# MYSQL

## 一、基础命令

mysqld --skip-grant-tables 安全模式下启动服务，不需要密码

1.登陆，登出

mysql -u<user> -p<password> [-h<远程ip>] 登陆

exit 登出

2.修改密码

mysqladmin -u<user> -p <old password> -password <new password>

set password for ‘username’@’host’ =password(‘new password’);、

alter user ‘root’@‘localhost’ identified by ‘new password’；

3.建表与授权

create user ‘userName’ @‘host’ identified by ’password’;

host可以指定登陆主机的localhost/ip，任意主机登陆用通配符表示 ‘%’

grant privileges on db\_name.table\_name to ‘username’@’host’;

例: GRANT SELECT, INSERT ON test.user TO 'pig'@'%';

例: GRANT privileges ON databasename.tablename TO 'username'@'host' WITH GRANT OPTION;该命令可以授权用户“授权的权限”

REVOKE privilege ON databasename.tablename FROM 'username'@'host';

DROP USER 'username'@'host';

4.建表建库

create database db\_name ;

create table table\_name(

id int not null primary key auto\_increment,

name varchar(255),

foreign key (name) references tab\_name (col\_name));

drop table tab\_name;直接删表结构。

truncate table tab\_name; 删除所有表数据，删除索引，不可回滚，不触发触发器

delete from tab\_name where condition; 不加where子句，则全部删除，可以回滚。

drop database db\_name;

5.show

show databases；

use db\_name;

show tables;

show columns from tab\_name from db\_name;/ showcolumns from db\_name.tab\_name;表的列信息

show index from tab\_name;显示索引

show grants for user\_name;显示用户的权限

show engines; 显示引擎

6.备份

导出数据库

mysqldump -u <user> -p<password> db\_name>outfilename.sql

mysqldump -u root -proot zj>zj.sql

导出一个表

mysqldump -u <user> -p<password> db\_name tab\_name> outfile\_name.sql

mysqldump -u user\_name -p database\_name table\_name > outfile\_name.sql

导出数据结构

mysqldump -u <user> -p <password> -d db\_name> outfile\_name.sql

-d没有数据

导入

Source path

7.alter

修改表名

alter table rename to old\_tab\_name;

新增字段

alter table tab\_name add col\_name col\_type;

删除字段

alter table tab\_name drop col\_name;

修改字段类型

alter table tab\_name modify col\_name new\_col\_type;

重命名字段（change子句后面可以接建表时字段的任何子句）

Alter table tab\_name change col\_name new\_col\_name new\_col\_type;

修改表名

alter table tab\_name rename to new\_name;

1. root登陆
2. 创建用户zj，密码zj
3. 修改zj密码为zhaojing
4. 授权zj账户权限为 所有库的所有表的所有权限
5. 退出，并且用zj登陆
6. 查看当前用户
7. 查看当前用户的权限
8. 查看所有数据库
9. 创建数据库zj，并且查看
10. 选择进入数据库zj
11. 查看所有表
12. 建表student（id主键自增非空，name）
13. 插入数据2条
14. 建表teacher（id非空主键，name，sid ）
15. 向teacher插入两条数据
16. 增加teacher表主键自增
17. 向teacher插入两条数据
18. 查看表teacher的索引
19. 查增加student分数字段score int
20. 删除teacher字段name
21. 修改student字段score为varchar 使用modify
22. 看student的列信息
23. 备份表（结构和数据）
24. 备份表结构
25. 删除teacher表id为1的数据
26. 删除teacher表所有数据，并且删除索引，腾出空间
27. 删除teacher表
28. 恢复备份
29. 删除库
30. 删除用户

1.mysql -uroot -proot

2. create user 'zj'@'%' identified by 'zj;

3. alter user 'zj'@'%' identified by 'zhaojing';

或set password for ‘zj’@’%’ = password(‘zhaojing’)；

4. grant all privileges on zj.\* to 'zj'@'%';

5.exit; mysql -uzj -pzhaojing

6. select user();

7. show grants for zj;

8.show databases;

9.create database zj;

10.use zj;

11.show tables;

12.create table student(

id int not null primary key auto\_increment,

name varchar(255)

);

13.insert into sudent (name) values(‘zj’),(‘zp’);

14.create table teacher(

id int not null primary key ,

name varchar(255) ,

sid int,

);

15.insert into teacher(id,name,sid) values(1,’yxy’,1),(2,’sy’,2);

16.alter table teacher change id id int primary key auto\_increment;

17.insert into teacher(name,sid) values(‘yx’,1),(‘sx’,2);

18.show index from teacher;

19.alter table student add score int;

20.alter table teacher drop name;

21.alter table student modify score varchar(255);

22.show colmuns from zj.student;

23.mysqldump -uroot -proot zj student >student.sql;

24.mysqldump -uroot -proot -d zj>table\_zj.sql

25.delete from teacher where id=1;

26. truncate table teacher;

27.drop table teacher;

28. source c://user/teacher.sql

29.drop database zj;

30.drop user zj;

## 二、数据库引擎

### 2．1 各引擎的特性

#### 2..1.1 MyISAM

是mysql的默认引擎，不支持事务，不支持外键，访问速度快。

对事务完整性没有要求，或以insert,select为主的的应用都可以使用。

每个MyISAM磁盘上存储3个文件，文件名和表名相同，扩展名分别是

**.frm** （存储表定义

**.MYD** （MYData，存储数据

**MYI**  （MYIndex，存储索引

数据文件和索引文件可以放在不同的目录，平均分布IO，获取更快的速度

创建表时可以，用DATADIRECTORY和INDEXDIRECTORY指定索引位置，必须是绝对路径。

MYISAM表支持三种不同的存储格式，即：

**静态表**（固定长度，使用char(n)的时候，存储快，易缓存，出现故障易恢复，空间占用大

**动态表**（即使用varchar(n)的时候，空间占用少，频繁更新删除产生碎片，需要定期执行OPTIMIZE TABLE 或 myisamchk -r 改善性能。

**压缩表** （myisampack工具创建 ，占用少，访问开支小

#### 2.1.2 InnoDB

提供具有提交、回滚和崩溃恢复的事务安全能力。对比MyISAM，InnoDB写处理效率差一些，且会占用更多磁盘空间以保留数据和索引。

不同于其它引擎的特性：

1、自增长列可手动插入，且定是索引（不插入或插入为0，则，最终值为自增的结果，如果是组合索引，必是组合索引的第一列

2、mysql支持外键的引擎只有InnoDB

创建外键的时候，要求父表有相应的索引。子表创建外键的时候也会相应创建对应的索引。

创建外键时，可以指定在删除、更新父表时，子表的行为。有：

RESTRICT 、NO ACTION 子表有相应数据时，不能删除或更新

CASCADE 相应更新、删除子表的数据

SET NULL 将子表相应字段设置为null

创建了外键，父表的主键和索引不能删除

当导入多个表，需要忽略导入顺序时，可以先关闭外键检查，

关闭：SET\_FOREIGN\_KEY\_CHECKS=0;

打开: SET\_FOREIGN\_KEY\_CHECKS=1;

同理，删除数据时，也可以这样操作。

3、存储方式

使用共享空间存储：表结构保存在.frm文件中，数据和索引保存在

Innodb\_data\_home\_dir和innodb\_data\_file\_path定义的表空间中，可以是多个文件

使用多表空间存储：表结构仍我保存在同.frm文件中，但是每个表的数据和索引单独保存在.ibd文件中，

要使用多表存储方式，需要设置参数innodb\_file\_per\_table,并重新启动服务。

多表空间没有大小限制，方便单表备份、恢复

#### 2.1.3 MEMORY

该引擎只在内存中建表，每个mermory表只实际对应一个磁盘文件，格式是.frm。

数据存于内存并且默认使用hash索引，访问快，但是服务关闭则数据丢失。

给memory表创建索引时，可以指定使用hash索引还是btree索引。

#### 2.1.4 MERGE

该引擎是对多个相同结构表（myisam）的逻辑组合。两个文件.frm和.MGR

.frm 表结构

.MGR 包含组合表的信息，包括 MERGE 表由哪些表组成、插入新的数据时的依据。可以通过修改.MRG 文件来修改 MERGE 表，但是修改后要通过 FLUSH TABLES 刷新。

Merge表并不能智能的插入某个表，只能在创建表的时候，设置insert\_method的值（first,last两个值）来指定插入。

可以对merge表进行drop操作，但是只能删除merge定义，对原表没有影响。

## 三、索引

对相关列使用索引是提高select查询速度的最佳途径。

Myisam和Innodb默认创建的是btree索引。Memory默认是hash索引。

### 3.1索引的原则

1、where子语的字段

2、使用唯一索引

3、利用最左前缀

4、勿过度使用索引

### 3.2 使用索引的情况

以下情况可能会保用索引

1、复合索引中，使用索引最左边的边进行查询，可能使用索引

2、like子句中，如果是常量并且只有%不在第一字符时，可能使用索引

3、如果对大的文本进行搜索，使用全文索引而不用使用 like ‘%…%’。

4、如果列名是索引，使用 column\_name is null 将使用索引。

### 3.3 有索引但不使用索引的情况

1、如果列名是索引，使用 column\_name is null 将使用索引。

2、如果使用 MEMORY/HEAP 表并且 where 条件中不使用“=”进行索引列，那么

不会用到索引

3、用 or分割开的条件，如果 or前的条件中的列有索引，而后面的列中没有索引，

那么涉及到的索引都不会被用到

4、如果不是复合索引的第一列

5、如果 like 是以％开始

6、如果列类型是字符串，那么一定记得在 where 条件中把字符常量值用引号引

起来，否则的话即便这个列上有索引，MySQL 也不会用到的

### 3.4 查看索引使用情况

show status like 'Handler\_read%';

若索引正在使用，handler\_read\_key 将会很高

Handler\_read\_rnd\_next 的值高则意味着查询运行低效，并且应该建立索引补救。这个值

的含义是在数据文件中读下一行的请求数。如果正进行大量的表扫描，

Handler\_read\_rnd\_next 的值较高，则通常说明表索引不正确或写入的查询没有利用索引

## 四、事务与锁

## 五、sql语句的优化

1、了解各种sql执行的频率

Show status like ‘Com\_%’;

Com\_select：执行 select 操作的次数，一次查询只累加 1。

 Com\_insert：执行 INSERT 操作的次数，对于批量插入的 INSERT 操作，只累加一次。

 Com\_update：执行 UPDATE 操作的次数。

 Com\_delete：执行 DELETE 操作的次数。

2、定位效率慢的语句

--log-slow-queries[=filen\_ame]

加入上面启动项后，mysql会将执行时间超过long\_query\_time（26章）秒的语句，记录到日志文件中。

3、分析低效的sql

Explain <select | insert …>

**select\_type**：表示 SELECT 的类型，常见的取值有

SIMPLE（简单表，即不使用表连接或者子查询）、

PRIMARY（主查询，即外层的查询）、

UNION（UNION 中的第二个或者后面的查询语句）、

SUBQUERY（子查询中的第一个 SELECT）

**table**：输出结果集的表

**type**：表示表的连接类型,只要不是all（全表扫描），都可以的

**possible\_keys**：表示查询时，可能使用的索引。

**key**：表示实际使用的索引。



**key\_len**：索引字段的长度。

 **rows**：扫描行的数量。

 **Extra**：执行情况的说明和描述。

4、确定问题并采取相应的优化措

## 六、大批量数据的导入

1、当用 load 命令导入数据的时候，适当的设置可以提高导入的速度。

对于 MyISAM 存储引擎的表，可以通过以下方式快速的导入大量的数据。

**ALTER TABLE tbl\_name DISABLE KEYS;**

**loading the data**

**ALTER TABLE tbl\_name ENABLE KEYS;**

DISABLE KEYS 和 ENABLE KEYS 用来打开或者关闭 MyISAM 表非唯一索引的更新。在导入

大量的数据到一个非空的 MyISAM 表时，通过设置这两个命令，可以提高导入的效率。对于导入大量数据到一个空的 MyISAM 表，默认就是先导入数据然后才创建索引的，所以不用进行设置。

2、在导入数据前执行 SET UNIQUE\_CHECKS=0，关闭唯一性校验，在导入结束后执行

SET UNIQUE\_CHECKS=1，恢复唯一性校验，可以提高导入的效率。

3、如果应用使用自动提交的方式，建议在导入前执行 SET AUTOCOMMIT=0，关闭自

动提交，导入结束后再执行 SET AUTOCOMMIT=1，打开自动提交，也可以提高导入的效率。

## 七、数据厍对象的优化

## 八、锁问题

## 九、连接

### 9.1 内连接

**内连接查询操作列出与连接条件匹配的数据行，它使用比较运算符比较被连接列的列值。**

select \* from book as a,stu as b where a.sutid = b.stuid

select \* from book as a inner join stu as b on a.sutid = b.stuid

内连接可以使用上面两种方式，其中第二种方式的inner可以省略。

### 9.2 外连接

#### 9.2.1左连接

是以左表为基准，将a.stuid = b.stuid的数据进行连接，然后将左表没有的对应项显示，右表的列为NULL

select \* from book as a **left join** stu as b on a.sutid = b.stuid

#### 9.2.2右连接 是以右表为基准，将a.stuid = b.stuid的数据进行连接，然以将右表没有的对应项显示，左表的列为NULL

select \* from book as a **right join** stu as b on a.sutid = b.stuid

#### 9.2.3全连接

完整外部联接返回左表和右表中的所有行。当某行在另一个表中没有匹配行时，则另一个表的选择列表列包含空值。如果表之间有匹配行，则整个结果集行包含基表的数据值。

select \* from book as a full **outer join** stu as b on a.sutid = b.stuid

### 9.3 交叉连接

交叉联接返回左表中的所有行，左表中的每一行与右表中的所有行组合。交叉联接也称作笛卡尔积。

select \* from book as a **cross join** stu as b order by a.id