

显示、创建、删除表

4.3.1 显示表

【例 4-6】显示当前数据库内的表，在 MySQL 命令行模式下输入 “SHOW TABLES;” 命令。



图 4-8 显示当前数据库中的表

在 MySQL 中，可以使用 DESCRIBE 或 SHOW CREATE TABLE 命令来查看数据表的结构。DESCRIBE/DESC 语句会以表格的形式来展示表的字段信息，包括字段名、字段数据类型、是否为空、是否为主键、是否有默认值等，语法格式如下：

DESCRIBE <表名>;

或简写成：

DESC <表名>;

【例 4-7】以表格形式显示 Lib 数据库中 book 表的结构。在 MySQL 命令行窗口中执行如下 SQL 语句：

DESC book;

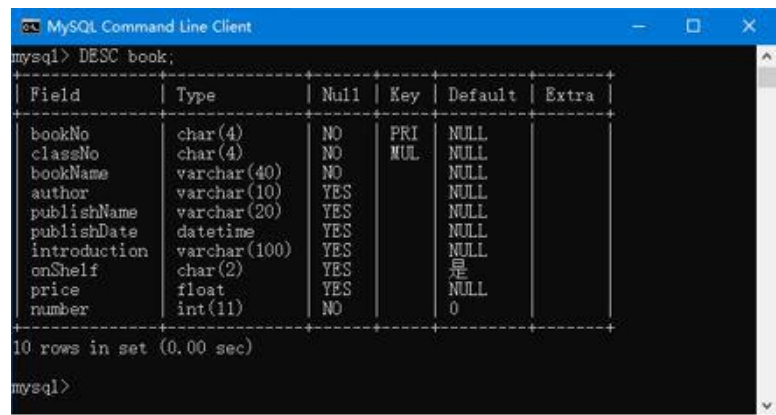


图 4-9 显示表结构

SHOW CREATE TABLE 命令会以 SQL 语句的形式来展示表信息。和 DESCRIBE 相比，SHOW CREATE TABLE 展示的内容更加丰富，它可以查看表的存储引擎和字符编码；另外，在 SHOW CREATE TABLE 语句的结尾处（分号前面）添加 \g 或者 \G 参数可以改变展示形式。

SHOW CREATE TABLE 的语法格式如下：

SHOW CREATE TABLE <表名>;

【例 4-8】以 SQL 语句形式显示 Lib 数据库中 book 表的结构。

在 MySQL 命令行窗口中，输入“SHOW CREATE TABLE book;”命令，则会显示一个两列的表格，第一列是表的名字，第二列是创建该表的“CREATE TABLE”的 SQL 语句。由于 SQL 语句比较长，所以在窗口中有换行的现象。而且还在末尾显示了表的存储引擎和字符编码分别是“InnoDB”和“utf8”。

```
-----+
| book | CREATE TABLE `book` (
| `bookNo` char(4) NOT NULL,
| `classNo` char(4) NOT NULL,
| `bookName` varchar(40) NOT NULL,
| `author` varchar(10) DEFAULT NULL,
| `publishName` varchar(20) DEFAULT NULL,
| `publishDate` datetime DEFAULT NULL,
| `introduction` varchar(100) DEFAULT NULL,
| `onShelf` char(2) DEFAULT '是',
| `price` float DEFAULT NULL,
| `number` int(11) NOT NULL DEFAULT '0',
| PRIMARY KEY (`bookNo`),
| KEY `FK_Book_BookClass` (`classNo`),
| CONSTRAINT `FK_Book_BookClass` FOREIGN KEY (`classNo`) REFERENCES `bookcla
| ss` (`classNo`)
| ) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
-----+
```

图 4-9 显示建表结构

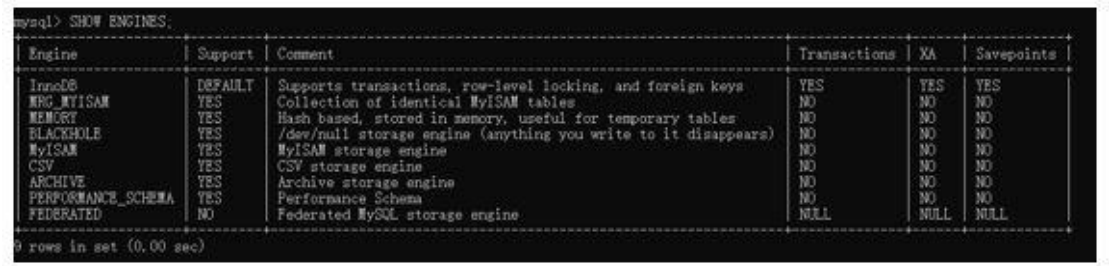
4.3.2 MySQL 存储引擎

数据库存储引擎是数据库底层软件组件，数据库管理系统使用数据引擎进行创建、查询、更新和删除数据操作。简而言之，存储引擎就是指表的类型。数据库的存储引擎决定了表在计算机中的存储方式。不同的存储引擎提供不同的存储机制、索引技巧、锁定水平等功能，使用不同的存储引擎还可以获得特定的功能。

现在许多数据库管理系统都支持多种不同的存储引擎。MySQL 的核心就是存储引擎。

MySQL 提供了多个不同的存储引擎，包括处理事务安全表的引擎和处理非事务安全表的引擎。在 MySQL 中，不需要在整个服务器中使用同一种存储引擎，针对具体的要求，可以对每一个表使用不同的存储引擎。

MySQL 5.7 支持的存储引擎有 InnoDB、MyISAM、Memory、Merge、Archive、CSV、BLACKHOLE 等。可以使用“SHOW ENGINES;”语句查看系统所支持的引擎类型，结果如图所示。



Engine	Support	Comment	Transactions	XA	Savepoints
InnoDB	DEFAULT	Supports transactions, row-level locking, and foreign keys	YES	YES	YES
MRG_MYISAM	YES	Collection of identical MyISAM tables	NO	NO	NO
MEMORY	YES	Hash based, stored in memory, useful for temporary tables	NO	NO	NO
BLACKHOLE	YES	/dev/null storage engine (anything you write to it disappears)	NO	NO	NO
MyISAM	YES	MyISAM storage engine	NO	NO	NO
CSV	YES	CSV storage engine	NO	NO	NO
ARCHIVE	YES	Archive storage engine	NO	NO	NO
PERFORMANCE_SCHEMA	YES	Performance Schema	NO	NO	NO
FEDERATED	NO	Federated MySQL storage engine	NULL	NULL	NULL

9 rows in set (0.00 sec)

图 4-9 显示当前系统支持的存储引擎

Support 列的值表示某种引擎是否能使用，YES 表示可以使用，NO 表示不能使用，DEFAULT 表示该引擎为当前默认的存储引擎。

表 4-10 存储引擎说明

存储引擎	描述
------	----

ARCHIVE	用于数据存档的引擎，数据被插入后就不能在修改了，且不支持索引。
CSV	在存储数据时，会以逗号作为数据项之间的分隔符。
BLACKHOLE	会丢弃写操作，该操作会返回空内容。
FEDERATED	将数据存储于远程数据库中，用来访问远程表的存储引擎。
InnoDB	具备外键支持功能的事务处理引擎
MEMORY	置于内存的表
MERGE	用来管理由多个 MyISAM 表构成的表集合
MyISAM	主要的非事务处理存储引擎
NDB	MySQL 集群专用存储引擎

4.3.3 创建表

在创建数据库之后，接下来就要在数据库中创建数据表。所谓创建数据表，指的是在已经创建的数据库中建立新表。数据表属于数据库，在创建数据表之前，应使用语句“USE<数据库>”指定操作在哪个数据库中进行，如果没有选择数据库，就会抛出“No database selected”的错误。

创建数据表的过程是规定数据列的属性的过程，同时也是实施数据完整性（包括实体完整性、引用完整性和域完整性）约束的过程。接下来我们介绍一下创建数据表的语法形式。

```
CREATE TABLE [IF NOT EXISTS]表名 (
    列名 1 数据类型 1 [完整性约束条件 1],
    列名 2 数据类型 2 [完整性约束条件 2],
    .....
    列名 n 数据类型 n
)[表引擎 表字符集];
```

【例 4-9】在 Lib 数据库中创建 book_bk 表，表的结构同 book 表。为了下面的知识点的介绍，给该创建表的命令加了“行号”。

在 MySQL 的命令行格式中，输入以下命令：

行号	命令内容
1	USE Lib;
2	CREATE TABLE book_bk (
3	bookNo char(4) NOT NULL,
4	classNo char(4) NOT NULL,
5	bookName varchar(40) NOT NULL,
6	author varchar(10) DEFAULT NULL,
7	publishName varchar(20) DEFAULT NULL,
8	publishDate datetime DEFAULT NULL,
9	introduction varchar(100) DEFAULT NULL,
10	onShelf char(2) DEFAULT '是',
11	price float DEFAULT NULL,
12	number int(11) NOT NULL DEFAULT '0',
13	PRIMARY KEY (bookNo),
14	KEY FK_Book_BookClass (classNo),
15	CONSTRAINT FK_Book_BookClass FOREIGN KEY (classNo) REFERENCES bookclass (classNo)

```
16 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

第 1 行是选择数据库 Lib；第 2 行是创建表的命令；第 3~12 行是对表内的各个列进行定义；第 13 行是定义表的主键；第 14~15 行是定义表的外键；第 16 行是定义表的存储引擎和字符集。

上述命令还可以这样写：

行号	命令内容
1	USE Lib;
2	CREATE TABLE book_bk (
3	bookNo char(4) NOT NULL PRIMARY KEY,
4	classNo char(4) NOT NULL,
5	bookName varchar(40) NOT NULL,
6	author varchar(10) DEFAULT NULL,
7	publishName varchar(20) DEFAULT NULL,
8	publishDate datetime DEFAULT NULL,
9	introduction varchar(100) DEFAULT NULL,
10	onShelf char(2) DEFAULT '是',
11	price float DEFAULT NULL,
12	number int(11) NOT NULL DEFAULT '0',
13	FOREIGN KEY(classNo) REFERENCES bookclass(classNo)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

请注意，在第 3 行，因为是单主键表，所以在声明 bookNo 列的时候直接定义主键了。如果一个表里的主键由两个以上的列构成联合主键，则应该用本例第一种方法；第 13 行是外键定义时一种省略的写法，单外键时可以这样定义，classNo 列是参照 bookClass 表的 classNo 列，没有外键名。如果一个表里有两个以上的列是参照其他表的主键而形成的外键，则应该用本例第一种方法，这样每一个外键就有外键名，方便以后进行维护操作。

4.3.4 复制表

复制表是指使用 CREATE TABLE 命令复制一个已经存在的表，其命令格式如下：

```
CREATE TABLE [IF NOT EXISTS] 新表名  
[ LIKE 参照表名 ] | [ AS (select 语句) ]
```

使用 LIKE 关键字创建一个与已有的表相同结构的新表，列名、数据类型、空指定和索引也将复制，但是表的内容不会复制，因此创建的新表是一个空表。

【例 4-10】复制一个 book 表，表名为 book_bk1，在命令行窗口中输入如下命令。

```
CREATE TABLE book_bk1 LIKE book;
```

使用 AS 关键字可以复制表的内容，但索引和完整性约束是不会复制的。

【例 4-10】复制一个 book 表，包括表内的数据，表名为 book_bk2，在命令行窗口中输入如下命令。

```
CREATE TABLE book_bk2 AS SELECT * FROM book;
```

可以分别用“SELECT * FROM book_bk1;”或“SELECT * FROM book_bk2;”进行查验。

4.3.5 删除表

在 MySQL 数据库中，对于不再需要的数据表，可以将其从数据库中删除。在删除表的同时，表的结构和表中所有的数据都会被删除，因此在删除数据表之前最好先备份，以免造成

无法挽回的损失。

删除表分两种情况，一是删除没有被关联的普通表；二是删除与其他表关联的表。

第一种情况，直接使用 DROP TABLE 语句可以删除一个或多个数据表，语法格式如下：

```
DROP TABLE [IF EXISTS] 表名 1 [ ,表名 2, 表名 3 ...]
```

对语法格式的说明如下：

- 表名 1, 表名 2, 表名 3 ... 表示要被删除的数据表的名称。DROP TABLE 可以同时删除多个表，只要将表名依次写在后面，相互之间用逗号隔开即可。
- IF EXISTS 用于在删除数据表之前判断该表是否存在。如果不加 IF EXISTS，当数据表不存在时 MySQL 将提示错误，中断 SQL 语句的执行；加上 IF EXISTS 后，当数据表不存在时 SQL 语句可以顺利执行，但是会发出警告（warning）。

两点注意：

- 用户必须拥有执行 DROP TABLE 命令的权限，否则数据表不会被删除。
- 表被删除时，用户在该表上的权限不会自动删除。

【例 4-11】删除刚才创建的表 book_bk，在命令行窗口中输入如下命令。

```
DROP TABLE book_bk;
```

第二种情况，数据表之间经常存在外键关联的情况，这时如果直接删除父表，会破坏数据表的完整性，也会删除失败。

删除父表有以下两种方法：

- 先删除与它关联的子表，再删除父表；但是这样会同时删除两个表中的数据。
- 将关联表的外键约束取消，再删除父表；适用于需要保留子表的数据，只删除父表的情况。

【例 4-12】删除表 bookClass。

由于当前数据库中的 book 表是 bookClass 的子表。如果要删除 bookClass 表，要分情况：

1、不需要保留表的数据，则先删除 book 表，然后再删除 bookClass 表，在命令行窗口中输入如下命令。

```
DROP TABLE book;
```

```
DROP TABLE bookClass;
```

2、需要保留子表的数据，则先删除 book 表中的外键，然后再删除 bookClass 表。请注意，在 MySQL 中删除外键其实是两个操作，一是删除外键，二是删除该外键所建的索引。在命令行窗口中输入如下命令。

```
ALTER TABLE book DROP FOREIGN KEY FK_Book_BookClass;
```

```
ALTER TABLE book DROP INDEX FK_Book_BookClass;
```

```
DROP TABLE bookClass;
```