

第02章_MySQL的数据目录

讲师：尚硅谷-宋红康（江湖人称：康师傅）

官网：<http://www.atguigu.com>

1. MySQL8的主要目录结构

```
[root@atguigu01 ~]# find / -name mysql
```

安装好MySQL 8之后，我们查看如下的目录结构：

1.1 数据库文件的存放路径

MySQL数据库文件的存放路径： `/var/lib/mysql/`

```
mysql> show variables like 'datadir';
+-----+-----+
| Variable_name | Value                |
+-----+-----+
| datadir       | /var/lib/mysql/     |
+-----+-----+
1 row in set (0.04 sec)
```

从结果中可以看出，在我的计算机上MySQL的数据目录就是 `/var/lib/mysql/`。

1.2 相关命令目录

相关命令目录： `/usr/bin`（`mysqladmin`、`mysqlbinlog`、`mysqldump`等命令）和 `/usr/sbin`。

```
[root@atguigu01 sbin]# cd /usr/bin
[root@atguigu01 bin]# find . -name "mysqladmin*"
./mysqladmin
[root@atguigu01 bin]# find . -name "mysqldump*"
./mysqldump
./mysqldumpslow
```

1.3 配置文件目录

配置文件目录： `/usr/share/mysql-8.0`（命令及配置文件），`/etc/mysql`（如`my.cnf`）



```
[root@atguigu01 mysql]# cd /usr/share/mysql-8.0/
[root@atguigu01 mysql-8.0]# ll
总用量 1072
drwxr-xr-x. 2 root root 4096 7月 27 23:00 bulgarian
drwxr-xr-x. 2 root root 4096 7月 27 23:00 charsets
drwxr-xr-x. 2 root root 4096 7月 27 23:00 czech
drwxr-xr-x. 2 root root 4096 7月 27 23:00 danish
-rw-r--r--. 1 root root 25575 4月 23 23:06 dictionary.txt
drwxr-xr-x. 2 root root 4096 7月 27 23:00 dutch
drwxr-xr-x. 2 root root 4096 7月 27 23:00 english
drwxr-xr-x. 2 root root 4096 7月 27 23:00 estonian
drwxr-xr-x. 2 root root 4096 7月 27 23:00 french
drwxr-xr-x. 2 root root 4096 7月 27 23:00 german
drwxr-xr-x. 2 root root 4096 7月 27 23:00 greek
drwxr-xr-x. 2 root root 4096 7月 27 23:00 hungarian
-rw-r--r--. 1 root root 3999 4月 23 23:06 innodb_memcached_config.sql
-rw-r--r--. 1 root root 2216 4月 24 00:28 install_rewriter.sql
drwxr-xr-x. 2 root root 4096 7月 27 23:00 italian
drwxr-xr-x. 2 root root 4096 7月 27 23:00 japanese
drwxr-xr-x. 2 root root 4096 7月 27 23:00 korean
-rw-r--r--. 1 root root 608148 4月 23 23:06 messages_to_clients.txt
-rw-r--r--. 1 root root 339567 4月 23 23:06 messages_to_error_log.txt
-rw-r--r--. 1 root root 1977 4月 24 00:28 mysql-log-rotate
drwxr-xr-x. 2 root root 4096 7月 27 23:00 norwegian
drwxr-xr-x. 2 root root 4096 7月 27 23:00 norwegian-ny
drwxr-xr-x. 2 root root 4096 7月 27 23:00 polish
drwxr-xr-x. 2 root root 4096 7月 27 23:00 portuguese
drwxr-xr-x. 2 root root 4096 7月 27 23:00 romanian
drwxr-xr-x. 2 root root 4096 7月 27 23:00 russian
```

2. 数据库和文件系统的关系

像 InnoDB, MyISAM 这样的存储引擎都是把表存储到磁盘上，操作系统用来管理磁盘的结构被称为文件系统

2.1 查看默认数据库

查看一下在我的计算机上当前有哪些数据库：

```
mysql> SHOW DATABASES;
```

可以看到有4个数据库是属于MySQL自带的系统数据库。

- **mysql**

MySQL 系统自带的核心数据库，它存储了MySQL的用户账户和权限信息，一些存储过程、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**

MySQL 系统自带的数据库，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。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**。在系统数据库 **information_schema** 中提供了一些以 **innodb_sys** 开头的表，用于表示内部系统表。

```
mysql> USE information_schema;
Database changed

mysql> SHOW TABLES LIKE 'innodb_sys%';
+-----+
| Tables_in_information_schema (innodb_sys%) |
```

```

+-----+
| INNODB_SYS_DATAFILES          |
| INNODB_SYS_VIRTUAL            |
| INNODB_SYS_INDEXES            |
| INNODB_SYS_TABLES             |
| INNODB_SYS_FIELDS             |
| INNODB_SYS_TABLESPACES        |
| INNODB_SYS_FOREIGN_COLS       |
| INNODB_SYS_COLUMNS            |
| INNODB_SYS_FOREIGN            |
| INNODB_SYS_TABLESTATS         |
+-----+
10 rows in set (0.00 sec)

```

- **performance_schema**

MySQL 系统自带的数据库，这个数据库里主要保存MySQL服务器运行过程中的一些状态信息，可以用来 **监控 MySQL 服务的各类性能指标**。包括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，内存的使用情况等信息。

- **sys**

MySQL 系统自带的数据库，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，帮助系统管理员和开发人员监控 MySQL 的技术性能。

2.2 数据库在文件系统中的表示

看一下我的计算机上的数据目录下的内容：

```

[root@atguigu01 mysql]# cd /var/lib/mysql
[root@atguigu01 mysql]# ll
总用量 189980
-rw-r-----. 1 mysql mysql      56 7月 28 00:27 auto.cnf
-rw-r-----. 1 mysql mysql    179 7月 28 00:27 binlog.000001
-rw-r-----. 1 mysql mysql    820 7月 28 01:00 binlog.000002
-rw-r-----. 1 mysql mysql    179 7月 29 14:08 binlog.000003
-rw-r-----. 1 mysql mysql    582 7月 29 16:47 binlog.000004
-rw-r-----. 1 mysql mysql    179 7月 29 16:51 binlog.000005
-rw-r-----. 1 mysql mysql    179 7月 29 16:56 binlog.000006
-rw-r-----. 1 mysql mysql    179 7月 29 17:37 binlog.000007
-rw-r-----. 1 mysql mysql  24555 7月 30 00:28 binlog.000008
-rw-r-----. 1 mysql mysql    179 8月  1 11:57 binlog.000009
-rw-r-----. 1 mysql mysql    156 8月  1 23:21 binlog.000010
-rw-r-----. 1 mysql mysql    156 8月  2 09:25 binlog.000011
-rw-r-----. 1 mysql mysql   1469 8月  4 01:40 binlog.000012
-rw-r-----. 1 mysql mysql    156 8月  6 00:24 binlog.000013
-rw-r-----. 1 mysql mysql    179 8月  6 08:43 binlog.000014
-rw-r-----. 1 mysql mysql    156 8月  6 10:56 binlog.000015
-rw-r-----. 1 mysql mysql    240 8月  6 10:56 binlog.index
-rw-----. 1 mysql mysql   1676 7月 28 00:27 ca-key.pem
-rw-r--r--. 1 mysql mysql   1112 7月 28 00:27 ca.pem
-rw-r--r--. 1 mysql mysql   1112 7月 28 00:27 client-cert.pem
-rw-----. 1 mysql mysql   1676 7月 28 00:27 client-key.pem
drwxr-x---. 2 mysql mysql   4096 7月 29 16:34 dbtest
-rw-r-----. 1 mysql mysql  196608 8月  6 10:58 #ib_16384_0.dblwr
-rw-r-----. 1 mysql mysql  8585216 7月 28 00:27 #ib_16384_1.dblwr
-rw-r-----. 1 mysql mysql    3486 8月  6 08:43 ib_buffer_pool
-rw-r-----. 1 mysql mysql 12582912 8月  6 10:56 ibdata1
-rw-r-----. 1 mysql mysql 50331648 8月  6 10:58 ib_logfile0

```

```

-rw-r-----. 1 mysql mysql 50331648 7月 28 00:27 ib_logfile1
-rw-r-----. 1 mysql mysql 12582912 8月 6 10:56 ibtmp1
drwxr-x---. 2 mysql mysql 4096 8月 6 10:56 #innodb_temp
drwxr-x---. 2 mysql mysql 4096 7月 28 00:27 mysql
-rw-r-----. 1 mysql mysql 26214400 8月 6 10:56 mysql.ibd
srwxrwxrwx. 1 mysql mysql 0 8月 6 10:56 mysql.sock
-rw-----. 1 mysql mysql 5 8月 6 10:56 mysql.sock.lock
drwxr-x---. 2 mysql mysql 4096 7月 28 00:27 performance_schema
-rw-----. 1 mysql mysql 1680 7月 28 00:27 private_key.pem
-rw-r--r--. 1 mysql mysql 452 7月 28 00:27 public_key.pem
-rw-r--r--. 1 mysql mysql 1112 7月 28 00:27 server-cert.pem
-rw-----. 1 mysql mysql 1680 7月 28 00:27 server-key.pem
drwxr-x---. 2 mysql mysql 4096 7月 28 00:27 sys
drwxr-x---. 2 mysql mysql 4096 7月 29 23:10 temp
-rw-r-----. 1 mysql mysql 16777216 8月 6 10:58 undo_001
-rw-r-----. 1 mysql mysql 16777216 8月 6 10:58 undo_002

```

这个数据目录下的文件和子目录比较多，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。

以我的 `temp` 数据库为例，在MySQL5.7 中打开：

```

[root@atguigu02 mysql]# cd ./temp
[root@atguigu02 temp]# ll
总用量 1144
-rw-r-----. 1 mysql mysql 8658 8月 18 11:32 countries.frm
-rw-r-----. 1 mysql mysql 114688 8月 18 11:32 countries.ibd
-rw-r-----. 1 mysql mysql 61 8月 18 11:32 db.opt
-rw-r-----. 1 mysql mysql 8716 8月 18 11:32 departments.frm
-rw-r-----. 1 mysql mysql 147456 8月 18 11:32 departments.ibd
-rw-r-----. 1 mysql mysql 3017 8月 18 11:32 emp_details_view.frm
-rw-r-----. 1 mysql mysql 8982 8月 18 11:32 employees.frm
-rw-r-----. 1 mysql mysql 180224 8月 18 11:32 employees.ibd
-rw-r-----. 1 mysql mysql 8660 8月 18 11:32 job_grades.frm
-rw-r-----. 1 mysql mysql 98304 8月 18 11:32 job_grades.ibd
-rw-r-----. 1 mysql mysql 8736 8月 18 11:32 job_history.frm
-rw-r-----. 1 mysql mysql 147456 8月 18 11:32 job_history.ibd
-rw-r-----. 1 mysql mysql 8688 8月 18 11:32 jobs.frm
-rw-r-----. 1 mysql mysql 114688 8月 18 11:32 jobs.ibd
-rw-r-----. 1 mysql mysql 8790 8月 18 11:32 locations.frm
-rw-r-----. 1 mysql mysql 131072 8月 18 11:32 locations.ibd
-rw-r-----. 1 mysql mysql 8614 8月 18 11:32 regions.frm
-rw-r-----. 1 mysql mysql 114688 8月 18 11:32 regions.ibd

```

在MySQL8.0中打开：

```
[root@atguigu01 mysql]# cd ./temp
[root@atguigu01 temp]# ll
总用量 1080
-rw-r-----. 1 mysql mysql 131072 7月 29 23:10 countries.ibd
-rw-r-----. 1 mysql mysql 163840 7月 29 23:10 departments.ibd
-rw-r-----. 1 mysql mysql 196608 7月 29 23:10 employees.ibd
-rw-r-----. 1 mysql mysql 114688 7月 29 23:10 job_grades.ibd
-rw-r-----. 1 mysql mysql 163840 7月 29 23:10 job_history.ibd
-rw-r-----. 1 mysql mysql 131072 7月 29 23:10 jobs.ibd
-rw-r-----. 1 mysql mysql 147456 7月 29 23:10 locations.ibd
-rw-r-----. 1 mysql mysql 131072 7月 29 23:10 regions.ibd
```

2.3 表在文件系统中的表示

2.3.1 InnoDB存储引擎模式

1. 表结构

为了保存表结构，InnoDB 在 数据目录 下对应的数据库子目录下创建了一个专门用于 描述表结构的文件，文件名是这样：

```
表名.frm
```

比方说我们在 atguigu 数据库下创建一个名为 test 的表：

```
mysql> USE atguigu;
Database changed

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

那在数据库 atguigu 对应的子目录下就会创建一个名为 test.frm 的用于描述表结构的文件。frm文件的格式在不同的平台上都是相同的。这个后缀名为.frm是以 二进制格式 存储的，我们直接打开是乱码的。

2. 表中数据和索引

① 系统表空间 (system tablespace)

默认情况下，InnoDB会在数据目录下创建一个名为 ibdata1、大小为 12M 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12M？注意这个文件是 自扩展文件，当不够用的时候它会自己增加文件大小。

当然，如果你想让系统表空间对应文件系统上多个实际文件，或者仅仅觉得原来的 ibdata1 这个文件名难听，那可以在MySQL启动时配置对应的文件路径以及它们的大小，比如我们这样修改一下my.cnf 配置文件：

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

这样在MySQL启动后就会创建这两个512M大小的文件作为系统表空间，其中autoextend表明这两个文件如果不够用会自动扩展data2文件的大小，需要注意的是，在一个MySQL服务器中，系统表空间只有一份，从MySQL5.5.7到MySQL5.6.6之间的各个版本中，我们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MySQL5.6.6以及之后的版本中，InnoDB并不会默认的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。使用 **独立表空间** 来存储表数据的话，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，文件名和表名相同，只不过添加了一个 **.ibd** 的扩展名而已，所以完整的文件名称长这样：

```
表名.ibd
```

比如：我们使用了 **独立表空间** 去存储 **atguigu** 数据库下的 **test** 表的话，那么在该表所在数据库对应的 **atguigu** 目录下会为 **test** 表创建这两个文件：

```
test.frm
test.ibd
```

其中 **test.ibd** 文件就用来存储 **test** 表中的数据和索引。

③ 系统表空间与独立表空间的设置

我们可以自己指定使用 **系统表空间** 还是 **独立表空间** 来存储数据，这个功能由启动参数 **innodb_file_per_table** 控制，比如说我们想刻意将表数据都存储到 **系统表空间** 时，可以在启动MySQL服务器的时候这样配置：

```
[server]
innodb_file_per_table=0 # 0：代表使用系统表空间； 1：代表使用独立表空间
```

默认情况：

```
mysql> show variables like 'innodb_file_per_table';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| innodb_file_per_table | ON    |
+-----+-----+
1 row in set (0.01 sec)
```

④ 其他类型的表空间

随着MySQL的发展，除了上述两种老牌表空间之外，现在还新提出了一些不同类型的表空间，比如通用表空间（general tablespace）、临时表空间（temporary tablespace）等。

2.3.2 MyISAM存储引擎模式

1. 表结构

在存储表结构方面，**MyISAM** 和 **InnoDB** 一样，也是在 **数据目录** 下对应的数据库子目录下创建了一个专门用于描述表结构的文件：

```
表名.frm
```

2. 表中数据和索引

在MyISAM中的索引全部都是 **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，同时表数据都存放在对应的数据库子目录下。假如 **test** 表使用MyISAM存储引擎的话，那么在它所在数据库对应的 **atguigu** 目录下会为 **test** 表创建这三个文件：



test.frm	存储表结构
test.MYD	存储数据 (MYData)
test.MYI	存储索引 (MYIndex)

举例：创建一个 **MyISAM** 表，使用 **ENGINE** 选项显式指定引擎。因为 **InnoDB** 是默认引擎。

```
CREATE TABLE `student_myisam` (  
  `id` bigint NOT NULL AUTO_INCREMENT,  
  `name` varchar(64) DEFAULT NULL,  
  `age` int DEFAULT NULL,  
  `sex` varchar(2) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MYISAM AUTO_INCREMENT=0 DEFAULT CHARSET=utf8mb3;
```

2.4 小结

举例：数据库 **a**，表 **b**。

1、如果表 **b** 采用 **InnoDB**，data/a 中会产生 1 个或者 2 个文件：

- **b.frm**：描述表结构文件，字段长度等
- 如果采用 **系统表空间** 模式的，数据信息和索引信息都存储在 **ibdata1** 中
- 如果采用 **独立表空间** 存储模式，data/a 中还会产生 **b.ibd** 文件（存储数据信息和索引信息）

此外：

① MySQL 5.7 中会在 data/a 的目录下生成 **db.opt** 文件用于保存数据库的相关配置。比如：字符集、比较规则。而 MySQL 8.0 不再提供 db.opt 文件。

② MySQL 8.0 中不再单独提供 b.frm，而是合并到 b.ibd 文件中。

2、如果表 **b** 采用 **MyISAM**，data/a 中会产生 3 个文件：

- MySQL 5.7 中：**b.frm**：描述表结构文件，字段长度等。
MySQL 8.0 中 **b.xxx.sdi**：描述表结构文件，字段长度等
- **b.MYD** (MYData)：数据信息文件，存储数据信息(如果采用独立表存储模式)
- **b.MYI** (MYIndex)：存放索引信息文件

3. 疑问

.frm 在 MySQL 8 中不存在了。那去哪里了呢？

这就需要解析 ibd 文件。Oracle 官方将 frm 文件的信息及更多信息移动到叫做序列化字典信息（Serialized Dictionary Information, SDI），SDI 被写在 ibd 文件内部。MySQL 8.0 属于 Oracle 旗下，同理。

为了从 IBD 文件中提取 SDI 信息，Oracle 提供了一个应用程序 ibd2sdi。

=== ibd2sdi 官方文档 ===

这个工具不需要下载，MySQL 8 自带的有，只要你配好环境变量就能到处用。

(1) 查看表结构

到存储 ibd 文件的目录下，执行下面的命令：

```
ibd2sdi --dump-file=student.txt student.ibd
```

```
[root@bogon atguigu]# ibd2sdi --dump-file=student.txt student.ibd  
[root@bogon atguigu]# ls  
a.ibd student.ibd student.txt test_frm.ibd
```

这样 ibd2sdi 就会把 ***.ibd 里存储的表结构以 json 的格式保存在 student.txt 中。