IT-234 – database concepts

UNIT 7 – MODIFY DATA IN AN EXISTING TABLE

overview

One of the primary functions of any database is to be able to manipulate the data stored within its tables.

Permitted users must be able to insert, update, and delete data as necessary in order to keep the database current and ensure that only the appropriate data is being stored.

overview

SQL provides three statements for basic data manipulation: INSERT, UPDATE, and DELETE.

In this unit, we examine each of these statements and demonstrate how they can be used in an SQL environment to modify data in the database.

overview

After completing this unit, you should be able to:

- Use DML commands to modify an existing table.

SQL - INSERT

The SQL INSERT INTO Statement is used to add new rows of data to a table in the database.

There are two basic syntaxes of the INSERT INTO statement.

```
INSERT INTO TABLE_NAME (column1, column2, column3,...columnN)
VALUES (value1, value2, value3,...valueN);
```

Here, column1, column2, column3,...columnN are the names of the columns in the table into which you want to insert the data.

SQL - INSERT

You may not need to specify the column(s) name in the SQL query if you are adding values for all the columns of the table.

Make sure the order of the values is in the same order as the columns in the table.

```
INSERT INTO TABLE_NAME VALUES (value1,value2,value3,...valueN);
```

Customers

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
89	White Clover Markets	Karl Jablonski	305 - 14th Ave. S. Suite 3B	Seattle	98128	USA
90	Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240	Finland
91	Wolski	Zbyszek	ul. Filtrowa 68	Walla	01-012	Poland

```
INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)
VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway');
```

Customers

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
89	White Clover Markets	Karl Jablonski	305 - 14th Ave. S. Suite 3B	Seattle	98128	USA
90	Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240	Finland
91	Wolski	Zbyszek	ul. Filtrowa 68	Walla	01-012	Poland
92	Cardinal	Tom B. Erichsen	Skagen 21	Stavanger	4006	Norway

SQL – INSERT example

SQL – INSERT example

Customers

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
89	White Clover Markets	Karl Jablonski	305 - 14th Ave. S. Suite 3B	Seattle	98128	USA
90	Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240	Finland
91	Wolski	Zbyszek	ul. Filtrowa 68	Walla	01-012	Poland

```
INSERT INTO Customers (CustomerName, City, Country)
VALUES ('Cardinal', 'Stavanger', 'Norway');
```

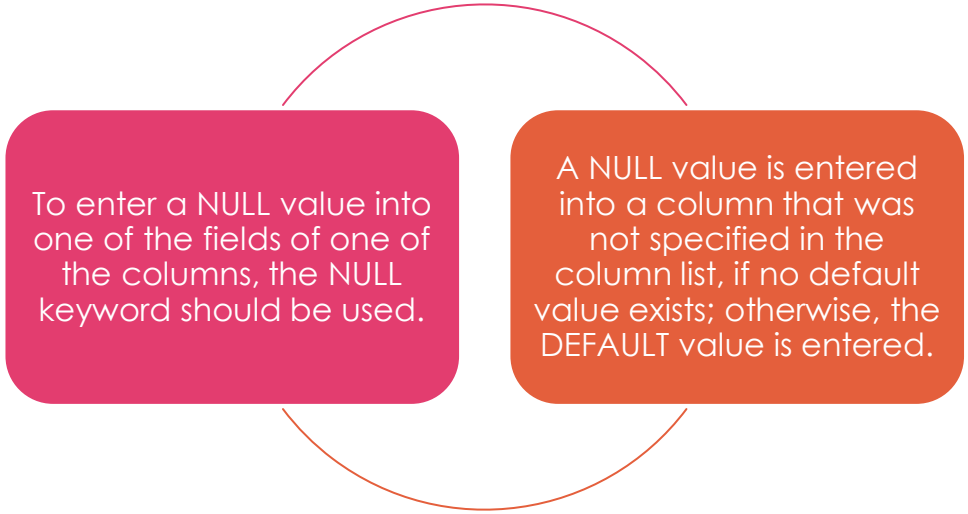
Customers

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
89	White Clover Markets	Karl Jablonski	305 - 14th Ave. S. Suite 3B	Seattle	98128	USA
90	Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240	Finland
91	Wolski	Zbyszek	ul. Filtrowa 68	Walla	01-012	Poland
92	Cardinal	null	null	Stavanger	null	Norway

SQL – INSERT – Important points

- The order of the values must match the order of the columns in the column list.
- The value type must match the type of columns in the column list.
- The number of values – must match the number of values in the column list.
- You can use the GETDATE () function instead of specifying a date.

SQL – INSERT – Important points



SQL – INSERT INTO SELECT

To insert data from other tables into a table, you use the following SQL Server INSERT INTO SELECT statement:

```
INSERT  [ TOP ( expression ) [ PERCENT ] ]
INTO target_table (column_list)
query
```

In this syntax, the statement inserts rows returned by the query into the target_table.

SQL – INSERT INTO SELECT

The query is any valid SELECT statement that retrieves data from other tables.

It must return the values that are corresponding to the columns specified in the column_list.

SQL – INSERT INTO SELECT

The TOP clause part is optional → it allows you to specify the number of rows returned by the query to be inserted into the target table.

If you use the PERCENT option, the statement will insert the percent of rows instead.

Note that it is a best practice to always use the TOP clause with the ORDER BY clause.

Insert all rows from another table example

The following statement inserts all addresses from the `customers` table into the `addresses` table:

```
INSERT INTO sales.addresses (street, city, state, zip_code)
SELECT
    street,
    city,
    state,
    zip_code
FROM
    sales.customers
ORDER BY
    first_name,
    last_name;
```

```
CREATE TABLE sales.addresses (
    address_id INT IDENTITY PRIMARY KEY,
    street VARCHAR (255) NOT NULL,
    city VARCHAR (50),
    state VARCHAR (25),
    zip_code VARCHAR (5)
);
```

SQL – INSERT INTO SELECT - Example

SQL – INSERT INTO SELECT - Example

To verify the insert, you use the following query:

```
SELECT
    *
FROM
    sales.addresses;
```

Here is the result:

address_id	street	city	state	zip_code
1	807 Grandrose Ave.	Yonkers	NY	10701
2	26 Market Drive	Forest Hills	NY	11375
3	60 Myers Dr.	Amityville	NY	11701
4	9782 Indian Spring Lane	Harlingen	TX	78552
5	167 James St.	Los Banos	CA	93635
6	755 East Henry Lane	Central Islip	NY	11722
7	8165 Baker Ave.	Franklin Square	NY	11010
8	669 S. Gartner Street	San Pablo	CA	94806
9	683 West Kirkland Dr.	East Northport	NY	11731
10	684 Westport Drive	Ballston Spa	NY	12020
11	720 Thompson Lane	Rego Park	NY	11374
12	973 Yukon Avenue	Encino	CA	91316
13	55 Cambridge Street	Plainview	NY	11803

SQL – INSERT INTO SELECT - Example

Insert some rows from another table example

Sometimes, you just need to insert some rows from another table into a table. In this case, you limit the number of rows returned from the query by using conditions in the `WHERE` clause.

The following statement adds the addresses of the stores located in `Santa Cruz` and `Baldwin` to the `addresses` table:

```
INSERT INTO
    sales.addresses (street, city, state, zip_code)
SELECT
    street,
    city,
    state,
    zip_code
FROM
    sales.stores
WHERE
    city IN ('Santa Cruz', 'Baldwin')
```

SQL - UPDATE

The UPDATE statement is used to modify the existing records in a table.

UPDATE Syntax:

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

Be careful when updating records in a table!

- Notice the WHERE clause in the UPDATE statement.
- The WHERE clause specifies which record(s) that should be updated.
- If you omit the WHERE clause, all records in the table will be updated!

SQL – UPDATE example

Customers

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

```
UPDATE Customers
SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'
WHERE CustomerID = 1;
```

Customers

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Alfred Schmidt	Obere Str. 57	Frankfurt	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

SQL – UPDATE example

Customers

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Alfred Schmidt	Obere Str. 57	Frankfurt	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

```
UPDATE Customers
SET ContactName='Juan'
WHERE Country='Mexico';
```

Customers

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Alfred Schmidt	Obere Str. 57	Frankfurt	12209	Germany
2	Ana Trujillo Emparedados y helados	Juan	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Juan	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

Customers

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Alfred Schmidt	Obere Str. 57	Frankfurt	12209	Germany
2	Ana Trujillo Emparedados y helados	Juan	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Juan	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

Be careful when updating records. If you omit the `WHERE` clause, ALL records will be updated!

```
UPDATE Customers
SET ContactName='Juan';
```

Customers

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Juan	Obere Str. 57	Frankfurt	12209	Germany
2	Ana Trujillo Emparedados y helados	Juan	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Juan	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Juan	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Juan	Berguvsvägen 8	Luleå	S-958 22	Sweden

SQL – UPDATE example

SQL - delete

The DELETE statement is used to delete existing records in a table.

DELETE Syntax:

```
DELETE FROM table_name WHERE condition;
```

Be careful when deleting records in a table!

- Notice the WHERE clause in the DELETE statement.
- The WHERE clause specifies which record(s) that should be deleted.
- If you omit the WHERE clause, all records in the table will be deleted!

Customers

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

```
DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';
```

Customers

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

SQL – delete example

Customers

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

```
DELETE FROM Customers;
```

Customers

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
NO DATA FOUND						

SQL – delete example