[回顾 1](#_Toc14707227)

[查询优化之索引 2](#_Toc14707228)

[索引 2](#_Toc14707229)

[单独创建索引 7](#_Toc14707230)

[Git 7](#_Toc14707231)

[引言 7](#_Toc14707232)

[Linux系统本地git 9](#_Toc14707233)

[Linux系统本地远程git 14](#_Toc14707234)

[Windows系统git 19](#_Toc14707235)

# 回顾

1. 用户管理; 备注: 使用root用户连接MySQL成功后

* account\_name格式: ‘user\_name’@’host\_name’
  1. 创建用户: mysql> CREATE USER ……
  2. 修改用户: mysql> ALTER USER …….
  3. 删除用户: mysql> DROP USER ……

1. 权限管理
   1. 授予权限: mysql> GRANT 权限类型 ON 权限范围 TO account\_name WITH GRANT OPTION;
   2. 撤销权限: mysql> REVOKE 权限类型 ON 权限范围 FROM account\_name;
2. 索引
   1. 为表的某个字段设置主键, MySQL默认为其创建主键索引(唯一; 不空; 默认升序) 🡺 WHERE / ORDER BY字句中包含主键字段, 查询效率非常高
   2. 语法:

CREATE TABLE tbl\_name (

column\_name1 column\_type 完全约束,

column\_name2 column\_type 完全约束,

……

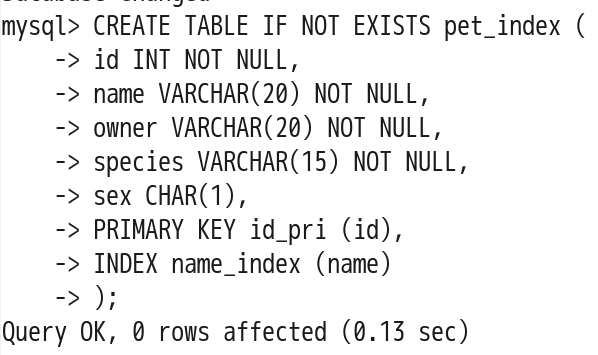
PRIMARY KEY [主键索引名] (column\_name)

)

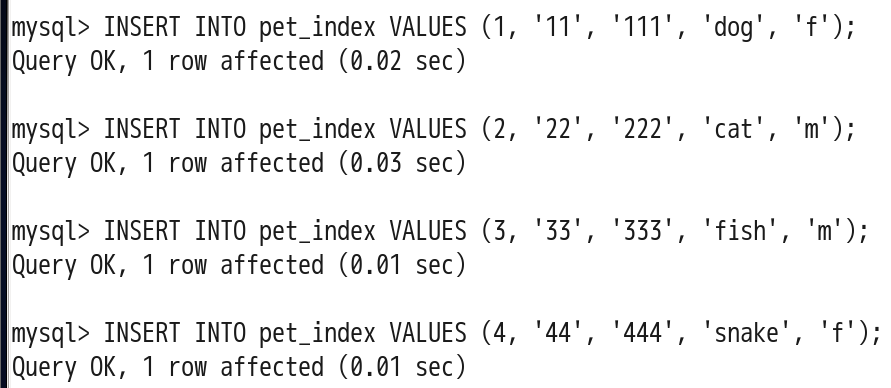
# 查询优化之索引

## 索引

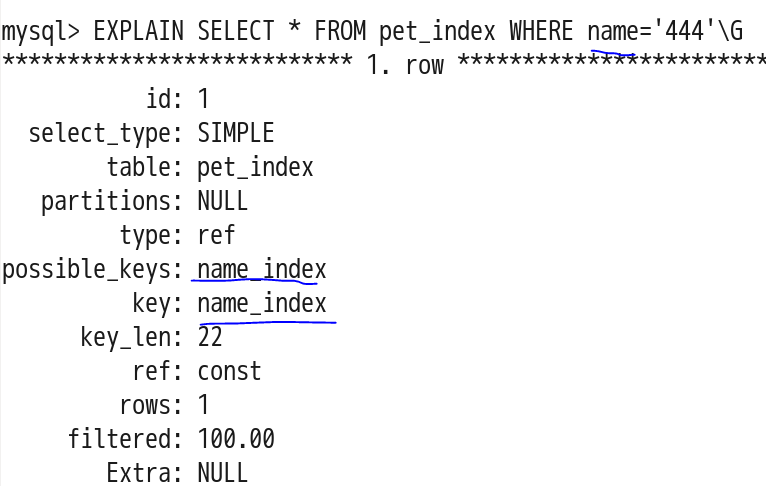
1. 思考题: 如果该字段很少被查询(WHERE / ORDER BY), 同时为字段创建索引(数据结构B+tree), 是否影响MySQL性能?
   1. 增/删/改: 除了对表中相应记录做增/删/改操作, 还需要额外的对索引(B+tree)做相应的操作
2. 主键索引: 字段值唯一(没有重复) + 不允许有空值 + 查询字段频率高
3. 普通索引: 字段值有重复 + 查询字段频率高
   1. 连接数据库服务器
   2. 切换数据库TI
   3. 创建表pet\_index; 为name字段创建普通索引, 其中name\_index是普通索引名



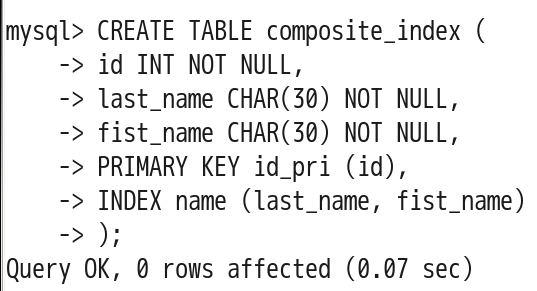
* 1. 插入四条记录



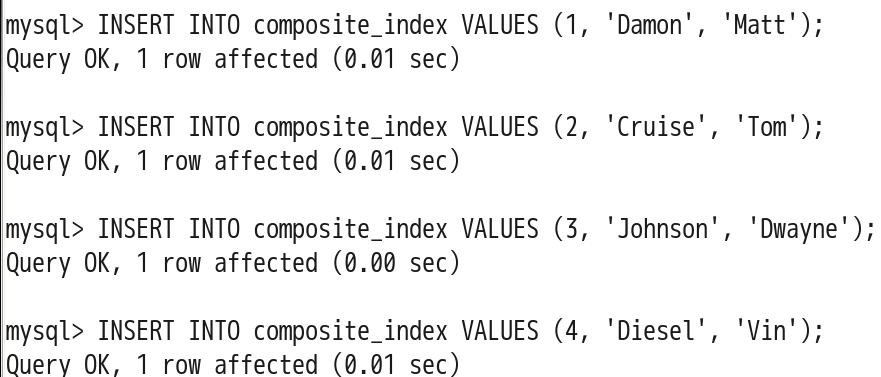
* 1. EXPLAIN语法分析SELECT语句使用是否该索引



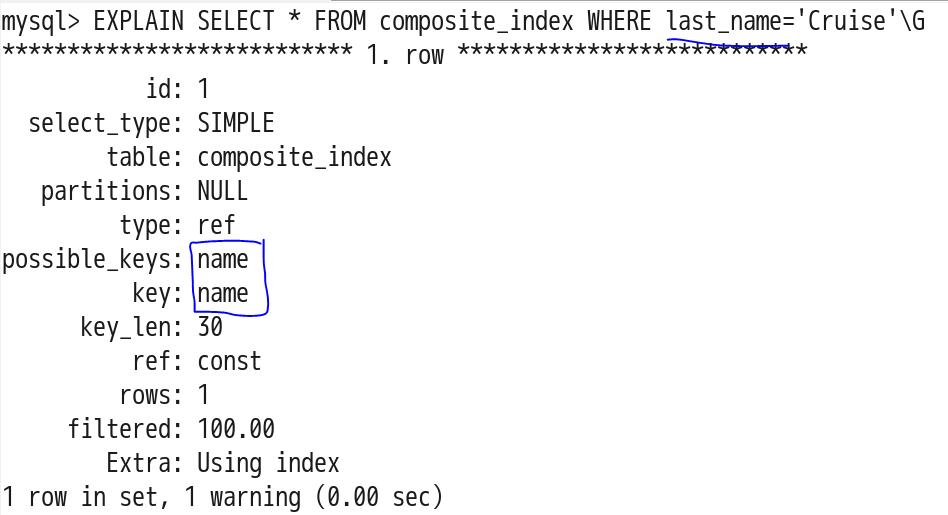
1. 唯一索引: 字段值唯一(没有重复) + 允许有空值 + 查询字段频率高
2. 组合索引: 为多个字段创建索引
   1. 创建表composite\_index, 并为last\_name和fist\_name两个字段创建组合索引, 索引名为name

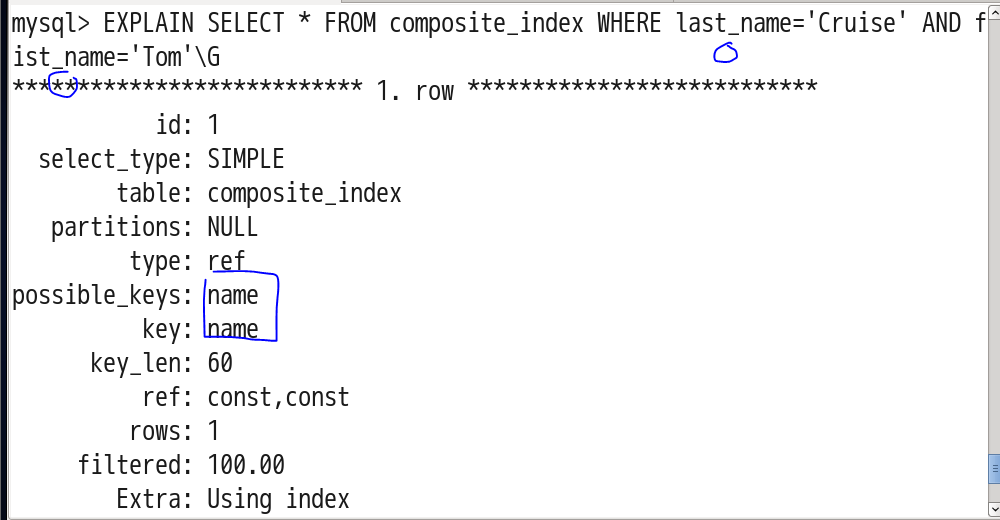


* 1. 插入四条记录

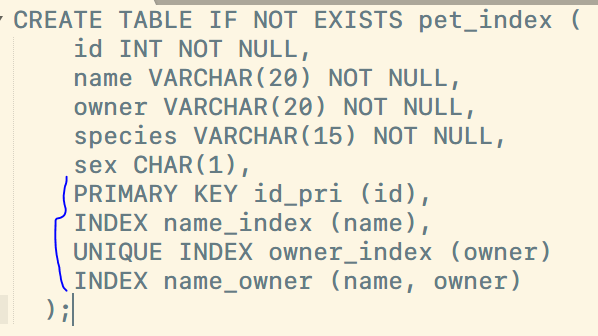


* 1. EXPLAIN语句分析SELECT是否使用组合索引





1. InnoDB存储引擎: 主键索引, 普通索引, 唯一索引, 组合索引: 关键词如下:



## 单独创建索引

1. 前提: 已经创建完表结构; 如何单独创建索引?
2. 单独创建索引: 普通索引, 唯一索引, 组合索引
   1. 语法:

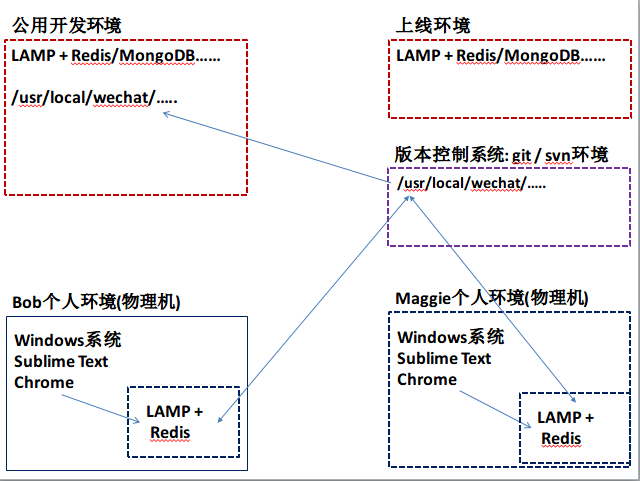
mysql> CREATE [UNIQUE] INDEX index\_name ON tbl\_name (column\_name1 [, column\_name2] …..)

1. 分析: EXPLAIN语句语法一致

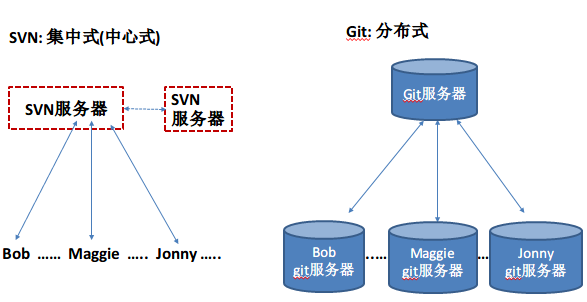
# Git

## 引言

1. 服务器端开发人员技能之一: 熟练掌握git / svn 🡺 团队协作开发
2. 工作中涉及的多个环境



1. 版本控制系统: 记录一个或者多个文件/目录变化的系统, 以便于将来查看任意版本内容; 或者回退到指定版本对应内容.
2. Git: Linus(开发Linux系统内核), 开源(C语言), 分布式版本控制系统
3. Git和svn区别
   1. svn: 中心式(集中式)版本控制系统
   2. git: 分布式版本控制系统



1. 为什么学git? 学习曲线陡(“难”)

## Linux系统本地git

1. 环境搭建: 安装git软件包
   1. $ sudo yum install git
2. 课堂执行命令:
   1. 创建目录gittest, 并切换目录

$ mkdir ~/gittest

$ cd ~/gittest

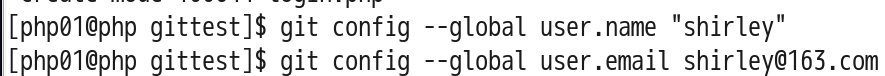
* 1. 初始化本地仓库; $ git init
  2. vi编辑文件login.php, 添加一行echo语句, 保存退出

$ vi login.php

* 1. 添加到缓存区; $ git add login.php
  2. 添加到本地仓库区, 并设置备注消息(双引号内)

$ git commit -m “first commit”

* 1. 设置提交用户为shirley(任意); [邮箱地址shirley@163.com](mailto:邮箱地址shirley@163.com)(任意)



* 1. 再编辑login.php(添加一行echo); 添加到缓存区; 提交到本地仓库

$ vi login.php

$ git add login.php

$ git commit -m “second commit”

1. 练习:
   1. 第三次提交: ~/gittest/public/register.php, 添加一行echo语句

$ cd ~/gittest

$ mkdir public

$ vi public/register.php

$ git add \*

$ git commit -m “third commit”

* 1. 第四次提交:

1. ~/gittest/login.php, 再添加一行echo语句

$ vi ~/gittest/login.php

1. ~/gittest/public/chat.php, 添加一行echo语句

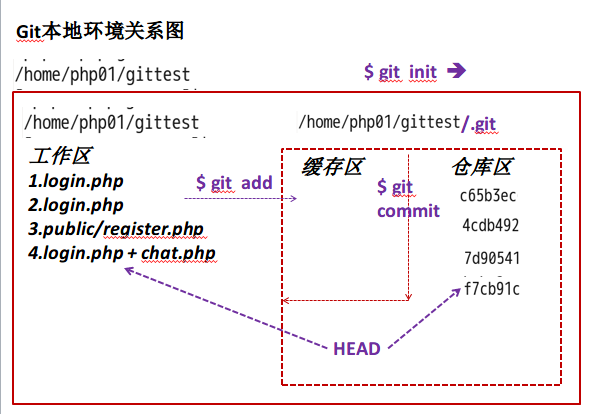
$ vi ~/gittest/public/chat.php

1. 添加到缓存区, 提交到仓库去

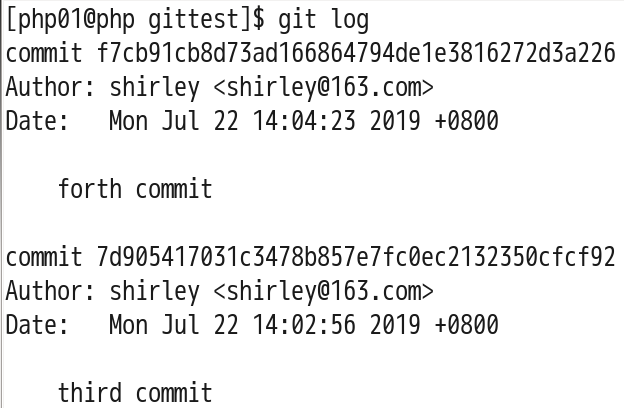
$ git add \*

$ git commit -m “forth commit”

1. Git本地环境中三个区的关系



1. 回退到指定某个版本
   1. 获取回退到指定版本的提交字符串(前7位)



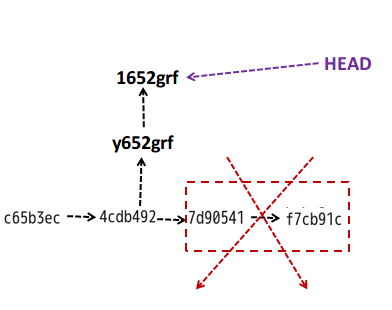
* 1. Ctrl + shift + t重新开启一个终端; git reset回退到指定版本(7d90541是对应提交字符串的前7位)

$ git reset --hard 7d90541

* 1. 再reset到最后一次(第四次提交) 🡺 ~/gittest目录又重新回到第四次提交的状态

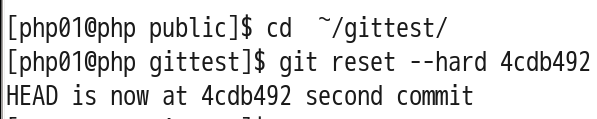


1. 练习:
   1. 已知: ~/gittest目录: 已经提交四次;
   2. 需求: 先回退到第二次提交状态; 创建~/gittest/group.php, 添加一行echo, 提交一次; 修改login.php, 添加一行echo, 提交一次



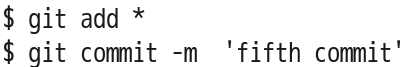
* 1. 练习命令如下:

1. Git reset回退到第二次提交; 其中4cdb492是第二次提交对应的前7位字符串

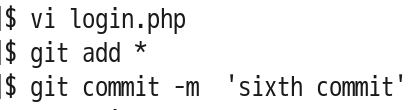


1. 创建group.php, 添加并提交



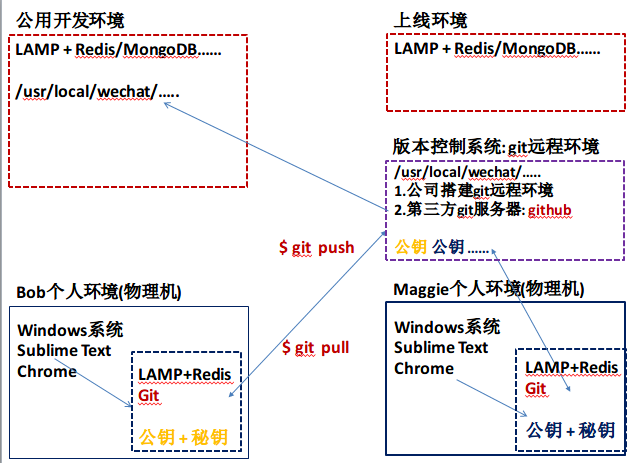


1. 编辑login.php, 添加并提交



## Linux系统本地远程git

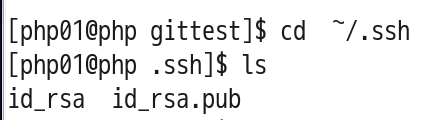
1. 流程图:



1. 准备工作:
   1. 第一步: [www.github.com](http://www.github.com)网站注册账号和登录密码
2. 点击, 输入用户名 + 邮箱地址 + 登录密码 🡺 点击 “Create an account”
3. 登录邮箱, 点击激活链接 / 按钮
4. 点击, 登录github网站
   1. 第二步: Linux系统, 命令生成公钥和秘钥(字符串文件)
5. Linux系统终端, 输入下面命令(四次回车); 其中双引号的邮箱地址可以随便写



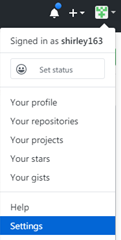
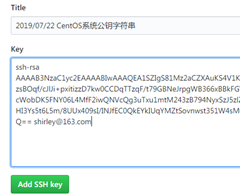
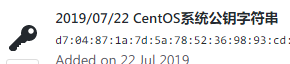
1. 切换目录

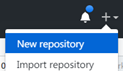
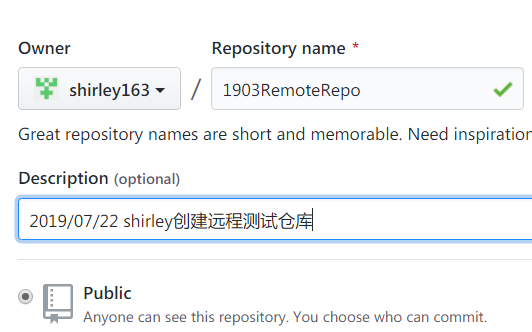
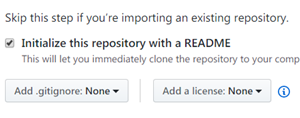
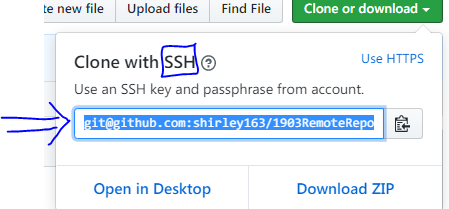


1. cat查看公钥文件内容, 并选中, 鼠标右键复制



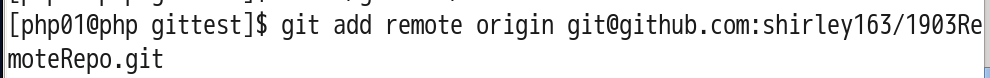
* 1. 第三步: 公钥上传到github远程服务器

1. 登录github网站 🡺  🡺  🡺 🡺 粘贴之前复制的id\_rsa.pub文件的所有内容 🡺  🡺  🡺 
2. 需求: Linux系统~/gittest目录(git本地仓库)内容, 和远程github仓库同步(一致)
   1. 第一步: github网站创建远程仓库 🡺 URI / URL地址

 🡺 🡺  🡺  🡺  🡺  🡺 复制蓝色URI地址

* 1. 第二步: 本地仓库(~/gittest)和远程仓库建立连接

1. 切换目录; $ cd ~/gittest
2. Git命令建立和远程仓库的连接(粘贴之前复制的URI); 🡺 只能建立连接一次



* 如果上面建立连接错误, 需要先移除连接

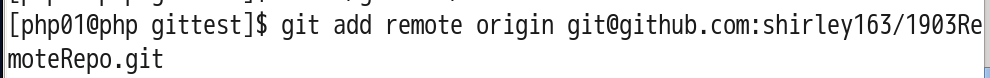
1. Git 1.7.10(低版本)运行:



1. Git高版本



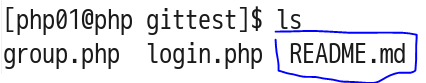
* 再执行建立正确连接命令



* 1. 第三步: 将远程仓库内容pull下来(下载); 输入yes, 回车即可

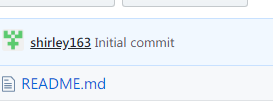
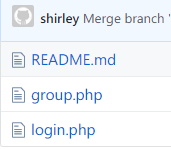


* ls命令查看结果:



* 1. 第四步: 将本地仓库内容push上去(上传)



 🡺 刷新github网站, 查看结果 🡺 

## Windows系统git

1. 任务:
   1. 第一步: 下载安装git.exe (最新版本)
2. 官网链接(点击: 下一步/next):

<https://git-scm.com/download/win>

* 1. 第二步: 下载安装GUI软件

1. TortoiseGit官方链接(点击: 下一步/next):

<https://download.tortoisegit.org/tgit/2.8.0.0/TortoiseGit-2.8.0.0-64bit.msi>

/PSD1903/24\_Git/Day01/Day01-PM3.zip