

Database solutions

Selected SQL commands – part 2

Marzena Nowakowska

Faculty of Management and Computer Modelling

Kielce University of Technology

room: 3.21 C

Changing data in a table

UPDATE table-expression

SET field_1 = expr_1 [,field_2 = expr_2, ...] [**WHERE** condition];

By using a single UPDATE statement, one record, selected records, or all records in a table can be changed. WHERE clause is optional. This clause specifies the rows in the table that are to be changed.

Examples

UPDATE Readers

SET City = 'Kielce', Street = 'ul. Wesola 5'

WHERE R_id = 'X001';

UPDATE Readers

SET R_id = 'Inn' & Right(R_id,Len(R_id)-1)

WHERE R_id Like 'X*';

R_id-> 'X0012' ; Len(R_id) = 5, Right('X0012', 4) -> '0012',
'Inn' & '0012' = 'Inn0012'

.... SET price = price * 1.5;

Changing data in a table - continuation

The WHERE clause that is used to restrict the rows to which an UPDATE statement applies can contain a subselect (a subquery). A subselect enables to update rows in one table based on the contents of another table.

Example

```
UPDATE Product SET Sale_price = Sale_Price *1.1
WHERE Vendor_id IN
  (SELECT Vendor_id FROM Vendors
   WHERE Vendor_Name='Cumulonimbus Corporation');
```

Homework: prepare another UPDATE command that gives the same result

The *table_expression* term in the UPDATE command syntax may refer to joined table in order to select appropriate records the fields of which have to be changed.

Example

```
UPDATE
  Location INNER JOIN Department ON Location.Location_ID = Department.Location_ID
SET Department.Dept_name = "BO-" & Dept_name
WHERE Location.City="Boston";
```

Adding a record to a table

Among variety of ways of feeding the database with records, INSERT command is a classical way. There are two versions of the command.

1.

```
INSERT INTO table [(field1[, field2[, ...]])] VALUES (value1[, value2[, ...]);
```

It is possible to omit field names, provided that values for all the fields appear in the command and they follow the table field order.

Examples

```
INSERT INTO Customer
```

```
VALUES (1, 'Taylor', 'David', '235 Nutley Ave.', 'Hamilton', 'NJ', '07110', '(201) 555-1963');
```

```
INSERT INTO Customer (Customer_ID, Last_N, First_N, Street, City, State,  
Zip_Code, Phone)
```

```
VALUES (1, 'Taylor', 'David', '235 Nutley Ave.', 'Hamilton', 'NJ', '07110', '(201) 555-1963');
```

In the following example the fields: Street, State, Zip_Code will have null values.

```
INSERT INTO Customer (Customer_ID, Last_N, First_N, City, Phone)
```

```
VALUES (1, 'Taylor', 'David', 'Hamilton', '(201) 555-1963');
```

Adding records to a table

A way of putting a block of records (sometimes one-element block) to a table from another table is nest a SELECT command within an INSERT command. This method (a subselect) duplicates the selected data in other table.

2.

```
INSERT INTO table [(field1[, field2[, ...]])] SELECT field_1[, field_2[, ...]  
FROM table_expression;
```

Examples

```
INSERT INTO Return( Book_id, Reader_id, Borrow_date, Return_date )  
    SELECT Borrowing.Book_id, Borrowing.Reader_id,  
           Borrowing.Borrow_dat, Date() AS Return  
    FROM Borrowing WHERE (Borrowing.Borrow_date) <= #2000-12-31#);
```

```
INSERT INTO ARCHIVES ( Id, Surname,Forename, Brt_dt, Emp_dt, Fr_dt )  
    SELECT EMPLOYEES.Id, Surname, Forename, PERSONAL_DATA.Br_dt,  
           Emp_dt, Date()  
    FROM  
           EMPLOYEES INNER JOIN PERSONAL_DATA ON EMPLOYEES.Id =  
           PERSONAL_DATA .Id  
    WHERE (EMPLOYEES.Id = "EM_001");
```

Deleting records

Records are deleted by using a single DELETE command. If WHERE clause is not added to the command, all records are deleted.

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

Examples

```
DELETE FROM Customer WHERE Last_N = 'Taylor';
```

```
DELETE *  
FROM Employee  
WHERE Employee.Dept_ID In  
      (SELECT Department.Dept_ID FROM  
       Location INNER JOIN Department ON Location.Location_ID =  
       Department.Location_ID  
       WHERE Location.City="BOSTON");
```