

# **INSERT. UPDATE. DELETE. TRUNCATE**

**Сичная Мария,  
начальник Бюро (кастомизации)  
платежных приложений**

Синтаксис SQL оператора INSERT при вставке одной записи в таблицу:

```
INSERT INTO table  
(column1, column2, ... )  
VALUES  
(expression1, expression2, ... );
```

**table** – таблица, в которую нужно вставить записи

**column1, column2** – столбцы в table для вставки значений

**expression1, expression2** – эти значения присваиваются столбцам в таблице: expression1 -> column1, expression2 -> column2

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

```
INSERT INTO dept  
VALUES (50, 'DEVELOPMENT', 'MAGNITOGORSK');
```

```
INSERT INTO dept (dname, deptno) VALUES ('ENGINEERING', 60);
```

```
INSERT INTO dept (deptno) VALUES (70);
```

```
INSERT INTO dept VALUES (80, null, null);
```

Синтаксис Oracle оператора INSERT при вставке нескольких записей с помощью оператора SELECT:

```
INSERT INTO table  
(column1, column2, ... column_n )  
SELECT expression1, expression2, ... expression_n  
FROM source_table  
[WHERE conditions];
```

***source\_table*** — исходная таблица, из которой будут вставляться данные

***[WHERE conditions]*** — необязательный. Условия, которые должны быть выполнены для вставляемых записей

```
CREATE TABLE dept_copy AS  
(SELECT * FROM dept  
WHERE 1 = 0); - копирование таблицы  
без данных
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

```
INSERT INTO dept_copy  
SELECT * FROM dept;
```

```
CREATE TABLE dept_mgn AS  
(SELECT deptno, dname FROM dept WHERE 1=0); -  
копирование конкретных столбцов из таблицы без данных
```

```
INSERT INTO dept_mgn (deptno, dname)  
SELECT deptno, dname FROM dept  
WHERE loc = 'MAGNITOGORSK';
```

Синтаксис SQL оператора INSERT при вставке одной записи в таблицу:

***UPDATE table  
SET column1 = expression1,  
column2 = expression2  
[WHERE conditions];***

***table*** – таблица, в которой меняются данные

***column1, column2*** – столбцы, которые нужно обновить

***expression1, expression2*** – новые значения для ***column1, column2***

***[WHERE conditions]*** – необязательный. Условия, которые должны быть выполнены для обновления столбцов. Если не предусмотрено никаких условий, то **все записи в таблице будут обновлены!**

<https://oracleplsql.ru/update.html>

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

```
UPDATE dept  
  SET loc = 'WASHINGTON'  
WHERE deptno = 10;
```

```
UPDATE dept  
  SET dname = 'ENGINEERING',  
      loc = 'MAGNITOGORSK'  
WHERE deptno > 10;
```

Синтаксис Oracle оператора UPDATE при обновлении одной таблицы с данными из другой таблицы:

```
UPDATE table1  
SET column1 = (SELECT expression1  
FROM table2  
WHERE conditions)  
[WHERE conditions];
```

в таблице **table1** меняется столбец **column1** значением **expression1** из таблицы **table2** с условием **WHERE conditions** (соответствие должно быть один к одному)

**[WHERE conditions]** – необязательный. Условия, которые должны быть выполнены для обновления столбцов. Если не предусмотрено никаких условий, то **все записи в таблице будут обновлены!**



DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

```
UPDATE dept_copy  
  SET (dname, loc) = (SELECT dname, loc  
                      FROM dept  
                      WHERE dept.deptno = dept_copy.deptno)  
WHERE deptno > 10;
```

Синтаксис оператора DELETE:

***DELETE FROM table  
[WHERE conditions];***

***table*** – таблица, из которой удаляются записи

***[WHERE conditions]*** – необязательный. Условия, которые должны быть выполнены для удаляемых записей. **Если не предусмотрено никаких условий, то все записи из таблицы будут удалены!**

<https://oracleplsql.ru/delete.html>

```
DELETE FROM dept  
WHERE deptno = 20;
```

```
DELETE FROM dept  
WHERE UPPER(dname) LIKE '%S'  
AND deptno > 20;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Oracle оператор **COMMIT** фиксирует все изменения для текущей транзакции. Как только COMMIT выполнится, остальным пользователям будут доступны внесенные изменения.

```
INSERT INTO dept  
VALUES (50, 'DEVELOPMENT', 'MAGNITOGORSK');
```

```
UPDATE dept SET loc = 'WASHINGTON'  
WHERE deptno = 10;
```

```
DELETE FROM dept  
WHERE deptno = 20;
```

```
COMMIT;
```

В Oracle оператор **ROLLBACK** используется для отмены работы, выполняемой текущей транзакцией или транзакции, которая сомнительна.

```
DELETE FROM dept  
WHERE deptno = 20;
```

```
ROLLBACK;
```

```
DELETE FROM dept  
WHERE deptno = 30;
```

```
COMMIT;
```

```
SELECT * FROM dept;
```

DEPTNO	DNAME	LOC
50	DEVELOPMENT	MAGNITOGORSK
20	RESEARCH	DALLAS
10	ACCOUNTING	WASHINGTON
40	OPERATIONS	BOSTON

TRUNCATE TABLE используется для удаления всех записей из таблицы.

***TRUNCATE TABLE table\_name;***

*Команда TRUNCATE это не команда DML – это команда DDL. Когда DML команды работают с данными, они добавляют, изменяют или удаляют данные как часть транзакции. Эти транзакции можно подтверждать или отменять при помощи COMMIT и ROLLBACK. Это очень полезное свойство работы с данными, но его реализация заставляет БД делать много дополнительной работы, которая не видна пользователю. DDL команды не управляются пользовательскими транзакциями. Несмотря на то, что в БД они выполняются как транзакции, разработчик не может управлять ими. Когда команда выполнена – изменения вступили в силу. Но по сравнению с DML командами – DDL команды очень быстрые.*

TRUNCATE TABLE dept;

DELETE FROM dept;

<https://oracleplsql.ru/commit.html>

**СПАСИБО ЗА ВНИМАНИЕ !**  
**ВОПРОСЫ ?**

**[www.compassplus.ru](http://www.compassplus.ru)**