

CH9 SQL Environment & Persistent Stored Modules

SQL环境与持续存储模块

一、Three-Tier Architecture

1. 结构

- Web servers 为用户提供接口
- Application servers 执行逻辑事务调度
- Database servers 提供应用层所需接口

2. Distributed Database & Centralized database

集中式数据库：访问速度快、可扩展性好

分布式数据库：更高的访问速度、更强的可扩展性、更高的并发访问量

3. Distributed Database

- 物理分布，逻辑集中 DDBS
- 物理分布，逻辑分布 FDBS 联邦式

二、SQL Environments

- Schemas 表、视图、断言、触发器等信息的集合
- Catalogs Schemas的集合
- Cluster Catalogs的集合
- Connections 连接
- Sessions 数据库操作
- Modules

三、Stored Procedures

1. Definition

把功能强大的过程放到Data Server端，使得所有客户端可调用(Reuse)

2. Creating

```
1 CREATE PROCEDURE <name>(<parameter list>)      -- 过程
2 CREATE FUNCTION <name>(<parameter list>) RETURNS -- 函数
```

mode-name-type元组：IN s CHAR

IN = input only, does not change value, is the default and can be omitted

OUT = output only

INOUT = both

函数的参数只可以使用IN

3. Invoking PSM

```
1 CALL <procedure name>(<argument list>);
```

函数可以在任意时刻调用，只要返回值正确。

4. PSM Statements

```
1 RETURN <expression>          -- 不会结束函数执行
2 DECLARE <name> <type>         -- 使用之前需要声明
3 BEGIN ... END;                -- 过程块
4 SET <variable> = <expression>; -- 赋值
5 Statement labels              -- IF LOOP等
```

5. IF

```
1 IF <condition> THEN
2     <statements>
3 ELSEIF <condition> THEN
4     <statements>
5 ELSE
6     <statements>
7 END IF;
```

6. LOOP

```
1 -- 普通循环
2 <loop name>: LOOP
3     <statements>
4 END LOOP;
5
6 LEAVE <loop name>;          -- break
7
8 -- for循环
9 FOR <loop name> AS <cursor name> CURSOR FOR
10    <query>
11 DO
12    <statements>
13 END FOR;
14
15 -- while循环
16 WHILE <condition> DO
17    <statements>
18 END WHILE;
19
20 -- do-while循环
21 REPEAT
22    <statements>
23 UNTIL <condition>
24 END REPEAT;
```

7. Query

使用情况：

- 子查询可在任意合法地方使用
- 产生一个值的查询可以成为赋值语句的右值
- 单行SELECT ... INTO
- 可在游标中使用

SELECT ... INTO

将SELECT返回的单条结果通过INTO赋值给目标变量。

```
1 | SELECT price INTO p FROM sells WHERE bar='Joe' AND beer='Bud';
```

8. Cursors

游标可以遍历查询结果的所有元组（迭代器）

```
1 | DECLARE c CURSOR FOR <query>;    -- 声明
2 |
3 | OPEN c;    -- 打开
4 | CLOSE c;  -- 关闭
5 |
6 | FETCH FROM c INTO x_1, x_2, ..., x_n;    -- next(iter)将下一条元组信息返回到后面的x
    序列中
```

9. Breaking Cursor Loops

每条SQL语句返回一个状态（5位字符串）：

00000 = "OK"

02000 = "Failed to find a tuple"

可以通过 SQLSTATE 获取状态

```
1 | DECLARE NotFound CONDITION FORM SQLSTATE '02000';
2 |
3 | Loop: LOOP
4 | FETCH ...
5 | IF NotFound THEN LEAVE Loop;
6 | END Loop;
```

10. Exceptions

错误处理

```
1 | DECLARE <where to go> HANDLER FOR <condition list> <statement>
```

where to go : CONTINUE, EXIT, UNDO

condition list : 激活处理器的条件

statement : 错误发生时执行的语句

```
1 DECLARE EXIT HANDLER FOR NotFound, ToMany
2     RETURN NULL;
```

11. Call

```
1 INSERT INTO StarsIn(movieTitle, movieYear, starName)
2 VALUES('a', Getyear('a'), 'Dc');
```