# 事务

## 什么是事务

1）事务指的是一组操作，它可以由一条或多条语句SQL语句组成，这些语句要么全 部执行，要么全部不执行。

1. 事务可以使数据操作更加安全，保证了在同一个事务中的操作具有同步性。

3）事务具有预览作用。在事务未提交之前，可以对事务中操作的执行结果进行预览，如使用SELECT语句进行预览。

## 事务的四大特性ACID

1. 原子性：原子性是指一个事务必须被视为一个不可分割的最小单位。只有事务中所 有的SQL语句都执行成功，该事务才执行成功。事务中如果有任何一个SQL 语句执行失败，已经执行成功的SQL语句也必须撤销，数据库的状态会到 开启事务前的转态。
2. 一致性：
3. 隔离性：隔离性可以理解为并发控制。当多个用户并发访问数据库时，数据库为一 个用户开启的事务，不能被其他用户开启的事务的操作所影响，多个并发 事务之间要相互隔离。这一点从事务的隔离控制中可以体现出现。
4. 持久性：事务执行成功后所做的修改会永久保存到数据库。

原子性（Atomicity）、一致性（Consistency）、隔离性（Isolation）、持久性（Durability）。

## 开启事务

使用事务必须先开启事务。开启事务的语法格式为：

**START TRANSACTION ;**

或者使用**begin**命令开启；

## 提交事务

在MySQL中直接书写的SQL语句都是自动提交的，而事务中的执行的所有SQL语句最后都需要使用COMMIT语句手动进行提交，并且只有事务提交后这些SQL的执行才会真正的生效。其语法格式为：

**COMMIT ;**

操作完事务时，一定要使用COMMIT语句提交事务，否则意味着该事务操作是失败的。

## 回滚事务

回滚事务也称取消事务。指的是在事务未提交前撤销事务中已执行的操作。其语法格式 为：  
 **ROLLBACK ;**

## 结束事务

开启事务后，当提交事务COMMIT或者回滚事务ROLLBACK事务后，则意味着该事务结束，如果还需要进行事务操作，此时需要重新开启一个事务。

数据库在操作事务的过程中如果中途退出或者电脑关机，也意味着该事务回滚了。

## 事务的隔离级别一

数据库是运行多线程并发访问的，是多用户的，所以容易出现多个线程同时开启事务的情况，这样会出现脏读、重复读以及幻读的情况。为了避免这种情况的发生，需要为事务设置隔离级别，用于隔离每个线程中开启的事务。

在MySQL中，事务有四种隔离级别：

1. READ UNCOMMITTED：READ UNCOMMITTED（读未提交）是事务中最低的隔离级别， 也称脏读。该级别下的事务可以读取到其他事务中未提交的数据。相当危险、一般 很少使用。
2. READ COMMITTED：READ COMMITTED（读提交），该级别下的事务可以读取到其他 事务中已提交的数据。可以避免脏读，但会出现不可重复读和幻读的情况。
3. REPEATABLE READ：REPEATABLE READ（可重复读）是MySQL默认的事务隔离级别， 确保同一事务中读取数据时，会看到同样的数据行，不管其他事务是否提交。可以 避免重复读的情况。
4. SERIALZABLE：SERIALIZABLE（可串行化）是事务的最高隔离级别。该隔离级别的事 务必须提交后。其他事务才会执行，也就是其他事务必须等待此隔离级别的事务提 交后才可以执行。所以会出现停滞现象。

## 事务的隔离级别二

1. 读未提交：我的事务能够读取到其他未提交事务所做的修改。
2. 读提交：我的事务只能读取到其他已提交的事务所做的修改。
3. 重复读：不管其他事务是否提交。我开启的事务只会读取到我开启事务时表中的数 据。等我此次事务提交后，才能读取到其他已提交事务所做的操作。
4. 可串行化：我开启事务后，其他事务中的修改操作进入停滞状态，必须等我的事 务提交后其他事务中的修改操作才会执行。

# 存储过程

## 什么是存储过程

在开发过程中，经常会遇到重复使用某一功能的情况，为次，MySQL引入了存储过程。存储过程就是一条或多条SQL语句的集合，当对数据库进行一系列复杂操作时，存储过程可以将这些复杂操作封装成一个代码块，以便重复使用，类似于方法。

## 创建存储过程

要使用存储过程，首先要创建一个存储过程。创建存储过程的语法格式为：

**CREATE PROCEDURE 存储过程名称([参数列表])**

**BEGIN**

**SQL语句**

**......**

**END;**

1. CREATE PROCEDURE用于创建存储过程。
2. 参数列表由一个或多个变量组成，表示调用该存储过程时需要传入的参数。其中 每个参数变量的格式为[IN|OUT|INOUT] 参数名 参数类型，多个参数之间用逗号隔 开。IN表示输入参数、OUT表示输出参数、INOUT表示既可以输入也可以输出。
3. BEGIN和END关键字表示SQL语句开始和结束，将多个要执行的SQL语句括起来。 类似与方法中的花括号，

在创建存储过程时，为了避免与存储过程中SQL语句结束符;相冲突。需要使用 DELIMIETER改变存储过程的结束符，创建存储过程后再使用DELIMITER恢复结束符。

## 调用存储过程

使用CALL语句调用存储过程。其语法格式为：

**CALL 存储过程名( [参数列表] ) ;**

如果调用的存储过程定义了参数，那么在调用该存储过程时也必须传入相应类型、个数 的参数。

## 查看返回结果

当调用的存储过程在定义时定义参数了并且存在OUT类型的参数时，表示此存储过程 拥有返回结果。此时可以使用SELECT语句查看存储过程的返回结果，其语法格式为：

**SELECT @实参名 ;**

其中参数名是调用该存储过程时传入给OUT类型参数的实参名。

## 查看存储过程

创建存储过程后，在MySQL中会存储存储过程的状态信息，如该存储过程所属的数据 库，创建时间等等。用户可以使用SHOW STATUS 语句或SHOW CREATE语句查看，也 可以直接从系统的information\_schema数据中查询。

1. 使用SHOW STATUS语句可以查看到处在任何数据库中存储过程的状态。其语法格 式为：

**SHOW PROCEDURE STATUS [LIKE ‘匹配字符串];**

如果没有加上LIKE关键字，则查询系统中存在的所有存储过程，如果加了LIKE关键 字，则查询符合匹配字符串的存储过程。

2）使用SHOW CREATE PROCEDURE语句可以查看到任何数据库中指定的存储过程的状 态。其语法格式为：

**SHOW CREATE PROCEDURE 数据库名.存储过程名 ;**

如果是查看当前数据库中则数据库名可以省略。

3）MySQL中存储过程的信息都存储在information\_schema数据库下的Routines表中。 可以通过查询该表的记录来查询存储过程的信息，语法格式为：

**SELECT \* FROM information\_schema.Routines**

**[WHERE ROUTINE\_NAME=’ 存储过程名 ’**

**AND ROUTINE\_TYPE=’ PROCEDURE ’ ;]**

WHERE为可选参数，如未选，则查询系统所有的存储过程。如选，则查询指定存储 过程。

## 修改存储过程

在MySQL中可以使用ALTER语句修改存储过程的特性，其语法格式为：

**ALTER PROCEDURE 存储过程名 [特性1,...] ;**

其特性如下

1. CONTAINS SQL表示子程序中包含SQL语句，但不包含读或写数据的语句；
2. NO SQL表示子程序中不包含SQL语句；
3. READS SQL DATA表示子程序中包含读写数据的语句；
4. MODIFIES SQL DATA表示子程序中包含写数据的语句；
5. SQL SECURITY{ DEFINEY | INVOKER }指明谁有权限来执行；DEFINEY表示只有定义者自 己才能够执行，INVOKER表示调用者可以执行。
6. COMMENT ‘ string ’表示注释信息。

在MySQL中暂还不提供对存储过程代码的修改，如果一定要修改存储过程代码，可以先将存储过程删除之后，再重新编写代码创建一个新的存储过程。

## 删除存储过程

使用DROP语句删除废旧的存储过程。其语法格式为：

**DROP PROCEDURE 存储过程名 ;**

**变量**

在编写存储过程时，有时会使用变量保存处理过程中的值或者在接收调用该存储过 程传入的参数。此时就需要定义变量。这些变量的作用域是在BEGIN...END程序中。

1）要想在存储过程中使用变量，首先需要定义变量。在存储过程中定义变量的语法格 式为：

**DECLARE 变量名 变量类型 [DEFAULT value] ;**

DEFAULT为可选参数，用于定义变量的默认值。

1. 定义变量后，为变量赋值可以改变变量的值，在MySQL中有两种方式为变量 赋值。

·通过SET语句赋值，其语法格式为：

**SET 变量名=值 ;**

·通过SELECT…INTO为一个或多个变量赋值，其语法格式为：

**SELECT 字段名1... INTO 变量名... FROM 表名**

**WHERE 条件表达式 ;**

这种方式只能将指定记录的某字段值赋给变量。

## 流程控制

MySQL中的流程控制语句包括：IF语句、CASE语句、LOOP语句、LEAVE语句、ITERATE 语句、REPEAT语句、WHILE语句。

1. IF语句是根据条件的判断结果来执行相应的语句。其语法格式为：

**IF 条件 THEN SQL语句列表**

**[ ELSEIF 条件 THEN SQL语句列表]**

**[ ELSE SQL语句列表]**

**END IF ;**

SQL语句列表应使用BEGIN...AND括起来。并且，每一个IF语句需要使用END IF 来结束，不可省略。

1. CASE语句是另一个条件判读的语句，其语法格式为：

**CASE 条件**

**WHEN 值 THEN SQL语句列表;**

**[WHEN 值 THEN SQL语句列表]**

**[ELSE SQL语句列表]**

**END CASE ;**

当CASE后面的条件与WHEN后面的值相同时，则其THEN后面的SQL语句列表被执 行

1. WHILE语句用于创建循环。WHILE语句在执行时，先对表达式进行判断，如果为真， 则执行循环体，否则退出循环。其语法格式为：

**[别名：]WHILE 条件表达式**

**SQL语句列表 ;**

**END WHILE [别名];**

1. REPEATE语句用于创建循环。先执行循环体，然后对条件表达式进行判断，如果表 达式为真，则循环接收；否则重复执行循环中的语句。其语法格式为：

**[别名：]REPEAT**

**SQL语句列表 ;**

**UNTIL 条件表达式;**

**END REPEATE[别名] ;**

1. LOOP语句是另一中创建循环的语句，但LOOP语句只是创建循环，并不进行条件判 断，通常在LOOP语句结束语句后面循环调用LOOP循环，在循环体中设置IF语句 跳出循环。其语法格式为：

**[别名：]LOOP**

**SQL语句列表 ;**

**END LOOP [别名] ;**

1. LEAVE语句用于退出循环，一般与放在条件语句中。其语法格式为：

**LEAVE 循环别名 ;**

1. ITERATE语句用于跳出当前循环不再往下执行，开始下一次循环。其语法格式为：

**ITERATE 循环别名 ;**