

Chapter 26: INSERT

Section 26.1: INSERT data from another table using SELECT

```
INSERT INTO Customers (FName, LName, PhoneNumber)
SELECT FName, LName, PhoneNumber FROM Employees
```

This example will insert all Employees into the Customers table. Since the two tables have different fields and you don't want to move all the fields over, you need to set which fields to insert into and which fields to select. The correlating field names don't need to be called the same thing, but then need to be the same data type. This example is assuming that the Id field has an Identity Specification set and will auto increment.

If you have two tables that have exactly the same field names and just want to move all the records over you can use:

```
INSERT INTO Table1
SELECT * FROM Table2
```

Section 26.2: Insert New Row

```
INSERT INTO Customers
VALUES ('Zack', 'Smith', 'zack@example.com', '7049989942', 'EMAIL');
```

This statement will insert a new row into the Customers table. Note that a value was not specified for the Id column, as it will be added automatically. However, all other column values must be specified.

Section 26.3: Insert Only Specified Columns

```
INSERT INTO Customers (FName, LName, Email, PreferredContact)
VALUES ('Zack', 'Smith', 'zack@example.com', 'EMAIL');
```

This statement will insert a new row into the Customers table. Data will only be inserted into the columns specified - note that no value was provided for the PhoneNumber column. Note, however, that all columns marked as `not null` must be included.

Section 26.4: Insert multiple rows at once

Multiple rows can be inserted with a single insert command:

```
INSERT INTO tbl_name (field1, field2, field3)
VALUES (1,2,3), (4,5,6), (7,8,9);
```

For inserting large quantities of data (bulk insert) at the same time, DBMS-specific features and recommendations exist.

MySQL - [LOAD DATA INFILE](#)

MSSQL - [BULK INSERT](#)
