뒤에 tinyint(자리수) 형태로 선언하면 됩니다. tinyint 형은 1바이트 크기의 저장공간으로 -128 ~ 127 사이의 값을 저장할 수 있습니다.

**테이블 생성시)**

|  |
| --- |
| CREATE TABLE `tableName` (  컬럼명 **tinyint(2)**,  ….  } |

tinyint 정수 데이터형은 테이블 생성시에 같이 지정하여 선언할 수 있습니다. 또는 기존 테이블에 새로운 데이터 타입의 컬럼을 추가할 수 있습니다.

**컬럼추가)**

|  |
| --- |
| ALTER TABLE 테이블명 ADD 컬럼이름 **tinyint(2)**; |

**06.1.3 SMALLINT**

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

smallint 형은 기존 16비트타입의 값을 가지는 정수 데이터형 입니다. 정수형 자료의 선언은 컬럼명 뒤에 smallint(자리수) 형태로 선언을 합니다. smallint 형은 2바이트 크기의 저장공간으로 -32768 ~ 32767 사이의 값을 저장할 수 있습니다.

**테이블 생성시)**

|  |
| --- |
| CREATE TABLE `tableName` (  컬럼명 **smallint(5)**,  ….  } |

smallint 정수 데이터형은 테이블 생성시에 같이 지정하여 선언할 수 있습니다. 또는 기존 테이블에 새로운 데이터 타입의 컬럼을 추가할 수 있습니다.

**컬럼추가)**

|  |
| --- |
| ALTER TABLE 테이블명 ADD 컬럼이름 **smallint(5)**; |

**06.1.4 MEDIUMINT**

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

mediumint 형은 3바이트의 크기로 -8388608 ~ 8388607 사이의 값을 저장하는 자료형 입니다. 정수형 자료의 선언은 컬럼명 뒤에 mediumint(자리수) 형태로 선언을 합니다.

**테이블 생성시)**

|  |
| --- |
| CREATE TABLE `tableName` (  컬럼명 **mediumint(9)**,  ….  } |

smallint 정수 데이터형은 테이블 생성시에 같이 지정하여 선언할 수 있습니다. 또는 기존 테이블에 새로운 데이터 타입의 컬럼을 추가할 수 있습니다.

**컬럼추가)**

|  |
| --- |
| ALTER TABLE 테이블명 ADD 컬럼이름 **mediumint(9)**; |

**06.1.5 BIGINT**

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

bigint형은 8바이트 크기로 -92233772036854775808 ~ 9223372036854775807 사이의 정수값 저장하는 자료형 입니다. bitint형은 가장 큰 숫자의 정수값을 표현을 할 수 있습니다. 정수형 자료의 선언은 컬럼명 뒤에 bigint(자리수) 형태로 선언을 합니다.

**테이블 생성시)**

|  |
| --- |
| CREATE TABLE `tableName` (  컬럼명 **bigint(19)**,  ….  } |

bigint 정수 데이터형은 테이블 생성시에 같이 지정하여 선언할 수 있습니다. 또는 기존 테이블에 새로운 데이터 타입의 컬럼을 추가할 수 있습니다.

**컬럼추가)**

|  |
| --- |
| ALTER TABLE 테이블명 ADD 컬럼이름 **bigint(19)**; |

**06.1.6 FLOAT**

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

float형은 가장 일반적으로 많이 사용을 하는 부동소수형 데이터 타입니다. 우리가 일반적으로 알고 있는 소수점을 포함하는 실수를 저장을 합니다.

컴퓨터에서 실수의 표현은 2진법의 컴퓨터 데이터 비트의 한계에 따라서 값이 정확하지 않은 근사값으로 표현을 합니다. float형은 정밀도가 표시되지 않은 경우에는 4바이트의 크기를 가지며, 유효 자리수 8개를 유지합니다.

컬럼명 뒤에 float() 형태로 선언을 하면됩니다.

**테이블 생성시)**

|  |
| --- |
| CREATE TABLE `tableName` (  컬럼명 **float(3,2)**,  ….  } |

float 데이터형은 테이블 생성시에 같이 지정하여 선언할 수 있습니다. 또는 기존 테이블에 새로운 데이터 타입의 컬럼을 추가할 수 있습니다.

**컬럼추가)**

|  |
| --- |
| ALTER TABLE 테이블명 ADD 컬럼이름 **float(3,2)**; |

부동소수점의 값은 컴퓨터의 근사값 입니다. 만일, MYSQL을 마스터-슬레이브 형태의 다수 모델로 설계하여 처리를 할 때 유효자리수를 넘어서는 부분들은 서로 달라질 수 있습니다.

**06.1.7 DOUBLE**

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

double 형은 float형보다 더 큰 부동소수점을 표기하고 저장을 할 수 있습니다. double 형은 8바이트의 저장공간을 사용합니다. 또한 유효 자리수를 16개를 유지합니다.

컬럼명 뒤에 double() 형태로 선언을 하면됩니다.

**테이블 생성시)**

|  |
| --- |
| CREATE TABLE `tableName` (  컬럼명 **double(4,3)**,  ….  } |

double 데이터형은 테이블 생성시에 같이 지정하여 선언할 수 있습니다. 또는 기존 테이블에 새로운 데이터 타입의 컬럼을 추가할 수 있습니다.

**컬럼추가)**

|  |
| --- |
| ALTER TABLE 테이블명 ADD 컬럼이름 **double(4,3)**; |

**06.1.8 DECIMAL**

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

float형과 double형과 같은 데이터타입은 부종 소수점으로 유효범위 외의 값들은 정확하지 않습니다. decimal 타입은 고정소수점 형태의 데이터 타입입니다. decimal 타입은 고정 소수점으로 최대자리수는 65자, 소수점은 30자리 까지 지정할 수 있습니다.

컬럼명 뒤에 double() 형태로 선언을 하면됩니다.

**테이블 생성시)**

|  |
| --- |
| CREATE TABLE `tableName` (  컬럼명 **decimal(10,2)**,  ….  } |

decimal 데이터형은 테이블 생성시에 같이 지정하여 선언할 수 있습니다. 또는 기존 테이블에 새로운 데이터 타입의 컬럼을 추가할 수 있습니다.

**컬럼추가)**

|  |
| --- |
| ALTER TABLE 테이블명 ADD 컬럼이름 **decimal(10,2)**; |

**06.2 문자 자료형**

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

데이터베이스에서 가장 많은 자료를 포함하고 있는 데이터 타입은 문자형 입니다. MYSQL은 다양한 문자들을 저장하고 처리를 할 수 있습니다. 문자의 데이터형은 문자의 길이에 많은 상관 관계가 있습니다.

대표적으로 가장 많이 사용을 하는 문자형 데이터 타입은 CAHR 와 VARCHAR가 입니다. 이 두개는 문자를 저장할 때 고정된 크기로 저장공간을 사용할지 여부의 차이입니다. 또한 255 이상의 긴 문자열의 경우에는 text, longtext 를 사용하여 저장을 합니다.

**06.2.1 CHAR**

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

char형은 고정된 크기의 문자열을 저장할 수 있는 공간을 할당합니다. char데이터형을 사용할 때 컬럼명 뒤에 char() 형태로 선언을 합니다. 뒤에 붙은 ()안에 저장하고자 하는 데이터의 크기를 지정합니다. 최대 크기는 255까지 입니다.

char로 선언된 데이터 크기 보다 작은 글자가 입력된 경우에는 나머지 영역을 공백으로 처리합니다. 그만큼 데이터의 공간 손실이 발생할 수 있습니다. 하지만, char는 varchar 테이터 타입보다 검색속도 및 처리가 빠른 것이 장점입니다.

**테이블 생성시)**

|  |
| --- |
| CREATE TABLE `tableName` (  컬럼명 **char(15)**,  ….  } |

char 데이터형은 테이블 생성시에 같이 지정하여 선언할 수 있습니다. 또는 기존 테이블에 새로운 데이터 타입의 컬럼을 추가할 수 있습니다.

**컬럼추가)**

|  |
| --- |
| ALTER TABLE 테이블명 ADD 컬럼이름 **char(15)**; |

**06.2.2 VARCHAR**

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

varchar는 가변 문자열을 처리하는 데이터형 입니다. varchar는 char 형 보다 데이터의 저장공간을 유연하게 처리를 합니다. varchar데이터형을 사용할 때 컬럼명 뒤에 varchar() 형태로 선언을 하면 됩니다.

varchar가 char 보다 유연한 것은 괄호 () 안에 문자크기 보다 적은 값을 입력하여도 공백을 처리하지 않고 데이터의 용량을 줄여서 저장한 다는 것입니다. 따라서 불특정한 데이터의 경우 varchar는 데이터의 저장공간을 절약을 할 수 있습니다.

varchar는 가변적인 문자열의 길이를 관리하기 위해서 설정한 최대 크기보다 추가로 1~2 바이트의 추가 공간이 필요로 합니다.

**테이블 생성시)**

|  |
| --- |
| CREATE TABLE `tableName` (  컬럼명 **varchar(255)**,  ….  } |

varchar 데이터형은 테이블 생성시에 같이 지정하여 선언할 수 있습니다. 또는 기존 테이블에 새로운 데이터 타입의 컬럼을 추가할 수 있습니다.

컬럼추가)

|  |
| --- |
| ALTER TABLE 테이블명 ADD 컬럼이름 **varchar(255)**; |

**06.2.3 TEXT**

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

char와 varchar의 데이터형은 최대 문자를 저장하고 처리하는데 길이의 한계가 있습니다. 보다 많은 데이터를 저장하고 처리하기 위해서는 text형 데이터 타입을 사용합니다. text형 데이터 타입은 최대 65535 크기의 문자를 저장할 수 있습니다.

**테이블 생성시)**

|  |
| --- |
| CREATE TABLE `tableName` (  컬럼명 **text**,  ….  } |

text 데이터형은 테이블 생성시에 같이 지정하여 선언할 수 있습니다. 또는 기존 테이블에 새로운 데이터 타입의 컬럼을 추가할 수 있습니다.

**컬럼추가)**

|  |
| --- |
| ALTER TABLE 테이블명 ADD 컬럼이름 **text**; |

**06.2.4 LONGTEXT**

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

Longtext 형은 text형 보다 더 큰 크기의 문자를 저장할 수 있습니다. 저장 처리할 수 있는 최대크기는 4294967295 입니다.

**테이블 생성시)**

|  |
| --- |
| CREATE TABLE `tableName` (  컬럼명 **longtext**,  ….  } |

longtext 데이터형은 테이블 생성시에 같이 지정하여 선언할 수 있습니다. 또는 기존 테이블에 새로운 데이터 타입의 컬럼을 추가할 수 있습니다.

**컬럼추가)**

|  |
| --- |
| ALTER TABLE 테이블명 ADD 컬럼이름 **longtext**; |

**06.3 날짜/시간 자료형**

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

데이터베이스에서 각각의 데이터 자체만으로도 의미가 있겠지만, 데이터의 생성 및 처리에 대한 시간 또한 중요합니다. MYSQL에서는 데이터의 날짜와 시간을 저장할 수 있는 다양한 데이터 타입을 제공합니다.

**06.3.1 DATETIME**

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

datetime 형은 날짜와 시간을 같이 표시하는 데이터 타입 입니다. 년도-월-일 시간:분:초 까지 같이 저장을 합니다.

**테이블 생성시)**

|  |
| --- |
| CREATE TABLE `tableName` (  컬럼명 **datetime**,  ….  } |

**컬럼추가)**

|  |
| --- |
| ALTER TABLE 테이블명 ADD 컬럼이름 **datetime**; |

**06.3.2 DATE**

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

date형은 날짜만 표시합니다. 년도-월-일 까지 표시합니다.

**테이블 생성시)**

|  |
| --- |
| CREATE TABLE `tableName` (  컬럼명 **date**,  ….  } |

**컬럼추가)**

|  |
| --- |
| ALTER TABLE 테이블명 ADD 컬럼이름 **date**; |

**06.3.3 YEAR**

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

연도만 표시합니다.

**테이블 생성시)**

|  |
| --- |
| CREATE TABLE `tableName` (  컬럼명 **year(4)**,  ….  } |

**컬럼추가)**

|  |
| --- |
| ALTER TABLE 테이블명 ADD 컬럼이름 **year(4)**; |

**06.3.4 TIME**

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

time 형은 시간을 표시합니다. 시간은 콜론기호를 구분으로 시간:분:초 로 표시합니다. 최대 표시 가능한 시간은 -838:59:59 ~ 838:59:59 까지 입니다.

**테이블 생성시)**

|  |
| --- |
| CREATE TABLE `tableName` (  컬럼명 **time**,  ….  } |

**컬럼추가)**

|  |
| --- |
| ALTER TABLE 테이블명 ADD 컬럼이름 **time**; |

**06.3.5 TIMESTAMP**

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

TIMESTAMP는 년, 월, 일,시, 분, 초 값 을 숫자값으로 변환하여 저장을 합니다. Timestamp는 항상 UTC 타입존을 기준으로 저장이 됩니다.

**테이블 생성시)**

|  |
| --- |
| CREATE TABLE `tableName` (  컬럼명 **timestamp**,  ….  } |

**컬럼추가)**

|  |
| --- |
| ALTER TABLE 테이블명 ADD 컬럼이름 **timestamp**; |