**9\_RDMBS103数据导入导出+管理表记录+匹配条件+mysql管理工具**

**1 案例1：数据导入导出**

**1.1 问题**

修改检索目录为/myload

将/etc/passwd文件导入db3库的user表里，并添加行号字段。

将db3库user表所有记录导出, 存到/myload/user.txt 文件里。

步骤一：修改检索目录为/myload

**1）修改配置文件，重启服务**

]# mkdir /myload

]# chown mysql /myload

]# vim /etc/my.cnf

[mysqld]

secure\_file\_priv="/myload” #保存并退出

]# systemctl restart mysqld #重启服务

mysql> show variables like “secure\_file\_priv”; #查看

+------------------+-----------------------+

| Variable\_name | Value |

+------------------+-----------------------+

| secure\_file\_priv | /myload/ |

+------------------+-----------------------+

**2）新建db3库、user表**

[root@dbsvr1 ~]# mysql -u root –p123456

mysql> create database db3;

mysql> create table db3.user(

name char(50),

password char(1),

uid int,

gid int,

comment char(150),

homedir char(50),

shell char(50)

);

Query OK, 0 rows affected (0.70 sec)

**步骤二：将/etc/passwd文件导入db3库的user表里，并添加行号字段。**

**1）拷贝文件到检索目录下**

[root@dbsvr1 ~]#

[root@dbsvr1 ~]# cp /etc/passwd /myload/

**2）导入数据**

[root@dbsvr1 ~]# mysql –uroot –ptarena

mysql>**load data infile** "/myload/passwd"

**into table** db3.user

**fields terminated by** ":"

**lines terminated by** "\n" ; #导入数据

mysql> select \* from db3.user; #查看导入后表记录

mysql> alter table db3.user

-> add

-> id int primary key auto\_increment first;

#给user表第1列添加id字段并设置为主键且自增减.

mysql> select \* from db3.user; #查看表记录

**步骤三：将db3库user表所有记录导出, 存到/myload/user.txt 文件里。**

**1）查询要导出的数据**

    mysql> select \* from db3.user ；

**2）导出数据**

**命令格式:**

* **sql查询命令 into outfile “目录/文件”;**
* **sql查询命令 into outfile “目录/文件”**

**fields terminated by “分隔符”;**

* **sql查询命令 into outfile “目录/文件”**

**fields terminated by “分隔符”**

**lines terminated by “\n”;**

**注意事项:**

**导出数据行数由SQL查询决定;**

**导出的是表记录,不包括字段名;**

**导出时自动创建存储数据的文件;**

**存储数据文件,具有唯一性.**

mysql> **select \* from** db3.user

**into outfile** "/myload/user1.txt";

**3）查看文件内容**

]# cat /myload/user1.txt

**二 管理表记录**

**2.1增加表记录**

**插入1条表记录给所有字段赋值**

**insert into 库.表 values(值列表);**

**插入多条表记录给所有字段赋值**

**insert into 库.表 values(值列表),(值列表)...;**

**插入1条表记录给指定字段赋值**

**insert into 库.表(字段) values(值列表);**

**插入多条表记录给指定字段赋值**

**insert into 库.表(字段) values(值列表),(值列表)...;**

**2.2 查询表记录**

**select 字段1,字段2,...字段N from 库.表;**

**select 字段1,字段2,...字段N from 库.表 where 条件表达式;**

**字段控制显示的列,条件控制显示的行.**

**2.3 更新表记录**

**批量更新: update 库.表 set 字段名=值,字段名=值,...;**

**条件匹配更新:update 库.表 set 字段名=值,字段名=值,... where 条件;**

**2.4 删除表记录**

**条件匹配删除:delete from 库.表 where 条件;**

**删除所有记录:delete from 库.表;**

**三 匹配条件**

**适用于select update delete**

**3.1基本匹配条件**

**数值比较,字段必须是数值类型,比较类型: = != > >= < <=**

**字符比较,字段必须是字符类型,比较类型: = != is null is not null**

**逻辑匹配,多个判断条件时使用,匹配类型: and or !或not**

**范围内匹配/去重显示**

**in(值列表) 在...里**

**not in(值列表) 不在...里**

**between 数字 and 数字 在...之间**

**distinct 字段名 去重显示**

**3.2高级匹配条件**

**3.2.1 模糊查询**

用法: **where 字段名 like “通配符”;** \_表示1个字符,%表示0~n个字符.

**3.2.2 正则表达式**

用法:**where 字段名 regexp ‘正则表达式’**

^ $ . [] \* |

**3.2.3 四则运算: + - \* / % ()提高优先级**

**四 操作查询结果**

聚集函数(对数据做统计的命令叫函数,**字段名为数值类型**)

**avg(字段名) sum(字段名) min(字段名) max(字段名) conut(字段名)**

**4.1查询结果排序(通常是数值型字段)**

**order by 字段名 [asc|desc];** 默认为升序 asc:升序 desc:降序

**4.2 查询结果分组(通常是字符型字段)**

**group by 字段名;**

**4.3 查询结果过滤**

**having 条件表达式;**

**4.4 限制查询结果显示行数**

**limit 数字;** #显示头几行

**limit 数字1,数字2;** 数字1 为起始行,数字2为总行数,0表示第一行.

#显示指定范围内的行

**五 MYSQL管理工具**

命令行

安装软件,由软件提供管理数据库的页面(web页面或软件提供的图形窗口)

类型 界面 操作系统 说明

mysql 命令行 跨平台 MySQL官方bundle包自带

MySQL-workbench 图形 跨平台 MySQL官方提供

MySQL-front 图形 Windows 开源,轻量级客户端软件

Navicat 图形 Windows 专业 功能强大 商业版

phpMyAdmin 浏览器 跨平台 开源,需要LAMP平台

**五 案例**

**5.1匹配条件练习**

1）常用的表记录统计函数

查询stu\_info表一共有多少条记录（本例中为5条）：

mysql> SELECT count(\*) FROM stu\_info;

计算stu\_info表中各学员的平均年龄、最大年龄、最小年龄：

mysql> SELECT avg(age),max(age),min(age) FROM stu\_info;

计算stu\_info表中男学员的个数：

mysql> SELECT count(gender) FROM stu\_info WHERE gender='boy';

2）字段值的数值比较

列出stu\_info表中年龄为21岁的学员记录：

mysql> SELECT \* FROM stu\_info WHERE age=21;

列出stu\_info表中年龄超过21岁的学员记录：

mysql> SELECT \* FROM stu\_info WHERE age>21;

列出stu\_info表中年龄大于或等于21岁的学员记录：

mysql> SELECT \* FROM stu\_info WHERE age>=21;

列出stu\_info表中年龄在20岁和24岁之间的学员记录：

mysql> SELECT \* FROM stu\_info WHERE age BETWEEN 20 and 24;

3）多个条件的组合

列出stu\_info表中年龄小于23岁的女学员记录：

mysql> SELECT \* FROM stu\_info WHERE age < 23 AND gender='girl';

列出stu\_info表中年龄小于23岁的学员，或者女学员的记录：

mysql> SELECT \* FROM stu\_info WHERE age < 23 OR gender='girl';

如果某个记录的姓名属于指定范围内的一个，则将其列出：

mysql> SELECT \* FROM stu\_info WHERE name IN

-> ('Jim','Tom','Mickey','Minnie');

4）使用SELECT做数学计算

计算1234与5678的和：

mysql> SELECT 1234+5678;

计算1234与5678的乘积：

mysql> SELECT 1234\*5678;

计算1.23456789除以3的结果：

mysql> SELECT 1.23456789/3;

输出stu\_info表各学员的姓名、15年后的年龄：

mysql> SELECT name,age+15 FROM stu\_info;

5）使用模糊查询，LIKE

以下划线 \_ 匹配单个字符，% 可匹配任意多个字符。

列出stu\_info表中姓名以“J”开头的学员记录：

mysql> SELECT \* FROM stu\_info WHERE name LIKE 'J%';

列出stu\_info表中姓名以“J”开头且只有3个字母的学员记录：

mysql> SELECT \* FROM stu\_info WHERE name LIKE 'J\_\_';

6）使用正则表达式，REGEXP

列出stu\_info表中姓名以“J”开头且以“y”结尾的学员记录：

mysql> SELECT \* FROM stu\_info WHERE name REGEXP '^J.\*y$';

效果等同于：

mysql> SELECT \* FROM stu\_info WHERE name Like 'J%y';

列出stu\_info表中姓名以“J”开头或者以“y”结尾的学员记录：

mysql> SELECT \* FROM stu\_info WHERE name REGEXP '^J|y$';

效果等同于：

mysql> SELECT \* FROM stu\_info WHERE name Like 'J%' OR name Like '%y';

7）按指定的字段排序，ORDER BY

列出stu\_info表的所有记录，按年龄排序：

mysql> SELECT \* FROM stu\_info ORDER BY age;

因默认为升序（Ascend）排列，所以上述操作等效于：

mysql> SELECT \* FROM stu\_info ORDER BY age ASC;

若要按降序（Descend）排列，则将ASC改为DESC即可：

mysql> SELECT \* FROM stu\_info ORDER BY age DESC;

8）限制查询结果的输出条数，LIMIT

查询stu\_info表的所有记录，只列出前3条：

mysql> SELECT \* FROM stu\_info LIMIT 3;

列出stu\_info表中年龄最大的3条学员记录：

mysql> SELECT \* FROM stu\_info GROUP BY age DESC LIMIT 3;

9）分组查询结果，GROUP BY

针对stu\_info表，按性别分组，分别统计出男、女学员的人数：

mysql> SELECT gender,count(gender) FROM stu\_info GROUP BY gender;

+--------+---------------+

| gender | count(gender) |

+--------+---------------+

| boy | 3 |

| girl | 2 |

+--------+---------------+

2 rows in set (0.00 sec)

列出查询字段时，可以通过AS关键字来指定显示别名，比如上述操作可改为：

mysql> SELECT gender AS '性别',count(gender) AS '人数'

-> FROM stu\_info GROUP BY gender;

+--------+--------+

| 性别 | 人数 |

+--------+--------+

| boy | 3 |

| girl | 2 |

+--------+--------+

**六 MySQL管理工具**

**6.1 问题**

部署LAMP+phpMyAdmin平台

**6.2 方案**

安装httpd、mysql、php-mysql及相关包

启动httpd服务程序

解压phpMyAdmin包，部署到网站目录

配置config.inc.php，指定MySQL主机地址

浏览器访问、登录使用

**6.3 步骤**

**步骤一：准备软件的运行环境 lamp**

[root@mysql6~]# rpm -q httpd php php-mysql #检测是否安装软件包

[root@mysql6~]# yum -y install httpd php php-mysql

[root@mysql6~]# systemctl start httpd #启动服务

[root@mysql6~]# systemctl enable httpd #设置开机自启

**步骤二：测试运行环境**

[root@mysql6~]# vim /var/www/html/test.php #编辑页面测试文件

[root@mysql6~]# cat /var/www/html/test.php #查看页面测试文件

[root@mysql6~]# yum -y install elinks #安装测试网页工具

[root@mysql6~]# elinks --dump http://localhost/test.php

Ok #验证测试页面成功

**步骤三：安装软件包**

1）物理机传输解压包给虚拟机192.168.4.6

[root@room9pc桌面]# scp phpMyAdmin-2.11.11-all-languages.tar.gz 192.168.4.6:/root/

2）虚拟机192.168.4.6解压phpMyAdmin-2.11.11-all-languages.tar.gz压缩包

[root@mysql6~]# tar -zxf phpMyAdmin-2.11.11-all-languages.tar.gz -C /var/www/html/ #-C 表示改变至目录

[root@mysql6~]# cd /var/www/html/

[root@mysql6~]# mv phpMyAdmin-2.11.11-all-languages phpmyadmin #改变目录名

[root@mysql6~]# chown -R apache:apache phpmyadmin/

#改变phpmyadmin目录权限

**步骤四：修改软件的配置文件定义管理的数据库服务器**

切换到部署后的phpmyadmin程序目录，拷贝配置文件，并修改配置以正确指定MySQL服务器的地址

[root@mysql6html]# cd phpmyadmin

[root@mysql6 phpmyadmin]# cp config.sample.inc.php config.inc.php #备份主配置文件

[root@mysql6 phpmyadmin]# vim config.inc.php #编辑主配置文件

17 $cfg['blowfish\_secret'] = 'plj123'; #给cookie做认证的值，可以随便填写

31 $cfg['Servers'][$i]['host'] = 'localhost'; #指定主机名，定义连接哪台服务器

**步骤五：在客户端访问软件 管理数据库服务器**

1）在客户端访问软件,打开浏览器输入http://192.168.4.6/phpmyadmin(数据库服务器地址) 访问软件，用户名是root，密码是123456