



Базы данных и Информационные системы

10/15 **Data**Manipulation**L**anguage

Кузиков Б.О.
Сумы, СумГУ
2013



Задачи

- ▶ **Сегодня мы должны узнать:**
 - ▶ Как вставить данные в таблицу
 - ▶ Как обновить данные в таблице
 - ▶ Как удалить данные из таблицы

Основные выражения DML

- ▶ Select
- ▶ Insert
- ▶ Update
- ▶ Delete
- ▶ Merge

INSERT

- ▶ Для вставки строк в таблицу используется предложение INSERT:

```
INSERT INTO      table [(column [, column...])]  
VALUES          (value [, value...]);
```

- ▶ При этом за один раз вставляется только одна строка



Insert into ... values (...)

- ▶ Insert into ... values (...) вставляет 1 строку

```
SQL> SELECT INTO DEPT (deptno,dname)
      2      VALUES (1, 'New dep' )
```

- ▶ В качестве значений можно использовать
 - ▶ Выражения (в т.ч. с функциями)
 - ▶ Null
 - ▶ Default

```
SQL> SELECT INTO DEPS (deptno, dname,loc)
      2      VALUES (20, 'New dep', DEFAULT)
```

Вставка строк с NULL значениями

- ▶ Неявный метод: просто пропустить столбцы с NULL-значениями

```
INSERT INTO      departments (department_id,  
                               department_name)  
VALUES (30, 'Purchasing');
```

1 rows inserted

- ▶ Явный способ: вместо значений подставить константу NULL

```
INSERT INTO      departments  
VALUES (100, 'Finance', NULL, NULL);
```

Insert into ...

- ▶ Если в новой строке указаны значения всех атрибутов перечень атрибутов можно не указывать

```
SQL> SELECT INTO DEPT (deptno,dname,loc)
      2          VALUES (1, 'New dep', null)
```

=

```
SQL> SELECT INTO DEPT
      2          VALUES (1, 'New dep', null)
```

Вставка специальных значений

- ▶ Функция SYSDATE содержит текущую дату

```
INSERT INTO employees (empno, ename,  
    hire_date, mng, depno)  
VALUES      (113, 'Louis',  
            SYSDATE, 205, 110);
```

1 rows inserted

Вставка определенных значений времени

```
INSERT INTO employees (empno, ename,  
    hire_date, mng, depno)  
VALUES      (113, 'Louis',  
    TO_DATE('FEB 3, 2013', 'MON DD, YYYY'),  
    205, 110);
```

1 rows inserted



Вставка значений из запроса

- ▶ Команда вставки значений из запроса имеет синтаксис:

```
INSERT INTO      table [(column [, column...])]  
select-query
```

- ▶ Таким образом можно вставить несколько строк



Копирование строк из другой таблицы

▶ Пример:

```
INSERT INTO sales_reps(id, name, salary, commission_pct)
SELECT empno, ename, salary, commission_pct
FROM emp
WHERE job LIKE '%REP%';
```

4 rows inserted

▶ Помните:

- ▶ При такой записи не используется предложение VALUES
- ▶ Количество и тип столбцов таблицы, в которую вставляет значения INSERT должно совпадать с столбцами подзапроса



UPDATE

- ▶ Для обновления существующих значений используется предложение UPDATE:

```
UPDATE    table  
SET       column = value [, column = value, ...]  
[WHERE    condition];
```

- ▶ UPDATE может изменять несколько строк за один раз (если это требуется).



Updating Rows in a Table

- ▶ Для того чтобы указать, какие строки нужно обновить используется условие WHERE:

```
UPDATE emp  
SET      depno = 50  
WHERE    empno = 113;
```

1 rows updated

- ▶ Если не указать условие WHERE – обновятся значения всех строк:

```
UPDATE    copy_emp  
SET       depno = 110;
```

22 rows updated

- ▶ Указание SET *column_name* = NULL устанавливает значение атрибута в NULL.



Использование подзапросов при обновлении

- Установить служащему 113 должность и зарплату служащего 205.

```
UPDATE emp
SET job = (SELECT job
            FROM emp
            WHERE empno = 205),
    salary = (SELECT salary
              FROM emp
              WHERE empno = 205)
WHERE empno = 113;
```

1 rows updated

Обновление строк на основе других таблиц

► Пример

```
UPDATE copy_emp
SET   depno = (SELECT depno
                FROM emp
                WHERE empno = 100)
WHERE job = (SELECT job
              FROM emp
              WHERE empno = 200);
```

1 rows updated

DELETE

- ▶ Предложение DELETE удаляет строки из таблицы:

```
DELETE [FROM]    table  
[WHERE           condition];
```



Удаление строк из таблицы

- ▶ Для определения, какие строки удалятся используйте WHERE:

```
DELETE FROM departments  
WHERE department_name = 'Finance';
```

```
1 rows deleted
```

- ▶ Если не указать условия удаления удалятся все строки:

```
DELETE FROM copy_emp;
```

```
22 rows deleted
```



Условие удаления может содержать по-запрос

```
DELETE FROM employees
WHERE department_id =
    (SELECT department_id
     FROM departments
     WHERE department_name
           LIKE '%Public%');
```

```
1 rows deleted
```



Delete vs Drop

- ▶ В чем разница между:

```
SQL> drop table departments;  
SQL> delete from departments;
```

TRUNCATE

- ▶ Очищает таблицу
- ▶ Относится к data definition language (DDL), а не DML
- ▶ Синтаксис:

```
TRUNCATE TABLE table_name;
```

- ▶ Пример:

```
TRUNCATE TABLE copy_emp;
```



Merge

```
SQL> select * from master_tab;
```

```
PID SALES STATUS
```

```
-----  
1      12     CURR
```

```
2      13     NEW
```

```
3      15     CURR
```

```
SQL> select * from delta_tab;
```

```
PID SALES STATUS
```

```
-----  
2      24     CURR
```

```
3       0     OBS
```

```
4      42     CURR
```

```
SQL>
```

Merge

```
SQL> MERGE INTO master_tab m
      2          USING delta_tab d
      3          ON (m.pid = d.pid)
      4      WHEN MATCHED
      5      THEN  UPDATE set m.sales = m.sales+d.sales
      6                  , m.status = d.status
      7          DELTET where m.status = 'OBS'
      8      WHEN NOT MATCHED
      9      THEN INSERT VALUES (d.pid,d.sales,'NEW');
```

3 rows merged.

```
SQL> select * from master_tab;
```

PID	SALES	STATUS
-----	-------	--------

1	12	CURR
---	----	------

2	37	CURR
---	----	------

4	42	NEW
---	----	-----

```
SQL>
```