1、MySQL的安装

1、Ubuntu安装MySQL服务 RedHat(红帽)、CentOS、Ubuntu

1、安装服务端

sudo apt-get install mysql-server

2、安装客户端

sudo apt-get install mysql-client

2、Windows安装MySQL服务

1、下载MySQL安装包(windows)

mysql-installer\*\*\*5.7.\*\*\*.msi

2、双击、按照教程安装即可

2、启动和连接MySQL服务

1、服务端启动

1、查看MySQL服务状态

sudo /etc/init.d/mysql status

2、停止、启动、重启MySQL服务

sudo /etc/init.d/mysql stop | start | restart

2、客户端连接

1、命令格式

mysql -h主机地址 -u用户名 -p密码

mysql -hlocalhost -uroot -p123456

2、本地连接可以省略 -h 选项

mysql -u用户名 -p密码

mysql -uroot -p123456

3、库的管理

1、库的基本操作

1、查看已有的库

show databases;

2、创建库(指定默认字符集)

create database 库名 default charset=utf8;

3、查看创建库的语句

show create database 库名;

4、查看当前所在库

select database();

5、切换库

use 库名;

6、查看库中已有表

show tables;

7、删除库

drop database 库名;

4、表的管理

1、表的基本操作

1、创建表(指定字符集)

create table 表名(

字段名 数据类型,

字段名 数据类型,

... ...

);

2、查看创建表的语句(字符集)

show create table 表名;

3、查看表结构

desc 表名;

4、删除表

drop table 表名;

5、更改库、表的默认字符集

1、方法

通过更改MySQL服务的配置文件来实现

2、步骤

1、获取root权限

sudo -i

2、切换到配置文件所在路径

cd /etc/mysql/mysql.conf.d

3、备份

cp -p mysqld.cnf mysqld.cnf.bak

(-p选项连同原文件权限一起复制)

4、用vi打开mysqld.cnf

vi mysqld.cnf

[mysqld]

character\_set\_server = utf8 保存退出

a -> 写入 -> ESC -> shift + : -> wq

5、重启mysql服务

/etc/init.d/mysql restart | reload(重载配置文件)

6、退出超级用户 exit

7、登录到mysql验证：mysql -uroot -p123456

create database 库名;

show create database 库名;

6、表记录管理

1、在表中插入记录

1、insert into 表名 values(值1),(值2),...,(值N);

2、insert into 表名(字段名列表) values(值1),...(值N);

2、查询表记录

1、select \* from 表名;

2、select 字段1,字段名2,...,字段名N from 表名;

7、表字段的操作

1、语法：alter table 表名 执行动作;

1、添加字段

alter table 表名 add 字段名 数据类型 first | after 字段名;

2、删除字段

alter table 表名 drop 字段名;

3、修改字段数据类型

alter table 表名 modify 字段名 新数据类型;

# 修改数据类型时会受到表中原有数据的限制

4、修改字段名

alter table 表名 change 旧名 新名 数据类型;

5、修改表名

alter table 表名 rename 新表名;

MySQL-Day01回顾

1、MySQL的特点

1、关系型数据库

2、跨平台

3、支持多种编程语言

2、MySQL的启动和连接

1、服务端启动

sudo /etc/init.d/mysql start|stop|restart|reload

2、客户端连接

mysql -h主机地址 -u用户名 [-P3306] -p密码

3、注意

1、MySQL中的数据是以文件的形式存储在数据库目录/var/lib/mysql

2、关系型数据库的核心内容是 关系 即 二维表

3、库的管理

1、创建库(指定字符集)

create database 库名 default charset = utf8;

2、查看创建库的语句

show create database 库名;

3、查看当前所在库

select database();

4、切换库

use 库名;

5、查看库中已有表

show tables;

6、删除库

drop database 库名;

4、表的管理

1、创建表(指定字符集)

CREATE TABALE 表名(字段名 数据类型,...)DEFAULT CHARSET = UTF8;

2、查看创建表的语句(字符集和存储引擎)

show create table 表名;

3、查看表结构

desc 表名;

4、删除表

drop table 表名;

5、表记录的管理

1、插入记录

insert into 表名 values(),(),...;

insert into 表名(字段名列表) values(),(),...;

2、查询表记录

select \* from 表名;

select 字段名1,字段名2,... from 表名;

6、表字段管理

1、添加(add)

alter table 表名 add 字段名 数据类型;

alter table 表名 add 字段名 数据类型 first;

alter table 表名 add 字段名 数据类型 after 字段名;

2、修改(modify)

alter table 表名 modify 字段名 新数据类型;

3、删除(drop)

alter table 表名 drop 字段名;

4、字段名重命名(change)

alter table 表名 change 旧字段名 新字段名 数据类型;

5、表的重命名(rename)

alter table 表名 rename 新表名;

7、如何更改默认字符集

1、步骤

1、sudo -i

2、cd /etc/mysql/mysql.conf.d/

3、cp -p mysqld.cnf mysqld.cnf.bak

4、vi mysqld.cnf

[mysqld]

character\_set\_server = utf8

5、/etc/init.d/mysql restart | reload

8、客户端把数据存储到数据库服务器上的过程

1、连接到数据库

2、选择库

3、创建/修改表

4、断开连接

9、名词

DB DBMS(MySQL、Oracle...) DBS(数据库系统)

10、数据类型

1、数值类型

1、整型

1、int(4个字节)

2、tinyint(1个字节) 有符号、无符号unsigned

2、浮点型

1、float(4个字节,最多显示7个有效位)

2、double(8个字节,最多显示15个有效位)

3、decimal(M+2个字节,最多显示28个有效位)

2、字符类型

1、char(定长,浪费存储空间,但是性能高)

2、varchar(变长,节省存储空间,但是性能低)

3、枚举类型

1、enum(最多有65535个不同值)

2、set(最多有64个不同值)

## 在插入记录时 "值1,值2,..."

4、日期时间类型

1、year

2、date

3、time

4、datetime # 不给值默认返回NULL

5、timestamp # 不给值默认返回系统当前时间

11、where条件子句(配合查询、修改、删除操作)

1、语法格式

select \* from 表名 where 条件;

12、表记录管理

1、删除表记录

1、delete from 表名 where 条件;

2、注意

delete语句后如果不佳where条件子句,会将表中所有记录全部删除

2、更新表记录

1、update 表名 set 字段名1=值1,字段名2=值2,... where 条件;

2、注意

update语句后如果不加where条件子句会将表中所有记录全部更改

13、运算符操作

1、数值比较&字符比较

1、数值比较运算符：=、!=、>、>=、<、<=

2、字符比较运算符：=、!=

3、练习

1、找出攻击力高于150的英雄的名字和攻击力值

2、找出防御力不等于100的英雄信息

2、逻辑比较

1、运算符：

and (两个或者多个条件同时满足)

or (两个或者多个条件有一个满足就可以)

2、练习

1、找出攻击值大于200的蜀国英雄的名字和攻击值

2、查找蜀国和魏国的英雄信息

3、将吴国英雄中攻击值为110的英雄的攻击值设置为100，防御值设置为60

3、范围内比较

1、运算符

between and、 in 、not in

2、语法

between 值1 and 值2

in(值1,值2,...,值N)

not in(值1,值2,...,值N)

3、练习

1、查找攻击值在100-200之间的蜀国英雄信息

2、找到蜀国和吴国以外国家的女英雄信息

4、匹配空、非空

1、空 ：is null

2、非空 ：is not null

3、示例

1、查找姓名为NULL的蜀国女英雄信息

2、查找姓名为""的英雄的id、姓名和国家

4、注意

1、null ：空值,必须用 is 或者 is not 去匹配

2、"" ：空字符串,用 = 或者 != 去匹配

5、模糊比较

1、where 字段名 like 表达式;

2、表达式

1、\_ : 匹配单个字符

2、% : 匹配0到多个字符

3、示例

1、select name from sanguo where name like "\_%\_";

# 匹配名字中至少有两个字的英雄

2、select name from sanguo where name like "%";

# NULL不会被统计

3、select name from sanguo where name like "\_\_\_";

# 匹配名字为三个字的英雄

4、select name from sanguo where name like "赵%";

# 匹配姓赵的英雄

6、正则匹配查询 regexp

1、where 字段名 regexp "正则表达式";

2、正则表达式符号

^ : 以...开头

$ : 以...结尾

. : 匹配任意一个字符

[]: 包含...内容

[0-9]:匹配带数字的

[a-z]:匹配带小写字母的

[A-Z]

\* : 星号前面的字符出现0个或者多次 "^赵.\*"

3、示例

1、select name from sanguo where name regexp "[0-9]"; # 匹配名字中带数 字的

2、select name from sanguo where name regexp "^[0-9]"; # 匹配名字中以数 字开头的记录

3、select name from sanguo where name regexp "[0-9]$"; # 匹配名字中以 数字结尾的记录

4、select name from sanguo where name regexp "^司.\*懿$"; # 匹配以 司 开 头,以 懿 结尾的记录

5、select name from sanguo where name regexp "^...$"; # ... ... ...

14、SQL查询

1、总结(执行顺序)

3、select ... 聚合函数 from 表名

1、where ...

2、group by ...

4、having ...

5、order by ...

6、limit ...

2、order by

1、作用：给查询的结果进行排序

2、语法格式：order by 字段名 排序方式;

3、排序方式

1、ASC(默认) : 升序

2、DESC : 降序

4、示例

1、将英雄按防御值从低到高排序

select \* from sanguo order by fangyu asc;

2、将蜀国英雄按攻击值从高到低排序

select \* from sanguo where country = "蜀国" order by gongji desc;

3、将魏蜀两国的男英雄中名字为三个字的英雄按防御值升序排列

select \* from sanguo where country in("魏国","蜀国") and sex="男" and name like "\_\_\_" order by fangyu asc;

3、limit(永远放在SQL语句的最后写)

1、作用：限制显示查询记录的个数

2、用法

1、limit n ：-->显示n条记录

2、limit m,n ：-->从第(m+1)条开始,显示n条记录

limit 4,5 ：显示 第5名-第9名

## m的值是从0开始计数的

3、示例

1、查找防御值倒数第2名到倒数第4名的蜀国英雄记录

select \* from sanguo where country="蜀国" order by fangyu asc limit 1,3;

2、查找攻击值前3名且名字不为空值的蜀国英雄的姓名、攻击值和国家

select name,gongji,country from sanguo where name is not null and country="蜀国" order by gongji desc limit 3;

4、聚合函数

1、分类

1、avg(字段名) : 求字段的平均值

2、sum(字段名) : 求和

3、max(字段名) : 求最大值

4、min(字段名) : 求最小值

5、count(字段名):统计该字段记录的个数

2、示例

1、攻击力最强值是多少

select max(gongji) as best from sanguo;

2、统计一下id,name两个字段分别有多少条记录

select count(id),count(name) from sanguo;

# 空值NULL不会被统计,空字符串""会被统计

3、统计蜀国英雄中攻击值大于200的英雄的数量

select count(\*) from sanguo where country="蜀国" and gongji>200;

5、group by(先分组,再聚合)

1、作用 ：给查询的结果进行分组

2、示例

1、统计sanguo表中一共有几个国家

select country from sanguo group by country;

2、计算所有国家的平均攻击力

select country,avg(gongji) from sanguo group by country;

执行过程：

1、先分组(此时未去重) group by country

2、再聚合(求每组的平均攻击值)

3、去重

3、查找所有国家中 英雄数量最多的 前2名 的国家名称及英雄数量

select country,count(\*) as numbers from sanguo group by country order by numbers desc limit 2;

3、注意

1、group by之后的字段名必须要为select之后的字段名

2、如果select之后的字段没有在group by语句之后,则必须要对该字段进行 聚合处理(聚合函数)

6、having

1、作用 ：对查询的结果进行进一步筛选

2、示例

1、找出平均攻击力大于105的国家的前2名,显示国家名称和平均攻击力

select country,avg(gongji) as pjgj from sanguo group by country having pjgj > 105 order by pjgj desc limit 2;

3、注意

1、having语句通常与group by语句联合使用,用来过滤由group by语句返回 的记录集

2、having语句的存在弥补了where条件子句不能与聚合函数联合使用的不 足,where操作的是表中实际存在的字段,having操作的是聚合函数生成的显示 列

7、distinct

1、作用：不显示字段的重复值

2、示例

1、sanguo表中一共有哪些个国家

select distinct country from sanguo;

2、计算蜀国一共有多少个英雄

select count(distinct name) from sanguo where country = "蜀国";

3、注意

1、distinct处理的是distinct和from之间的所有字段,所有字段的值必须完全 相同才可以去重

2、distinct不能对任何字段做聚合处理

8、查询表记录时可以做数学运算

1、运算符

+ - \* / %

2、示例

1、查询时显示所有英雄的攻击力 \* 10

select name,gongji\*10,country from sanguo;

15、约束

1、作用

1、为了保证数据的完整性、一致性、有效性,可以限制无效的数据插入到数据表中

2、约束分类

1、默认约束(default)

1、作用

在插入记录时,如果不给该字段赋值,则使用默认值

2、格式

字段名 数据类型 default 默认值,

2、非空约束

1、作用

不允许该字段的值有空值NULL记录

2、格式

字段名 数据类型 not null,

16、索引

1、定义

对数据库中表的一列或者多列的值进行排序的一种结构(MySQL中索引用Btree方 式)

2、索引的优点

可以加快数据的检索速度

3、索引的缺点

1、当对表中的数据进行增加、修改、删除的时候,索引需要动态维护,降低了数据 的维护速度

2、索引需要占用物理存储空间(数据库目录/var/lib/mysql)

4、索引示例

1、运行 insert\_.py 文件,插入100万条记录

2、开启性能分析

show variables like "%pro%";

set profiling = 1;

3、执行查询语句

select name from t1 where name="lucy99999";

4、查看性能分析结果

show profiles;

5、在name字段创建索引

create index name on t1(name);

6、执行查询语句

select name from t1 where name="lucy99999";

8、查看性能分析结果

show profiles;

9、关闭性能分析

set profiling = 0;

5、索引分类

1、普通索引

2、唯一索引

3、主键索引

4、外键索引

6、普通索引(index)

1、使用规则

1、一个表中可以有多个index字段

2、字段的值可以有重复,也可以为NULL值

3、经常把做查询条件的字段设置为index字段

4、index字段的key标志为: MUL

2、创建

1、创建表时创建index

create table t1(

... ...,

... ...,

index(id),

index(name));

2、在已有表中添加索引字段

1、语法格式

create index 索引名 on 表名(字段名);

# 索引名一般和字段名一样

3、查看

1、desc 表名; ->查看KEY标志为 MUL

2、show index from 表名\G;

4、删除

drop index 索引名 on 表名;

注意：

删除普通索引只能一个一个删除

7、唯一索引(unique)

1、使用规则

1、一个表中可以有多个 unique 字段

2、unique字段的值不允许重复,可以为空值NULL

3、unique的KEY标志是 UNI

2、创建(基本等同index创建)

1、创建表时创建

unique(字段名),

unique(字段名)

2、已有表中创建

create unique index 索引名 on 表名(字段名);

3、查看、删除唯一索引

desc 表名;

show index from 表名;

drop index 索引名 on 表名;

8、主键索引(primary key) && 自增长属性(auto\_increment)

1、使用规则

1、一个表中只能有一个主键字段

2、对应字段的值不允许重复 且 不能为空值NULL

3、主键字段的KEY标志为 PRI

4、把表中能够唯一标识一条记录的字段设置为主键,通常把表中记录编号的字 段设置为主键

2、创建主键(PRI)

1、创建表时创建

1、字段名 数据类型 primary key auto\_increment,

2、

id int auto\_increment,

... ..., # 设置自增长起始值

primary key(id))auto\_increment=10000;

2、删除主键

1、先删除自增长属性(modify)

alter table 表名 modify id int;

2、删除主键

alter table 表名 drop primary key;

3、在已有表中添加主键

alter table 表名 add primary key(字段名);

9、外键

1、定义

让当前表字段的值在另一个表的范围内选择

2、语法格式

foreign key(参考字段名)

references 被参考表名(被参考字段名)

on delete 级联动作

on update 级联动作

3、案例

表1、缴费信息表(财务)

学号 姓名 班级 缴费金额

1 唐伯虎 AID01 28000

2 点秋香 AID01 20000

表2、学生信息表(班主任)

学号 姓名 缴费金额

1 唐伯虎 28000

2 点秋香 20000

1、创建缴费信息表

create table jftab(

id int primary key,

name char(15),

class char(5),

money int

)default charset=utf8;

insert into jftab values

(1,"唐伯虎","AID01",28000),

(2,"点秋香","AID01",20000),

(3,"祝枝山","AID01",22000);

2、创建学生信息表(从表)

create table bjtab(

stu\_id int,

name char(15),

money int,

foreign key(stu\_id) references jftab(id)

on delete cascade

on update cascade

);

4、级联动作

1、cascade ：数据级联更新

当主表删除记录 或者 更新被参考字段的值时,从表会级联更新

2、restrict 默认

1、当删除主表记录时,如果从表中有相关联记录则不允许主表删除

2、更新同理

3、set null

1、当主表删除记录时,从表中相关联记录的参考字段值自动设置为NULL

2、更新同理

4、no action

on delete no action on update no action

同 restrict,都是立即检查外键限制

5、删除外键

alter table 表名 drop foreign key 外键名;

1、外键名的查看方式

show create table 表名;

6、已有表中添加外键

## 会受到表中原有数据的限制

alter table 表名 add foreign key(参考字段名)

references 被参考表名(被参考字段名)

on delete 级联动作

on update 级联动作;

7、外键使用规则

1、两张表被参考字段和参考字段数据类型要一致

2、被参考字段必须是 key 的一种,通常是 primary key

17、数据导入

1、作用：把文件系统的内容导入到数据库中

2、语法

load data infile "文件名"

into table 表名

fields terminated by "分隔符"

lines terminated by "\n"

3、示例

把 /etc/passwd 文件中的内容导入到db2库下的userinfo表

tarena : x : 1000 : 1000 :

用户名 密码 UID号 GID号

tarena,,, : /home/tarena : /bin/bash

用户描述 用户主目录 登录权限

/bin/false

/usr/sbin/nologin

4、操作步骤

1、在数据库中创建对应的表

create table userinfo(

username char(20),

password char(1),

uid int,

gid int,

comment varchar(50),

homedir varchar(50),

shell varchar(50)

);

2、将要导入的文件拷贝到数据库的默认搜索路径中

1、查看数据库的默认搜索路径

show variables like "secure\_file\_priv";

/var/lib/mysql-files

2、Linux命令行输入：

sudo cp /etc/passwd /var/lib/mysql-files/

3、执行数据导入语句

load data infile "/var/lib/mysql-files/passwd"

into table userinfo

fields terminated by ":"

lines terminated by "\n";

5、练习：将AID1709.csv文件导入到数据库中

# csv文件单元格之间以 , 分隔

/var/lib/mysql-files/AID1709.csv

ls -l AID1709.csv

rw-------

chmod 666 AID1709.csv

1、在数据库中创建对应的表

id 姓名 成绩 手机号 班级

create table scoretab(

id int,

name varchar(20),

score float(5,2),

phone char(11),

class char(7)

)default charset=utf8;

2、把导入的文件复制到数据库的默认搜索路径中

cp 源文件 目标路径

cp /home/tarena/AID1709.csv /var/lib/mysql-flies/

######## 用 TAB 键 补齐路径 #######

3、执行数据导入语句

load data infile "/var/lib/mysql-files/AID1709.csv"

into table scoretab

fields terminated by ","

lines terminated by "\n";

# 修改文件权限 chmod 666 AID1709.csv

17、数据导出

1、作用

将数据库表中的记录保存到系统文件里

2、语法格式

select ... from 表名

into outfile "文件名"

fields terminated by "分隔符"

lines terminated by "\n";

3、把userinfo表中的username、password和uid导出到文件user.txt

select username,password,uid from userinfo

into outfile "/var/lib/mysql-files/user.txt"

fields terminated by ","

lines terminated by "\n";

1、sudo -i

2、cd /var/lib/mysql-files/

3、cat user.txt

4、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对应的数据库搜索路径

18、表的复制

1、语法格式

create table 表名 select 查询命令;

2、示例

1、复制userinfo表中的全部记录,userinfo2

create table userinfo2 select \* from userinfo;

2、复制userinfo表中username,password,uid三个字段的第2-10条记录,userinfo3

create table userinfo3 select username,password,uid from userinfo limit 1,9;

3、复制表结构

create table 表名 select 查询命令 where false;

4、注意

复制表的时候不会把原有表的 key 属性复制过来

19、嵌套查询(子查询)

1、定义

把内层的查询结果作为外层的查询条件

2、示例

1、把uid的值小于 uid 平均值的用户名和uid号显示出来

select username,uid from userinfo

where uid < (select avg(uid) from userinfo);

20、连接查询

1、内连接

1、定义

从表中删除与其他被连接的表中没有匹配到的行

2、语法格式

select 字段名列表 from 表1

inner join 表2 on 条件 inner join 表3 on 条件;

3、示例

1、显示省市的详细信息

select sheng.s\_name,city.c\_name from sheng

inner join city on sheng.s\_id=city.cfather\_id;

2、显示省市县详细信息

select sheng.s\_name,city.c\_name,xian.x\_name from sheng

inner join city on sheng.s\_id=city.cfather\_id

inner join xian on city.c\_id=xian.xfather\_id;

2、外连接

1、左连接

1、定义

以左表为主显示查询结果

2、语法格式

select 字段名列表 from 表1

left join 表2 on 条件;

3、示例

1、以省表为主显示省市详细信息

select sheng.s\_name,city.c\_name from sheng

left join city on sheng.s\_id=city.cfather\_id;

2、显示省市区详细信息,要求县全部显示

select sheng.s\_name,city.c\_name,xian.x\_name from sheng left join city

on sheng.s\_id=city.cfather\_id

right join xian on city.c\_id=xian.xfather\_id;

3、显示省市区详细信息,要求 市 全部显示

select sheng.s\_name,city.c\_name,xian.x\_name from sheng

right join city on sheng.s\_id=city.cfather\_id

left join xian on city.c\_id=xian.xfather\_id;

#### 结果集 ####

2、右连接

用法同左连接,以右表为主显示查询结果

21、多表查询

1、select 字段名列表 from 表名列表; # 笛卡尔积

2、select 字段名列表 from 表名列表 where 条件;

等同于 内连接 inner join