

存储过程

简介

存储过程(stored procedure)是一组预先编译好的Transact-SQL代码。它可以对数据库进行查询和更新。存储过程类似计算机高级语言的过程，也可以带参数并返回处理结果。

- 存储过程的优点
 1. 执行速度快
 2. 有利于应用程序执行SQL语句
 3. 减少网络通信量
- 存储过程的分类
 1. 系统存储过程
 2. 用户自定义存储过程
 3. 扩展存储过程

使用

1. 无参存储过程

```
-- 创建
delimiter $$;    -- 定义以 $$ 作为结束标志
CREATE PROCEDURE sp_fun1()
begin
    select * from t_student;
end;$$
delimiter;

-- 调用
call sp_fun1();

-- 查看
show PROCEDURE CODE sp_fun1;

-- 删除
drop PROCEDURE if exists sp_fun1;
```

2. 带参存储过程

```

drop PROCEDURE sp_fun2;

delimiter $$;
CREATE PROCEDURE sp_fun2(in id varchar(35)) -- in表示传入参数，变量id类型
varchar(35)
begin
    select * from t_student where student_id = id;
end;$$
delimiter;

call sp_fun2('s01');

```

3. 传出参数

```

CREATE PROCEDURE sp_fun3(out _count int)    -- out表传出参数，变量_count类型int
begin
    select count(*) into _count from t_student;
end;$$
delimiter;

call sp_fun3(@_count);
select @_count;

```

例子

```

-- 循环添加数据
delimiter $$;
CREATE PROCEDURE sp_fun4()
begin
    declare i int;
    set i = 1;
    while (i <= 5) do
        insert into t_student values(CONCAT('s', i), CONCAT('name', i), 'cr01');
        set i = i + 1;
    end while;
end;$$
delimiter;

call sp_fun4(); -- 调用循环添加数据

delimiter $$;
CREATE PROCEDURE sp_fun5(in flag int)
begin
    if (flag = 1) then
        select *from t_student;
    end if;
end;$$
call sp_fun5(0);

```

触发器

简介

触发器(trigger)是一种特殊的存储过程，它是基于表创建的并可以看作是表定义的一部分。

- 触发器的作用
 1. 禁止无效的修改
 2. 级联修改相关表格
 3. 执行较为复杂的约束操作
- 触发器的优点
 1. 维护数据库中的数据完整性
 2. 自动跟踪对表的更新情况

使用

```
delimiter $$;  
CREATE TRIGGER tri_fun1  
BEFORE INSERT  
ON t_student FOR EACH ROW  
begin  
    SET NEW.student_id = CONCAT('2019c', NEW.student_id);  
end;$$  
delimiter;  
  
show triggers from student; -- 查询触发器
```

事务

简介

事务日志是一个数据库的数据修改历史，在创建一个数据库时，会自动创建一个事务日志文件采用提前写的方式，即对数据库的修改，首先会写入到日志中，然后才写入数据库。每一个数据修改都会标出一个开始点和结束点，如果由于各种原因未达到结束点，则会恢复前面所做的所有操作

使用

```
-- 事务 transaction
-- 开始事务 START TRANSACTION
-- 提交事务 commit
-- 回滚 rollback

START TRANSACTION; -- 开始事务

-- 执行的操作
insert into t_student values('03', 'name01', 'cr01');
insert into t_student values('04', 'name01', 'cr01');

commit; -- 提交
rollback; -- 回滚
```