

# INSERT statement in SQL Server

The INSERT statement in SQL Server (Transact-SQL) is used to insert one or more records into a table.

The INSERT statement in SQL Server (Transact-SQL) is used to insert one or more records into a table.

## INSERT command syntax

### Insert a record

In its simplest form, the INSERT command syntax inserts a record with the VALUES keyword

```
INSERT INTO bang
(cot1, cot2, ...)
VALUES
(bieuthuc1, bieuthuc2, ...),
(bieuthuc1, bieuthuc2, ...),
. ;
```

The full syntax of the INSERT command inserts a record using VALUES keyword

```
INSERT INTO bang
(cot1, cot2, ...)
VALUES
( DEFAULT | NULL | bieuthuc1,
  DEFAULT | NULL | bieuthuc2,
...
);
```

Or the syntax to insert a record using the keyword DEFAULT VALUES

```
INSERT INTO bang
(cot1, cot2, ...)
DEFAULT VALUES;
```

### Insert multiple records

In the simplest form, the syntax for inserting multiple records is by sub-select

```
INSERT INTO bang
(cot1, cot2, ...)
SELECT bieuthuc1, bieuthuc2, ...
FROM bang_nguon
[WHERE dieukien];
```

Full syntax for inserting multiple desks with sub-select

```
INSERT [TOP ( tri_dau) [PERCENT] ]  
INTO bang  
(cot1, cot2, ...)  
SELECT biethuc1, biethuc2, ...  
FROM bang_nguon  
[WHERE dieukien];
```

## Variable name or variable value

### **state**

Table to insert records into.

### **cot1, cot2**

Columns in the table to insert values

### **biethuc1, biethuc2**

Value to specify in the column in the table. cot1 will be assigned the value of biethuc1, cot2 will be assigned the value of biethuc2 .

### **TOP (giatri\_dau)**

Option. If specifically, it will insert the first value of the row based on giatri\_dau. For example, TOP (10) will insert the first 10 rows from the result set.

### **PERCENT**

Option. If specified, the first rows are based on the percentage of giatri\_dau of the result set. For example, TOP (10) PERCENT will insert 10% of the first value in the result set.

### **bang\_nguon**

Source table (original table) to insert data from another table.

### **WHERE dieukien**

Option. Sending conditions are met so that the record is inserted.

### **Note**

1. When inserting records into a table with the INSERT statement, you must provide values ??for NOT NULL columns.
2. You can remove columns from the INSERT statement if the column allows NULL values.

**For example - use the keyword VALUES**

```

INSERT INTO nhanvien
(nhanvien_id, ten, ho)
VALUES
(10, 'Anderson', 'Sarah');

```

This INSERT statement will result in 1 record being inserted into the table. This record must have nhanvien\_id of 10, they are Anderson and their name is Sarah.

This syntax can be used to insert more than 1 record at a time. Eg:

```

INSERT INTO nhanvien
(nhanvien_id, ho, ten)
VALUES
(10, 'Anderson', 'Sarah'),
(11, 'Johnson', 'Dale');

```

For example, the INSERT statement above shows that it is possible to insert more than 1 record with the VALUES keyword. In this example, 2 records are inserted into the table. The first record has 10 minutes, they are Anderson and their name is Sarah. The second record has nhanvien\_id is 11, they are Johnson and the name is Dale.

The above command is equivalent to the INSERT statements below.

```

INSERT INTO nhanvien
(nhanvien_id, ho, ten)
VALUES
(10, 'Anderson', 'Sarah');

```

```

INSERT INTO nhanvien
(nhanvien_id, ho, ten)
VALUES
(11, 'Johnson', 'Dale');

```

### **For example - use the keyword DEFAULT VALUES**

```

INSERT INTO nhanvien
(nhanvien_id, ho, ten)
DEFAULT VALUES;

```

This INSERT statement inserts a record into the table. New records are created by default values for employee ID, family and name information fields.

### **For example - use the SELECT command**

It is possible to create a more complex INSERT statement with the SELECT statement as shown below.

```

INSERT INTO danhba
(danhba_id, ho, ten)
SELECT nhanvien_id, ho, ten
FROM nhanvien
WHERE nhanvien_id = 100;

```

By placing the SELECT statement in the INSERT command, you can make more inserts faster.

With this type of insert, you may want to check the number of rows to be inserted, determine the number of rows to be inserted by running the SELECT command before inserting.

```
SELECT count (*)  
FROM nhanvien  
WHERE nhanvien_id = 100;
```

Previous article: [BETWEEN conditions in SQL Server](#)

The following article: [UPDATE command in SQL Server](#)

You finished reading the article "**INSERT statement in SQL Server**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.