四、DML语言--数据的更新

# 第一节 数据更新

## 数据操作语言DML（Data Manipulation Language）

DML：用于操作数据库中的数据，包括：

* 1. 插入新数据（INSERT）
  2. 修改已有数据（UPDATE）
  3. 删除不再需要的数据（DELETE）
  4. 数据合并（MERGE）

## 数据插入(INSERT语句)

在DML语言中，除了对数据的查询操作外，还有对数据的更新操作，其语法格式是：

|  |
| --- |
| INSERT INTO 表名称[(字段1，字段2，…)]VALUES(值1，值2，…); |

更新数据，需要对不同的数据类型采用其各自的格式，以下为常用的几种数据类型的格式：

|  |  |
| --- | --- |
| 数字 | 直接写数字 |
| 字符串 | 用单引号括住表示，注意，要用英文格式的单引号 |
| 日期 | 按照数据库默认的格式字符串表示日期，例如：28-12月-11 |
| 用TO\_DATE函数，将字符串变为日期格式数据； |
| 若设置时间为当前系统时间或与之有关联，则可以用SYSDATE表示； |

### 补：Oracle中表的复制操作:此操作为Oracle数据库中的特有操作；

|  |
| --- |
| **CREATE TABLE myemp AS (SELECT \* FROM emp);** |
|  |

**注**：

* insert语句每次只能向表中插入一条记录；
* 缺省字段名列表时，应为新插入记录中的每个字段设定新值，且其字段顺序与表中字段顺序一致；
* 可在insert语句中指定赋值字段列表，只为部分字段显式设定新值，其余字段将被缺省赋值为null；

插入数据操作，在myemp表中插入几条记录；

1）插入部分数据（缺少mgr）

|  |
| --- |
| INSERT INTO myemp(empno,ename,job,hiredate,sal,comm,deptno)  VALUES(8000,'李兵','程序员',TO\_DATE('1990-1-11','yyyy-mm-dd'),5000,1000,30); |

2）插入全部数据

|  |
| --- |
| INSERT INTO myemp(empno,ename,job,mgr,hiredate,sal,comm,deptno)  VALUES(8000,'李兵','程序员',7369,TO\_DATE('1990-1-11','yyyy-mm-dd'),5000,1000,30); |

3）省略字段列表

|  |
| --- |
| INSERT INTO myemp  VALUES(8001,'易小龙','程序员',7369,TO\_DATE('1990-1-11','yyyy-mm-dd'),5000,1000,30); |
|  |
|  |

**注**：使用省略字段列表的写法插入数据，只能按照字段在数据库中的顺序插入数据，且必须插入全部数据，使用这种方式即不方便，也不利于维护，故不推荐使用；

INSERT语句也可以用于子查询；此时不必再给出values子句，子查询中的值列表应与insert子句中的字段列表相对应；

1. 创建myemp表并复制emp表中的数据；

|  |
| --- |
| DROP TABLE myemp PURGE;  CREATE TABLE myemp AS (  SELECT \* FROM emp WHERE 1=2  );  INSERT INTO myemp (SELECT \* FROM emp); |
|  |

## 数据更新（UPDATE语句）

更新是指对数据库表中已有数据进行修改的操作；

语法：

|  |
| --- |
| UPDATE 表名称  SET 更新字段1=更新值1，更新字段2=更新值2，…  [WHERE 更新条件(s)] |

1. 修改全部记录，更新数据表myemp，使所有员工的薪水的增加50%；

|  |
| --- |
| UPDATE myemp SET sal=sal\*1.5; |

**注**：若数据修改未指定条件，则表示修改所有记录中的数据；

1. 修改特定记录，更新数据表myemp，将8001号雇员的职务改为‘项目经理’，薪水改为7500，

去掉奖金；

|  |
| --- |
| UPDATE myemp SET sal=7500,job='项目经理',comm=null WHERE empno=8001; |
|  |

**注**：

* 更新操作的性能并不是很好，如果数据量极大，则其性能会受到影响；
* update语句每次可更新多条记录；
* 可使用where子句限定要更新的记录，如果缺省where子句，则更新表中的所有记录；

## 数据删除（DELETE）

删除是指从数据库表中删除部分记录或全部记录；

格式：

|  |
| --- |
| **DELETE FROM myemp**  **[WHERE 删除条件(s)];** |

1. 删除myemp表中，所有在1987年雇佣的雇员；

|  |
| --- |
| DELETE FROM myemp WHERE TO\_CHAR(hiredate,'yyyy')='1987'; |
|  |

1. 删除myemp表中的全部数据；

|  |
| --- |
| DELETE FROM myemp; |
|  |

**注**：

* delete语句每次可删除多条记录；
* 可使用where子句限定要删除的记录，如果缺省where子句，则删除表中的所有记录；
* 删除操作要尽可能的少用，在系统开发中提供删除操作时，应该提示用户是否确认删除，以防数据被误删；
* 与数据查询相比，数据更新操作更加有风险，故操作时要更加慎重、仔细，在系统开发中应提供给用户确认提示，并提供相应机制恢复数据；

## 数据合并（MERGE语句）

merge语句用于进行数据合并——根据条件在表中执行数据的修改或插入操作，如果要插入的记录在

目标表中已经存在，则执行更新操作、否则执行插入操作；语法：

|  |
| --- |
| **MERGE INTO TABLE [ALIAS]**  **USING (TABLE | VIEW | SUB\_QUERY) [ALIAS]**  **ON (JOIN\_CONDITION)**  **WHEN MATCHED THEN**  **UPDATE SET COL1 = COL1\_VAL, COL2 = COL2\_VAL**  **WHEN NOT MATCHED THEN**  **INSERT (COLUMN\_LIST) VALUES (COLUMN\_VALUES);** |

1. MERGE语句示例：

|  |
| --- |
| CREATE TABLE test1(EID NUMBER(10), name VARCHAR2(20),birth date,salary NUMBER(8,2));  INSERT INTO test1 VALUES (1001, '张三', '20-5月-70', 2300);  INSERT INTO test1 VALUES (1002, '李四', '16-4月-73', 6600);  SELECT \* FROM test1;  CREATE TABLE test2(eid NUMBER(10), name VARCHAR2(20),birth date,salary NUMBER(8,2));  SELECT \* FROM test2;  MERGE INTO test2  USING test1  ON(test1.EID = test2.EID )  WHEN MATCHED THEN  UPDATE SET name = test1.name, birth = test1.birth, salary = test1.SALARY  WHEN NOT MATCHED THEN  INSERT (eid, name, birth, salary) VALUES(test1.eid, test1.name, test1.birth, test1.salary);  SELECT \* FROM test2; |
|  |