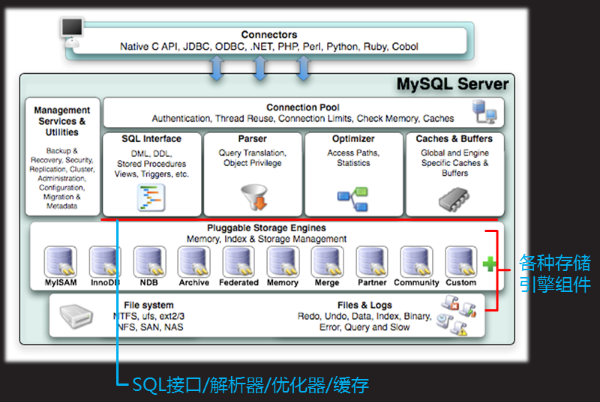
**mysql存储引擎**

mysql工作原理

MySQL体系结构

8个结构构成：管理工具、连接池、sql接口、分析器、优化器、缓存、存储引擎、硬盘



mysql存储引擎

作为可插拔式的组件提供

mysql服务软件自带的功能程序，处理表的处理器

不同的存储引擎有不同的功能和数据存储方式

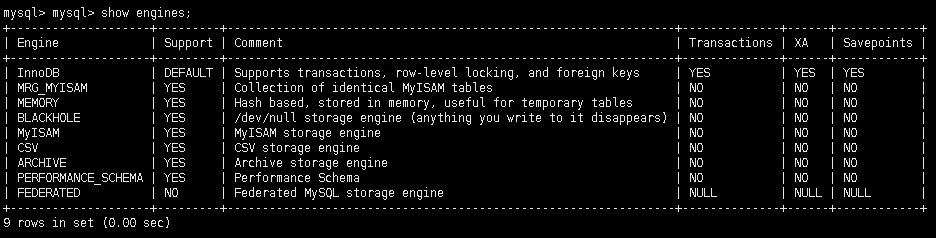
默认的存储引擎

mysql 5.0\5.1--->myisam

mysql 5.5/5.6--->innodb

列出可用的存储引擎类型

show engines或show engines\G



存储引擎的配置

修改表的存储引擎

建表时手动指定

未指定时，使用默认存储引擎

show create table 表名\G;可确认

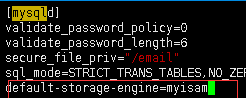
设置默认存储引擎

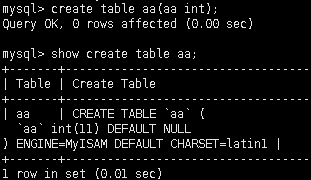
vim /etc/my.cnf配置文件

[mysqld]

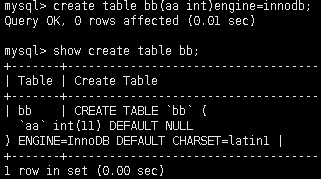
default-storage-engine=存储引擎名称

systemctl restart mysqld

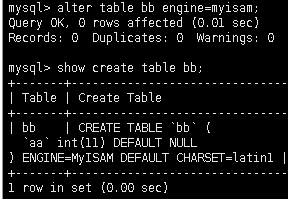




创建表时，指定默认的引擎



修改表的存储引擎



存储引擎特点

myisam存储引擎

主要特点

支持表级锁（当有人往表中加数据，将表加锁）

不支持事务、事务回滚、外键

相关表文件

表名.frm：存放表结构

表名.MYI：存索引信息

表名.MYD：存表数据

innodb存储引擎

主要特点

支持行级锁定（只给被访问的行加锁）

支持事务、事务回滚、外键

相关表文件

表名.frm：存放表结构

表名.ibd：存放数据+索引

ibdata1

ib\_logfile0

ib\_logfile1三个是事务日志

mysql锁机制：为了解决并发访问冲突问题

锁粒度

表级锁：一次直接对整张进行加锁

行级锁：只锁定某一行

页级锁（给内存里的东西加锁，1M就是一页）：对整个页面（mysql管理数据的基本存储单位）进行加锁

锁类型（根据访问去分）

读锁（共享锁）：支持并发读（进行select操作是读锁）

写锁（互斥锁、排它锁）：是独占锁，上锁期间其他线程不能读表或写表

查看当前的锁状态（进行insert update delete是写锁）

检查table\_local开头的变量，%作通配符

事务特性（ACID）

事务（一次sql操作，从开始到结束的过程）

事务回滚：在事务执行过程中，任意一步操作失败，恢复所有的操作

事务日志：ibdata1 ib\_logfile0 ib\_logfile1，后两个是所有已经提交的sql命令，第一个是未提交的sql命令及执行Sql命令的序号（通过LSN日志序列号）

atomic：原子性

事务的整个操作是一个整体，不可分割，要么全部成功，要么全部失败。

consistency：一致性

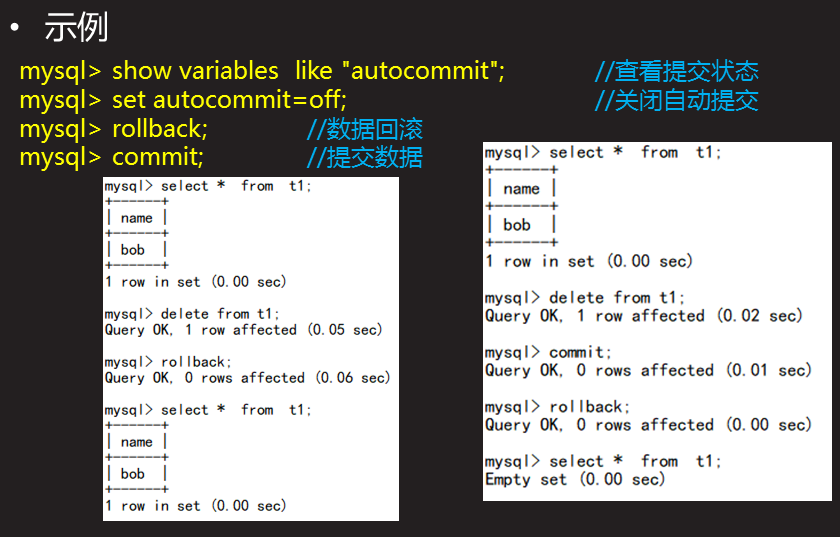
事务操作的前后，表中的记录没有变化。

isolation：隔离性

事务操作是相互隔离不受影响的

durability：持久性

数据一旦提交，不可改变，永久改变表数据



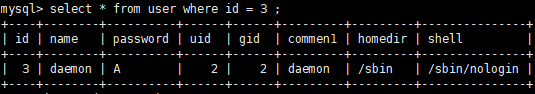
**匹配条件**

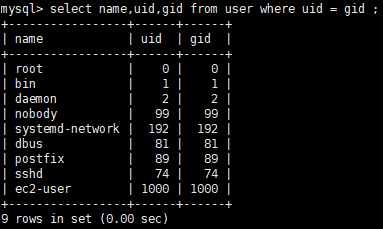
基本匹配条件

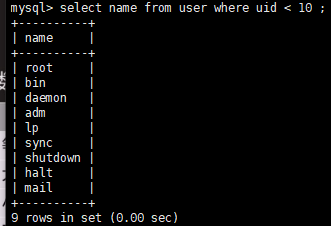
数值比较

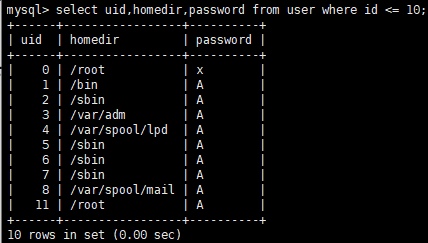
字段类型必须是数据数值类型

|  |  |
| --- | --- |
| 类型 | 用途 |
| = | 等于 |
| >、>= | 大于、大于或等于 |
| <、<= | 小于、小于或等于 |
| != | 不等于 |

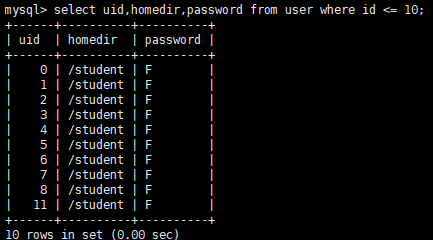










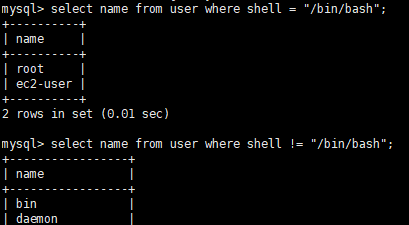


字符比较/匹配空/非空

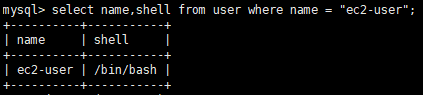
字符比较时，字段类型必须是字符类型

|  |  |
| --- | --- |
| 类型 | 用途 |
| = | 相等 |
| != | 不相等 |
| is null | 匹配空 |
| is not null | 非空 |

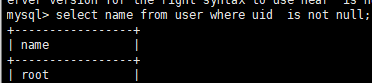
where 字段名 符号 “值”

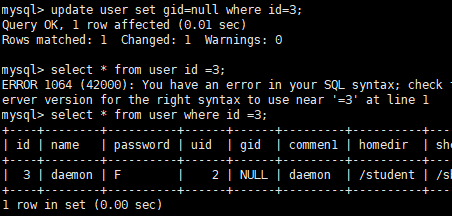








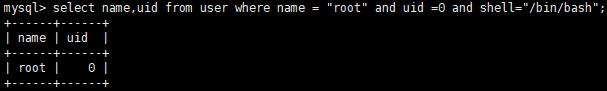


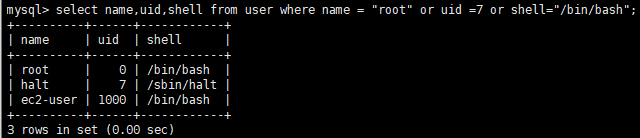


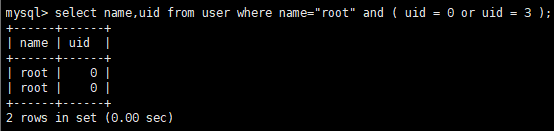
逻辑匹配

多个判断条件时使用

|  |  |
| --- | --- |
| 类型 | 用途 |
| OR | 逻辑或，多个条件满足一个即可 |
| AND | 逻辑与，多个条件都得成立 |
| ！ | 逻辑非， |
| () | 提高优先级 |



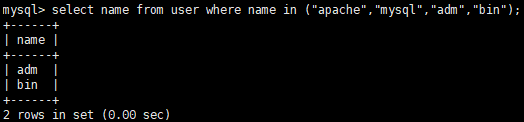


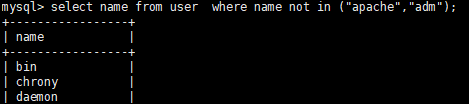


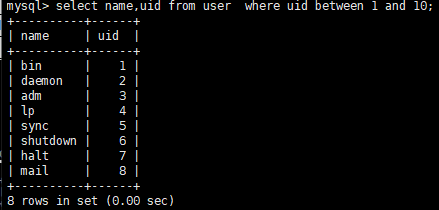
范围内匹配/去重显示

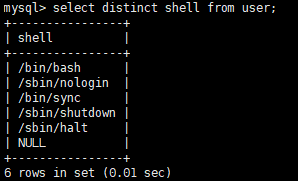
匹配范围内的任意一个值即可

|  |  |
| --- | --- |
| 类型 | 用途 |
| in （值列表） | 在...里... |
| not in (值列表) | 不在...里... |
| between 数字1 and 数字2 | 在..之间.. |
| distinct字段名 | 去重显示，不显示字段的重复值，只显示第一次出现的 |









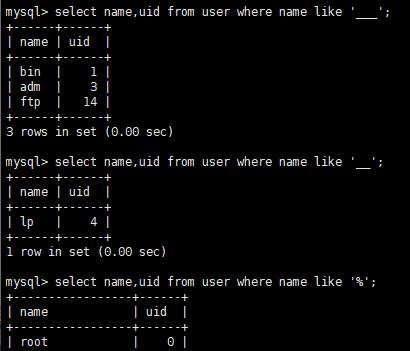
高级匹配条件

模糊查询

基本用法

where 字段名 like ‘通配符’

\_匹配单个字符、%匹配0-N个字符

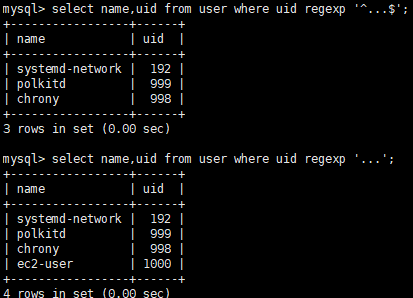


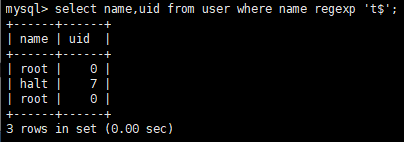
正则匹配

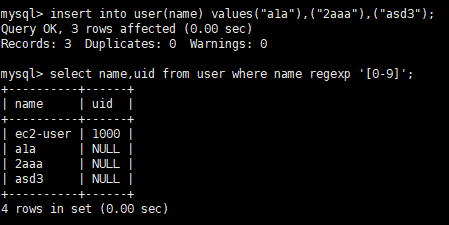
基本用法

where 字段名 regexp ‘正则表达式’

正则元字符 ^ $ . [] \* |







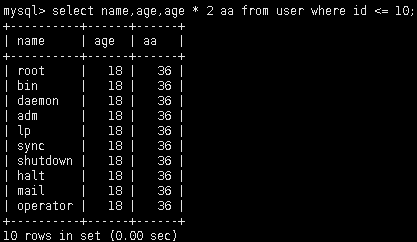
四则运算

运算操作

字段必须是数值类型

|  |  |
| --- | --- |
| 类型 | 用途 |
| + | 加法 |
| - | 减法 |
| \* | 乘法 |
| / | 除法 |
| % | 取余 |





操作查询结果

聚集函数（字段类型必须是数值型的）

mysql内置数据统计函数

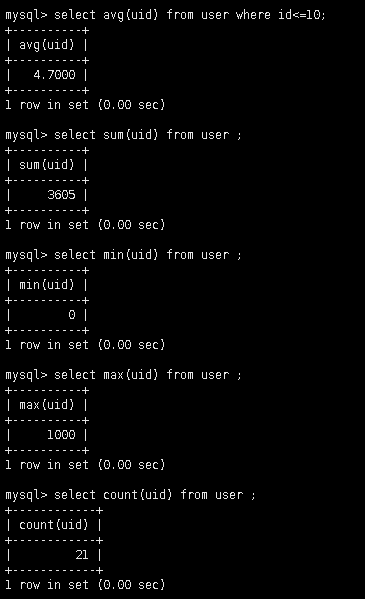
avg（字段名） //统计字段平均值

sum（字段名） //统计字段之和

min（字段名） //统计字段的最小值

max（字段名） //统计字段最大值

count（字段名） //统计字段值个数

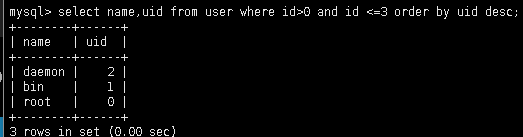


查询结果排序（默认升序）

基本用法

sql查询结果

order by 字段名（通常是数值类型字段） [asc|desc];

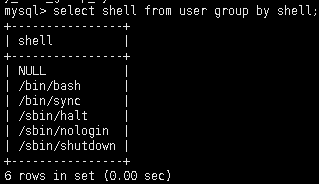


查询结果分组

基本用法

sql查询结果

group by 字段名（通常是字符类型字段）;



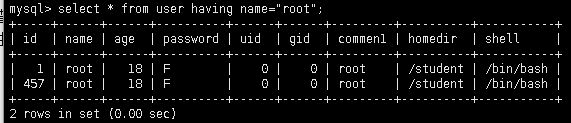
查询结果过滤

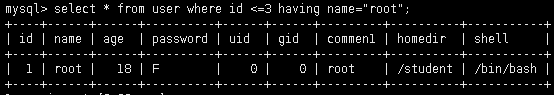
基本用法

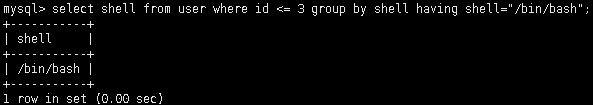
sql查询 having 条件表达式;

sql查询 where 条件 having 条件表达式;

sql查询 group by 字段名 having 条件表达式;







限制查询结果显示行数

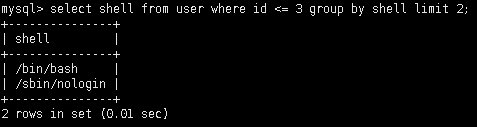
基本用法

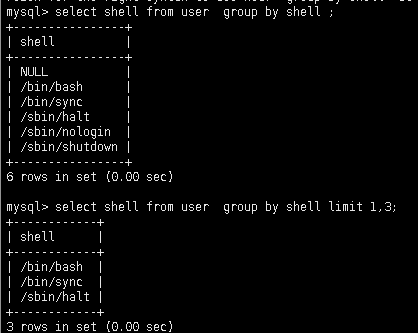
sql查询 limit N; //显示查询结果前N条记录

sql查询 limit N,M; //显示指定范围内的查询记录

sql查询 where 条件查询 limit 3; //显示查询结果前3条记录

sql查询 where 条件查询 limit 3,3; //从第4条开始，共显示3条



从第一行显示显示三行