

UPDATE-Anweisung in SQL

Die **UPDATE**-Anweisung in SQL wird verwendet, um vorhandene Daten in einer Tabelle zu ändern. Mit der **UPDATE**-Anweisung können Sie eine oder mehrere Spalten einer oder mehrerer Zeilen in einer Tabelle aktualisieren.

Grundsyntax

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

- **table_name**: Der Name der Tabelle, die aktualisiert werden soll.
- **SET column1 = value1, column2 = value2, ...**: Die Spalten und ihre neuen Werte, die aktualisiert werden sollen.
- **WHERE condition**: Eine Bedingung, die bestimmt, welche Zeilen aktualisiert werden. Ohne die **WHERE**-Klausel werden alle Zeilen in der Tabelle aktualisiert.

Beispiele

Beispiel 1: Einfache Aktualisierung

Angenommen, wir haben eine Tabelle **employees**:

employee_id	name	department	salary
1	Alice Smith	HR	60000
2	Bob Johnson	IT	80000
3	Charlie Lee	IT	75000

Um das Gehalt von **Charlie Lee** auf **77000** zu aktualisieren, verwenden wir:

```
UPDATE employees
SET salary = 77000
WHERE name = 'Charlie Lee';
```

Ergebnis:

employee_id	name	department	salary
1	Alice Smith	HR	60000
2	Bob Johnson	IT	80000
3	Charlie Lee	IT	77000

Beispiel 2: Aktualisierung mehrerer Spalten

Um den Namen und die Abteilung von Alice Smith zu ändern, verwenden wir:

```
UPDATE employees
SET name = 'Alice Johnson', department = 'Finance'
WHERE employee_id = 1;
```

Ergebnis:

employee_id	name	department	salary
1	Alice Johnson	Finance	60000
2	Bob Johnson	IT	80000
3	Charlie Lee	IT	77000

Beispiel 3: Aktualisierung ohne WHERE-Klausel

Wenn keine WHERE-Klausel angegeben wird, werden alle Zeilen in der Tabelle aktualisiert. Angenommen, wir wollen allen Mitarbeitern eine Gehaltserhöhung von 5000 geben:

```
UPDATE employees
SET salary = salary + 5000;
```

Ergebnis:

employee_id	name	department	salary
1	Alice Johnson	Finance	65000
2	Bob Johnson	IT	85000
3	Charlie Lee	IT	82000

Beispiel 4: Bedingte Aktualisierung

Angenommen, wir wollen das Gehalt aller Mitarbeiter in der IT-Abteilung auf 90000 setzen:

```
UPDATE employees
SET salary = 90000
WHERE department = 'IT';
```

Ergebnis:

employee_id	name	department	salary
-------------	------	------------	--------

employee_id	name	department	salary
1	Alice Johnson	Finance	65000
2	Bob Johnson	IT	90000
3	Charlie Lee	IT	90000

Beispiel für die Verwendung von **ON UPDATE RESTRICT** in SQL

Erstellung der Tabellen

```
-- Kategorie-Tabelle anlegen
CREATE TABLE categories (
  id INT PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(200)
);

-- Produkttabelle anlegen (1-N zu Kategorie) mit ON UPDATE RESTRICT
CREATE TABLE products (
  id INT PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(200),
  category_id INT,
  FOREIGN KEY (category_id) REFERENCES categories(id) ON UPDATE RESTRICT ON
  DELETE CASCADE
);

-- Versuch, die id der Kategorie 'Motorcycles' zu ändern
UPDATE categories
SET id = 4
WHERE id = 2;
```