

11.1.1.숫자 타입 개요

이 섹션에서는 숫자 데이터 타입에 대해서 간략히 설명을 할 예정이며, 보다 자세한 정보를 얻고자 한다면 Section 11.2, “숫자 타입”을 참조하기 바란다. 스토리지 요구 사항에 대해서는 Section 11.5, “데이터 타입 스토리지 요구 사항”을 참조하기 바란다.

M은 출력할 수 있는 최대 폭을 나타낸다. 최대 유효 출력 폭은 255이다. 출력 폭은 스토리지의 크기 또는 하나의 타입이 가질 수 있는 값의 범위와는 관련이 없다.

만일 여러분이 숫자 컬럼에 대해서 ZEROFILL을 지정하면, MySQL은 자동으로 UNSIGNED 속성을 컬럼에 추가한다.

SERIAL은 BIGINT UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE의 별칭이다.

정수 컬럼의 정의에서의 SERIAL DEFAULT VALUE는 NOT NULL AUTO_INCREMENT UNIQUE의 별칭이다.

Warning: 정수간의 뺄셈을 할 때 이들 정수 중의 하나가 UNSIGNED로 되어 있는 경우에는, 결과 값은 NO_UNSIGNED_SUBTRACTION SQL 모드가 활성화 되어 있지 않는 한 부호가 표시되지 않게 된다. Section 12.8, “Cast Functions and Operators”를 참조.

- BIT[(M)]

비트-필드 타입. M은 값 당 비트 수를 나타내며, 1에서 64 사이의 값을 갖는다. M이 생략되면 디폴트는 1이 된다.

이 데이터 타입은 MyISAM에 대해서는 5.0.3에서 추가 되었고, MEMORY, InnoDB, 그리고 BDB에 대한 확장으로 5.0.5에서 추가 되었다. 5.0.3 이전에는, BIT는 TINYINT(1)와 동의어 이다.

- TINYINT[(M)] [UNSIGNED] [ZEROFILL]

매우 작은 정수. 부호 범위는 -128에서 127까지 이다. 부호가 없다면 범위는 0에서 255까지 이다.

- BOOL, BOOLEAN

이 타입들은 TINYINT(1)과 동의어이다. 0의 값은 거짓(false)로 간주된다. 0이 아닌 값은 진실(true)로 처리된다.

나중에 출시될 버전에서는, 전체 불리언 타입에 대해서는 표준 SQL을 준수해서 처리되도록 할 예정이다.

- SMALLINT[(M)] [UNSIGNED] [ZEROFILL]

작은 정수. 부호가 있는 범위는 -32768에서 32767까지이다. 부호가 없다면 0에서 65535까지임.

- MEDIUMINT[(M)] [UNSIGNED] [ZEROFILL]

중간 크기의 정수. 부호가 있으면 -8388608에서 8388607까지이고, 부호가 없다면 0에서 16777215까지임.

- INT[(M)] [UNSIGNED] [ZEROFILL]

일반적인 크기의 정수. 부호 범위는 -2147483648에서 2147483647까지이고, 없으면 0에서 4294967295까지임.

- INTEGER[(M)] [UNSIGNED] [ZEROFILL]

이 타입은 INT과 동의어임.

- BIGINT[(M)] [UNSIGNED] [ZEROFILL]

큰 크기의 정수. 부호 범위는 -9223372036854775808에서 9223372036854775807까지이고, 없으면 0에서 18446744073709551615까지임.

BIGINT 컬럼에 관련해서는 몇 가지를 유의해야 한다:

- 모든 계산(arithmetic)은 부호화 된 BIGINT 또는 DOUBLE 값을 사용해서 해야 하며, 따라서 비트 함수를 사용할 때를 제외하고는 9223372036854775807 (63 bits) 보다 큰 부호가 없는 정수를 사용할 수는 없다! 만일 이렇게 하지 않으면, BIGINT 값을 DOUBLE로 변환할 때 절사(rounding) 에러가 나오기 때문에 결과가 틀려질 수도 있게 된다.

MySQL에서는 아래와 같은 경우에 BIGINT를 사용할 수 있다:

- 정수를 사용해서 부호가 없는 큰 정수를 BIGINT 컬럼 안에 저장할 때.
- MIN(col_name) 또는 MAX(col_name)에서, col_name이 BIGINT 컬럼을 참

조할 때.

- 양 쪽의 피 연산자가 모두 정수인 경우에 연산자를 사용할 때(+, -, *, 등등).

- 스트링을 사용해서 BIGINT 컬럼 안에 정확한 정수 값을 저장할 수 있다. 이럴 경우, MySQL은 스트링-숫자 변환을 실행한다.
- -, +, 그리고 * 연산자는 양 쪽의 피 연산자가 모두 정수 값일 경우에 BIGINT 연산을 사용한다. 이것은, 만일 여러분이 두 개의 큰 정수 값을 곱할 경우 (또는 정수를 리턴하는 함수의 결과), 9223372036854775807 보다 큰 결과가 나올 때 예상치 못한 결과를 가질 수도 있음을 의미한다

- FLOAT[(M,D)] [UNSIGNED] [ZEROFILL]

작은(단일 정밀도 - single-precision) 부동 소수점 수. 사용 가능한 값은 -3.402823466E+38 에서 -1.175494351E-38, 0 이고, 그리고 1.175494351E-38 에서 3.402823466E+38 까지다. 이 범위는 이론적인 한계이고, IEEE 표준에 근거를 하고 있다. 실제로는 여러분이 사용하는 H/W 또는 OS에 근거하여 약간 작게 된다.

M은 전체 자릿 수이고 D는 소수점 뒷자리를 가리킨다. 만일 M 과 D가 생략되면, H/W가 허용하는 범위로 값을 저장한다. 단일 정밀도 부동 소수점 숫자는 대략 7 개의 십진 자리수를 갖는다.

UNSIGNED를 지정 하면, 음수 값을 허용하지 않는다.

FLOAT를 사용하면 예상하지 못한 결과가 나올 수도 있는데, 그 이유는 MySQL에서의 모든 계산은 이중 정밀도(double-precision)로 실행되기 때문이다. Section A.5.7, "Solving Problems with No Matching Rows"를 참조.

- DOUBLE[(M,D)] [UNSIGNED] [ZEROFILL]

일반 크기 (이중 정밀도)의 부동 소수점 숫자. 사용 가능한 값은 -1.7976931348623157E+308 에서 -2.2250738585072014E-308, 0, 그리고 2.2250738585072014E-308 에서 1.7976931348623157E+308 까지다. 이 범위는 이론적인 한계이고, IEEE 표준에 근거를 하고 있다. 실제로는 여러분이 사용하는 H/W 또는 OS에 근거하여 약간 작게 된다.

M은 전체 자릿 수이고 D는 소수점 뒷자리를 가리킨다. 만일 M 과 D가 생략되면, H/W가 허용하는 범위로 값을 저장한다. 이중 정밀도 부동 소수점 숫자는 대략 15 개의 십진 자리수를 갖는다.

UNSIGNED를 지정 하면, 음수 값을 허용하지 않는다.

- DOUBLE PRECISION[(M,D)] [UNSIGNED] [ZEROFILL], REAL[(M,D)] [UNSIGNED] [ZEROFILL]

이 타입은 DOUBLE과 동의어다. 예외 사항: 만일 REAL_AS_FLOAT SQL 모드를 활성화 하면, REAL은 DOUBLE이 아니라, FLOAT에 대한 동의어가 된다.

- FLOAT(p) [UNSIGNED] [ZEROFILL]

부동 소수점 숫자. p 는 비트 정밀도를 가리키지만, MySQL은 결과 데이터 타입으로 FLOAT 또는 DOUBLE을 사용할 지를 결정할 때에만 이 값을 사용한다. 만일 p 가 0 에서 24 사이라면, 데이터 타입은 아무런 M 또는 D 값을 사용하지 않는 FLOAT가 된다. 만일 p 가 25 에서 53 사이의 값이라면, 데이터 타입은 아무런 M 또는 D 값을 사용하지 않는 DOUBLE이 된다. 결과 컬럼의 범위는 단일 정밀도 FLOAT 또는 이중 정밀도 DOUBLE 데이터 타입에 대한 것과 동일하다.

FLOAT(p) 선택스는 ODBC와의 호환성을 위해 제공된 것이다.

- DECIMAL[(M[,D])] [UNSIGNED] [ZEROFILL]

MySQL 5.0.3 및 이후 버전에만 해당:

묶음(packed) 고정 소수점 숫자. M 은 전체 자릿수를 가리키고 (정밀도) D 는 소수점 뒷자리수를 나타낸다 (Scale). 소수점 및 음수 부호 '-'는 M 에 포함되지 않는다. 만일 D 가 0 이라면, 값은 소수점 또는 분수를 가지지 않는다. The maximum number of for DECIMAL 에 대한 최대 (M) 자리수는 65 (5.0.3 에서 5.0.5 까지는 64)다. (D)의 최대 지원 숫자는 30 이다. 만일 D 가 생략되면, 디폴트는 0 이다. 만일 M 가 생략되면, 디폴트는 10 이다.

UNSIGNED를 지정하면, 음수 값을 사용할 수 없다.

DECIMAL 컬럼을 사용하는 모든 기본 계산(+, -, *, /)는 65 자리수를 가지고 계산된다.

MySQL 5.0.3 이전:

묶이지 않은(unpacked) 고정 소수점 숫자. CHAR 컬럼처럼 동작 함; “묶이지 않은(unpacked)”의 의미는, 숫자가 각각의 자리값 별로 하나의 문자를 사용해서 스트링으로 저장되는 것을 의미한다. M 은 전체 자릿수이며 D 는 소수점 이하 자릿수를 나타낸다. 소수점 및 음수 부호 '-'는 M 에 포함되지 않는다. 만일 D 가 0 이면, 소수점을 없는 것이다. DECIMAL 값의 최대 범위는 DOUBLE과 동일하지만, 주어진

DECIMAL 컬럼의 실제 범위는 M 과 D 에 의해 제한된다. 만일 D 가 생략되면, 디폴트는 0 이 된다. 만일 M 이 생략되면, 디폴트는 10이 된다.

UNSIGNED가 지정되면, 음수 값을 사용할 수 없다.

테이블에 있는 DECIMAL 컬럼에 대한 서버의 동작은 테이블을 생성한 MySQL 버전에 의존한다. 만일 서버가 MySQL 5.0.3 또는 그 이상 버전이기는 하지만, 테이블에 있는 DECIMAL 컬럼이 5.0.3 이전에 생성된 것이라면, 이전 버전의 특징이 이 컬럼에 적용된다. 테이블을 새로운 버전의 DECIMAL 포맷으로 변환 시키고자 한다면, mysqldump를 사용해서 덤프를 하고 이것을 다시 읽어 온다.

- DEC[(M[,D])] [UNSIGNED] [ZEROFILL], NUMERIC[(M[,D])] [UNSIGNED] [ZEROFILL], FIXED[(M[,D])] [UNSIGNED] [ZEROFILL]

위의 타입들은 DECIMAL과 동의어다. FIXED 동의어는 다른 데이터 베이스 시스템과의 호환을 위해서 사용하는 것이다.