

**Created by: Ahsan Arif**

## UPDATE Statement

---

The UPDATE statement allows you to update a single record or multiple records in a table.

The syntax for the UPDATE statement is:

```
UPDATE table  
SET column = expression  
WHERE predicates;
```

### Example #1 - Simple example

Let's take a look at a very simple example.

```
UPDATE suppliers  
SET name = 'HP'  
WHERE name = 'IBM';
```

This statement would update all supplier names in the suppliers table from IBM to HP.

### Example #2 - More complex example

You can also perform more complicated updates.

You may wish to update records in one table based on values in another table. Since you can't list more than one table in the UPDATE statement, you can use the EXISTS clause.

For example:

```
UPDATE suppliers  
    ( SELECT customers.name  
SET supplier_name FROM customers  
=          WHERE customers.customer_id =  
            suppliers.supplier_id)  
WHERE EXISTS  
    ( SELECT customers.name  
      FROM customers  
      WHERE customers.customer_id = suppliers.supplier_id);
```

Whenever a supplier\_id matched a customer\_id value, the supplier\_name would be overwritten to the customer name from the customers table.

### Practice Exercise #1:

Based on the *suppliers* table populated with the following data, update the *city* to "Santa Clara" for all records whose *supplier\_name* is "NVIDIA".

```
CREATE TABLE suppliers
( supplier_id      number(10)      not null,
  supplier_name    varchar2(50)    not null,
  city             varchar2(50),
  CONSTRAINT suppliers_pk PRIMARY KEY
  (supplier_id)
);
```

```
INSERT INTO suppliers (supplier_id, supplier_name, city)
VALUES (5001, 'Microsoft', 'New York');
```

```
INSERT INTO suppliers (supplier_id, supplier_name, city)
VALUES (5002, 'IBM', 'Chicago');
```

```
INSERT INTO suppliers (supplier_id, supplier_name, city)
VALUES (5003, 'Red Hat', 'Detroit');
```

```
INSERT INTO suppliers (supplier_id, supplier_name, city)
VALUES (5004, 'NVIDIA', 'New York');
```

### Solution:

The following SQL statement would perform this update.

```
UPDATE suppliers
SET city = 'Santa Clara'
WHERE supplier_name = 'NVIDIA';
```

The *suppliers* table would now look like this:

SUPPLIER_ID	SUPPLIER_NAME	CITY
5001	Microsoft	New York
5002	IBM	Chicago
5003	Red Hat	Detroit
5004	NVIDIA	Santa Clara

## Practice Exercise #2:

Based on the *suppliers* and *customers* table populated with the following data, update the *city* in the *suppliers* table with the *city* in the *customers* table when the *supplier\_name* in the *suppliers* table matches the *customer\_name* in the *customers* table.

```
CREATE TABLE suppliers
( supplier_id      number(10)      not null,
  supplier_name    varchar2(50)    not null,
  city             varchar2(50),
  CONSTRAINT suppliers_pk PRIMARY KEY
  (supplier_id)
);
```

```
INSERT INTO suppliers (supplier_id, supplier_name, city)
VALUES (5001, 'Microsoft', 'New York');
```

```
INSERT INTO suppliers (supplier_id, supplier_name, city)
VALUES (5002, 'IBM', 'Chicago');
```

```
INSERT INTO suppliers (supplier_id, supplier_name, city)
VALUES (5003, 'Red Hat', 'Detroit');
```

```
INSERT INTO suppliers (supplier_id, supplier_name, city)
VALUES (5005, 'NVIDIA', 'LA');
```

```
CREATE TABLE customers
( customer_id      number(10)      not null,
  customer_name    varchar2(50)    not null,
  city             varchar2(50),
  CONSTRAINT customers_pk PRIMARY KEY
  (customer_id)
);
```

```
INSERT INTO customers (customer_id, customer_name, city)
VALUES (7001, 'Microsoft', 'San Francisco');
```

```
INSERT INTO customers (customer_id, customer_name, city)
VALUES (7002, 'IBM', 'Toronto');
```

```
INSERT INTO customers (customer_id, customer_name, city)
VALUES (7003, 'Red Hat', 'Newark');
```

**Solution:**

The following SQL statement would perform this update.

```
UPDATE
suppliers
SET city =          ( SELECT customers.city
                     FROM customers
                     WHERE customers.customer_name =
suppliers.supplier_name)
WHERE EXISTS
( SELECT customers.city
  FROM customers
  WHERE customers.customer_name = suppliers.supplier_name);
```

The *suppliers* table would now look like this:

<b>SUPPLIER_ID</b>	<b>SUPPLIER_NAME</b>	<b>CITY</b>
5001	Microsoft	San Francisco
5002	IBM	Toronto
5003	Red Hat	Newark
5004	NVIDIA	LA