

# 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

```
1 [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文件，之后在运行过程中产生的数据也都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**。

MySQL把数据都存到哪个路径下呢？其实 **数据目录** 对应这一个系统变量 `datadir`，我们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可以了。

```
1 mysql> show variables like 'datadir';
2 +-----+-----+
3 | Variable_name | Value          |
4 +-----+-----+
5 | datadir       | /var/lib/mysql/ |
6 +-----+-----+
7 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在磁盘上的，操作系统用来管理磁盘的结构被称为 **文件系统**，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表存储在文件系统上的。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。本章学习一下 InnoDB 和 MyISAM 这两个存储引擎的数据如何在文件系统中存储。

### 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：

```
1 | 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。

```

1  mysql> USE information_schema;
2  Database changed
3
4  mysql> SHOW TABLES LIKE 'innodb_sys%';
5  +-----+
6  | Tables_in_information_schema (innodb_sys%) |
7  +-----+
8  | INNODB_SYS_DATAFILES                      |
9  | INNODB_SYS_VIRTUAL                        |
10 | INNODB_SYS_INDEXES                        |
11 | INNODB_SYS_TABLES                         |
12 | INNODB_SYS_FIELDS                         |
13 | INNODB_SYS_TABLESPACES                    |
14 | INNODB_SYS_FOREIGN_COLS                   |
15 | INNODB_SYS_COLUMNS                        |
16 | INNODB_SYS_FOREIGN                        |
17 | INNODB_SYS_TABLESTATS                     |
18 +-----+
19 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

使用 `CREATE DATABASE` 数据库名语句创建一个数据库的时候，在文件系统上实际发生了什么呢?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，每当新建一个数据库时，MySQL会帮我们做这两件事儿:

1. 在 `数据目录` 下创建一个和数据库名同名的子目录。
2. 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(仅限MySQL5.7及之前版本)，这个文件中包含了 `该数据库的各种属性`，比如该数据库的字符集和比较规则。

看一下我的计算机上的数据目录下的内容:

```

1  [root@atguigu01 mysql]# cd /var/lib/mysql
2  [root@atguigu01 mysql]# ll
3  total 106572
4  -rw-r-----. 1 mysql mysql    56 Jun 22 04:35 auto.cnf
5  -rw-r-----. 1 mysql mysql   479 Jun 22 04:46 binlog.000001
6  -rw-r-----. 1 mysql mysql   820 Jun 22 04:46 binlog.000002
7  -rw-r-----. 1 mysql mysql    16 Jun 22 04:38 binlog.index
8  -rw-----. 1 mysql mysql   1676 Jun 22 04:35 ca-key.pem
9  -rw-r--r--. 1 mysql mysql   1108 Jun 22 04:35 ca.pem
10 -rw-r--r--. 1 mysql mysql   1108 Jun 22 04:35 client-cert.pem
11 -rw-----. 1 mysql mysql   1676 Jun 22 04:35 client-key.pem
12 -rw-r-----. 1 mysql mysql 6291456 Jun 22 04:49 #ib_16384_0.dblwr
13 -rw-r-----. 1 mysql mysql 14680064 Jun 22 04:35 #ib_16384_1.dblwr
14 -rw-r-----. 1 mysql mysql    6337 Jun 22 04:35 ib_buffer_pool

```

```

15 -rw-r-----. 1 mysql mysql 12582912 Jun 22 04:46 ibdata1
16 -rw-r-----. 1 mysql mysql 50331648 Jun 22 04:46 ib_logfile0
17 -rw-r-----. 1 mysql mysql 50331648 Jun 22 04:46 ib_logfile1
18 -rw-r-----. 1 mysql mysql 12582912 Jun 22 04:38 ibtmp1
19 drwxr-x---. 2 mysql mysql      4096 Jun 22 04:38 #innodb_redo
20 drwxr-x---. 2 mysql mysql      187 Jun 22 04:38 #innodb_temp
21 drwxr-x---. 2 mysql mysql      143 Jun 22 04:35 mysql
22 -rw-r-----. 1 mysql mysql 29360128 Jun 22 04:46 mysql.ibd
23 srwxrwxrwx. 1 mysql mysql        0 Jun 22 04:38 mysql.sock
24 -rw-----. 1 mysql mysql        6 Jun 22 04:38 mysql.sock.lock
25 -rw-r-----. 1 mysql mysql      124 Jun 22 04:35 mysql_upgrade_history
26 drwxr-x---. 2 mysql mysql     8192 Jun 22 04:35 performance_schema
27 -rw-----. 1 mysql mysql     1680 Jun 22 04:35 private_key.pem
28 -rw-r--r--. 1 mysql mysql      452 Jun 22 04:35 public_key.pem
29 -rw-r--r--. 1 mysql mysql     1108 Jun 22 04:35 server-cert.pem
30 -rw-----. 1 mysql mysql     1680 Jun 22 04:35 server-key.pem
31 drwxr-x---. 2 mysql mysql       28 Jun 22 04:35 sys
32 -rw-r-----. 1 mysql mysql 16777216 Jun 22 04:49 undo_001
33 -rw-r-----. 1 mysql mysql 16777216 Jun 22 04:40 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：

```

1 [root@atguigu02 mysql]# cd ./temp
2 [root@atguigu02 temp]# ll
3 总用量 1144
4 -rw-r-----. 1 mysql mysql   8658 8月 18 11:32 countries.frm
5 -rw-r-----. 1 mysql mysql 114688 8月 18 11:32 countries.ibd
6 -rw-r-----. 1 mysql mysql    61 8月 18 11:32 db.opt
7 -rw-r-----. 1 mysql mysql   8716 8月 18 11:32 departments.frm
8 -rw-r-----. 1 mysql mysql 147456 8月 18 11:32 departments.ibd
9 -rw-r-----. 1 mysql mysql   3017 8月 18 11:32 emp_details_view.frm
10 -rw-r-----. 1 mysql mysql   8982 8月 18 11:32 employees.frm
11 -rw-r-----. 1 mysql mysql 180224 8月 18 11:32 employees.ibd
12 -rw-r-----. 1 mysql mysql   8660 8月 18 11:32 job_grades.frm
13 -rw-r-----. 1 mysql mysql  98304 8月 18 11:32 job_grades.ibd
14 -rw-r-----. 1 mysql mysql   8736 8月 18 11:32 job_history.frm
15 -rw-r-----. 1 mysql mysql 147456 8月 18 11:32 job_history.ibd
16 -rw-r-----. 1 mysql mysql   8688 8月 18 11:32 jobs.frm
17 -rw-r-----. 1 mysql mysql 114688 8月 18 11:32 jobs.ibd
18 -rw-r-----. 1 mysql mysql   8790 8月 18 11:32 locations.frm
19 -rw-r-----. 1 mysql mysql 131072 8月 18 11:32 locations.ibd
20 -rw-r-----. 1 mysql mysql   8614 8月 18 11:32 regions.frm
21 -rw-r-----. 1 mysql mysql 114688 8月 18 11:32 regions.ibd

```

在MySQL8.0中打开：

```

1 [root@atguigu01 mysql]# cd ./temp
2 [root@atguigu01 temp]# ll
3 总用量 1080
4 -rw-r-----. 1 mysql mysql 131072 7月 29 23:10 countries.ibd
5 -rw-r-----. 1 mysql mysql 163840 7月 29 23:10 departments.ibd
6 -rw-r-----. 1 mysql mysql 196608 7月 29 23:10 employees.ibd
7 -rw-r-----. 1 mysql mysql 114688 7月 29 23:10 job_grades.ibd
8 -rw-r-----. 1 mysql mysql 163840 7月 29 23:10 job_history.ibd
9 -rw-r-----. 1 mysql mysql 131072 7月 29 23:10 jobs.ibd
10 -rw-r-----. 1 mysql mysql 147456 7月 29 23:10 locations.ibd
11 -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

我们的数据其实都是以 `记录的形式` 插入到表中的，每个表的信息其实可以分为两种：

1. 表结构的定义
2. 表中的数据

`表结构` 就是该表的名称，表里边有多少列，每个列的数据类型，约束条件和索引，使用的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了。

### 2.3.1InnoDB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：

```
1 表名 .frm
```

比方说我们在 `atguigu` 数据库下创建一个名为 `test` 的表：

```

1 mysql> USE atguigu;
2 Database changed
3
4 mysql> CREATE TABLE test (
5     -> c1 INT
6     -> );
7 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

储备知识：（索引章节会讲到）

- 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，数据页之间不必要是物理连续的，因为数据页之间有 `双向链表` 来维护着这些页的顺序。
- 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。

为了更好的管理这些页，InnoDB提出了一个表空间或者文件空间（英文名：table space 或者 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，我们的表数据就存放在某个表空间下的某些页里。这里表空间有几种不同的类型：

### ①系统表空间（systemtablespace）

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：

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

这样在MySQL启动之后就会创建这两个512M大小的文件作为系统表空间，其中的 autoextend 表明这两个文件如果不够用会自动扩展data2文件的大小。

需要注意的一点是，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：

```
1 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：

```
1 test.frm
2 test.ibd
```

其中 test.ibd文件就用来存储 test表中的数据 and 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：

```
1 [server]
2 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：



```

1 mysql> show variables like 'innodb_file_per_table';
2 +-----+-----+
3 | Variable_name | value |
4 +-----+-----+
5 | innodb_file_per_table | ON |
6 +-----+-----+
7 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（generaltablesapce）、临时表空间（temporarytablesapce）等。

### 3.图解

#### MySQL5.7下数据库的目录结构（InnoDB）

```

mysql> use dbtest1;
Database changed
mysql> show tables;
+-----+
| Tables_in_dbtest1 |
+-----+
| emp1               |
| emp2               |
+-----+
2 rows in set (0.00 sec)

```

查看dbtest1中的表的情况

```

[root@atguigu06 ~]# cd /var/lib/mysql
[root@atguigu06 mysql]# ll
总用量 122952
-rw-r-----. 1 mysql mysql      56 12月  8 21:15 auto.cnf
-rw-r-----. 1 mysql mysql    1680 12月  8 21:15 ca-key.pem
-rw-r--r--. 1 mysql mysql    1112 12月  8 21:15 ca.pem
-rw-r--r--. 1 mysql mysql    1112 12月  8 21:15 client-cert.pem
-rw-r-----. 1 mysql mysql    1680 12月  8 21:15 client-key.pem
drwxr-x---. 2 mysql mysql    4096 12月  9 15:12 dbtest1
drwxr-x---. 2 mysql mysql    4096 12月  9 17:38 dbtest2
-rw-r-----. 1 mysql mysql      406 12月  9 15:03 ib_buffer_pool
-rw-r-----. 1 mysql mysql 12582912 12月  9 17:39 ibdata1
-rw-r-----. 1 mysql mysql 50331648 12月  9 17:39 ib_logfile0
-rw-r-----. 1 mysql mysql 50331648 12月  8 21:15 ib_logfile1
-rw-r-----. 1 mysql mysql 12582912 12月  9 17:38 ibtmp1
drwxr-x---. 2 mysql mysql    4096 12月  8 21:15 mysql
srwxrwxrwx. 1 mysql mysql      0 12月  9 15:03 mysql.sock
-rw-r-----. 1 mysql mysql      5 12月  9 15:03 mysql.sock.lock
drwxr-x---. 2 mysql mysql    4096 12月  8 21:15 performance_schema
-rw-r-----. 1 mysql mysql    1680 12月  8 21:15 private_key.pem
-rw-r--r--. 1 mysql mysql     452 12月  8 21:15 public_key.pem
-rw-r--r--. 1 mysql mysql    1112 12月  8 21:15 server-cert.pem
-rw-r-----. 1 mysql mysql    1676 12月  8 21:15 server-key.pem
drwxr-x---. 2 mysql mysql   12288 12月  8 21:15 sys

[root@atguigu06 mysql]# cd ./dbtest1
[root@atguigu06 dbtest1]# ll
总用量 220
-rw-r-----. 1 mysql mysql      61 12月  9 15:11 db.opt
-rw-r-----. 1 mysql mysql    8588 12月  9 15:12 emp1.frm
-rw-r-----. 1 mysql mysql   98304 12月  9 15:13 emp1.ibd
-rw-r-----. 1 mysql mysql    8588 12月  9 15:08 emp2.frm
-rw-r-----. 1 mysql mysql   98304 12月  9 15:08 emp2.ibd

```

数据库的相关配置：字符集、比较规则

表结构

数据信息和索引信息

## MySQL8.0下数据库的数据目录 (InnoDB)

```
mysql> use dbtest1;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_dbtest1 |
+-----+
| emp1               |
| mytbl2             |
+-----+
2 rows in set (0.00 sec)
```

查询dbtest1的表情况

```
-rw-r-----. 1 mysql mysql      56 12月  8 20:59 auto.cnf
-rw-r-----. 1 mysql mysql     475 12月  9 09:01 binlog.000001
-rw-r-----. 1 mysql mysql    7190 12月  9 18:52 binlog.000002
-rw-r-----. 1 mysql mysql      32 12月  9 09:01 binlog.index
-rw-r-----. 1 mysql mysql    1676 12月  8 20:59 ca-key.pem
-rw-r-----. 1 mysql mysql    1112 12月  8 20:59 ca.pem
-rw-r-----. 1 mysql mysql    1112 12月  8 20:59 client-cert.pem
-rw-r-----. 1 mysql mysql    1676 12月  8 20:59 client-key.pem
drwxr-xr-x.  2 mysql mysql     4096 12月  9 20:39 dbtest1
-rw-r-----. 1 mysql mysql   196608 12月  9 18:52 #ib_16384_0.dblwr
-rw-r-----. 1 mysql mysql   8585216 12月  8 20:59 #ib_16384_1.dblwr
-rw-r-----. 1 mysql mysql     5928 12月  8 20:59 ib_buffer_pool
-rw-r-----. 1 mysql mysql  12582912 12月  9 18:52 ibdata1
-rw-r-----. 1 mysql mysql   50331648 12月  9 18:52 ib_logfile0
-rw-r-----. 1 mysql mysql   50331648 12月  8 20:59 ib_logfile1
-rw-r-----. 1 mysql mysql  12582912 12月  9 09:01 ibtmp1
drwxr-xr-x.  2 mysql mysql     4096 12月  9 09:01 #innodb_temp
drwxr-xr-x.  2 mysql mysql     4096 12月  8 20:59 mysql
-rw-r-----. 1 mysql mysql  25165824 12月  9 18:50 mysql.ibd
srwxrwxrwx.  1 mysql mysql        0 12月  9 09:01 mysql.sock
-rw-r-----. 1 mysql mysql        5 12月  9 09:01 mysql.sock.lock
drwxr-xr-x.  2 mysql mysql     4096 12月  8 20:59 performance_schema
-rw-r-----. 1 mysql mysql    1680 12月  8 20:59 private_key.pem
-rw-r-----. 1 mysql mysql     452 12月  8 20:59 public_key.pem
-rw-r-----. 1 mysql mysql    1112 12月  8 20:59 server-cert.pem
-rw-r-----. 1 mysql mysql    1680 12月  8 20:59 server-key.pem
drwxr-xr-x.  2 mysql mysql     4096 12月  8 20:59 sys
-rw-r-----. 1 mysql mysql  16777216 12月  9 18:52 undo_001
-rw-r-----. 1 mysql mysql  16777216 12月  9 18:52 undo_002
[root@atguigu05 mysql]# cd ./dbtest1
[root@atguigu05 dbtest1]# ll
总用量 160
-rw-r-----. 1 mysql mysql 114688 12月  9 15:38 emp1.ibd
-rw-r-----. 1 mysql mysql 114688 12月  9 18:52 mytbl2.ibd
```

1. 数据库的存储配置被放在具体的表里面  
2. xxx.frm和xxx.ibd合二为一

### 4、疑问

.frm 在MySQL8中不存在了，去哪里了？

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

这个工具不需要下载, MySQL8自带的有, 只需要你配好环境变量就能到处用。

查看表结构: 到存储ibd文件的目录下, 执行下面的命令

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

结果如图所示

```
[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 xxx.ibd 里存储的表结构以 json 的格式保存在 student.txt 中



```

    "ibd2sdi":
    {
      "type":1,
      "id":362,
      "object":{
        "mysql_version_id":80026,
        "dd_version":80023,
        "sdi_version":80019,
        "dd_object_type":"Table",
        "dd_object":{
          "name":"student",
          "mysql_version_id":80026,
          "created":20210726084458,
          "last_altered":20210726084458,
          "hidden":1,
          "options":"avg_row_length=0;checksum=0;delay_key_write=0;encrypt_type=N;key_block_size=0
keys_disabled=0;pack_record=1;stats_auto_recalc=0;stats_sample_pages=0;",
          "columns":{
            {
              "name":"id",
              "type":9,
              "is_nullable":false,
              "is_zerofill":false,
              "is_unsigned":false,
              "is_auto_increment":true,
              "is_virtual":false,
              "hidden":1,
              "ordinal_position":1,
              "char_length":20,
              "numeric_precision":19,
              "numeric_scale":0,
              "numeric_scale_null":false,
              "datetime_precision":0,
              "datetime_precision_null":1,

```

可点击key和value值进行编辑

图中标记部分从上到下分别表示

- 表
- 列
- 列名
- 列的长度

通过上面的测试结果可以发现，MySQL8把之前版本的frm文件合并到ibd文件中了。

## 2.3.2MyISAM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：

1 | 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：

1	test.frm	存储表结构
2	test.MYD	存储数据 (MYData)
3	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。

```

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

```

### 3、图解

#### MySQL中数据库的目录结构（MYISAM）

```

mysql> 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;
Query OK, 0 rows affected (0.01 sec)

mysql> show tables;
+-----+
| Tables_in_dbtest1 |
+-----+
| emp1               |
| emp2               |
| student_myisam     |
+-----+
3 rows in set (0.00 sec)

```

使用MYISAM创建表student\_myisam

```

[root@atguigu06 dbtest1]# ll MySQL5.7
总用量 236
-rw-r-----. 1 mysql mysql 61 12月 9 15:11 db.opt 数据库配置信息
-rw-r-----. 1 mysql mysql 8588 12月 9 15:12 emp1.frm
-rw-r-----. 1 mysql mysql 98304 12月 9 15:13 emp1.ibd
-rw-r-----. 1 mysql mysql 8588 12月 9 15:08 emp2.frm
-rw-r-----. 1 mysql mysql 98304 12月 9 15:08 emp2.ibd
-rw-r-----. 1 mysql mysql 8642 12月 9 21:30 student_myisam.frm 表结构
-rw-r-----. 1 mysql mysql 0 12月 9 21:30 student_myisam.MYD 表数据
-rw-r-----. 1 mysql mysql 1024 12月 9 21:30 student_myisam.MYI 索引

```

1. 在InnoDB中表数据和索引是合在一起的，在MYISAM中，表数据和索引是分开的

```

-rw-r-----. 1 mysql mysql 114688 12月 9 15:38 emp1.ibd
-rw-r-----. 1 root root 11108 12月 9 21:27 emp1.txt
-rw-r-----. 1 mysql mysql 114688 12月 9 18:52 mytbl2.ibd
-rw-r-----. 1 mysql mysql 4330 12月 9 21:30 student_myisam_363.sdi
-rw-r-----. 1 mysql mysql 0 12月 9 21:30 student_myisam.MYD
-rw-r-----. 1 mysql mysql 1024 12月 9 21:30 student_myisam.MYI

```

MySQL8.0

相当于MySQL5.7下的.frm

MySQL8.0中：

进入数据库目录

```

-rw-r-----. 1 mysql mysql 4343 8月 19 20:27 student_myisam1_389.sdi
-rw-r-----. 1 mysql mysql 0 8月 19 20:27 student_myisam1.MYD
-rw-r-----. 1 mysql mysql 1024 8月 19 20:27 student_myisam1.MYI
-rw-r-----. 1 mysql mysql 4342 8月 19 20:29 student_myisam_390.sdi
-rw-r-----. 1 mysql mysql 0 8月 19 20:29 student_myisam.MYD
-rw-r-----. 1 mysql mysql 1024 8月 19 20:29 student_myisam.MYI

```

包含三个文件：

```

1 student_myisam_390.sdi -- 存储元数据
2 student_myisam.MYD     -- 存储数据
3 student_myisam.MYI     -- 存储索引

```

对于InnoDB表，SDI与InnoDB用户表空间中的数据一起存储。对于MyISAM和其他存储引擎，它被写入数据目录中的.sdi文件。

在MySQL8.0中，MyISAM存储引擎不提供分区支持。在以前版本的MySQL中创建的分区MyISAM表不能在MySQL8.0中使用。

在MySQL5.7中:

查看文件目录, 如图所示:

```
-rw-r-----. 1 mysql mysql 114688 8月 18 11:32 regions.ibd
-rw-r-----. 1 mysql mysql 8657 8月 19 20:32 student_myisam.frm
-rw-r-----. 1 mysql mysql 0 8月 19 20:32 student_myisam.MYD
-rw-r-----. 1 mysql mysql 1024 8月 19 20:32 student_myisam.MYI
[root@atquigu02 tmp]#
```

包含三个文件:

```
1 student_myisam.frm --存储表结构
2 student_myisam.MYD --存储数据
3 student_myisam.MYI --存储索引
```

可以发现, 在之前的数据库版本中, MyISAM引擎已存在frm文件, 但是在MySQL8之后也和InnoDB引擎一样去掉了, 放入了sdi文件中。

## 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 MySQL5.7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MySQL8.0不再提供 `db.opt` 文件。

② MySQL8.0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:

- MySQL5.7中: `b.frm`: 描述表结构文件, 字段长度等。
- MySQL8.0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

## 2.5 视图在文件系统中的表示

我们知道MySQL中的 视图 其实是 虚拟的表, 也就是某个查询语句的一个别名而已, 所以在存储视图的时候是不要存储真实的数据的, 只需要把它的结构存储起来就行了。和表一样, 描述视图结构的文件也会被存储到所属数据库对应的子目录下边, 只会存储一个 视图名. `frm` 的文件。如下图中

的: `emp_details_view.frm`

```
[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
```

## 2.6 其他的文件

除了我们上边说的这些用户自己存储的数据以外，**数据目录**下还包括为了更好运行程序的一些额外文件，主要包括这几种类型的文件：

- 服务器进程文件

我们知道每运行一个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写入到一个文件中。

- 服务器日志文件

在服务器运行过程中，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。后面讲解。

- 默认/自动生成的SSL和RSA证书和密钥文件

主要是为了客户端和服务器安全通信而创建的一些文件。