

## 第8章 数据的家——MySQL的数据目录

### 8.1 数据库和文件系统的关系

我们知道，像 InnoDB、MyISAM 这样的存储引擎都是把表存储在磁盘上，而操作系统又是使用文件系统来管理磁盘，所以用专业一点的话来表述就是：像 InnoDB、MyISAM 这样的存储引擎都是把数据存储在文件系统上。当我们想读取数据的时候，这些存储引擎会从文件系统中把数据读出来返回给我们；当我们想写入数据的时候，这些存储引擎会把这些数据又写回文件系统。本章就是要唠叨一下 InnoDB 和 MyISAM 这两个存储引擎的数据是如何在文件系统中存储的。



本章以 MySQL 5.7.22 为例，因此某些内容在其他的 MySQL 版本中可能会有些出入，请大家注意。

### 8.2 MySQL 数据目录

MySQL 服务器程序在启动时，会到文件系统的某个目录下加载一些数据，之后在运行过程中产生的数据也会存储到这个目录下的某些文件中。这个目录就称为数据目录。本章的内容就要详细唠唠这个目录下具体都有哪些重要的东西。

#### 8.2.1 数据目录和安装目录的区别

我们之前只接触过 MySQL 的安装目录（在安装 MySQL 时可以自己指定），而且前面的章节中已经重点强调过这个安装目录下非常重要的 bin 目录。它里边存储了许多用来控制客户端程序和服务器程序的命令（许多可执行文件，比如 mysql、mysqld、mysqld\_safe 等，有好几十个）。而数据目录是用来存储 MySQL 在运行过程中产生的数据。大家一定要把安装目录与本章要讨论的数据目录区分开！一定要区分开！

#### 8.2.2 如何确定 MySQL 中的数据目录

说了半天，MySQL 到底把数据存到哪个路径下呢？其实数据目录对应着一个系统变量 `datadir`。在使用客户端与服务器建立连接之后，查看这个系统变量的值就知道了：

```
mysql> SHOW VARIABLES LIKE 'datadir';
+-----+-----+
| Variable_name | Value                               |
+-----+-----+
| datadir       | /usr/local/var/mysql/             |
+-----+-----+
1 row in set (0.00 sec)
```

从上述结果可以看出，在我的计算机上，MySQL 的数据目录就是 /usr/local/var/mysql/。大家可以用自己的计算机试试看。

## 8.3 数据目录的结构

MySQL 在运行过程中都会产生哪些数据呢？当然会包含我们创建的数据库、表、视图和触发器等用户数据。除了这些用户数据，为了让程序更好地运行，MySQL 也会创建一些额外的数据。我们接下来细细地品味一下这个数据目录中的内容。

### 8.3.1 数据库在文件系统中的表示

每当使用“CREATE DATABASE 数据库名”语句创建一个数据库的时候，在文件系统中实际发生了什么呢？其实很简单，每个数据库都对应数据目录下的一个子目录，或者说对应一个文件夹。每当我们新建一个数据库时，MySQL 会帮我们做两件事：

- 在数据目录下创建一个与数据库名同名的子目录（或者说是文件夹）；
- 在与该数据库名同名的子目录下创建一个名为 db.opt 的文件。这个文件中包含了该数据库的一些属性，比如该数据库的字符集和比较规则。

下面查看一下在我的计算机上当前有哪些数据库：

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| charset_demo_db   |
| dahaizi           |
| mysql             |
| performance_schema |
| sys               |
| xiaohaizi          |
+-----+
7 rows in set (0.00 sec)
```

可以看到，当前在我的计算机上有 7 个数据库，其中 charset\_demo\_db、dahaizi 和 xiaohaizi 数据库是我们自定义的，其余 4 个数据库是 MySQL 自带的系统数据库。再在我的计算机上看看一下数据目录中的内容：

```
.
├── auto.cnf
└── ca-key.pem
```

```

├─ ca.pem
├─ charset_demo_db
├─ client-cert.pem
├─ client-key.pem
├─ dahaizi
├─ ib_buffer_pool
├─ ib_logfile0
├─ ib_logfile1
├─ ibdata1
├─ ibtmp1
├─ mysql
├─ performance_schema
├─ private_key.pem
├─ public_key.pem
├─ server-cert.pem
├─ server-key.pem
├─ sys
├─ xiaohaizideMacBook-Pro.local.err
├─ xiaohaizideMacBook-Pro.local.pid
└─ xiaohaizi

```

6 directories, 16 files

当然，这个数据目录中的文件和子目录比较多，但是如果仔细看的话可以发现，除了 `information_schema` 这个系统数据库外，其他的数据库在数据目录下都有对应的子目录。这个 `information_schema` 比较特殊，设计 MySQL 的大叔对它的实现进行了特殊对待，没有在数据目录下为其建立相应的子目录。

### 8.3.2 表在文件系统中的表示

我们的数据其实都是以记录的形式插入到表中的。每个表的信息可以分为两种：

- 表结构的定义；
- 表中的数据。

表结构指的是该表的名称是啥、表里面有多少列、每个列的数据类型是啥、有啥约束条件和索引、用的是啥字符集和比较规则等各种信息。这些信息都体现在了我们的建表语句中。为了保存这些信息，InnoDB 和 MyISAM 这两种存储引擎都在数据目录下对应的数据库子目录中创建了一个专门用于描述表结构的文件，文件名是下面这样：

表名.frm

比如，我们在 `dahaizi` 数据库下创建一个名为 `test` 的表：

```

mysql> USE dahaizi;
Database changed

mysql> CREATE TABLE test (
    ->     c1 INT
    -> );
Query OK, 0 rows affected (0.03 sec)

```

则在数据库 `dahaizi` 对应的子目录下就会创建一个名为 `test.frm` 文件，用来描述表结构。值得注



意的是，这个后缀名为 .frm 的文件是以二进制格式存储的，若直接打开会显示乱码。大家还不赶紧在自己的计算机上创建个表试试，看看有没有生成对应的后缀名为 .frm 文件。

描述表结构的文件现在我们知道怎么存储了，那么表中的数据存到什么文件中了呢？在这个问题上，不同的存储引擎就产生了分歧。下面我们分别看一下 InnoDB 和 MyISAM 使用什么文件来保存表中的数据。

### 1. InnoDB 是如何存储表数据的

前面章节重点唠叨过 InnoDB 的一些实现原理，我们应该很熟悉下面这些内容。

- InnoDB 其实是使用页为基本单位来管理存储空间的，默认的页大小为 16KB。
- 对于 InnoDB 存储引擎来说，每个索引都对应着一棵 B+ 树，该 B+ 树的每个节点都是一个数据页。数据页之间没有必要是物理连续的，因为数据页之间有双向链表来维护这些页的顺序。
- InnoDB 的聚簇索引的叶子节点存储了完整的用户记录，也就是所谓的“索引即数据，数据即索引”。

为了更好地管理这些页，设计 InnoDB 的大叔提出了表空间（table space）或者文件空间（file space）的概念。这个表空间是一个抽象的概念，它可以对应文件系统上一个或多个真实文件（不同表空间对应的文件数量可能不同）。每一个表空间可以被划分为很多个页，表数据就存放在某个表空间下的某些页中。设计 InnoDB 的大叔将表空间划分为几种不同的类型，我们逐一细看。

#### （1）系统表空间（system tablespace）

这个系统表空间可以对应文件系统上一个或多个实际的文件。在默认情况下，InnoDB 会在数据目录下创建一个名为 ibdata1（在你的数据目录下找找看有没有）、大小为 12MB 的文件，这个文件就是对应的系统表空间在文件系统上的表示。怎么才 12MB？这么点儿还没插多少数据就用完了。这是因为这个文件是自扩展文件，也就是当不够用的时候它会自己增加文件大小。

当然，如果想让系统表空间对应文件系统上的多个实际文件，或者仅仅觉得原来的 ibdata1 这个文件名难听，那么可以在 MySQL 服务器启动时，配置对应的文件路径以及它们的大小。比如我们像下面这样修改配置文件：

```
[server]
innodb_data_file_path=data1:512M;data2:512M:autoextend
```

这样，在 MySQL 启动之后就会创建 data1 和 data2 这两个各自 512MB 大小的文件作为系统表空间。其中的 autoextend 表明，如果这两个文件不够用，则会自动扩展 data2 文件的大小。

我们也可以不把系统表空间对应的文件路径配置到数据目录下，甚至可以配置到单独的磁盘分区上，这时涉及的启动选项就是 innodb\_data\_file\_path 和 innodb\_data\_home\_dir。具体的配置逻辑挺绕的，这就不多唠叨了。知道通过修改哪个选项可以修改系统表空间对应的文件，然后在有需要的时候去查询官方文档就好了。

需要注意的一点是，在一个 MySQL 服务器中，系统表空间只有一份。从 MySQL 5.5.7 到 MySQL 5.6.5 之间的各个版本中，表中的数据都会被默认存储到这个系统表空间。

## (2) 独立表空间 (file-per-table tablespace)

在 MySQL 5.6.6 以及之后的版本中, InnoDB 不再默认把各个表的数据存储到系统表空间中, 而是为每一个表建立一个独立表空间。也就是说, 我们创建了多少个表, 就有多少个独立表空间。在使用独立表空间来存储表数据时, 会在该表所属数据库对应的子目录下创建一个表示该独立表空间的文件, 其文件名和表名相同, 只不过添加了一个 .ibd 扩展名。所以完整的文件名称长这样:

表名.ibd

假如我们使用独立表空间来存储 dahaizi 数据库下的 test 表, 那么在该表所在数据库对应的 dahaizi 目录下会为 test 表创建下面这两个文件:

- test.frm ;
- test.ibd。

其中, test.ibd 文件用来存储 test 表中的数据。当然也可以自己指定是使用系统表空间还是独立表空间来存储数据, 这个功能由启动选项 innodb\_file\_per\_table 控制。比如, 我们想刻意将表数据都存储到系统表空间, 则可以在启动 MySQL 服务器时这样配置:

```
[server]
innodb_file_per_table=0
```

当 innodb\_file\_per\_table 的值为 0 时, 表示使用系统表空间; 当 innodb\_file\_per\_table 的值为 1 时, 表示使用独立表空间。不过 innodb\_file\_per\_table 选项只对新建的表起作用, 对于已经分配了表空间的表并不起作用。如果想把已经存储到系统表空间中的表转移到独立表空间, 可以使用下面的语法:

```
ALTER TABLE 表名 TABLESPACE [=] innodb_file_per_table;
```

要把已经存储到独立表空间的表转移到系统表空间, 可以使用下面的语法:

```
ALTER TABLE 表名 TABLESPACE [=] innodb_system;
```

其中, 中括号扩起来的等号 = 可有可无。比如, 我们想把 test 表从独立表空间移动到系统表空间, 可以这么写:

```
ALTER TABLE test TABLESPACE innodb_system;
```

## (3) 其他类型的表空间

除了上述两种表空间之外, 还有一些不同类型的表空间, 比如通用表空间 (general tablespace)、undo 表空间 (undo tablespace)、临时表空间 (temporary tablespace) 等, 具体情况就不详细唠叨了, 等用到的时候再提。

## 2. MyISAM 是如何存储表数据的

唠叨完了 InnoDB 的系统表空间和独立表空间, 现在轮到 MyISAM 了。我们知道, 索引和数据在 InnoDB 中是同一回事, 而 MyISAM 中的索引相当于全部都是二级索引, 该存储引擎的数据和索引是分开存放的。所以在文件系统中也是使用不同的文件来存储数据文件和索引



文件。而且与 InnoDB 不同的是, MyISAM 并没有什么表空间一说, 表的数据和索引都存放到对应的数据库子目录下。假如 test 表使用的是 MyISAM 存储引擎, 那么在它所在数据库对应的 dahaizi 目录下会为 test 表创建下面这 3 个文件:

- test.frm ;
- test.MYD ;
- test.MYI。

其中, test.MYD 表示表的数据文件, 也就是插入的用户记录; test.MYI 表示表的索引文件, 我们为表创建的索引都会放到这个文件中。

### 8.3.3 其他的文件

除了上面说的这些用户自己存储的数据以外, 数据目录下还包含了一些确保程序更好运行的额外文件, 主要包括下面几种类型的文件。

- 服务器进程文件: 每运行一个 MySQL 服务器程序, 都意味着启动一个进程。MySQL 服务器会把自己的进程 ID 写入到这个文件中。
- 服务器日志文件: 在服务器运行期间, 会产生各种各样的日志, 比如常规的查询日志、错误日志、二进制日志、redo 日志等。这些日志各有各的用途, 我们之后会重点唠叨一些日志的用途, 现在先了解一下就可以了。
- SSL 和 RSA 证书与密钥文件: 主要是为了客户端和服务端安全通信而创建的一些文件, 现在看不懂可以忽略。

## 8.4 文件系统对数据库的影响

因为 MySQL 的数据都是存储在文件系统中, 因此就不得不受到文件系统的一些制约, 这在数据库和表的命名、表的大小和性能等方面具有明显的体现。

- 数据库名称和表名称不得超过文件系统所允许的最大长度。

每个数据库都对应数据目录的一个子目录, 数据库名称就是这个子目录的名称; 每个表都会在数据目录的子目录下产生一个与表名同名的 .frm 文件。如果使用 InnoDB 的独立表空间或者使用 MyISAM 存储引擎, 还会产生别的与表名同名的文件 (扩展名不一样)。这些目录或文件名的长度都受限于文件系统所允许的长度。

- 特殊字符的问题。

为了避免因为数据库名和表名出现某些特殊字符而造成文件系统不支持的情况, MySQL 会把数据库名和表名中所有除数字和拉丁字母以外的任何字符在文件名中都映射成 @ + 编码值的形式, 并将其作为文件名。比如, 我们创建的表的名称为 'test?', 由于 “?” 不属于数字或者拉丁字母, 因此会被映射成编码值, 所以这个表对应的 .frm 文件的名称就变成了 test@003f.frm。

- 文件长度受文件系统最大长度的限制。

对于 InnoDB 的独立表空间来说, 每个表的数据都会被存储到一个与表名同名的 .ibd 文件中; 对于 MyISAM 存储引擎来说, 数据和索引会分别存放到与表同名的 .MYD 和 .MYI 文件中。

这些文件会随着表中记录的增加而增大，它们的大小受限于文件系统支持的最大文件大小。

## 8.5 MySQL 系统数据库简介

前文提到了 MySQL 的几个系统数据库，这几个数据库包含了 MySQL 服务器运行过程中所需的一些信息以及一些运行状态信息，现在稍微了解一下。

- **mysql**：这个数据库相当重要，它存储了 MySQL 的用户账户和权限信息、一些存储过程和事件的定义信息、一些运行过程中产生的日志信息、一些帮助信息以及时区信息等。
- **information\_schema**：这个数据库保存着 MySQL 服务器维护的所有其他数据库的信息，比如有哪些表、哪些视图、哪些触发器、哪些列、哪些索引等。这些信息并不是真实的用户数据，而是一些描述性信息，有时候也称之为元数据。
- **performance\_schema**：这个数据库主要保存 MySQL 服务器运行过程中的一些状态信息，算是对 MySQL 服务器的一个性能监控。它包含的信息有统计最近执行了哪些语句，在执行过程的每个阶段都花费了多长时间，内存的使用情况等。
- **sys**：这个数据库主要是通过视图的形式把 **information\_schema** 和 **performance\_schema** 结合起来，让开发人员更方便地了解 MySQL 服务器的性能信息。

啥？这4个系统数据库就这样介绍完了？是的，我们这一节的标题中写的就是简介嘛！如果真的要唠叨一下这几个系统库的使用，恐怕又是一本书的篇幅了。这里只是因为介绍数据目录时遇到了它们，为了内容的完整性而向大家提一下，具体如何使用还是要查询相关文档。

## 8.6 总结

像 InnoDB、MyISAM 这样的存储引擎都是把数据存储在文件系统上。

MySQL 服务器程序在启动时会到数据目录中加载数据，运行过程中产生的数据也会被存储到数据目录中。系统变量 **datadir** 表明了数据目录的路径。

每个数据库都对应着数据目录下的一个子目录，该子目录中包含一个名为 **db.opt** 的文件。这个文件包含了该数据库的一些属性，比如该数据库的字符集和比较规则等。

对于 InnoDB 存储引擎来说：

- 如果使用系统表空间存储表中的数据，那么只会在该表所在数据库对应的子目录下创建一个名为“表名.frm”的文件，表中的数据会存储在系统表空间对应的文件中；
- 如果使用独立表空间存储表中的数据，那么会在该表所在数据库对应的子目录下创建一个名为“表名.frm”的文件和一个名为“表名.ibd”的文件，表中的数据会存储这个“表名.ibd”文件中。

对于 MyISAM 存储引擎来说，会在该表所在数据库对应的子目录下创建 3 个文件。

- 表名.frm：表示表的结构文件。
- 表名.MYD：表示表的数据文件。

- 表名 .MYI：表示表的索引文件。

数据目录中除了存储用户数据外，还需要存储一些额外的文件，包括：

- 服务器进程文件；
- 服务器日志文件；
- SSL 和 RSA 证书与密钥文件。

特定的文件系统会对 MySQL 服务器程序的运行产生一些影响，比如：

- 数据库名称和表名称不得超过文件系统所允许的最大长度；
- 特殊字符的问题；
- 文件长度受文件系统最大长度的限制。

为了存储 MySQL 服务器运行过程中所需的信息以及运行状态信息，设计 MySQL 的大叔设计了下面这些系统数据库：

- mysql；
- information\_schema；
- performance\_schema；
- sys。