实用数据库5

2018年3月13日 8:30

1. drop table t1 purge;
2. 循环插入 —— SQL无法支持，循环是一种控制结构
3. PL/SQL —— 块结构
    a. 声明部分 （可选）
    b. 可执行部分（必须）
    c. 异常处理部分 （可选）

4. 上述三个块的分类
    a. 匿名块 —— 不作为数据库的对象，不保存在数据库中

    Declare

    …

    声明部分：定义类型、游标

    …

    Begin

    …

    可执行部分

    …

    End

    ```
    BEGIN
      FOR i IN 1..10 loop
         INSERT INTO t1 values(i);
      END loop;
      COMMIT;
    END;
    ```

    ```
    SQL> BEGIN
      2    FOR i IN 1..10 loop
      3        INSERT INTO t1 values(i);
      4    END loop;
      5    COMMIT;
      6  END;
      7  /

    PL/SQL 过程已成功完成。
    ```

```
SQL> CALL p2();
调用完成。
SQL> EXECUTE p2;
PL/SQL 过程已成功完成。
```

创建调用过程：

```
create or replace procedure p1
AS
BEGIN
  FOR i IN 1..10 loop
     INSERT INTO t1 values(i);
  END loop;
  COMMIT;
END;
```

b. 命名块 —— 如存储过程、函数、package

5. 循环
   a. 绝对循环
   b. while循环
6. PL/SQL 可以嵌套SQL语句
7. 子查询
   a. 嵌套子查询
   b. 关联子查询

```
SQL> select a.id, b.id
  2  from t2 a LEFT OUTER JOIN t2 b
  3  on a.id = b.id - ROUND((select count(*) from t2)/2)
  4  where a.id <= ROUND((select count(*) from t2)/2);

        ID         ID
---------- ----------
         1          7
         2          8
         3          9
         4         10
         5         11
         6

已选择6行。
```

8. SQL 中运算尽可能放在右侧：

SELECT * FROM …

WHERE salary > (select MAX(…) from …);

9. Manager_id 存在NULL的情况

```
SQL> select last_name from employees
  2  where employee_id NOT IN(select NVL(manager_id, 0) from employees);
```

```
SQL> select last_name from employees e
  2  where not exists (select * from employees
  3                    where e.employee_id = manager_id);
```

10. 和用户交互：提交变量
    a. & 操作符

```
SQL> select empno, ename from emp where empno = &工号;
输入 工号 的值:  7566
原值    1: select empno, ename from emp where empno = &工号
新值    1: select empno, ename from emp where empno = 7566

     EMPNO ENAME
---------- ----------
      7566 JONES
```

```
SQL> select empno, ename from emp where ename = '&姓名';
输入 姓名 的值:  SMITH
原值    1: select empno, ename from emp where ename = '&姓名'
新值    1: select empno, ename from emp where ename = 'SMITH'

     EMPNO ENAME
---------- ----------
      7369 SMITH

SQL> select empno, ename from emp where ename = UPPER('&姓名');
输入 姓名 的值:  smith
原值    1: select empno, ename from emp where ename = UPPER('&姓名')
新值    1: select empno, ename from emp where ename = UPPER('smith')

     EMPNO ENAME
---------- ----------
      7369 SMITH
```

    b. Define 命令

```
SQL> define xs = sal
SQL> select &xs from emp;
原值    1: select &xs from emp
新值    1: select sal from emp

       SAL
----------
      2800
      1600
      1250
      2975
      1250
      2850
      2450
      3000
      5000
      1500
      1100

       SAL
----------
       950
      3000
      1300

已选择14行。
```

c. &&

```
SQL> select empno, ename, &c3
  2  from emp
  3  order by &c3;
输入 c3 的值:  sal
原值    1: select empno, ename, &c3
新值    1: select empno, ename, sal
输入 c3 的值:  sal
原值    3: order by &c3
新值    3: order by sal

     EMPNO ENAME                SAL
---------- ---------- ----------
      7900 JAMES                950
      7876 ADAMS               1100
      7521 WARD                1250
      7654 MARTIN              1250
      7934 MILLER              1300
      7844 TURNER              1500
      7499 ALLEN               1600
      7782 CLARK               2450
      7369 SMITH               2800
      7698 BLAKE               2850
      7566 JONES               2975

     EMPNO ENAME                SAL
---------- ---------- ----------
      7902 FORD                3000
      7788 SCOTT               3000
      7839 KING                5000
```

```
SQL> select empno, ename, &&c3
  2  from emp
  3  order by &c3;
```

```
SQL> select empno, ename, &&c3
  2  from emp
  3  order by &c3;
输入 c3 的值:  sal
原值    1: select empno, ename, &&c3
新值    1: select empno, ename, sal
原值    3: order by &c3
新值    3: order by sal
```

此过程中先定义了c3

```
SQL> define c3
DEFINE C3                 = "sal" (CHAR)
```

11. 一次更新多列:

```
SQL> UPDATE employees
  2  SET (job_id, salary) = (SELECT job_id, salary
  3                          FROM employees
  4                          WHERE employee_id = 205)
  5  WHERE employee_id = 114;

已更新 1 行。
```

12. 限制性插入
13. 默认值
    a. 默认的插入为NULL

```
SQL> INSERT INTO demo1 values(6, DEFAULT);

已创建 1 行。
```

更改此表某列的默认值

```
SQL> ALTER TABLE demo1 modify name DEFAULT 'Frank';

表已更改。

SQL> INSERT INTO demo1 values(7, DEFAULT);

已创建 1 行。

SQL> select * from demo1;

        ID NAME
---------- --------------------
         1 ABCDE
         2 AB_DE
         3 AB%DE
         4 AB__E
         5 AB_%E
         6
         7 Frank

已选择7行。
```

红框中的alter语句时DDL语句，默认会自动COMMIT。（DDL、DCL都自动提交）