Windows端启动与停止

启动：net start mysql

停止：net stop mysql

MySQL常用命令

显示当前服务器版本 SELECT VERSION();

显示当前日期时间 SELECT NOW();

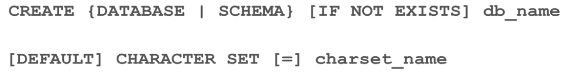
显示当前用户 SELECT USER();

MySQL语句的规范

关键字与函数名全部大写

数据库名称、表名称、字段名称全部小写

创建数据库



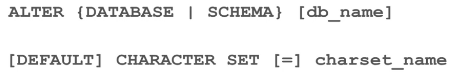
查看当前服务器下的数据表列表



查看警告信息：SHOW WARNINGS;

查看创建的数据库信息：SHOW CREATE DATABASE Lu;

修改数据库



删除数据库



数据类型



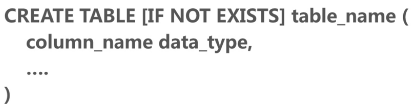


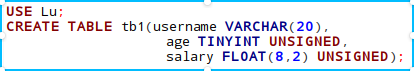






创建数据表





查看数据表



查看数据表结构





插入记录



或



记录查找

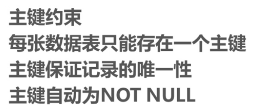


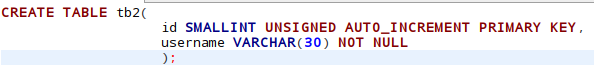


自动编号AUTO\_INCREMENT

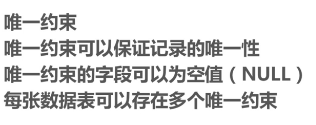


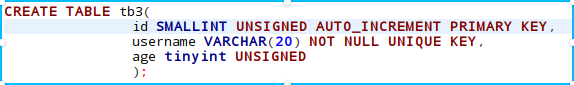
主键PRIMARY KEY





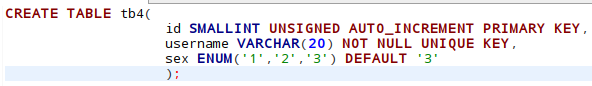
UNIQUE KEY



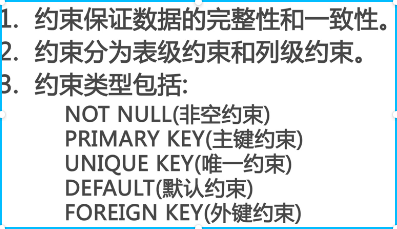


默认约束DEFAULT

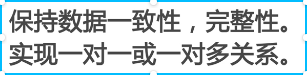




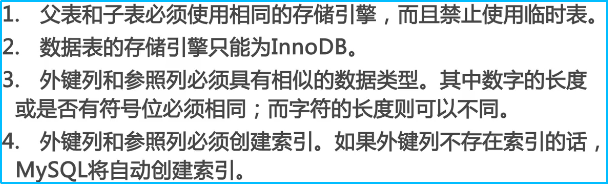
约束



外键约束FOREIGN KEYp



外键约束的要求



编辑数据表的默认存储引擎

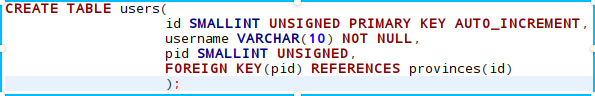
MySQL配置文件



创建第一张表：



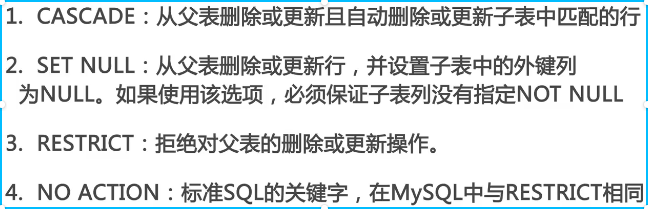
创建第二张表：pid参照id



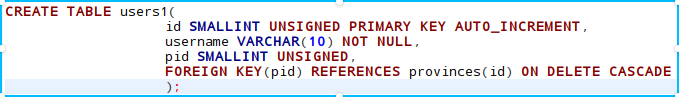
显示主键：



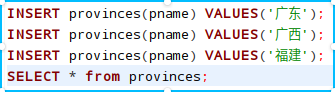
外键约束的参照操作



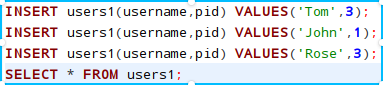
创建另一张表：



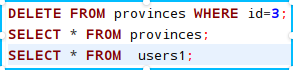
先往provinces表插入记录：



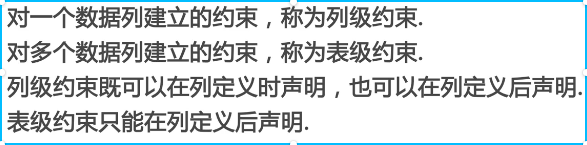
再往users1表插入记录：



删除provinces表的id=3，users1中的pid=3也会删除：

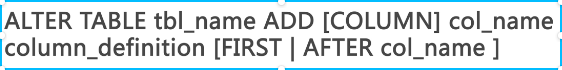


表级约束与列级约束



修改数据表

添加单列

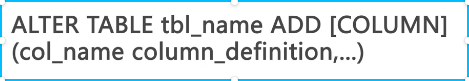








添加多列



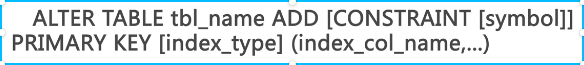
删除列



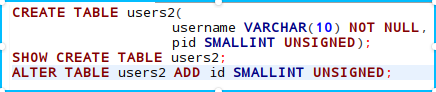




添加主键约束

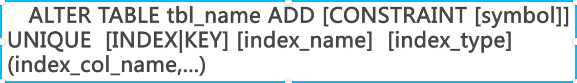


先创建一张废表users2：



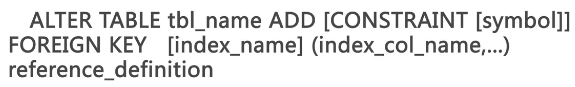


添加唯一约束





添加外键约束





添加、删除默认约束



创建一个列：



添加默认值：



删除默认值：



删除主键约束





删除唯一约束



查出所有唯一约束：





删除外键约束

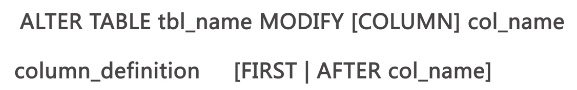




彻底删除索引：

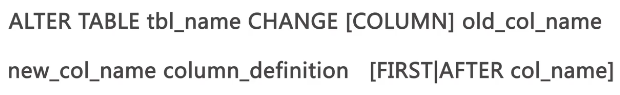


修改列定义



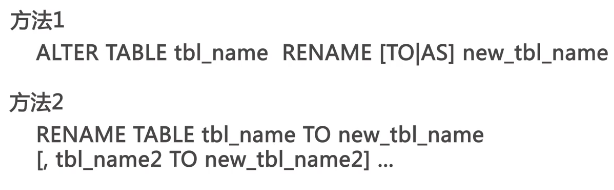


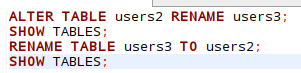
修改列名称

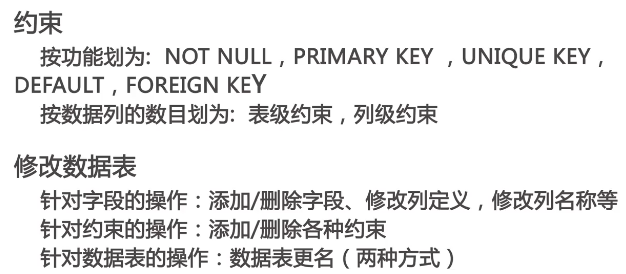




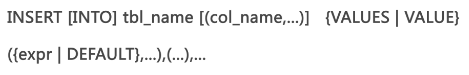
数据表更名



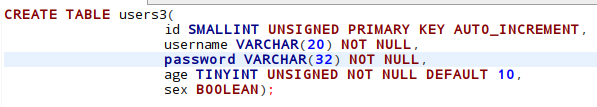


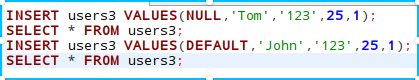


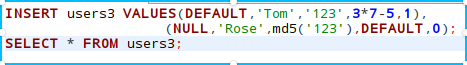
插入记录

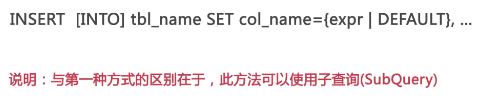


创建表：





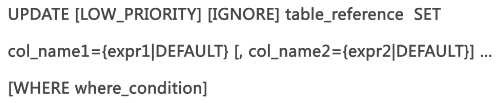








更新记录（单表更新）



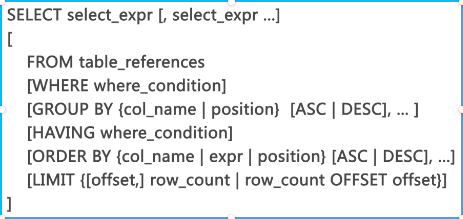
Id为偶数的年龄+10：



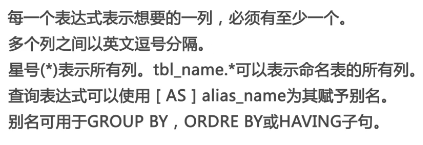
删除记录（单表删除）



查找记录



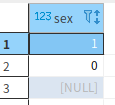
查询表达式



查询结果分组

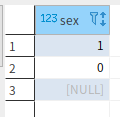


ASC:升序，DESC：降序

分组条件





对查询结果进行排序





限制查询结果返回的数量





创建一张test表：



将表users中年龄>=25的用户名插入表test：

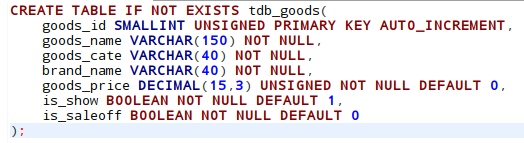


SET NAMES gbk:修改当前编码为gbk

使用比较运算符的子查询

运算符：=、 >、 <、 >=、 <=、 <>、 ！=、 <=>

创建表：



插入值：



求平均值：



对平均值四舍五入，且保留小数点后两位



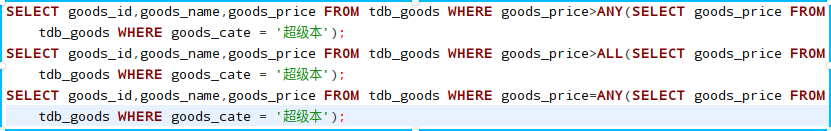
联合实现子查询：



用ANY、SOME或ALL修饰的比较运算符

ANY、SOME、ALL关键字





使用[NOT] IN的子查询

语法结构

=ANY运算符与IN等效。

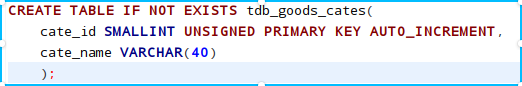
！=ALL或<>ALL运符与NOT IN等效。

使用[NOT] EXISTS的子查询

如果子查询返回任何行，EXISTS将返回TRUE；否则为FALSE。

使用INSERT...SELECT插入记录

创建表：



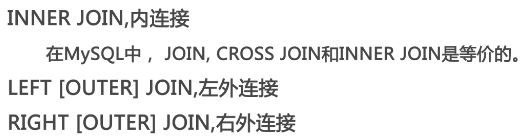






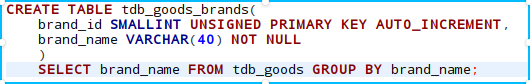
多表更新

连接类型





CREATE...SELECT



内连接INNER JOIN

连接条件

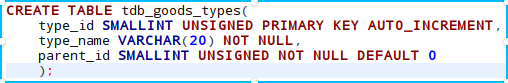
使用ON关键字来设定连接条件，也可以使用WHERE来代替。

通常使用ON关键字来设定连接条件，使用WHERE关键字进行结果集记录的过滤。

无限分类的数据表设计

自连

创建表：



插入数据



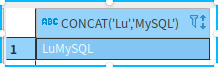




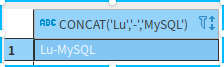
MySQL字符函数









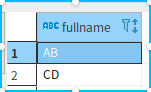


建表test2:

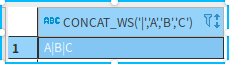






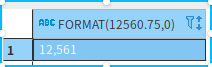






格式化：



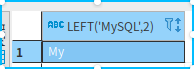


大小写：



获取左侧字符





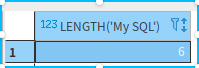
组合嵌套：

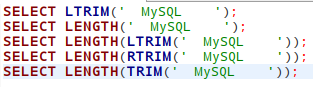








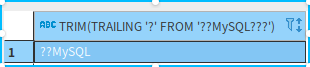
































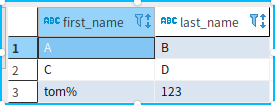




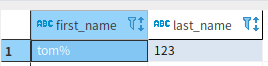


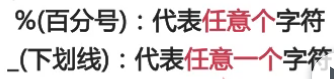












数值运算符和函数



向上取整：





向下取整：

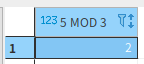




























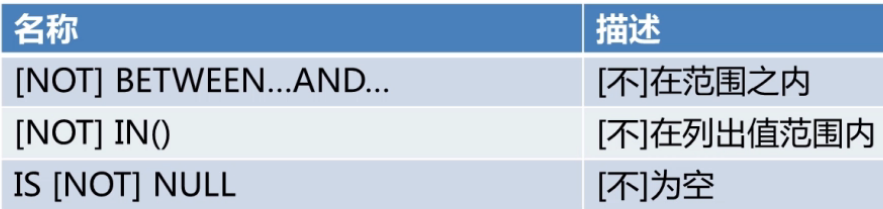








比较运算符和函数



























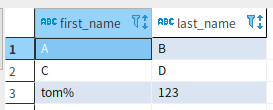










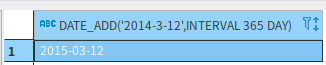


日期时间函数







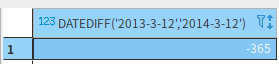








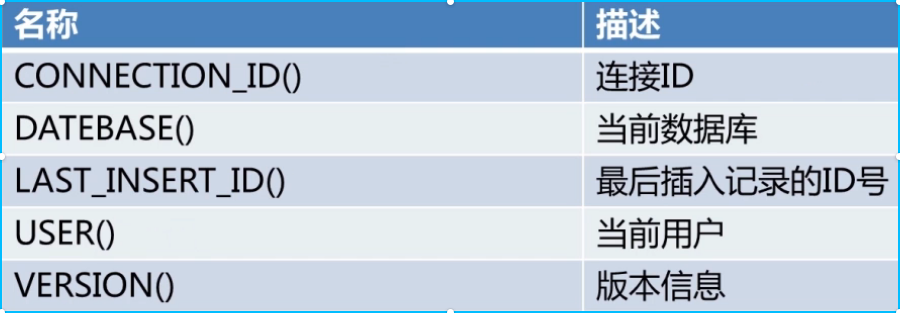






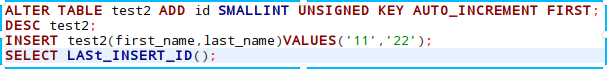


信息函数



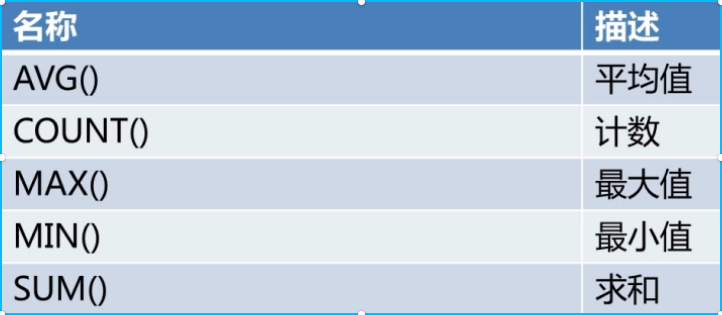




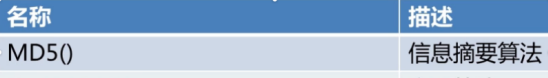




聚合函数



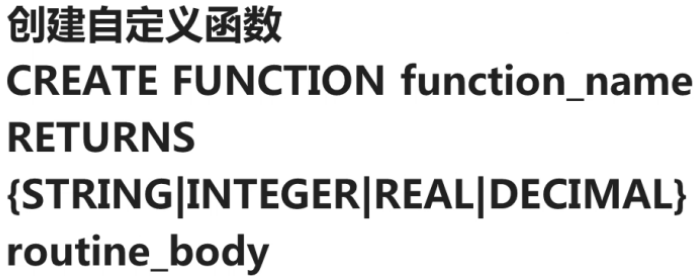
加密函数



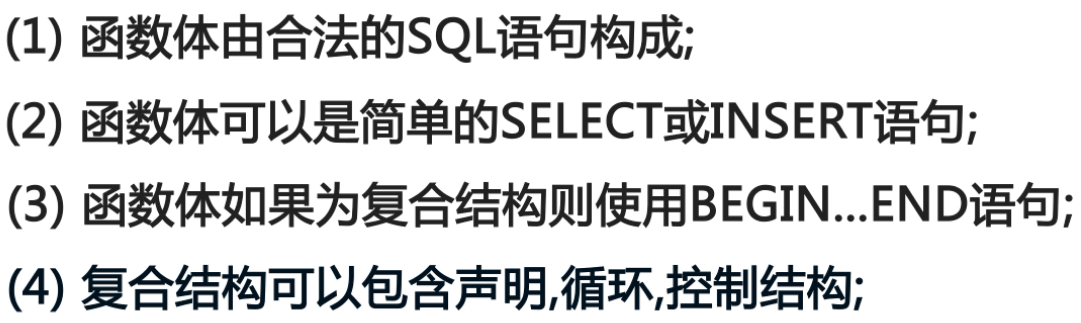


自定义函数

自定义函数的两个必要条件：（1）参数（2）返回值



函数体

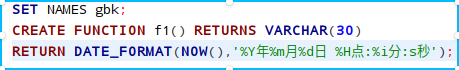


（1）创建不带参数的自定义函数

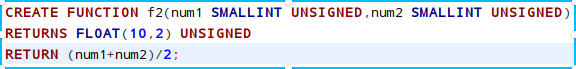




封装成：（有问题）



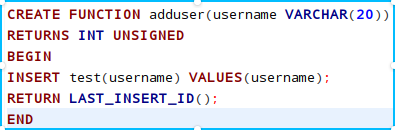
1. 创建带有参数的自定义函数：（有问题）



删除和调用：



1. 创建具有复合结构函数体的自定义函数：（有问题）



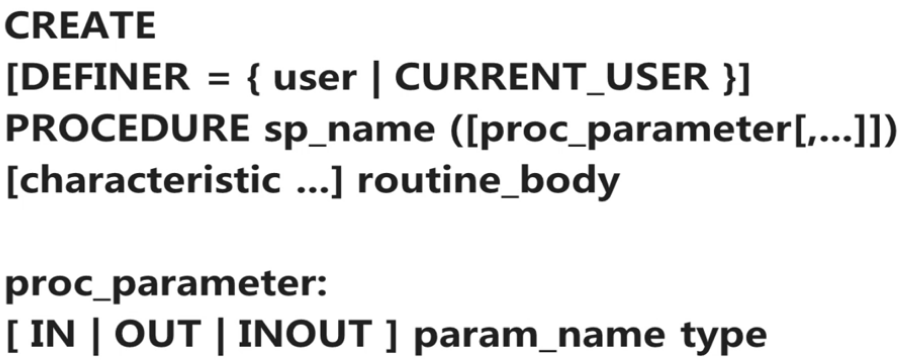
存储过程

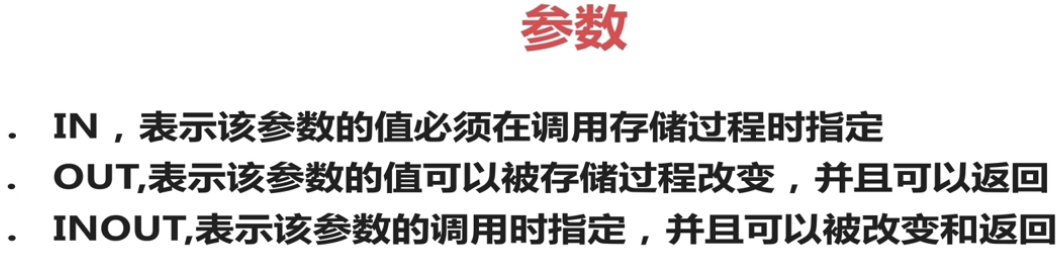


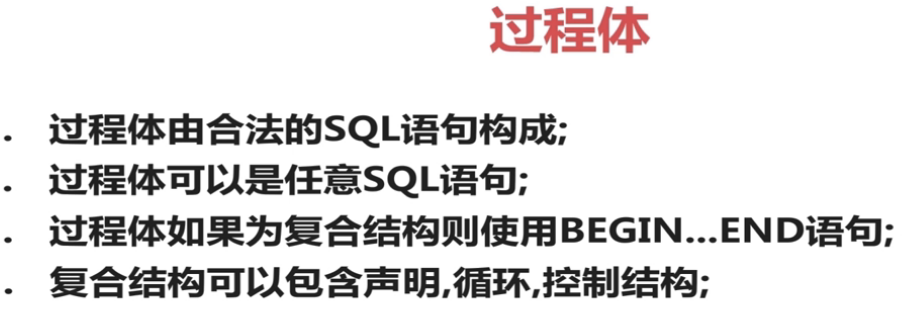
存储过程的优点

1. 增强SQL语句的功能和灵活性
2. 实现较快的执行速度
3. 减少网络流量

创建存储过程

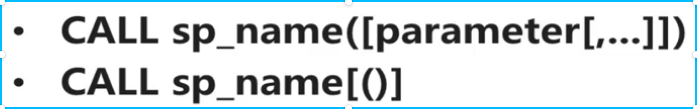








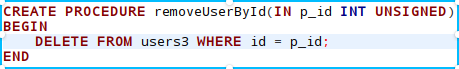
调用存储过程





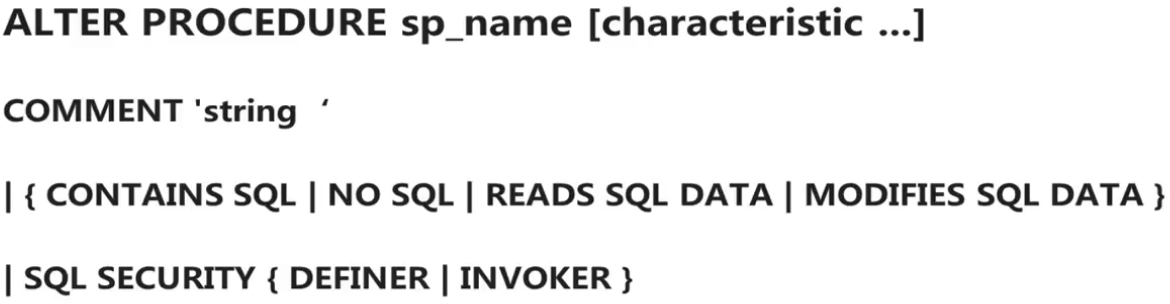
创建带有IN类型参数的存储过程

（有问题）





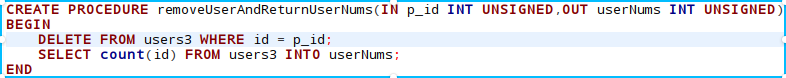
修改存储过程



删除存储过程

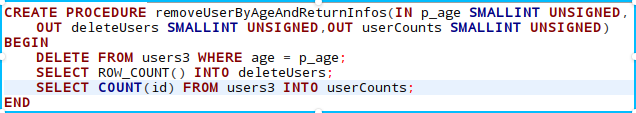


创建带有IN和OUT类型参数的存储过程



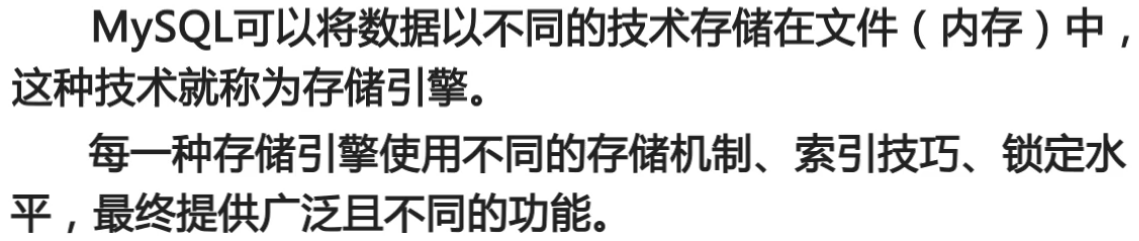


创建带有多个OUT类型参数的存储 过程





存储引擎



MySQL支持的存储引擎

- MyISAM

- InnoDB

- Memory

- CSV

- Archive

并发控制：当多个连接对记录进行修改时保证数据的一致性和完整新。

锁：

共享锁（读锁）：在同一时间段内，多个用户可以读取同一个资源，读取过程中数据不会发 生任何变化。

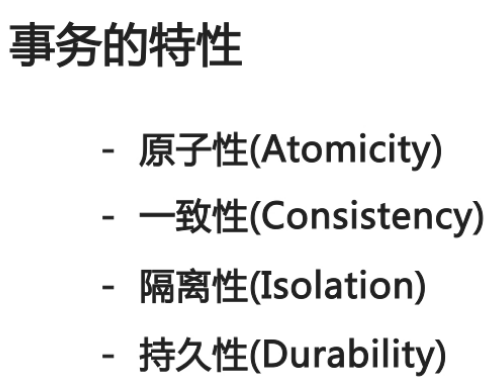
排它锁（写锁）：在任何时候只能有一个用户写入资源，当进行写锁时会阻塞其他的读锁或者写锁操作。

锁颗粒

- 表锁，是一种开销最小的锁策略。

- 行锁，是一种开销最大的锁策略。

事务：用于保证数据库的完整性



外键：保证数据一致性的策略。

索引：对数据表中一列或多列的值进行排序的一种结构。







修改存储引擎的方法



