

MySQL高级-Day01回顾

MySQL高级-Day02笔记

外键（见Day01笔记）

多表查询（见Day01笔记）

连接查询（见Day01笔记）

数据导入

数据导出

表的复制

锁（自动加锁和释放锁）

存储引擎

MySQL的用户账户管理

今日作业

MySQL高级-Day01回顾

- SQL查询总结

```
3、select ...聚合函数 from 表名
1、where ...
2、group by ...
4、having ...
5、order by ...
6、limit ...;
```

- 聚合函数（铁三角之一）

avg(...) sum(...) max(...) min(...) count(字段名) # 空值NULL不会被统计

- group by（铁三角之二）

给查询结果进行分组 如果select之后的字段名和group by之后的字段不一致,则必须对该字段进行聚合处理(聚合函数)

- having语句（铁三角之三）

对查询的结果进行进一步筛选 注意 1、having语句通常和group by语句联合使用,过滤由group by语句返回的记录集
2、where只能操作表中实际存在字段,having可操作由聚合函数生成的显示列

- distinct

select distinct 字段1,字段2 from 表名;

- 查询时做数学运算

select 字段1*2,字段2+5 from 表名;

update 表名 set attack=attack*2 where 条件;

- 索引(BTree)

优点：加快数据检索速度 缺点：占用物理存储空间,需动态维护,占用系统资源 SQL命令运行时间监测

mysql>show variables like '%pro%';

1、开启：mysql> set profiling=1; 2、查看：mysql> show profiles; 3、关闭：mysql> set profiling=0;

- 普通(MUL)、唯一(UNI,字段值不能重复,可为NULL)

创建 index(字段名),index(字段名) unique(字段名),unique(字段名) create [unique] index 索引名 on 表名(字段名);

查看 desc 表名; show index from 表名\G;

Non_Unique:1 :index Non_Unique:0 :unique

删除 drop index 索引名 on 表名; (只能一个一个删)

MySQL高级-Day02笔记

外键（见Day01笔记）

==1、原理==

==2、级联动作，以及每个级联动作特点==

多表查询（见Day01笔记）

==1、笛卡儿积是如何匹配的==

==2、多表查询：在笛卡儿积的基础上进行进一步条件筛选==

连接查询（见Day01笔记）

==全程重点，整体分类，具体语法实现==

数据导入

==掌握大体步骤==

==source 文件名.sql==

作用

把文件系统的内容导入到数据库中 语法（方式一）

load data infile "文件名" into table 表名 fields terminated by "分隔符" lines terminated by "\n" 示例
scoretable.csv文件导入到数据库db2的表aid1903中

- 1、将scoretable.csv放到数据库搜索路径中
mysql>show variables like 'secure_file_priv';
/var/lib/mysql-files/
Linux: sudo cp /home/tarena/scoreTable.csv /var/lib/mysql-files/
- 2、在数据库中创建对应的表
create table scoretab(
rank int,

```
name varchar(20),
score float(5,2),
phone char(11),
class char(7)
)charset=utf8;
```

3、执行数据导入语句

```
load data infile '/var/lib/mysql-files/scoreTable.csv'
into table scoretab
fields terminated by ','
lines terminated by '\n'
```

4、练习

添加id字段, 要求主键自增长, 显示宽度为3, 位数不够用0填充

```
alter table scoretab add id int(3) zerofill primary key auto_increment first;
```

语法 (方式二)

source 文件名.sql

数据导出

作用

将数据库中表的记录保存到系统文件里

语法格式

select ... from 表名 into outfile "文件名" fields terminated by "分隔符" lines terminated by "分隔符";

练习

1、把sanguo表中英雄的姓名、攻击值和国家三个字段导出来, 放到 sanguo.csv中

```
select name,attack,country from country.sanguo
into outfile '/var/lib/mysql-files/sanguo.csv'
fields terminated by ','
lines terminated by '\n';
```

2、将mysql库下的user表中的 user、host两个字段的值导出到 user2.txt, 将其存放在数据库目录下

```
select user,host from mysql.user
into outfile '/var/lib/mysql-files/user2.txt'
fields terminated by ' '
lines terminated by '\n';
```

注意

- 1、导出的内容由SQL查询语句决定
- 2、执行导出命令时路径必须指定在对应的数据库目录下

表的复制

==1、表能根据实际需求复制数据==

==2、复制表时不会把KEY属性复制过来==

语法

```
create table 表名 select 查询命令;
```

练习

- 1、复制sanguo表的全部记录和字段, sanguo2

```
create table sanguo2 select * from country.sanguo;
```
- 2、复制sanguo表的前3条记录, sanguo3
- 3、复制sanguo表的 id,name,country 三个字段的前3条记录, sanguo4

```
create table sanguo select id,name,country from sanguo limit 3;
```

注意

复制表的时候不会把原有表的 KEY 属性复制过来

复制表结构 create table 表名 select 查询命令 where false;

锁（自动加锁和释放锁）

==全程重点，理论和锁分类及特点==

目的

解决客户端并发访问的冲突问题

锁类型分类

读锁(共享锁): select 加读锁之后别人不能更改表记录,但可以进行查询
写锁(互斥锁、排他锁): 加写锁之后别人不能查、不能改

锁粒度分类

表级锁: myisam 行级锁: innodb

存储引擎

基本操作

- 1、查看所有存储引擎
- 2、查看已有表的存储引擎
- 3、创建表时指定存储引擎
- 4、已有表指定存储引擎

常用存储引擎特点

InnoDB特点

- 1、支持行级锁
- 2、支持外键、事务操作
- 3、表字段和索引同存储在一个文件中

MyISAM特点

- 1、支持表级锁
- 2、表字段和索引分开存储

如何决定使用哪个存储引擎

- 1、执行查操作多的表用 MyISAM
- 2、执行写操作多的表用 InnoDB

MySQL的用户账户管理

开启MySQL远程连接

添加授权用户

- 1、用root用户登录mysql
`mysql -uroot -p123456`
- 2、授权
`grant 权限列表 on 库.表 to "用户名"@"%" identified by "密码" with grant option;`

权限列表

all privileges 、select 、insert

库.表： *.* 代表所有库的所有表

示例

- 1、添加授权用户tiger, 密码123, 对所有库的所有表有所有权限
- 2、添加用户rabbit, 对db2库有所有权限

今日作业

- 1、把 /etc/passwd 文件的内容导入到数据库的表中

```
tarena:x:1000:1000:tarena,,,:/home/tarena:/bin/bash
```

- 2、Day01的md文件中的外键及查询作业题