《一头扎进 Mysql》视频教程

第一章 Mysql 简介及安装和配置

Java1234_小锋

扣扣:527085608

Java1234 官方群 1，2，3：（已满）

Java1234 官方群 4：45982738

第一节：Mysql 简介

百度百科

第二节：Mysql 安装及配置

1，Mysql5.1 下载及安装

2，Mysql 数据库编码配置　utf-8

3，Mysql 图形界面 Sqlyog 下载及安装

《一头扎进 Mysql》视频教程

第二章 Mysql 数据类型简介

Java1234_小锋

扣扣:527085608

Java1234 官方群 1，2，3：（已满）

Java1234 官方群 4：45982738

第一节：整数类型、浮点数类型和定点数类型

1，整数类型

整数类型	字节数	无符号(unsigned)范围	有符号(signed)范围(默认)
TINYINT	**1**	**0~255**	**-128~127**
SMALLINT	**2**	**0~65535**	**-32768~32767**
MEDIUMINT	**3**	**0~16777215**	**-8388608~8388607**
INT	**4**	**0~4294967295**	**-2147483648~2147483647**
INTEGER	**4**	**0~4294967295**	**-2147483648~2147483647**
BIGINT	**8**	**0~18446744073709551615**	**-9223372036854775808 ~9223372036854775807**

2，浮点数类型和定点数类型

类型	字节数	无符号(unsigned)范围	有符号(signed)范围(默认)
FLOAT	**4**	**0,** **1.175494351E-38** **~3.402823466E+38**	**-3.402823466E+38** **~1.175494351E-38,** **0,** **1.175494351E-38** **~3.402823466E+38**
DOUBLE	**8**	**0,** **2.2250738585072014E-308** **~1.7976931348623157E-308**	**-1.7976931348623157E+308** **~2.2250738585072014E-308,** **0,** **2.2250738585072014E-308** **~1.7976931348623157E+308**
DECIMAL(M,D)	**M+2**	**同 Double**	**同 Double**

M 表示：数据的总长度(不包括小数点)；
D 表示：小数位；
例如　decimal(5,2)　　123.45
存入数据的时候，按四舍五入计算

第二节：日期与时间类型

类型	字节数	取值范围	零值
YEAR	1	1910~2155	0000
DATE	4	1000-01-01~9999-12-31	0000:00:00
TIME	3	-838:59:59~838:59:59	00:00:00
DATETIME	8	1000-01-01 00:00:00 ~9999-12-31 23:59:59	0000-00-00 00:00:00
TIMESTAMP	4	19700101080001~20380119111407	00000000000000

第三节：字符串类型

类型	说明
CHAR	固定长度字符串
VARCHAR	可变长度字符串
TEXT	大文本（TINYTEXT,TEXT,MEDIUMTEXT,LONGTEXT）
ENUM	枚举类型（只能取一个元素）
SET	集合类型（能取多个元素）

第四节：二进制类型

类型	说明
BINARY(M)	字节数为 M，允许长度为 0~M 的定长二进制字符串
VARBINARY(M)	允许长度为 0~M 的变长二进制字符串，字节数为值的长度加 1
BIT(M)	M 位二进制数据，最多 255 个字节
TINYBLOB	可变长二进制数据，最多 255 个字节
BLOB	可变长二进制数据，最多（2^{16}-1）个字节
MEDIUMBLOB	可变长二进制数据，最多（2^{24}-1）个字节
LONGBLOB	可变长二进制数据，最多（2^{32}-1）个字节

《一头扎进 Mysql》视频教程

第三章　数据库基本操作

Java1234_小锋

扣扣:527085608

Java1234 官方群 1，2，3：（已满）

Java1234 官方群 4：45982738

第一节：数据库简介

数据库（Database）是按照数据结构来组织、存储和管理数据的仓库；

第二节：显示所有数据库

```
Show databases;
```

第三节：创建数据库

```
Create database  数据库命
```

第四节：删除数据库

```
Drop database  数据库名
```

《一头扎进 Mysql》视频教程

第四章　数据库表基本操作

Java1234_小锋

扣扣:527085608

Java1234 官方群 1，2，3：（已满）

Java1234 官方群 4：45982738

第一节：创建表

表是数据库存储数据的基本单位。个一个表包含若干字段或记录；

语法：

```
CREATE TABLE  表名(  属性名  数据类型  [完整性约束条件],
                     属性名  数据类型  [完整性约束条件],
                     .
                     .
                     属性名  数据表格  [完整性约束条件]
                    );
```

约束条件	说明
PRIMARY KEY	标识该属性为该表的主键，可以唯一的标识对应的记录
FOREIGN KEY	标识该属性为该表的外键，与某表的主键关联
NOT NULL	标识该属性不能为空
UNIQUE	标识该属性的值是唯一的
AUTO_INCREMENT	标识该属性的值自动增加
DEFAULT	为该属性设置默认值

创建图书类别表：t_bookType

```
CREATE TABLE t_booktype(
    id INT PRIMARY KEY AUTO_INCREMENT,
    bookTypeName VARCHAR(20),
    bookTypeDesc VARCHAR(200)
);
```

创建图书表：t_book

```
CREATE TABLE t_book(
id INT PRIMARY KEY AUTO_INCREMENT,
bookName VARCHAR(20),
author VARCHAR(10),
price DECIMAL(6,2),
bookTypeId INT,
CONSTRAINT `fk` FOREIGN KEY (`bookTypeId`) REFERENCES `t_bookType` (`id`)
);
```

第二节：查看表结构

1，查看基本表结构：　DESCRIBE(DESC) 表名；
2，查看表详细结构：　SHOW CREATE TABLE 表名；

第三节：修改表

1，修改表名　ALTER TABLE 旧表名　RENMAE 新表名 ；
2，修改字段　ALTER TABLE 表名　CHANGE 旧属性名　新属性名　新数据类型
3，增加字段　ALTER TABLE 表名　ADD 属性名 1　数据类型 [完整性约束条件] [FIRST | AFTER 属性名 2]
4，删除字段　ALTER TABLE 表名　DROP 属性名

第四节：删除表

1，删除表 DROP TABLE 表名；

《一头扎进 Mysql》视频教程

第五章　查询数据

Java1234_小锋

扣扣:527085608

Java1234 官方群 1，2，3：（已满）

Java1234 官方群 4：45982738

第一节：单表查询

5.1，查询所有字段

1，SELECT 字段 1，字段 2，字段 3...FROM 表名；
2，SELECT * FROM 表名；

5.2，查询指定字段

1，SELECT 字段 1，字段 2，字段 3...FROM 表名；

5.3，Where 条件查询

1，SELECT 字段 1，字段 2，字段 3...FROM 表名 WHERE 条件表达式；

5.4，带 IN 关键字查询

1，SELECT 字段 1，字段 2，字段 3...FROM 表名 WHERE 字段 [NOT] IN (元素 1，元素 2，元素 3)；

5.5，带 BETWEEN AND 的范围查询

1，SELECT 字段 1，字段 2，字段 3...FROM 表名 WHERE 字段 [NOT] BETWEEN 取值 1 AND 取值 2；

5.6，带 LIKE 的模糊查询

1，SELECT 字段 1，字段 2，字段 3...FROM 表名 WHERE 字段 [NOT] LIKE '字符串'；
"%"代表任意字符；
"_" 代表单个字符；

5.7，空值查询

1，SELECT 字段 1，字段 2，字段 3...FROM 表名 WHERE 字段 IS [NOT] NULL；

5.8，带 AND 的多条件查询

1，SELECT 字段 1，字段 2...FROM 表名 WHERE 条件表达式 1 AND 条件表达式 2 [...AND 条件表达式 n]

5.9，带 OR 的多条件查询

1，SELECT 字段 1，字段 2...FROM 表名 WHERE 条件表达式 1 OR 条件表达式 2 [...OR 条件表达式 n]

5.10，DISTINCT 去重复查询

```
SELECT DISTINCT  字段名  FROM 表名；
```

5.11，对查询结果排序

```
SELECT 字段 1，字段 2...FROM 表名  ORDER BY 属性名 [ASC|DESC]
```

5.12，GROUP BY 分组查询

```
GROUP BY 属性名 [HAVING 条件表达式][WITH  ROLLUP]
1，单独使用(毫无意义)；
2，与 GROUP_CONCAT()函数一起使用；
3，与聚合函数一起使用；
4，与 HAVING 一起使用(限制输出的结果)；
5，与 WITH ROLLUP 一起使用(最后加入一个总和行)；
```

5.13，LIMIT 分页查询

```
SELECT 字段 1，字段 2...FROM 表名 LIMIT 初始位置，记录数；
```

第二节：使用聚合函数查询

5.1，COUNT()函数

1，COUNT()函数用来统计记录的条数；
2，与 GOUPE BY 关键字一起使用；

5.2，SUN()函数

1，SUM()函数是求和函数；
2，与 GOUPE BY 关键字一起使用；

5.3，AVG()函数

1，AVG()函数是求平均值的函数；
2，与 GOUPE BY 关键字一起使用；

5.4，MAX()函数

1，MAX()函数是求最大值的函数；
2，与 GOUPE BY 关键字一起使用；

5.5，MIN()函数

1，MIN()函数是求最小值的函数；
2，与 GOUPE BY 关键字一起使用；

第三节：连接查询

连接查询是将两个或两个以上的表按照某个条件连接起来，从中选取需要的数据；

3.1，内连接查询

内连接查询是一种最常用的连接查询。内连接查询可以查询两个或者两个以上的表；

3.2，外连接查询

外连接可以查出某一张表的所有信息；
SELECT 属性名列表 FROM 表名 1 LEFT|RIGHT JOIN 表名 2 ON 表名 1.属性名 1=表名 2.属性名 2；

3.2.1 左连接查询

可以查询出“表名 1”的所有记录，而“表名 2”中，只能查询出匹配的记录；

3.2.2 右连接查询

可以查询出“表名 2”的所有记录，而“表名 1”中，只能查询出匹配的记录；

3.3，多条件连接查询

第四节：子查询

4.1 带 In 关键字的子查询

一个查询语句的条件可能落在另一个 SELECT 语句的查询结果中。

4.2 带比较运算符的子查询

子查询可以使用比较运算符。

4.3 带 Exists 关键字的子查询

假如子查询查询到记录，则进行外层查询，否则，不执行外层查询；

4.4 带 Any 关键字的子查询

ANY 关键字表示满足其中任一条件；

4.5 带 All 关键字的子查询

ALL 关键字表示满足所有条件；

第五节：合并查询结果

5.1 UNION

使用 UNION 关键字是，数据库系统会将所有的查询结果合并到一起，然后去除掉相同的记录；

5.2 UNION ALL

使用 UNION ALL，不会去除掉系统的记录；

第六节：为表和字段取别名

6.1 为表取别名

格式： 表名 表的别名

6.2 为字段取别名

格式： 属性名 [AS] 别名

《一头扎进 Mysql》视频教程

第六章　插入，更新和删除数据

Java1234_小锋

扣扣:527085608

Java1234 官方群 1，2，3：（已满）

Java1234 官方群 4：45982738

第一节：插入数据

1，给表的所有字段插入数据

```
格式：INSERT  INTO 表名  VALUES(值 1，值 2，值 3，...，值 n)；
```

2，给表的指定字段插入数据

```
格式：INSERT  INTO 表名(属性 1，属性 2，...，属性 n)  VALUES(值 1，值 2，值 3，...，值 n)；
```

3，同时插入多条记录

```
INSERT  INTO 表名 [(属性列表)]
        VALUES(取值列表 1)，(取值列表 2)
        ...，
        (取值列表 n)；
```

第二节：更新数据

```
UPDATE 表名
        SET 属性名 1=取值 1，属性名 2=取值 2，
        ...，
        属性名 n=取值 n
        WHERE 条件表达式；
```

第三节：删除数据

```
DELETE FROM 表名 [WHERE  条件表达式]
```

《一头扎进 Mysql》视频教程

第七章　索引

Java1234_小锋

扣扣:527085608

Java1234 官方群 1，2，3：（已满）

Java1234 官方群 4：45982738

第一节：索引的引入

索引定义：索引是由数据库表中一列或者多列组合而成，其作用是提高对表中数据的查询速度；
类似于图书的目录，方便快速定位，寻找指定的内容；

第二节：索引的优缺点

优点：提高查询数据的速度；
缺点：创建和维护索引的时间增加了；

第三节：索引实例

第四节：索引分类

1，普通索引
　　这类索引可以创建在任何数据类型中；
2，唯一性索引
　　使用 UNIQUE 参数可以设置，在创建唯一性索引时，限制该索引的值必须是唯一的；
3，全文索引
　　使用 FULLTEXT 参数可以设置，全文索引只能创建在 CHAR，VARCHAR，TEXT 类型的字段上。主要作用就是提高查询较大字符串类型的速度；只有 MyISAM 引擎支持该索引，Mysql 默认引擎不支持；
4，单列索引
　　在表中可以给单个字段创建索引，单列索引可以是普通索引，也可以是唯一性索引，还可以是全文索引；
5，多列索引
　　多列索引是在表的多个字段上创建一个索引；
6，空间索引
　　使用 SPATIAL 参数可以设置空间索引。空间索引只能建立在空间数据类型上，这样可以提高系统获取空间数据的效率；只有 MyISAM 引擎支持该索引，Mysql 默认引擎不支持；

第五节：创建索引

5.1　创建表的时候创建索引

```
CREATE TABLE  表名 (属性名    数据类型  [完整性约束条件],
                    属性名    数据类型  [完整性约束条件],
                    ....
                    属性名    数据类型
                    [UNIQUE | FULLTEXT | SPATIAL ] INDEX| KEY
                        [别名] (属性名 1 [(长度)] [ASC | DESC])
                    );
```

1，创建普通索引

2，创建唯一性索引

3，创建全文索引

4，创建单列索引

5，创建多列索引

6，创建空间索引

5.2　在已经存在的表上创建索引

```
CREATE [ UNIQUE | FULLTEXT | SPATIAL ] INDEX  索引名
  ON  表名   (属性名  [(长度)] [ ASC | DESC]);
```

5.3　用 ALTER TABLE 语句来创建索引

```
ALTER    TABLE  表名  ADD [ UNIQUE | FULLTEXT | SPATIAL ] INDEX
                索引名  (属性名  [(长度)] [ ASC | DESC]);
```

第六节：删除索引

```
DROP INDEX  索引名  ON  表名  ;
```

《一头扎进 Mysql》视频教程

第八章　　视图

Java1234_小锋

扣扣:527085608

Java1234 官方群 1，2，3：（已满）

Java1234 官方群 4：45982738

第一节：视图的引入

1，视图是一种虚拟的表，是从数据库中一个或者多个表中导出来的表。
2，数据库中只存放了视图的定义，而并没有存放视图中的数据，这些数据存放在原来的表中。
3，使用视图查询数据时，数据库系统会从原来的表中取出对应的数据。

第二节：视图的作用

1，使操作简便化；
2，增加数据的安全性；
3，提高表的逻辑独立性；

第三节：创建视图

```
CREATE [ ALGORITHM ={ UNDEFIEND | MERGE | TEMPTABLE }]
        VIEW  视图名  [ (  属性清单) ]
        AS SELECT  语句
        [ WITH [    CASCADED | LOCAL ] CHECK OPTION ];
```

ALGORITHM 是可选参数，表示视图选择的算法；
“视图名”参数表示要创建的视图的名称；
“属性清单”是可选参数，其指定了视图中各种属性的名词，默认情况下与 SELECT 语句中查询的属性相同；
SELECT 语句参数是一个完整的查询语句，标识从某个表查出某些满足条件的记录，将这些记录导入视图中；
WITH CHECK OPTION 是可选参数，表似乎更新视图时要保证在该视图的权限范围之内；

ALGORITHM 包括 3 个选项 UNDEFINED、MERGE 和 TEMPTABLE。其中，UNDEFINED 选项表示 MySQL 将自动选择所要使用的算法；MERGE 选项表示将使用视图的语句与视图定义合并起来，使得视图定义的某一部分取代语句的对应部分；TEMPTABLE 选项表示将视图的结果存入临时表，然后使用临时表执行语句；CASCADED 是可选参数，表示更新视图时要满足所有相关视图和表的条件，该参数为默认值；LOCAL 表示更新视图时，要满足该视图本身的定义条件即可；

3.1 在单表上创建视图

3.2 在多表上创建视图

第四节：查看视图

4.1 DESCRIBE 语句查看视图基本信息

4.2 SHOW TABLE STATUS 语句查看视图基本信息

4.3 SHOW CREATE VIEW 语句查看视图详细信息

4.3 在 views 表中查看视图详细信息

第五节：修改视图

5.1 CREATE OR REPLACE VIEW 语句修改视图

```
CREATE OR REPLACE [ ALGORITHM ={ UNDEFINED | MERGE | TEMPTABLE }]
                  VIEW 视图名 [( 属性清单 )]
                  AS SELECT 语句
                  [ WITH [ CASCADED | LOCAL ] CHECK OPTION ];
```

5.2 ALTER 语句修改视图

```
ALTER   [ ALGORITHM ={ UNDEFINED | MERGE | TEMPTABLE }]
                  VIEW 视图名 [( 属性清单 )]
                  AS SELECT 语句
                  [ WITH [ CASCADED | LOCAL ] CHECK OPTION ];
```

第六节：更新视图

更新视图是指通过视图来插入（INSERT）、更新（UPDATE）和删除（DELETE）表中的数据。因为视图是一个虚拟的表，其中没有数据。通过视图更新时，都是转换基本表来更新。更新视图时，只能更新权限范围内的数据。超出了范围，就不能更新。

6.1 插入(INSERT)

6.2 更新(UPDATE)

6.3 删除(DELETE)

第七节：删除视图

删除视图是指删除数据库中已存在的视图。删除视图时，只能删除视图的定义，不会删除数据；
DROP VIEW　[IF EXISTS] 视图名列表 [RESTRICT | CASCADE]

《一头扎进 Mysql》视频教程

第九章　　触发器

Java1234_小锋

扣扣:527085608

Java1234 官方群 1，2，3：（已满）

Java1234 官方群 4：45982738

第一节：触发器的引入

触发器（TRIGGER）是由事件来触发某个操作。这些事件包括 INSERT 语句、UPDATE 语句和 DELETE 语句。当数据库系统执行这些事件时，就会激活触发器执行相应的操作。

第二节：创建与使用触发器

2.1 创建只有一个执行语句的触发器

```
CREATE TRIGGER 触发器名  BEFORE | AFTER 触发事件
            ON 表名  FOR EACH ROW  执行语句
```

2.2 创建有多个执行语句的触发器

```
CREATE TRIGGER 触发器名  BEFORE | AFTER 触发事件
            ON 表名  FOR EACH ROW
            BEGIN
                执行语句列表
            END
```

第三节：查看触发器

3.1 SHOW TRIGGERS 语句查看触发器信息

3.2 在 triggers 表中查看触发器信息

第四节：删除触发器

```
DROP TRIGGER 触发器名;
```

《一头扎进 Mysql》视频教程

第十章　　MySQL 常用函数

Java1234_小锋

扣扣:527085608

Java1234 官方群 1，2，3：（已满）

Java1234 官方群 4：45982738

第一节：日期和时间函数

1，CURDATE() 返回当前日期；
2，CURTIME() 返回当前时间；
3，MONTH(d) 返回日期 d 中的月份值，范围是 1~12

第二节：字符串函数

1，CHAR_LENGTH(s) 计算字符串 s 的字符数；
2，UPPER(s) 把所有字母变成大写字母；
3，LOWER(s) 把所有字母变成小写字母；

第三节：数学函数

1，A BS(x) 求绝对值
2，SQRT(x) 求平方根
3，MOD(x,y) 求余

第四节：加密函数

1，PASSWORD(str) 一般对用户的密码加密 不可逆
2，MD5(str) 普通加密 不可逆
3，ENCODE(str，pswd_str) 加密函数，结果是一个二进制数，必须使用 BLOB 类型的字段来保存它；
4，DECODE(crypt_str，pswd_str) 解密函数；

《一头扎进 Mysql》视频教程

第十一章　　存储过程和函数

Java1234_小锋

扣扣:527085608

Java1234 官方群 1，2，3：（已满）

Java1234 官方群 4：45982738

第一节：存储过程和函数的引入

存储过程和函数是在数据库中定义一些 SQL 语句的集合，然后直接调用这些存储过程和函数来执行已经定义好的 SQL 语句。存储过程和函数可以避免开发人员重复的编写相同的 SQL 语句。而且，存储过程和函数是在 MySQL 服务器中存储和执行的，可以减少客户端和服务器端的数据传输；

第二节：创建存储过程和函数

2.1 创建存储过程

CREATE PROCEDURE sp_name([proc_parameter[,...]])
[characteristic...] routine_body
sp_name 参数是存储过程的名称；
proc_parameter 表示存储过程的参数列表；
characteristic 参数指定存储过程的特性；
routine_body 参数是 SQL 代码的内容，可以用 BEGIN...END 来标志 SQL 代码的开始和结束。
proc_parameter 中的每个参数由 3 部分组成。这 3 部分分别是输入输出类型、参数名称和参数类型。
[IN | OUT | INOUT] param_name type
其中，IN 表示输入参数；OUT 表示输出参数；INOUT 表示既可以是输入，也可以是输出；param_name 参数是存储过程的参数名称；type 参数指定存储过程的参数类型，该类型可以是 MySQL 数据库的任意数据类型；
Characteristic 参数有多个取值。其取值说明如下：
LANGUAGE SQL：说明 routine_body 部分是由 SQL 语言的语句组成，这也是数据库系统默认的语言。
[NOT] DETERMINISTIC ：指明存储过程的执行结果是否是确定的。DETERMINISTIC 表示结果是确定的。每次执行存储过程时，相同的输入会得到相同的输出。NOT DETERMINISTIC 表示结果是非确定的，相同的输入可能得到不同的输出。默认情况下，结果是非确定的。
{ CONTAINS SQL | NO SQL | READS SQL DATA | MODIFIES SQL DATA } ：指明子程序使用 SQL 语句的限制；CONTAINS SQL 表示子程序包含 SQL 语句，但不包含读或写数据的语句；NO SQL 表示子程序中不包含 SQL 语句；READS SQL DATA 表示子程序中包含读数据的语句；MODIFIES SQL DATA 表示子程序中包含写数据的语句。默认情况下，系统会指定为 CONTAINS SQL；
SQL SECURITY { DEFINER | INVOKER }；指明谁有权限来执行。DEFINER 表示只有定义者自己才能够执行；INVOKER 表示调用者可以执行。默认情况下，系统指定的权限是 DEFINER。
COMMENT ‘string’ ：注释信息；

2.2 创建存储函数

```
CREATE FUNCTION sp_name ( [func_parameter[,...]] )
        RETURNS type
        [ characteristic... ] routine_body
```

sp_name 参数是存储函数的名称；func_parameter 表示存储函数的参数列表；RETURNS type 指定返回值的类型；characteristic 参数指定存储过程的特性，该参数的取值与存储过程中的取值是一样的；routine_body 参数是 SQL 代码的内容，可以用 BEGIN...END 来标志 SQL 代码的开始和结束；

func_parameter 可以由多个参数组成，其中每个参数由参数名称和参数类型组成，其形式如下：

param_name type　其中，param_name 参数是存储函数的参数名称；type 参数指定存储函数的参数类型，该类型可以是 MySQL 数据库的任意数据类型；

2.3 变量的使用

1，定义变量

```
DECLARE var_name [,...]  type [ DEFAULT value ]
```

2，为变量赋值

```
SET var_name = expr [,var_name=expr] ...
```

```
SELECT col_name[,...] INTO var_name[,...]
        FROM table  name WHERE condition
```

2.4 游标的使用

查询语句可能查询出多条记录，在存储过程和函数中使用游标来逐条读取查询结果集中的记录。游标的使用包括声明游标、打开游标、使用游标和关闭游标。游标必须声明在处理程序之前，并且声明在变量和条件之后。

1，声明游标

```
DECLARE cursor_name CURSOR FOR select_statement ;
```

2，打开游标

```
OPEN cursor_name;
```

3，使用游标

```
FETCH cursor_name INTO var_name [,var_name ... ];
```

4，关闭游标

```
CLOSE cursor_name;
```

2.5 流程控制的使用

存储过程和函数中可以使用流程控制来控制语句的执行。MySQL 中可以使用 IF 语句、CASE 语句、LOOP 语句、LEAVE 语句、ITERATE 语句、REPEAT 语句和 WHILE 语句来进行流程控制。

1，IF 语句

```
IF search_condition THEN statement_list
    [ ELSEIF search_condition THEN statement_list ]...
    [ ELSE statement_list ]
END IF
```

2，CASE 语句

```
CASE case_value
    WHEN when_value THEN statement_list
    [WHEN when_value THEN statement_list]...
    [ELSE statement_list ]
END CASE
```

3，LOOP，LEAVE 语句

LOOP 语句可以使某些特定的语句重复执行，实现一个简单的循环。但是 LOOP 语句本身没有停止循环的语句，必须是遇到 LEAVE 语句等才能停止循环。LOOP 语句的语法的基本形式如下：

```
[begin_label：]LOOP
        Statement_list
END LOOP [ end_label ]
```

LEAVE 语句主要用于跳出循环控制。语法形式如下：

```
LEAVE label
```

4，ITERATE 语句

ITERATE 语句也是用来跳出循环的语句。但是，ITERATE 语句是跳出本次循环，然后直接进入下一次循环。基本语法：

```
ITERATE label ;
```

5，REPEAT 语句

REPEAT 语句是有条件控制的循环语句。当满足特定条件时，就会跳出循环语句。REPEAT 语句的基本语法形式如下：

```
[ begin_label : ] REPEAT
        Statement_list
        UNTIL search_condition
  END REPEAT [ end_label ]
```

6，WHILE 语句

```
[ begin_label : ] WHILE search_condition DO
        Statement_list
END WHILE [ end_label ]
```

第三节：调用存储过程和函数

第四节：查看存储过程和函数

第五节：修改存储过程和函数

第六节：删除存储过程和函数