1 存储引擎

MyISAM 支持表锁，但不支持事务

.frm 表结构定义文件

.MYD 表数据文件

.MYI 索引

InnoDB 支持事务，行锁，每表一个表空间

.frm 表结构文件

.ibd 表空间文件，存入了数据和索引

**每个表都可以定义一个存储引擎，表级别。**

**SHOW ENGINES;** 显示存储引擎

**SHOW TABLE STATUS;** 显示表的属性（包含引擎信息）

2 MySQL读取配置文件的顺序

/etc/my.cnf 🡪 /etc/mysql/my.cnf🡪 $MYSQL\_HOME/my.cnf🡪 default-extra-file=/path🡪 ~/my.cnf

3 数据类型的意义

1 定义存入值

2 定义占据的存储空间

3 定义定长还是变长

4 定义如何比较以及如何排序

5 定义是否可以索引

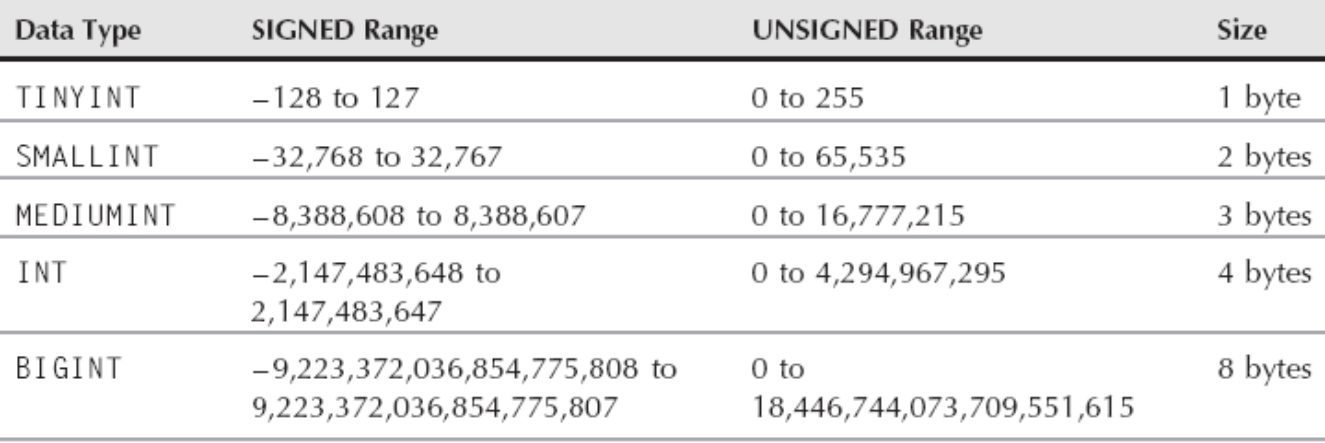
4 数据类型

1 数值型

INT 分为很多种，详细请看下个图片。

FLOAT 4个字节

DOUBLE 8个字节



2 字符型（D代表定长，Q代表区分大小写，N代表NOT）

CHAR D NQ 255

VARCHAR ND NQ 65535

BINARY D Q

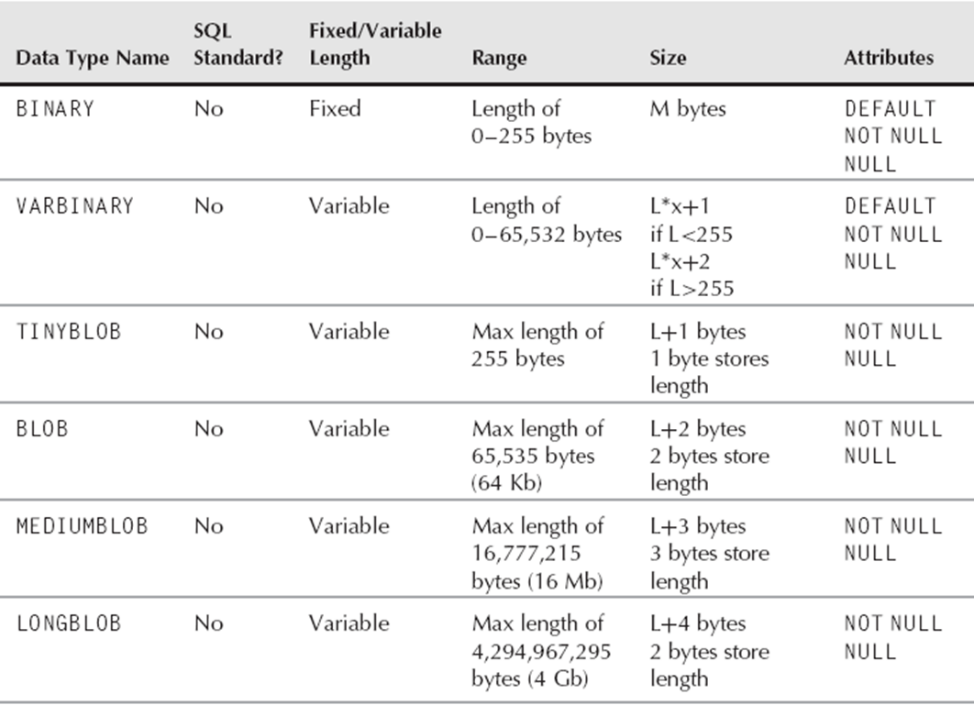
VARBINARY ND Q

**BLOG(二进制大对象，某个字段中本应该存入数值，但却存入了一个指针，指针指向某个文件，所以导致该字段无法创建索引)** Q

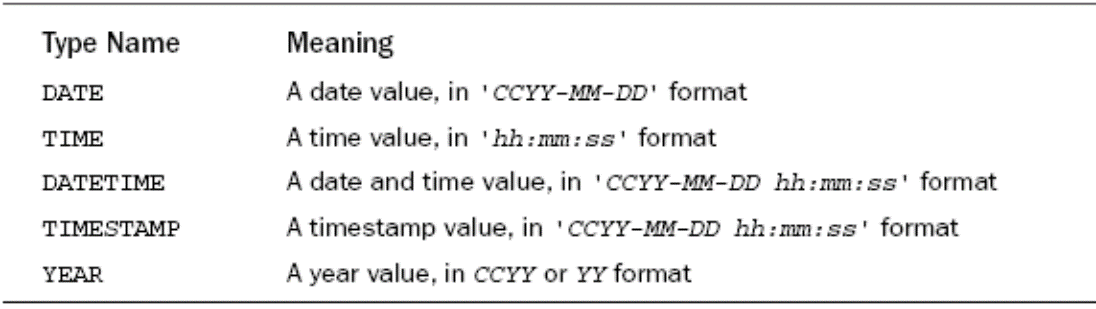
TEXT NQ

ENUM(枚举，例：只能输入是或者否)

SET （集合，可以输入是，否，是否，否是）



3 日期时间型



5 数据标识（修饰）符（character string type attributes）

**NOT NULL**: 不能为空

**NULL**：可以为空，默认情况下是NULL

**UNSIGNED**：无正负号 **仅数值型数据支持**

**DEFALLT**：该字段默认为

**COLLATION**：排序规则 **仅字符型数据支持**

**CHARACTER SET**：数据库所支持的字符集 **仅字符型数据支持**

**SHOW COLLATION ：显示排序规则**

**SHOW CHARACTER SET ：显示服务器端所支持的字符集**

**AUTO\_INCREMENT** 自动增长

except for BIT and DECIMAL

**the field is converted to use the NOT NULL attribute，必须非空**

**can only be one AUTO\_INCREMENT field per table**

By default, the sequence starts at 1 and increases by 1 for every insert. To change the number the sequence starts at, set the auto\_increment\_offset variable in the configuration file or set the dynamic session or global variable of the same name

6 SQL模型，The **sql\_mode** server variable can be set to change the default behavior of data validation and allowed SQL syntax，**它是一个服务器变量**。可以设置全局（global）也可以设置会话（per session）

**ANSI QUOTES** ：双引号和反引号作用相同，字符串只能被单引号引用

**IGNORF\_SPACE**：忽略多余的空白符

**STRICT\_ALL\_TABLES**：STRICT\_ALL\_TABLES—When this sql\_mode is not set, invalid data values are allowed, usually converted to valid data by using zero values or truncating the value. When this sql\_mode is set, invalid data values are rejected, and an error is thrown.

**STRICT\_TRANS\_TABLES**：invalid data values in a transactional table are rejected, and an error is thrown

**TRANITIONAL**：Syntax and behavior closely match traditional expectations

7 MySQL变量

1 服务器变量

按作用域可分为

全局变量（global）：和用户无关，全局有效

会话变量（per session）：和用户关联，只作用于当前用户。

按生效时间分两类

动态：立即生效

若调整变量为全局，则只对新建会话生效

若调整对象为会话，则只对当前会话生效。

静态：非立即生效

1 改变配置文件

2 通过参数传递给MySQL

查看全局变量：

1 **SHOW GLOBAL VARIABLES LIKE ’sql\_mode’;**

2 **SELECT @@global.sql\_mode;**

3 **SELECT Variable\_name, Variable\_value FROM INFORMATION\_SCHEMA.GLOBAL\_VARIABLES WHERE Variable\_name=’sql\_mode’;**

查看会话变量：

1 **SHOW [ SESSION ] VARIABLES LIKE ’sql\_mode’;**

2 **SELECT @@session.sql\_mode;**

3 **SELECT Variable\_name, Variable\_value FROM INFORMATION\_SCHEMA.SESSION\_VARIABLES WHERE Variable\_name=’sql\_mode’;**

设置服务器变量

**1 SET sql\_mode = 'TRADITIONAL';**

**2 SET GLOBAL sql\_mode = 'TRADITIONAL';**

**3 SELECT @@SESSION.sql\_mode;**