阅读记录

LINUX 一些命令

# 初始信息

虚拟机软件：VMware Server

root 用户　根

sudo -s 可以将自己提升为root用户　　　　　　　exit 回到先前用户状态 $转＃

sudo apt-get update 更新当前系统软件包信息

sudo gedit /etc/default/grub 查看修改参数 sudo update-grub

Linux的命令和文件是区分大小写的

date cal df free 显示时间 日历 显示磁盘剩余空间的数量 显示空闲内存

exit 离开会话框

# 基础篇

sudo dpkg -i linux-------.deb　安装

cd /etc 　进入目录　　　ls

cat fstab 查看fstab文件　Tab键会将文件名补全　Tab键两次展示目录下多个文件

×　？ [ ]　×匹配文件名中任意长度的字符串 ls \*.cpp 以cpp结尾的文件

？匹配文件名以text开头跟一个字符的文件　ls text?

[ ]匹配所有出现在方括号内的字符　　　　ls text[1A]

以’.’开头的文件名是隐藏文件 【ctrl+h显示隐藏文件】

虽然 Linux 支持长文件名,文件名可能包含空格,标点符号,但标点符号仅限使用“.”,“-”,下划线。最重要的是,不要在文件名中使用空格。如果你想表示词与词间的空格,用下划线字符来代替。

-表示范围

pwd 显示当前所在位置

cd /usr/bin

cd .. cd ../..　　cd 或cd ～ ..上一级目录　．当前目录

ls -F 在每个目录后加上/,在可执行后加×，链接文件加＠

ls -a 显示隐含的文件

ls～/usr 列出目录及存在的文件名

dir /etc/init.d/ 和ls 功能差不多

vdir=ls -l 列出目录和文件的完整信息

cat 查看文件内容　　　可以多个文件名　　　也可使用通配符

cat -n stack.h 在stack.h文件内容每一行前显示行号

使用more 命令显示长文件内容　一页一页的显示　空格键向下翻一页，enter键向下一行　q离开

head和tail 显示文件的开头结尾　　　还可以指定行　　　head -n 2 day weather

less　文件名　功能会像一个文本编辑器 浏览文件内容

page up 或page down G或g n h q 【向前或向后翻一页， 到首行或最后一行 查找下一个指定的字符 显示帮助屏幕 离开less程序】 less与more一样

file 确定文件类型

netstat -antp | grep 9000 查看端口号

netstat -aux|grep python

grep 寻找文件中信息 grep up day weather grep ‘red hat’ 文件　单引号将空格包进去

find /etc/ -name zip -print　路径，文件名，显示到标准输出

查找文件　locate 查找整个文件系统

查找特定程序whereis whereis -b find 只查找这个程序二进制可执行文件

who whomi uname -a uname -r

sudo groupadd workgroup 新建工作用户组

sudo useradd -G workgroup lucy 添加用户

sudo password lucy

sudo -s 切换root用户

root用户改变文件的读写执行权限

sudo chgrp workgroup work/ 给用户组所有权

sudo chomd g+rwx work/ 给用户添加权限

sudo chmod o-rwx work/　　　撤销用户权限

mkdir document picture mkdir ~/document/temp 主目录下建立

mkdir -p ~/tempx/job 建立完整目录

touch hello 建立文件夹

mv hello test/ 将hello文件移动到test下　　　-i 选择是否覆盖

cp 复制 -a -r -u(只复制目标文件中不存在的文件)

rmdir 只能删除空目录　　　　rm 一次性删除　　－i

sudo chown guest days 更改文件days的属主guest

符号链接（软连接）in file link ln -s days mydays in -s item link 【创建硬链接】

type 说明怎样解释一个命令 显示命令类型

type command

which 显示会执行哪个可执行程序

Which cd

man 显示命令手册

apropis 显示一系列适合的命令

info 显示命令info

whatis 显示一个命令的简介描述

alias 创建命令别名 label（as）

重定向标准输出> ls -l /usr/bin >-output.txt

进程：

ps aux 显示进程详情 nohup 环境路径 进程路径+进程名 & （在后台执行）

& 使进程在后台进行 fg %i （i为进程id）将进程返回到前台执行

kill killall

ping linuxcommand.org 尝试连接网站

tracepath/traceroute slashdot.org 显示连接经过的路由

locate搜索包含输入名所在路径

find 查找文件 wc查看数量

# 软件包安装

安装chrome for linux

sudo dpkg -i google-chrome-satble\_current\_i386.deb

卸载 dpkg -l | grep chrome

sudo apt-get update 更新软件包信息

sudo apt-get （install upgrade remove source clean） apt-get -h 展示完整用法

Ape-cache search (flight) 查询 搜索 依赖 显示show pkgnames列出名字

sudo apt upgrade // apt list // install // show // remove //

sudo apt autoremove 清理不再使用的依赖和库文件

tar zxvf ().tar.gz 解压缩

ext4fs 增加日志功能 保存日志文件利用其进行错误恢复

sudo mount dizhi1 dizhi2 挂载光盘

Ubuntu 使用uuid来标识文件系统，即通用的唯一标识符，是一个128位比特的数字，由系统自动生成和管理

df 查看文件系统 df -t (文件) 显示特定的文件

sudo fsck 检查修复文件系

gzip 压缩文件 bzip2

dump -0u -f 命令使用”备份级别”来实现增量备份

restore 配套使用进行恢复

cron 进行定时备份

升级root用户状态 sudo -s su 切换 exit

ps aux ps -ef | grep ps lax

kill -9 pid

共享文件 cd到要共享文件的目录下， python -m simplehttpserver 8888

nohup （） & 启动进程

scp (地址1) （地址2） 从地址1复制文件到地址2

ssh -l name ip 登录

yum提供了查找、安装、删除某一个、一组甚至全部软件的命令

1. 列出所有可更新的软件清单命令：yum check-update
2. 更新所有软件命令：yum update
3. 仅更新/安装指定的软件命令：yum update/install <package\_name>
4. 列出所有可安裝的软件清单命令：yum list
5. 删除软件包命令：yum remove <package\_name>
6. 查找软件包命令：yum search <keyword>
7. yum clean packages: 清除缓存目录下的软件包

yum clean headers: 清除缓存目录下的 headers

yum clean oldheaders: 清除缓存目录下旧的 headers

yum clean, yum clean all (= yum clean packages; yum clean oldheaders) :清除缓存目录下的软件包及旧的 headers

# Vim 的使用

命令模式()、输入模式(i)、底线命令模式（:）

i进行插入文件内容 Esc q！（强制退出不保存） / (搜索) ：

Ctrl+f 向前一页 Ctrl+b 向后一页 x删除光标所在位置字符 dd删除光标所在行

U撤销 ctrl+R 重做被撤销的操作 ：q退出vim q！清醒退出不保存

有字符串的 即有正则表达式 的

正则表达式

egrep “^a.\*t$” 文件夹 列出以a开头以t结尾 统计数量 | wc -w

. 用于匹配换行符之外的任意字符 × 重复0次或者多次 +一次或多次 ？0次或者一次 {m}重复m次 {m，n} 重复m到n次 {，n}最多n次 {m，}最少m次

1. z]取其中一个小写字符 [a-zA-Z] [0-9]

^[[:upper:]t$] 以一个大写字符开头



[^aeiou] 取反义除了aeiou之外的字母

^ht$ ^h|t$

5.Shell 编程

执行脚本 #！ /bin/bash echo打印 加执行权限chmod+x（）

source varible 影响其父shell脚本 export count=5 影响其子shell脚本

传递参数$0 存放脚本的名字

$\*包含参数列表 $@包含参数列表 $#包含参数的个数

for file in $@

do

ls -l $file

done

while循环用于不断执行一系列命令，也用于从输入文件中读取数据；命令通常为测试条件。其格式为：

while condition

do

command

done

在sh/bash里可不能这么写，如果else分支没有语句执行，就不要写这个else。

until 循环执行一系列命令直至条件为 true 时停止。

until 循环与 while 循环在处理方式上刚好相反。

一般 while 循环优于 until 循环，在某些时候也只是极少数情况下，until 循环更加有用。

until 语法格式:

until condition

do

command

done

condition 一般为条件表达式，如果返回值为 false，则继续执行循环体内的语句，否则跳出循环。

case

case ... esac

case ... esac 与其他语言中的 switch ... case 语句类似，是一种多分枝选择结构，每个 case 分支用右圆括号开始，用两个分号 ;; 表示 break，即执行结束，跳出整个 case ... esac 语句，esac（就是 case 反过来）作为结束标记。

Shell case语句为多选择语句。可以用case语句匹配一个值与一个模式，如果匹配成功，执行相匹配的命令。case语句格式如下：

case值 in

模式1）

Command1

Command2

；；

模式2）

Command1

Command2

；；

esac

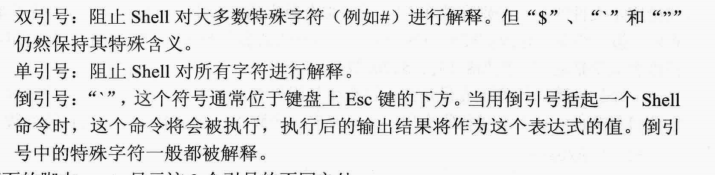
case工作方式如上所示。取值后面必须为单词in，每一模式必须以右括号结束。取值可以为变量或常数。匹配发现取值符合某一模式后，其间所有命令开始执行直至 ;;。

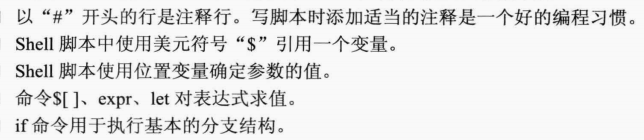
取值将检测匹配的每一个模式。一旦模式匹配，则执行完匹配模式相应命令后不再继续其他模式。如果无一匹配模式，使用星号 \* 捕获该值，再执行后面的命令。

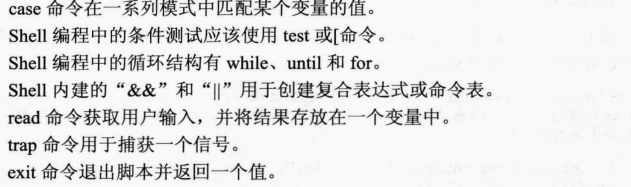
有时候需要在未达到循环结束条件时强制跳出循环，Shell使用两个命令来实现该功能：break和continue。

break命令允许跳出所有循环（终止执行后面的所有循环）。

continue命令与break命令类似，只有一点差别，它不会跳出所有循环，仅仅跳出当前循环。







readonly （命令） 将命令定义为只读变量，不能被改变

upset （命令）删除变量

shell 字符串可以使用单引号、双引号、拼接字符串

单引号字符串的限制：

单引号里的任何字符都会原样输出，单引号字符串中的变量是无效的；

单引号字串中不能出现单独一个的单引号（对单引号使用转义符后也不行），但可成对出现，作为字符串拼接使用。

双引号：

双引号里可以有变量

双引号里可以出现转义字符

获取字符串的长度：

string=”abcd”

echo ${#string} #输出为4

awk expr做基本运算 expr最常用 支持（算术运算符、关系运算符、布尔运算符、字符串运算符、文件测试运算符）

算数【+ - \* / % = == !=】 \*前边必须加（\）才能实现乘法运算

关系运算 【-eq 相等（返回True） -ne不相等（返回True） -gt/-lt 右边大于或小于左边（返回True） -ge/-le 左边大于或小于右边（返回True） 】

布尔运算 【！ 非运算 -o 或运算 -a 与运算 】

逻辑运算 【&& and || or】

字符串 【= ！= -z（字符串是否为0） -n（检查长度是否不为0） $(检查字符串为空)】

与echo一样的作用的 printf



Shell中的 test 命令用于检查某个条件是否成立，它可以进行数值、字符和文件三个方面的测

Git Magic

保存状态：

$ git init

$ git add .

$ git commit -m ‘menage’

恢复之前

$ git reset --hard

再次保存

$ git commit -a -m ”menage”

添加新文件

$ git add readme.txt Documentation

忘记某些文件

$ git rm ....h ....c

$ git rm -r 文件夹

重命名文件如添加新文件一样，先进行删除在进行添加文件

$ git mv bug.h feature.c

回滚某个时间点之后的所有操作

$ git log [显示了最近提交的列表，及他们的hash值] hash值前几个字符亦可以确定一个提交

$ git reset --hard 766f(hash的前几个字符)

切换分支时的检查

$ git checkout master [为避免git的抱怨，应该在每次运行checkout之前提 交（commit）或重置（reset）你的改动.]

选择只恢复特定文件，将其加载在命令后面

$ git checkout 82f5 some.file another.file 【在运行前最好git commit -a】

$ git checkout master~5 【回到倒数第五个保存状态】

撤销操作

$ git commit -a

$ git revert 1b6d

生成变更日志

$ git log >Changelog

下载一个git项目

$ git clone ................git(地址)

$ git pull 【更新项目到最新】

对比文件的差异

$ git diff 文件1 文件2

$ git diff “@{yesterday}” 【昨天的改变】

一个特定版本与倒数第二个变更之间

$ git diff 1b6d “master~2”

$ git whatchanged --since=”2 weeks ago”

$ git branch 【列出所有分支】

数据库sql

create table j建表使用字段类型，长度、是否为空、设置自动递增

cursor=db.cursor创建游标

cursor().fetchone fetchall rowcount（记录受影响行数）

execute()执行原声sql命令语句

db.close

try:

#执行sql语句

cursor.excute(sql)

#提交到数据库执行

db.commit

except:

#发生错误则回滚

db.rollback

**执行事务**

事务机制可以确保数据一致性。

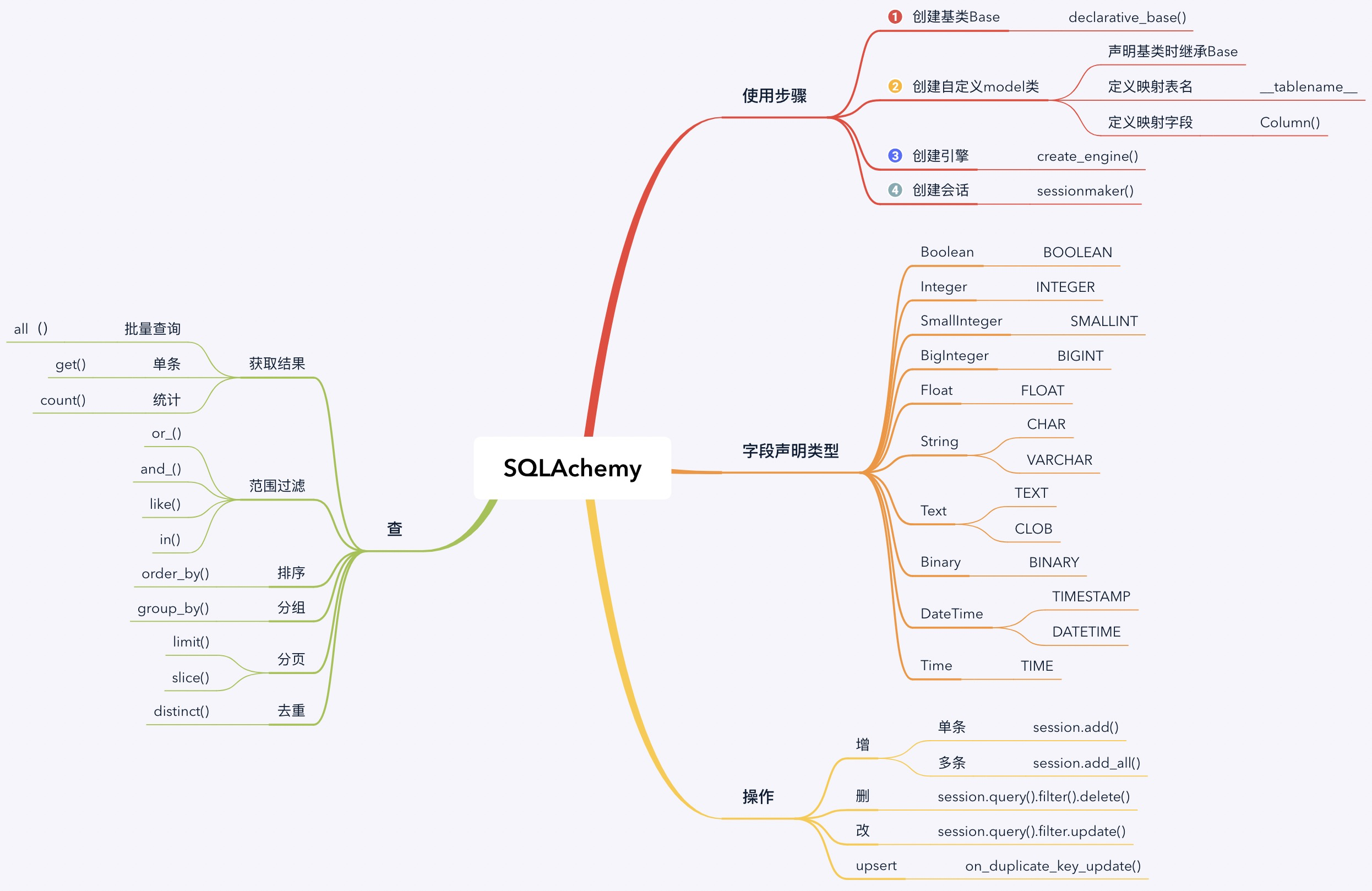
事务应该具有4个属性：原子性、一致性、隔离性、持久性。这四个属性通常称为ACID特性。

原子性（atomicity）。一个事务是一个不可分割的工作单位，事务中包括的诸操作要么都做，要么都不做。

一致性（consistency）。事务必须是使数据库从一个一致性状态变到另一个一致性状态。一致性与原子性是密切相关的。

隔离性（isolation）。一个事务的执行不能被其他事务干扰。即一个事务内部的操作及使用的数据对并发的其他事务是隔离的，并发执行的各个事务之间不能互相干扰。

持久性（durability）。持续性也称永久性（permanence），指一个事务一旦提交，它对数据库中数据的改变就应该是永久性的。接下来的其他操作或故障不应该对其有任何影响。



update

on\_duplicate\_key\_update()-不存在则插入，存在则更新

session.new的案例，主要作用：打印出已增加的数据，但是未提交至数据库的数据

session.dirty案例，主要作用：打印出已增加到session中，未提交至数据库，但是在这个过程中又被修改的数据，也叫脏读

print(session.new) print(session.dirty)

order\_by group\_by scalar行记录总数