目录

[一．linux入门： 2](#_Toc791809)

[1.工作 2](#_Toc791810)

[2.学习方法 2](#_Toc791811)

[vi编辑器：开发步骤 3](#_Toc791812)

[3.linux文件目录 3](#_Toc791813)

[4.linux java环境 5](#_Toc791814)

[1).jdk 5](#_Toc791815)

[2).tar.gz的安装 6](#_Toc791816)

[5.shell 6](#_Toc791817)

[6.shell编程 7](#_Toc791818)

[7．Linux网络设置 7](#_Toc791819)

[8. 任务调度 8](#_Toc791820)

[9. Linux进程 9](#_Toc791821)

[10.linux 启动 10](#_Toc791822)

[11. 其他 10](#_Toc791823)

[12. linux下java网络编程案例 11](#_Toc791824)

[二．mysql入门 12](#_Toc791825)

[1.记录 12](#_Toc791826)

[3.数据库管理系统/服务器 12](#_Toc791827)

[4.十个基本语局 13](#_Toc791828)

[5.基础增删改查 13](#_Toc791829)

[6.建表 14](#_Toc791830)

[1.列类型 14](#_Toc791831)

[2.创建表 15](#_Toc791832)

[7.列的增删改 16](#_Toc791833)

[8.查 17](#_Toc791834)

[9.左连接，右链接，内链接 18](#_Toc791835)

[10.union 19](#_Toc791836)

[11.mysql函数 19](#_Toc791837)

[12.视图 //映射 21](#_Toc791838)

[13.utf8/gb2312 22](#_Toc791839)

[14.mysql编码 22](#_Toc791840)

[17.回顾 23](#_Toc791841)

[三.mysql进阶 23](#_Toc791842)

[变量定义 23](#_Toc791843)

[存储过程 24](#_Toc791844)

[触发器 25](#_Toc791845)

[锁 26](#_Toc791846)

[慢查询 26](#_Toc791847)

[索引 26](#_Toc791848)

[表 27](#_Toc791849)

[优化 28](#_Toc791850)

[权限管理 28](#_Toc791851)

[监控 29](#_Toc791852)

# 一．linux入门：

1.工作：1.Linux程序员c/c++，java，php，jsp（Java Server Pages）

-Linux软件工程师pc

-Linux嵌入式开发

2.linux系统管理员

2.学习方法：

第一阶段：linux平台上的开发，vi，gcc,gdb,make,jdk,tomcat,mysql...和Linux基本操作

第二阶段：加厚c功底《c专家编程》/java

第三阶段：unix《unix环境高级编程》

第四阶段：linux应用系统开发/Linux嵌入式开发

思考——实践——再思考——再实践...

先how then why

先框架 再细节

做了才会 “做中学”

适当的囫囵吞枣

琢磨别人怎么做，而不是我认为该怎么做

1973 unix: ibm-aix, sun-solaris, hp-hpunix, bsd,minix-->pc 1991->1994 linux(1.0)

linux :

缺点：相对麻烦

优点：1.free 2.多线程/多用户 3.安全性 4.对内存和文件管理优越

操作：关机： shutdown -h now

shutdown -r now

reboot

vi编辑器：开发步骤：

1.vi hello.java

2.输入 i 进入插入模式

3保存：输入esc 进入命令模式

输入' ：' wq #保存退出 q! 退出不保存

4. ls -l

5.编译java：javac hello.java

6运行 ：java Hello

linux 开发c/c++

./a.out(c程序)

<vi编辑器的使用>

3.linux文件目录：级层树状目录结构

最上层是根目录“/”,然后再目录下创建其他目录。

/：根目录

1.root目录

root用户的文件

2.home目录

普通用户的文件

3.bin

普通命令的目录

4.sbin

要有一定权限才能使用的命令

5.mnt

默认挂载软驱和光驱

6.etc

配置相关文件

7.var

经常变化的文件

8.boot

引导文件

显示当前目录：pwd

相对路径：以当前目录为起点，去寻找某个文件（夹）

绝对路径：从根目录去访问某个文件（夹）

工作目录：默认的根目录

用户管理：

添加用户命令： useradd xiaoming

设密码： passwd xiaoming

删除 userdel 用户名 （删除用户）

userdel -r 用户名（删除用户及主目录）

运行级别：

命令： init [0123456]

0：关机

1：单用户

2：多用户没有网络

3：多用户有网络

4：系统未使用保留给用户

5：图形界面

6：系统重启

3,5常用

/etc/inittab id：5：initdefault

解决修改错误配置的方法

进入grub引导界面时，输入e

选中第二行 输入e

再输入 1（单用户级别）

再去修改运行级别

pwd 显示当前工作目录

cd 该变目录

cd（只有cd） : 到用户的目录

ls

ls -a(隐藏文件)

ls -l（长列表文件）

ls -ahl 文件所在组

1.mkdir :建立目录

2.rmdir ：删除空目录

3.touch :建立空文件

4.cp : 复制命令

cp -rf 源 目标 （从 源 到 目标）

5.mv : 移动文件和改文件名

6.rm : 删除文件

rm -rf 目录名

7.ln ：建立符号链接

ln -s 源 目标 (目标->源)

8.more ：显示文件内容

9.less ：分页

10.grep ：文本中查找关键词

grep -n

11.| :管道命令 再linux和unix中（把上一个命令的结果交给| 后的命令处理）

12.man ：帮助

13.find：搜索文件及目录

find 目录 -amin -10 （十分钟内）

find 目录 -amin +10 （十分钟前）

a：内存 c：硬盘

min 分钟 time 小时

14.重定向： ls -l > a.txt(覆盖写)

>> (追加写)

组：一个用户可在多个组中

添加组：groupadd 组名

查看组的信息： cat(vi) /etc/group

创建用户，并同时指定该用户分配到组中： useradd -g 组名 用户名

查看所有用户信息 ： cat（vi） /etc/passwd

-rw-r--r--

- rw- r-- r--

-:(文件类型)

d：文件夹

- ：普通文件

rw- ： 文件所有者对该文件的权限

r--

文件所在组对该文件的权限

r--

其他用户对该文件的权限

r：可读 （4）

w ：可写（2）

x ：可执行（1）

修改文件文件访问权限：

chmod 777 用户名

7 代表 r w x xiaoming

改变组：

usermod -g 组名 用户名

## 4.linux 编程环境

1).jdk 1.把文件 挂载 到linux上

卸载：umount 目录

2.把安装文件拷贝到 /home

cp 文件 /home

3. cd /home

4. 安装

./ 文件名.bin

5.看一个文件 /etc/profile （环境配置文件）

2).tar.gz的安装：

tar -zxvf ？？？.tar.gz

启动：(进入图形界面 ：startx)

cd vscode

./vscode

{ myeclipse java ee weblogicwebsphere}

& ： 后台运行

安装realpaly

1.把安装文件拷贝到 /home

2.安装 ./ 文件名.bin

硬盘分区：

基本分区

扩展分区: 分成逻辑分区才能使用

一个硬盘中：扩展分区 + 基本分区 <= 4

linux ：

硬盘分区 ：挂载在boot目录上

fdisk -l 分区情况 df -l 磁盘使用情况

/dev/sda(loop)

sd:企业级 hd：普通用户

a：盘号

1：第一个分区 1-4 为主分区 5以后为逻辑分区

一个独立且唯一的文件系统 根目录“/”

5.shell: 《linux命令，编辑器和shell编程》

操作：

ctrl + w 清楚光标前的单词

ctrl + u 清空行

ctrl + c 终止当前命令

ctrl + t 调换光标前的两个单词

ctrl + r 搜索先前使用的命令

ctrl + d 退出shell

种类：

Bourne /bin/sh

C /bin/csh

Kom /bin/ksh

1.env 显示当前环境变量

2.chsh -s 输入新的shell(shell 的修改) EX： /bin/csh

3.命令补全

按两次tab键

4.历史纪录

history

history 5 最近5个命令

！5 执行历史编号为5的命令

5.

/etc/bashrc 基本配置数据

/etc/profile 系统环境变量

6.配置.bashrc文件让用户bai自启动

vi /home/bai/.bashrc

7.临时加入一个路径

export PATH=$PATH:/root

8.显示变量内容

echo

9.取别名

alias llh='ls -l /home' (等号前后无空格)

6.shell编程：

变量：

系统变量（环境变量） / 用户自定义的变量（本地变量，shell变量）

变量创建：语法：varName=varValue （‘=’左右不能用空格）

通配符：

1. \* 多个字母或数字

2. ? 一个字母或数字

7．Linux网络设置：

ipconfig（Windows）

ifconfig（linux/unix）

tracert 目标ip/域名 ：追踪路由命令

ping 目标ip/域名： 测试两个目标是否畅通

127. . . 回路

子网：

前面网络号一样 就在一个子网内

代理：

你懂的

linux网络环境配置：

1.

1.root登陆

setup

text mode setup utiliy

2. /etc/rc.d/init.d/network restart

2.

/etc/sysconfig/network-scripts/ifcfg-eth0

/etc/rc.d/init.d/network restart

RPM管理： 公认标准

apache-1.3.23-11.i386.rpm

apache:软件名称

1.3.23-11 :版本号，主版本次版本

i386：硬件

rpm -qa （查询安装了那些rpm软件）

安装rpm

rpm -i RPM全路径

rpm -ivh RPM全路径

samba:

1.安装

是否安装？ rpm -q samba

2.把自己的安装文件挂载在linux下

3.把 samba 拷贝到/home

cp /mnt/cdrom/samba\* /home

4.创建一个用户 youyou

useradd youyou

passwd youyou

5.youyou设置samba密码

cat /etc/passwd | mksmbpasswd.sh > /etc/samba/smbpasswd

smbpasswd youyou

6.使用

service smb start

service smb stop

service smb restart

8. 任务调度：系统在特定时间执行特定命令

crontab

1.设置任务

crontab -e

2.每隔一定时间执行

date > /home/mydata1

希望每天凌晨2：00执行data >> /home/mydata2

crontab -e 中加入

0 2 \* \* \* data >> /home/mydata2

希望每分钟

\* \* \* \* \* data >> /home/mydata2

\*（每小时第几分钟） \*（每日第几小时） \*（每月的几号） \*（每年的几月） \*（每周第几天）

data MMDDHHMMCCYY.SS 设时间

cal year

3.调动多个任务

1.在crontab -e 直接写

2.把所有任务写入一个可执行文件（shell）

4.终止任务

crontab -r

crontab -l

9. Linux进程：正在执行的程序（有独立的地址空间）

前台：屏幕上可以操作

后台：实际在操作，屏幕上看不到（常驻在内存中）

ps -a（u/x）

ps -aux

终止进程：

kill 进程号

kill -9 进程号（加强版）

killall：进程以及子进程都杀掉

动态监控：top

k：每隔多长时间刷新

u :监视

线程： 轻量级的进程

没有独立的地址空间

线程不能独立存在，线程由进程创建

相对讲：线程耗费的cpu，内存小于进程

监控网络状态

1.netstat -anp

2.traceroute

3.route

mysql：

1.安装：

2.解压缩

3.

groupadd mysql

useradd -g mysql mysql

进入mysql文件夹

scripts/mysql——install——db --user=mysql

chown -R root . -R(递归) . : 当前目录

chown -R mysql data

chgrp -R mysql .

bin/mysql\_safe --user=mysql & 【& 以后台方式启动】

4.如何进入

1. cd bin

./mysql -u root -p

2.

vi .bash\_profile

5.mysql使用：

进入mysql

creat database 名字

create table users(userId varchar(8),userName varchar(8));

结合Java

mysql驱动 放到jdk下 主目录/jre/lib/ext/

备份：

mysqldump -u root -p密码 数据库名 > 目录

恢复：(建立一个空数据库)

mysql -u root -p密码 数据库名 < 目录

ssh客户端(secure shell)

1.默认linux安装 sshd 监听22端口

2.windows安装

3.linux启动：

使用ssh登陆到linux

10.linux 启动：

1.BIOS自检

2.启动GRUB/LILO

3.运行Linux内核并检测硬件

4.运行系统第一个进程init

5.init 读取/etc/inittab 的信息

6./etc/rc.d/rc.sysinit 系统初始化脚本

7./etc/rc.d/rcX.d/[KS]\* - 根据运行级别X配置服务

终止“K”开头的服务

启动“S”开头的服务

8./etc/rc.d/rc.local 执行本地特殊配置

9.其他

11. 其他

zip和uzip：

zip aa.zip 文件名 //压缩单个

zip aa.zip 文件1 文件2 //压缩多个

zip -r aa.zip 文件夹路径 //压缩文件夹

uzip 压缩包名

gzip和ugzip处理.gz文件

unzip file.zip

linux安装：

1.独立安装

2.虚拟机安装

tcp（传输控制）/ip（网络）:一组协议

1.history 1972华盛顿会议，制定通用通讯协议

1974美国国防部共享

2. 7层协议（理论）

4层（现实）应用，传输，网络，链路

书籍：协议，实现，TCP事务协议

## 12. linux下java网络编程案例

//server

import java.io.\*;

import java.net.\*;

public class MyServeTest {

public static void main(String []args){

try{

ServerSocket ss = new ServerSocket(8888);

//监听，直到客户端链接 一旦链接成功，就得到Socket

Socket s = ss.accept();

InputStreamReader isr = new InputStreamReader(s.getInputStream());

BufferedReader br = new BufferedReader(isr);

String kk = br.readLine();

System.out.println(kk);

//从服务器端，回一个消息

OutputStreamWriter osw = new OutputStreamWriter(s.getOutputStream());

PrintWriter pw = new PrintWriter(osw,true);

pw.println("I get your message"+kk);

}catch (Exception e){

e.printStackTrace();

}

}

}

//client

import java.io.\*;

import java.net.\*;

public class MyClient {

public static void main(String []args){

try{

Socket s = new Socket("192.168,222,66",8888);

//通过文件流发送消息

OutputStreamWriter osw = new OutputStreamWriter(s.getOutputStream());

BufferedWriter bw = new BufferedWriter(osw);

PrintWriter pw = new PrintWriter(bw,true);

//发送

pw.println("hello world");

//读取从服务器回送的信息

InputStreamReader isr = new InputStreamReader(s.getInputStream());

BufferedReader br = new BufferedReader(isr);

String res = br.readLine();

System.out.println("server response is "+res);

}catch (Exception e){

e.printStackTrace();

}

}

}

# 二．mysql入门

## 1.记录

记录的都是信息

2.信息时代：

更高效记录信息的方式

## 3.数据库管理系统/服务器

[数据库！=mysql，满足国际sql标准(未必100%遵循)]

多个库

库里多个表

学习mysql的原因

1.mysql与linux紧密

2.开源，免费

3.postgresql

4.叛本，不追求最新版本 {开源软件 有beta版，stable版，尽量用stable版}

mysql命令行与mysql服务器什么关系？

客户端 client --》 服务端 server的关系

其他mysql客户端：

navcat， mysqlfront

## 4.十个基本语局

0.连服务器 mysql -u root -p

(首先面对多个库)

1.查看所有库

show databases; (记得打';'号)

2.创建库

create database 数据库名 chatset(字符集)；

3.删除库

drop database 库名；

4.选库：

use 库名; (选完库后，面对表)

5.查看表（先选择库）

show tables；

6.简单的建表

create table stu(num int, name varchar(10))engine myisam charset utf8;

//engine 引擎与性能相关

7.删除表

drop table 表名；

8.表改名 表/列可以改名， database不能改名

rename table oldname to newname;

9.插入数据

insert into stu values (1,'zhangsan'),(2,'lisi'),(3,'wangwu');

select \* from 表名;

10.清空表数据

truncate 表名;

truncate 相当于删表再重建一个空的表

delete 删除所有数据行

全清空truncate更快点

5.基础增删改查(针对单个table)

tee 目录名，把sql及结果都输出到一个sql文件里

1.乱码？客户端与数据库字符集类型不同

set names 字符集名;

desc 表名；//查看表的信息

3.增删

增：

insert into 表名 (表头列表) values (值列表), (list2); //注意不要忘了写表名

添列：alter table 表名 add 列名 数据类型;

EX. insert into class values (1,'张三','男','百度', 8888.88, 200);

删：删除整行，不存在一行的某几列

delete from 表名 where expr;

4.改查

改：

update 表名 set 表头= ? where expression;

EX: update 表名 set name = '' where id = ?(表达式);

where后表达式:

为真 则该行发挥作用

(改哪张表 哪几列 哪一行生效 改成什么值)

查：

select 列名list（可倒序） from 表名 where expr;

暴力查找： select \* from 表名;//不要随便用

导入数据库：source 目录

6.建表 (声明字段的过程)：

1.列类型：

#### 1.数值型

整型 tinyint/smallint/mediuint/int/bigint

tinyint

{占据空间： 1字节(8个位)

范围 -128~127，0~255(unsigned)}

参数：(M),unsigned,zerofill

默认有符号

unsigned 无符号

M：宽度

zerofill 不够位数，0填充 （必是unsigned类型）

M必须与zerofill配合才有意义

浮点型

float(M，D) 10^38 ~ 10^-38 （M<=24 4个字节 M>24 8个字节）

decimal(M,D) 整数部分和小数部分分开存储，比float精确

M: 小数总位数（不包括小数点）精度

D: 小数点后位数 标度

#### 2.字符型

char ，varchar， text， blob

char(n): 定长字符串

不够n，在尾部用空格补齐

(都是定长，查找时，可用行数计算出偏移数)

varchar(n):

前面有长度标识符，不用空格补齐，但列前有标注占1~2字节

注意：char(M)， varchar(M) 限制的是字符数，不是字节数

text ：文本，存储文章，新闻内容

不必给默认值

blob：二进制类型，存储图像，音频等信息

意义： 防止因字符集问题导致信息丢失

#### 3.日期时间

date 日期 1000-01-01 到 9999-12-31

time 时间 00:00:00

datetime YYYY-MM-dd HH:mm:ss

year 1901-2155 (特殊：0000)

timestamp 当前时间 default CURRENT\_TIMESTAMP

#### 4.null类型 //空

Q：为什么建表时，default ''/0 not null？

A: 不想出现null

Q：为什么不想要null？

A： 1.比较要用特殊运算符 is null,is not null 碰到运算符，都返回null

2.效率不高，影响索引效率

因此建表时 default ''/0 not null

## 2.创建表

声明列过程

create table 表名(

列1名 列类型 列1参数，

列2名 列类型 列2参数，

......

)engine myisam/innodb charset utf8/gbk;

[把频繁用到的，考虑效率存到一张表

不常用和交占据空间的，存到另一张表中]

EX1. create table class(

id int auto\_increment not null primary key ,

name varchar(10) default '' not null,

gender char(1) default '' not null,

company varchar(20) default '' not null,

salary decinal(6,2) default 0.00,

fanbu snallint default 0

)engine myisam charset utf8;

EX2. create table website(

id int unsigned auto\_increment not null primary key,

name varchar(15) default '' not null,

gender char(1) not null,

weight tinyint unsigned default 0 not null,

birth date not null,

salary decimal(8,2) default 0.0 not null,

lastlogin datestamp default CURRENT\_TIMESTAMP not null,

intro varchar(100) deafault '' not null

)engine myisam charset utf8;

name和intro之外，其他都是定长的

为提高查询速度，让所有列都定长

优化:

create table website(

id int unsigned auto\_increment not null primary key,

name char(15) default '' not null,

intro varchar(100) deafault '' not null

)engine myisam charset utf8;

create table website(

id int unsigned auto\_increment not null primary key,

gender char(1) not null,

weight tinyint unsigned default 0 not null,

birth date default '1000-01-01'not null,

salary decimal(8,2) default 0.0 not null,

lastlogin int

)engine myisam charset utf8;

name 浪费部分空间，但提高速度 值

intro 简介 一旦注册改的频率不高

把频繁用到的，考虑效率存到一张表

不常用和交占据空间的，存到另一张表中

## 7.列的增删改

1.增 alter table 表名 add 列名称 列类型 列参数;

alter table 表名 add 列名称 列类型 列参数 after 某列;

//新建一个列，并在最前面 alter table 表名 add 列名称 列类型 列参数 first;

2.删 alter table 表名 drop 列名;

3.修改列类型

alter table 表名 modify 列名 新列类型 新列参数;

（不能该改列名）

4.修改列名及列类型

alter table 表名 change 旧列名 新列名 新列类型 新列参数;

//不匹配数据会丢失或修改不了

## 8.查

where 条件查询(针对 表 发挥作用)

运算符：

in 在某集合内

between 在某区间内

!= 或 <> 不等于

() : 集合

between ...and...

模糊查询

like

%:通配符

\_:匹配任意单一字符

查询模型：

列看成变量，where看成 if 表达式

列之间就能运算，术语 广义投影

起 列别名 (...) as ....

EX.where发挥作用时，表上并没有discount列，发挥完作用，形成的结果里才能discount

substring(),concact()

group by 分组与统计函数

统计函数: max，min,sum,avg,

count(\*)，count(1)//绝对行数

count(列名) //该列不为null的行数

即count(0/false)也会计数

//count(\*),count(1) 谁好呢？

对于myisam引擎，没有区别

对于innodb的表，count(\*)效率低

group by 分组查询

group by 列名;

//严格的讲，select a,b,c group 后 从a b c中选择

having 筛选

筛选得到结果集

order by 排序

对结果集排序

desc：降序排列

默认：升序

多个字段用 ',' 隔开

EX.列1 desc/asc，列2 desc/asc ....

//内层order by 能影响结果集才有意义

limit 限制结果条数

[offset],N

offset:默认为0 //跳过行数

N：取出行数

查询顺序：where > group by > having > order by > limit

// show create table 表名;

// 批量导入：insert 库名.表1名 select 表头列表 from 库名2.表名2;

子查询:

where

内层查询结果作为外层查询的比较条件

如果 where 列 = (内层 sql)，则内层sql返回的必是单行单列，单个值

如果 where 列 in (内层sql)，则内层sql只返回单列，可以多行

select id,name from good where id in (select max(ca\_id) from good order by id);

from

即内层sql的查询结果，当成一张临时表

exist

外层sql的结果，拿到内层sql测试，如果为真，取出该列

select id,name from good where exists (select \* from cat where cat.id =good.id);

## 9.左连接，右链接，内链接

1.表与集合的关系

一张表就是一个集合

每一行就是一个元素

Q：两行数据一样

A：mysql中有rowid区分

数据库中的笛卡尔乘积：

select \* from test0，test1;

//列名重复 用表名.列名区分

//没有利用索引

//效率低

2.左连接

语法: 假设A表不动，B表在A表右边滑动

A left join B on 条件

//条件为真，B表对应行取出

//形成一个结果集，可以看成一张表 A，B的列都可以查

EX. select goods\_id,cat\_id from good left join cat on good.cat\_id = cat.cat\_id;

//注意服务器内存占用

3.右链接

与左连接相反

4.内链接

左连接与右链接的交集 ∩

语法：A inner join B on 条件

5.外连接

左连接与右链接的并集 ∪

//mysql不支持

Q:左连接，右链接，内链接的区别

Answer: A left join B 等价于 B right join A

//左右链接可以互换，尽量用左链接，出于移植兼容性的考虑

## 10.union

合并多条sql语句的结果

语法： sql1 union sql2;

//unoin合并的是结果集，不区分来自哪张表

//要求：多张结果集的列数相同

//列名不一致，列名以第一条sql的列名为准

Q.union后结果重复

A. 比较常见，默认去重

union all 不去重

## 11.mysql函数

1.数学函数

abs(x)//绝对值

bin(x)//二进制

floor(x)//对小数取整

ceiling(x)//向上取整

rand()//随机0-1.0的的随机值

2.聚合函数

group\_concat(col,x)

//x为分割符 默认为',' 把col拼接到一起

3.字符串函数

ascii(char)//计算字符的ascii码

bit\_length(str)//字符串的比特长度

char\_length(str)//字符数

lenth(str)//字符串的字节长度

reverse(x)//反转字符串

locate(substr, str)，position(substr in str)//substr的第一次出现位置（从1开始数）

right(str,N) //从右数 截N个字符

4.日期时间函数

now()//当前时间datetime格式

curdate() //当前日期

curtime()//当前时间

dayofweek(date)//date在一星期的第几天

dayofmonth(date)，dayofyear(date)

week(date)//date在某年的第几个星期

month(date),

5.控制流函数

case 值 when 可能1 then 返回值1

when 可能2 then 返回值2

else 默认值 end

if(exp1,exp2,exp3) //exp1为真 返回exp2，反之返回exp3 三元运算符

if exp1 else exp2

语法： if condition1 then

exp1

elseif ...

else ...

end if

ifnull(exp1,exp2) // exp1为null时，返回exp2，反之返回exp1

//在存储过程中，加分隔符';'

6.加密函数

md5://不可逆

md5(str)

良好的加密：

1.不可逆

2.碰撞性低

7.系统调试函数

user()//返回用户及所在主机

database()//正在操作库名

version()//mysql版本

//注意事项

1.以合理的表结构减少函数的使用

EX. email 以@前后区分

2.使用时：

比如时间的格式化

优先在业务逻辑层处理

3.查询时使用函数的弊端

date\_format(A)

A列索引无法使用

即对某列使用函数，则该列索引无法使用，影响速度

8.自定义函数

1.查看能不能创建函数

show variables like '%fun%';

开启：

set global log\_bin\_trust\_function\_creators = 1;

2.创建语法

create function 函数名(变量1 类型1, 变量2 类型2...) returns 数据类型

begin

....

return 数据;

end;

3.删除

drop function if exists 函数名;

## 12.视图 //映射

视图view：

可以看成一张虚拟的表，是表通过某种运算的到的一个投影

语法：

create view 视图名 as select 语句;

用处：

1.简化查询

2.更精细的权限控制

3.数据多，分表是可以用到

表与视图数据相互影响的关系

Q.视图可更新？

A.有些可更新 //可更新要求：视图数据和表的数据一一对应

order by limit的结果不是一一对应的

一一对应：def：表中的行只能计算出视图中确定的一行

algorithm

三种模式：merge(合并) temptable(临时表) undefined(系统决定)

决定建临时表还是合并语句

//一些简单视图 并没有建临时表，而只是把条件存起来，下次来查询，把条件合并去查表

语法： create algorithm = merge/temptable(不可更新)/undefined view 视图名 as 语句;

管理：

存放位置： information\_schema 数据库下的views表里 //G为以列表现

with check option

更新视图的数据， 必须先满足视图条件，之后才能更新基表

## 13.utf8/gb2312

计算机： 0/1 人的世界：文字/图片/声音

字符集：2进制编码 到 字符的映射

acsii字符集 0xxx xxxx

GB2312

如何兼任ASCII又能双字节表示中文

干脆gb2312不占用0-127

但中文组合数少 1w+

事实上只有6763个汉字

GBK

第二位 小于128的也能使用

//兼容性问题

asci ：本地字符集

unicode:世界通用码表

用4个字节编号

常用的集中在前655355个字符中

只负责分配编号，而且都用4个字节分配编号

在不改变编号的基础上 简化字节

unicode transformation

一种压缩方式

utf8：网络传输数据方式

变长码 1-6个字节

unicode-- utf8

原文件-- 压缩文件

Q.如何确定字符的边界

A.按头部1的数目

Q.如何截取utf8无乱码

A.从头开始取一个字节，计算连续1的个数

容量上：gb2312 < gbk < utf8

乱码如何形成

1.转换出错（转码时信息丢失 不可修复）

2.解码时与实际编码不一致(可修复)

## 14.mysql编码

客户端-->连接器-->数据库server-->连接器-->输出

客户端发送的编码 set character\_set\_client = 编码方式;

连接器使用的编码 set character\_set\_connection = 编码方式;

返回数据的编码 set character\_set\_results = 编码方式;

三者相同时 简写 set names 编码方式;

编码大小:server>=connection>=client

不乱码：

1.正确指定客户端编码

2.合理选择连接器编码

3.正确指定返回内容编码

utf8的bom问题

windows记事本创建时，前面多了三个字节 来辨识编码信息

解决：保存为 utf8 而不是 utf8+

utf8，utf-8，UTF-8//一样的，别名而已

面试题：中文截取无乱码

如何判断一个utf8字符的字节数

字符转换为二进制，然后判断

16.存储引擎

myisam/innodb/bdb/memory/archive

mysql存储数据的不同方式

事务的原子性

start tansaction;

...

commit;

//有一个故障 就会回滚 rollback;

一致性：

操作前后值的变化，逻辑上成立

隔离性：

事务结束前，每一步操作带来的影响，别人看不见

持久性：

一旦完成，无法撤销

myisam引擎：隔离性没有完成

## 17.回顾

站在数据库使用者

学习mysql，增删改查，左右链接，子查询

数据库设计者：项目经理，架构师，

知识：引擎，索引，表的设计，读写分离

数据库管理者DBA：

权限管理，数据备份，运行监控，性能检测

# 三.mysql进阶

## 变量定义

会话变量：

属于某一客户端的变量（不同客户端之间相互独立）

show session variables;

set 变量名='';

set @@session.变量名='...';

全局变量：

针对数据库

show global variables;

set global 变量名='....';

set @@global.变量名='''';

select @@global.变量名;

## 存储过程

多个sql语局放在一起执行

优点

灵活/快/减少网络流量/安全机制

存储过程的创建：

1.选中数据库

2.改变分隔符

不让';'作为执行结束的标记

delimiter $$;

3.创建

create procedure 过程名()

begin

.....; //以;结尾

end

$$; //drop procedure 过程名;

4.恢复分割符

恢复delimiter ;

5.调用

call 过程名

局部变量的定义

declare 变量名 数据类型 default 默认值;

参数：

in：输入参数

在存储过程中修改的值，不能影响到存储过程外原有变量

out：

在存储过程中修改的值，能返回

在外赋值一直为null(不认可之前的赋值)

inout：

既能赋初值

又能返回

EX.//create procedure test\_0(in 变量名 类型)

begin

expr

end

分隔符

//set @p=10;

//call test\_0(@p);

流程控制：

while 语法：

while expr do

语句

end while;

repeat 语法:

repeat

内容

until 条件

end repeat;

loop 语法：

名字: loop

内容

if expr then

leave 名字;

end if;

end loop;

定义条件和处理：

//类似异常

语法： declare continue handler for sqlstate '错误代码值'

set 变量=变量值;

存储过程管理

show procedure status where db='数据库名';

当前存储过程的列表：

select specific\_name from mysql.proc;

查看存储过程的内容

select specific\_name,body from mysql.proc;

或 show create procedure 存储过程名;

删除

drop procedure if exists 过程名;

修改：

alter procedure 存储过程名 参数;

## 触发器

一种特殊的存储过程

监视地点：一般是表名

监视事件：update/delete/insert

触发时间：after/defore

触发事件：update/delete/insert

//不能直接调用,满足条件自动执行

创建语法：

create trigger trigger\_name trigger\_time trigger\_event

on table\_name for each row 内容;

update: 修改前的数据用old来表示 old.列名引用被修改之前行中的值

修改后，的数据用new来表示 new.列名引用被修改之后行中的值

管理：

show triggers;

desc information\_schema.TRIGGERS;

drop trigger 表名.触发器名;

## 锁

处理稀缺资源 //对有限资源进行保护，解决隔离和并发的矛盾

表级锁/行级锁/页面锁

myisam

共享读锁：//都能读

lock table 表名 read;//lock tables

unlock table 表名 read;//unlock tables

独占写锁：//只有一个能读

lock table 表名 write;//lock tables

unlock table 表名 write;//unlock tables

并发插入

concurrent\_insert = 0(不允许),1(默认),2(在表尾插入)

## 慢查询

超过指定时间的SQL语句查询称为'慢查询'

show variables like '%long%';

long\_query\_time

status：

show status like '%uptime%';

show status like 'com\_Select';

show status like 'connections';

启动慢查询

安全模式

## 索引

在关系型数据库中，建立在表中的一种数据结构，使表中sql语句执行更快

查看索引

show keys from 表名;

show index from 表名;

分类

主键索引

唯一索引

所在的列不能为空字符串，可以为null

create unique index 索引名称 on 表名;

普通索引

create index 索引名称 on 表名;

全文索引 //只有myisam支持

explain分析sql的执行计划

explain 语句;

原理：二叉树

分析 profile

select @@have\_profiling;

set profiling = 1;

show profile for query queryID;

## 表

分析表

analyze table 表名;

检查表

check table 表名;

优化表

optimize table 表名;

//只对myisam，innoDB，BDB起作用

//在执行时都对表进行锁定

//show table status 获取表的信息

分区 //数据量过大时，分划到几张表中

range分区

创建

语法：

create table 表名(

...

)

partition by range(列名)(

partition p0 values less than(.),

partition p1 values less than(..),

partition p2 values less than(....),

...

partition pn values less than MAXVALUE

);//主键必包含在分区中

list分区

枚举型

partition by list(列名)(

partition p0 values in (....),

partition p1 values in (...),

partition p2 values in (..),

...

);

hash分区

partition by hash(列名)

partiton 数量;

keys分区

管理：

删除/增加

## 优化

内存

myisam

使用key\_buffer缓存索引块

每个session独有的

read\_buffer\_size

read\_rnd\_buffer\_size 影响 order by语句

innodb

用内存区做io缓存池，缓存索引块和数据块

并发相关

max\_connections

thread\_cache\_size

应用

1.访问时采用连接池

2.采用缓存减少对mysql的访问

3.负载均衡

主服务器(写)与从服务器(读) 分离

## 权限管理

use mysql;

select host,user from user;

show grants for 'root@localhost';

账号名@主机名

//主机名 ip地址/localhost

赋权限

grant select on \*.\* to test@localhost indentified by 'root' with grant option;

监控：

自己写程序/采用商业解决方案/开源软件

定时维护

mysql定时器 5.1之后版本

show variables like '%sche%';

set global event\_scheduler = 1;

创建

create event 名字 on schedule every 大小 second（时间单位） on completion preserve disable

do call 存储过程名;

开启（关闭） ： alter event 定时器名 on completion preserve enable（disable）;

备份还原

mysqldump:

原理

1.查询出表的结构

2.创建表

3.insert语句，插入数据

语法：

mysqldump -u root -p 数据库名 table1，table2 > 目录/文件名.SQL //在命令行下执行

mysqldump -u root -p 数据库名 > 目录/文件名.SQL //备份所有表

mysqldump -u root -p --all-databases > 目录/文件名.SQL //备份所有数据库

还原：

mysql -u root -p 数据库名 < 目录/名字.SQL //版本号相同

# N．附录

准备添加的内容：shell编程+PGSQL