

# SQL **SELECT INTO** Statement

[< Previous](#)[Next >](#)

## The SQL SELECT INTO Statement

The **SELECT INTO** statement copies data from one table into a new table.

### **SELECT INTO** Syntax

Copy **all** columns into a new table:

```
SELECT *  
INTO newtable [IN externaldb]  
FROM oldtable  
WHERE condition;
```

Copy **only** some columns into a new table:

```
SELECT column1, column2, column3, ...  
INTO newtable [IN externaldb]  
FROM oldtable  
WHERE condition;
```

The new table will be created with the column-names and types as defined in the old table. You can create new column names using the **AS** clause.

# SQL SELECT INTO Examples

The following SQL statement creates a backup copy of Customers:

```
SELECT * INTO CustomersBackup2017  
FROM Customers;
```

The following SQL statement uses the **IN** clause to copy the table into a new table in another database:

```
SELECT * INTO CustomersBackup2017 IN 'Backup.mdb'  
FROM Customers;
```

The following SQL statement copies only a few columns into a new table:

```
SELECT CustomerName, ContactName INTO CustomersBackup2017  
FROM Customers;
```

The following SQL statement copies only the German customers into a new table:

```
SELECT * INTO CustomersGermany  
FROM Customers  
WHERE Country = 'Germany';
```

The following SQL statement copies data from more than one table into a new table:

```
SELECT Customers.CustomerName, Orders.OrderID  
INTO CustomersOrderBackup2017  
FROM Customers  
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
```

**Tip:** `SELECT INTO` can also be used to create a new, empty table using the schema of another. Just add a `WHERE` clause that causes the query to return no data:

```
SELECT * INTO newtable  
FROM oldtable  
WHERE 1 = 0;
```