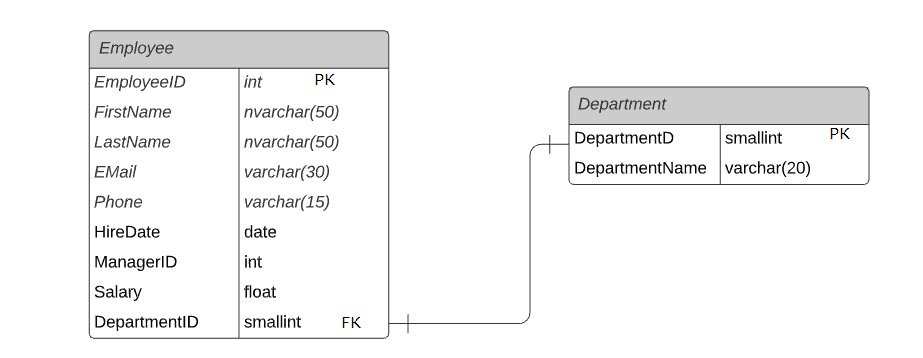
**SQl Server advanced 11**

# Foreign Keys in SQL Server:

Here you will learn what is a foreign key and how to established a relationship between two tables using a foreign key in the SQL Server database.

## What is Foreign Key?

The foreign key establishes the relationship between the two tables and enforces referential integrity in the SQL Server. For example, the following Employee table has a foreign key column DepartmentID that links to a primary key column of the Department table.

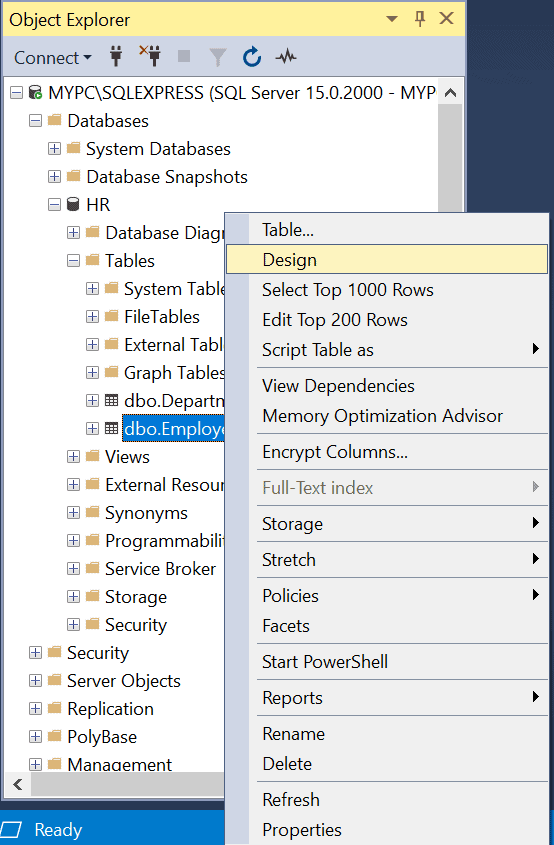
[](https://www.tutorialsteacher.com/Content/images/sqlserver/foreignkey5.png)Foreign Key Relationship in SQL Server

* A foreign key column can be linked to a primary key or a unique key column of the same or another table.
* The table having the foreign key constraint is called the child table, and the table being referenced by the foreign key is called the parent table. E.g. Employee is a child table and Department is a parent table.
* A value other than NULL is entered in the column of the foreign key constraint, that value must already exist in the referenced column of the parent table. Else you will get a foreign key violation error.
* Foreign key constraints can reference tables within the same database in the same server.
* Foreign key constraints can be defined to reference another column in the same table. This is referred to as a self-reference.
* A foreign key constraint on a single column (Column level constraint) can reference only one column in the parent table and should have the same data type as the referenced column.
* A foreign key constraint defined at the table level (on a combination of columns) should have the same number of reference columns as the number of columns defined in the constraint list. The data type of each column in the constraint must be the same as the corresponding column in the column list.
* There is no limit on the number of foreign key constraints a table can contain that references other tables. However, it is limited by the hardware configuration and the database design.
* Foreign key constraints are not enforced on temporary tables.

## Create a Foreign Key using SSMS

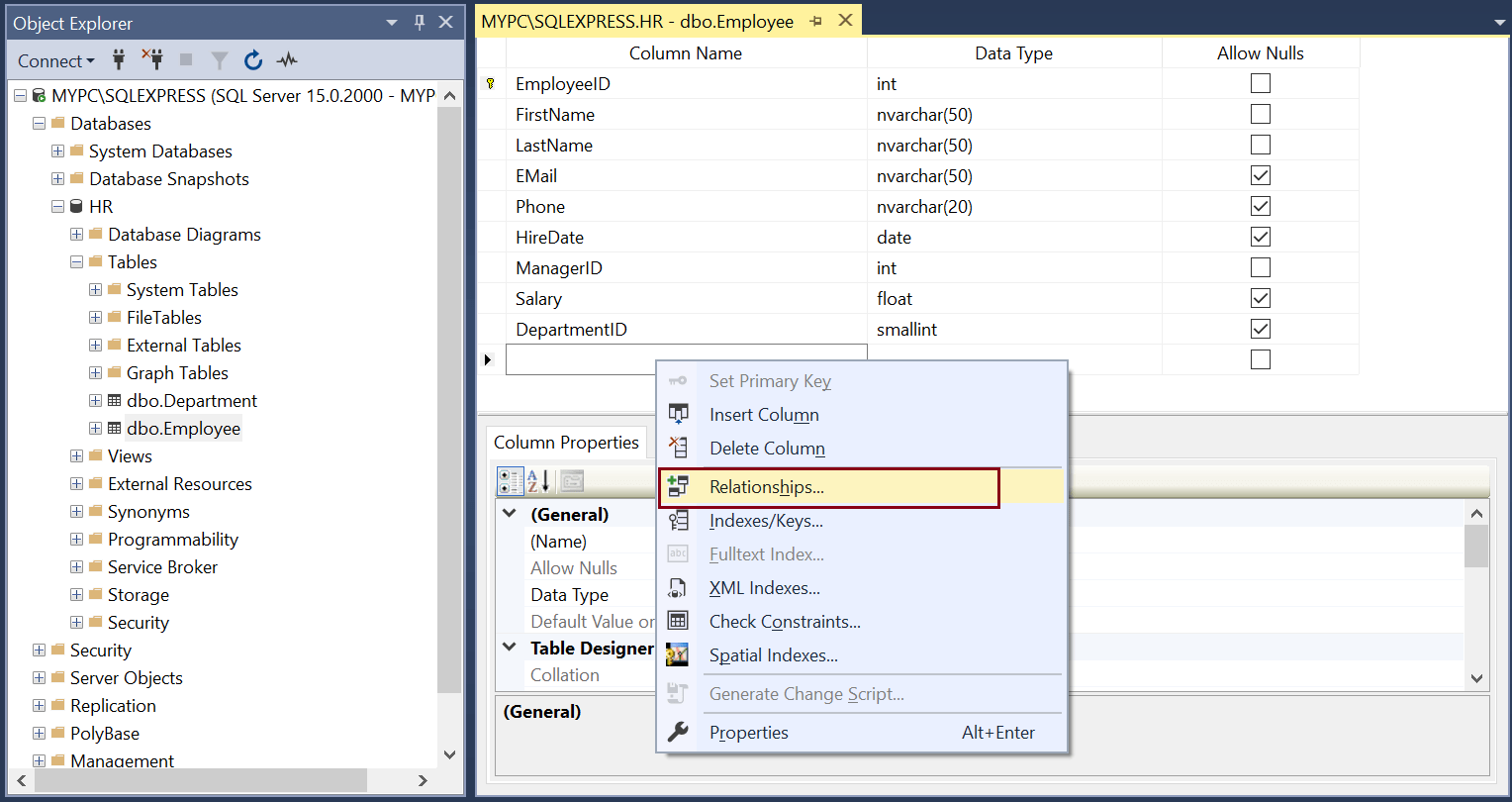
Here, we will configure the DepartmentID column as a foreign key in the Employee table that points to the DepartmentID PK column of the Department table using [SQL Server Management Studio](https://www.tutorialsteacher.com/sqlserver/sql-server-management-studio).

Open SSMS and expand the HR database. Right-click on the Employee table and click on the Design option, as shown below.

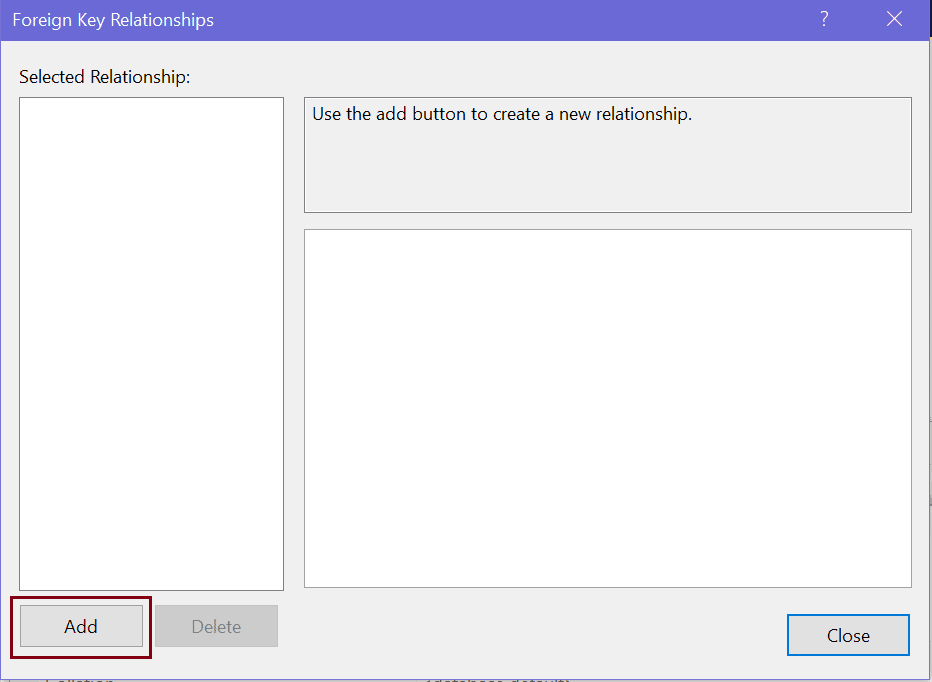
[](https://www.tutorialsteacher.com/Content/images/sqlserver/foreignkey1.png)Create a Foreign Key in SQL Server

This will open the Employee table in the design mode.

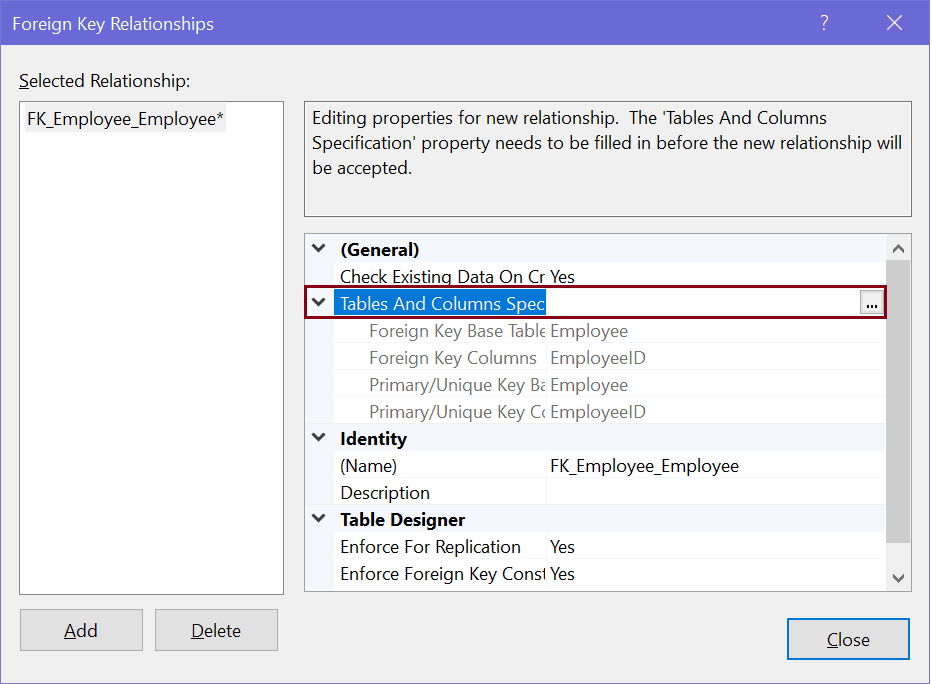
Now, right-click anywhere on the table designer and select Relationships..., as shown below.

[](https://www.tutorialsteacher.com/Content/images/sqlserver/foreignkey2.png)Define Relationships

This will open the Foreign Key Relationships dialog box, as shown below.

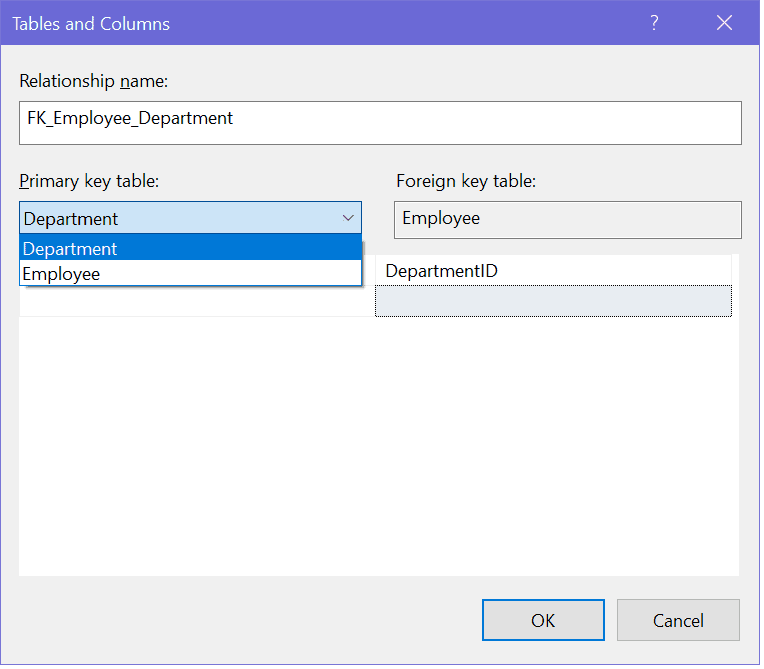
[](https://www.tutorialsteacher.com/Content/images/sqlserver/foreignkey9.png)Add Foreign Keys in SQL Server

Now, click on the Add button to configure a new foreign key, as shown below.

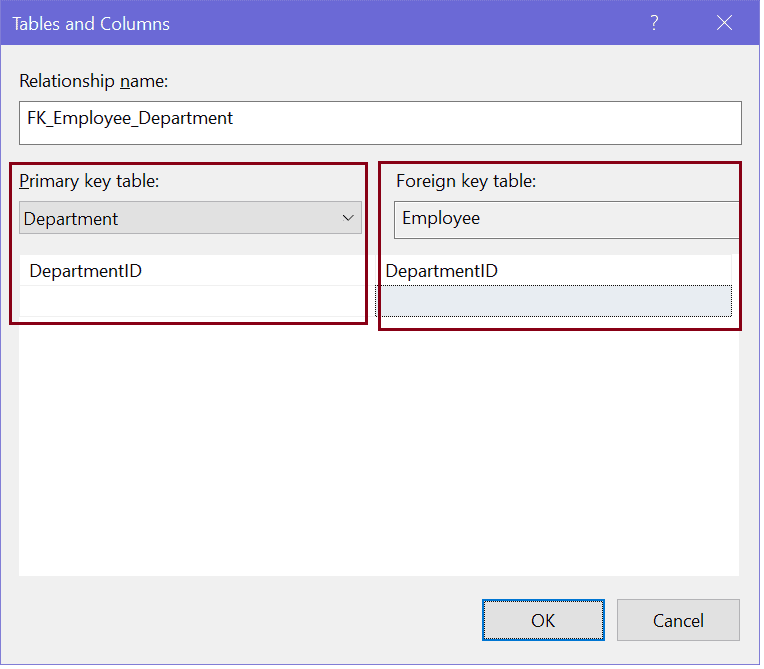
[](https://www.tutorialsteacher.com/Content/images/sqlserver/foreignkey3.png)Configure a Foreign Key in SQL Server

Now, to configure the primary key and foreign key relationship, click on the Tables and Column Specification [...] button. This will open Tables and Columns dialog box where you can select primary key and foreign key relationship.

Here, we are configuring the DepartmentID column in the Employee table as a foreign key, which points to the primary key column DepartmentID of the Department table. So, select primary table and key in the left side and foreign key table and column in the right side, as shown below.

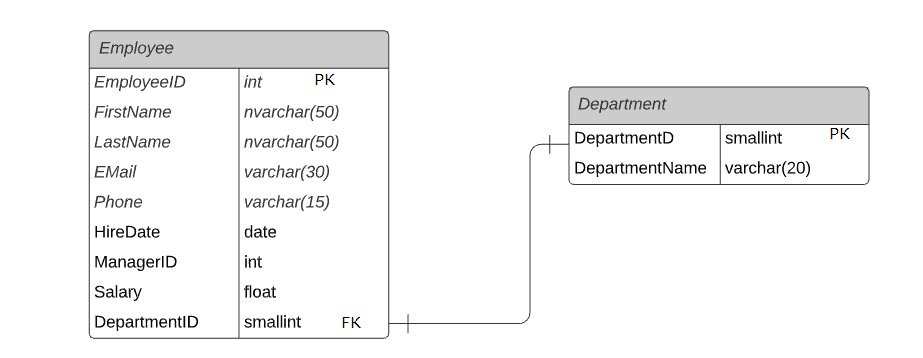
[](https://www.tutorialsteacher.com/Content/images/sqlserver/foreignkey10.png)Configure a Foreign Keys in SQL Server

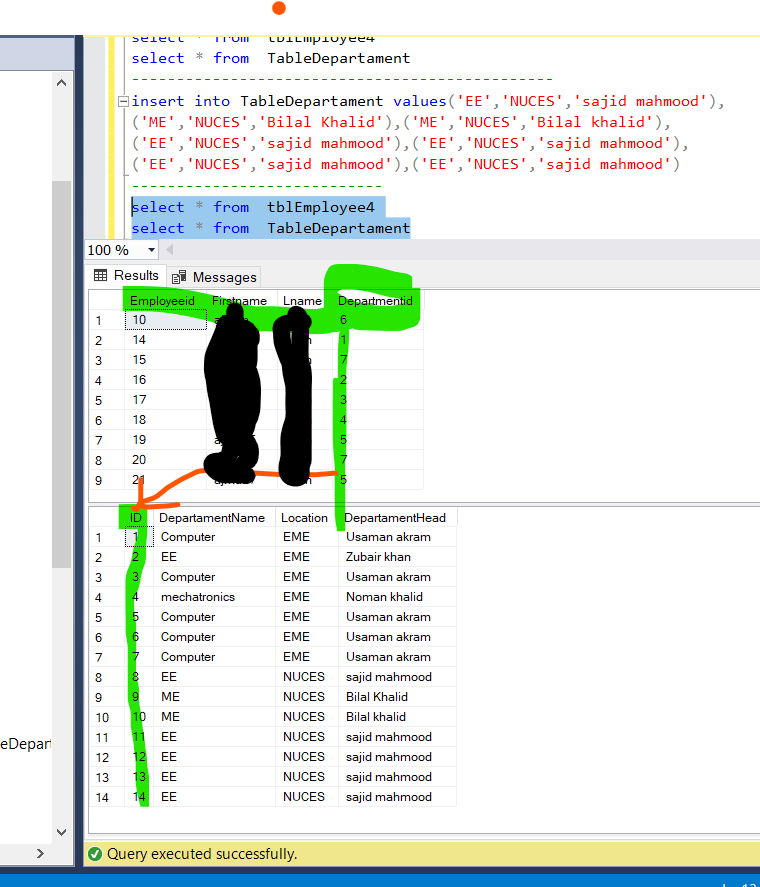
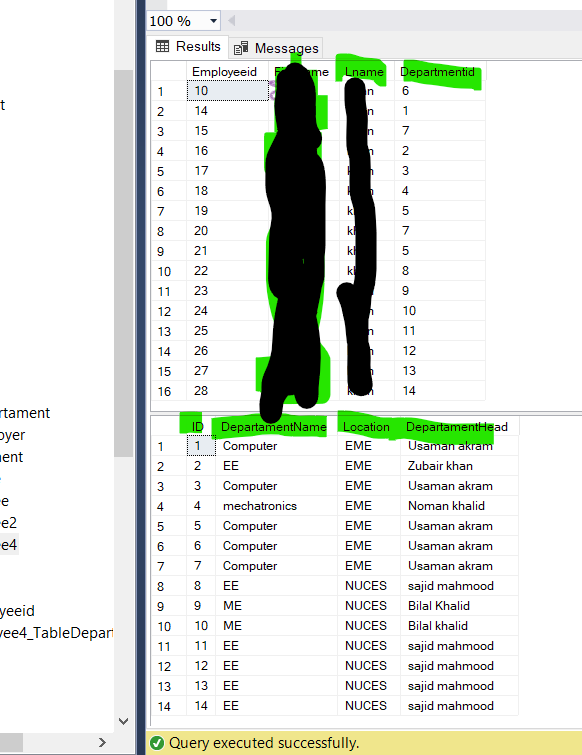
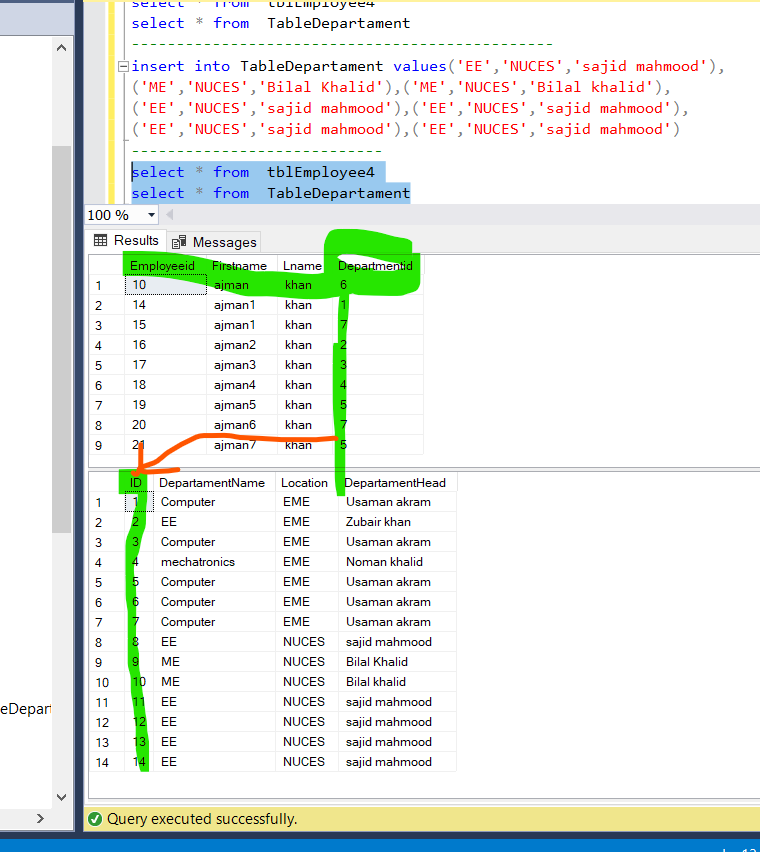
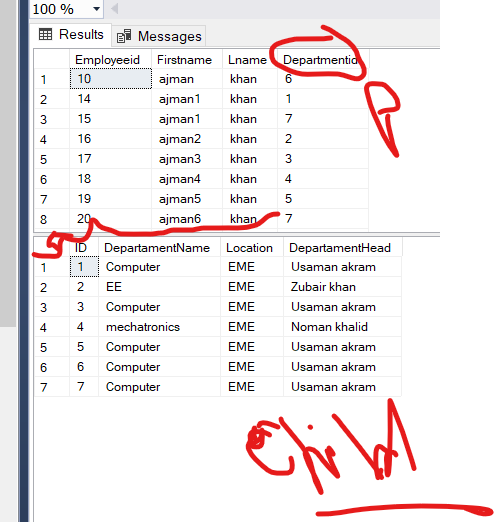
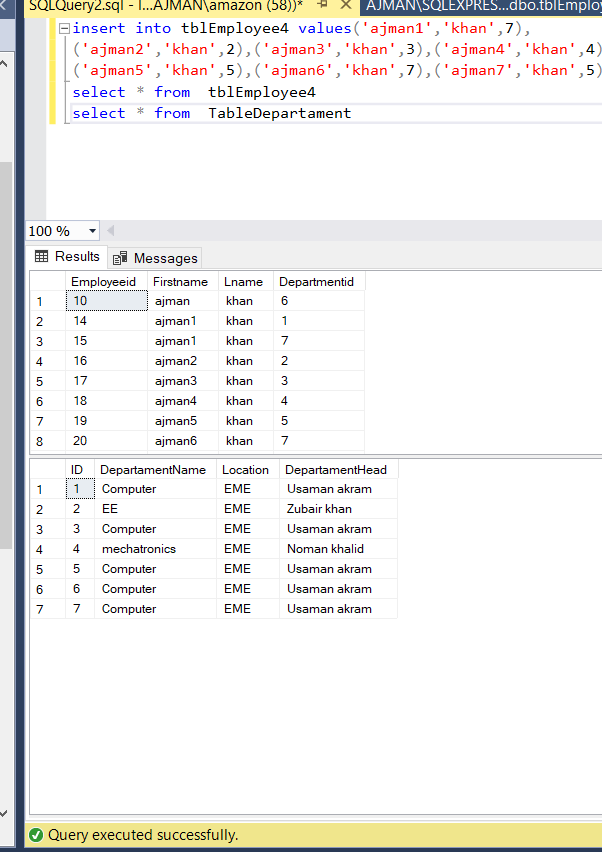
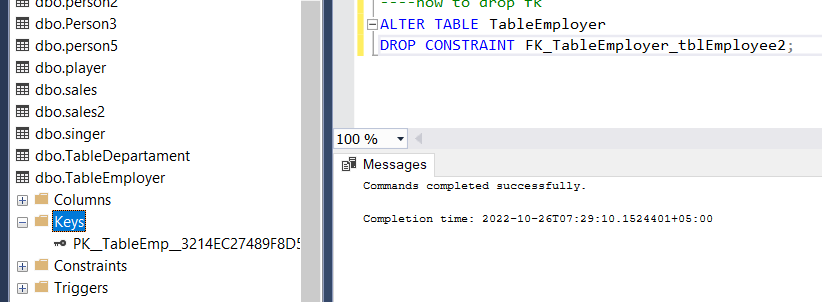
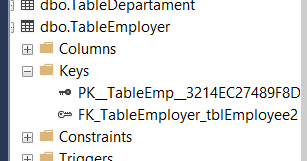
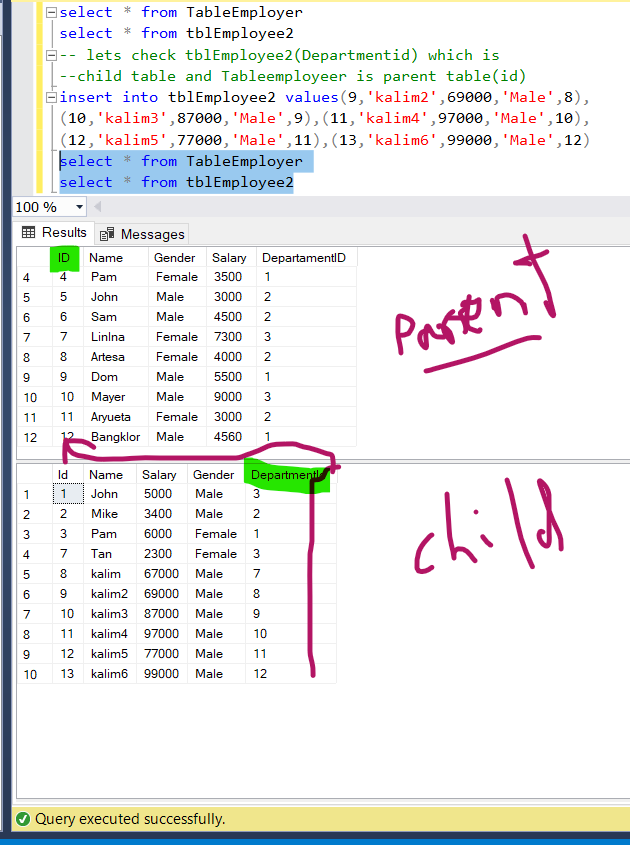
The following defines a foreign key DepartmentID in the Employee table.

[](https://www.tutorialsteacher.com/Content/images/sqlserver/foreignkey11.png)Configure a Foreign Key in SQL Server

Click OK to create the relationship and click on Close to close the dialog box.

Now, save your changes. This will create a one-to-many relationship between the Employee and Department table by setting a foreign key on the DepartmentID column in the Employee table, as shown below.

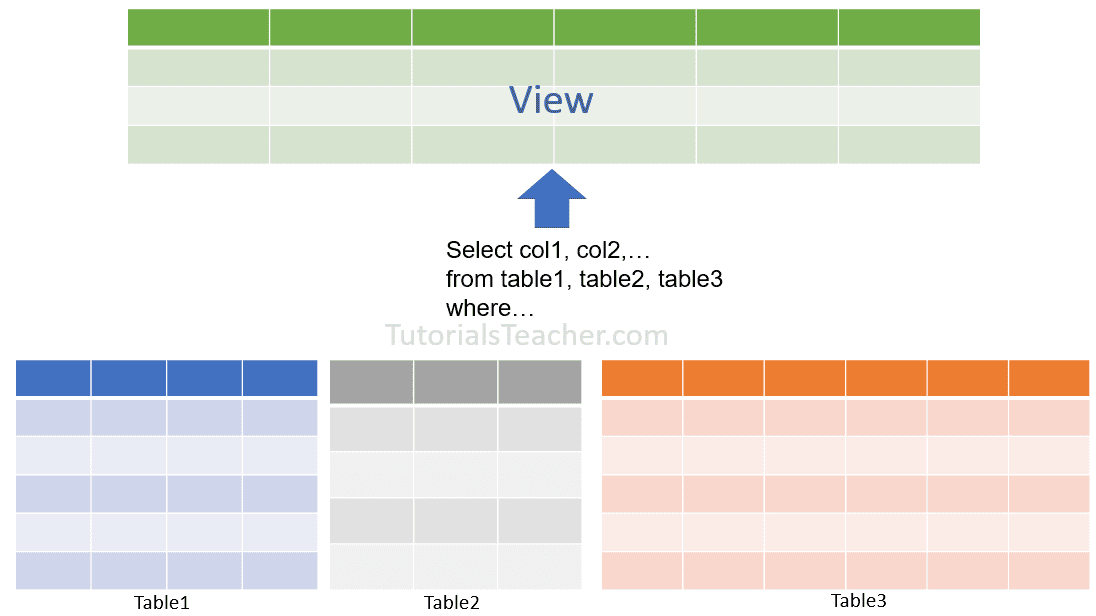
[](https://www.tutorialsteacher.com/Content/images/sqlserver/foreignkey5.png)Foreign Key Relationship in SQL Server



# SQL Server Views

In SQL Server, a view is a virtual table whose values are defined by a query. In another word, a view is a name given to a query that can be used as a table. The rows and columns of a view come from tables referenced by a query.

The following figure illustrates the view in SQL Server.

[](https://www.tutorialsteacher.com/Content/images/sqlserver/view.png)

## Types of Views

**User-defined Views:** As the name suggests, this type of view is defined by the DB users. The compiled query is stored in the database.

**Indexed Views:** When you create a unique clustered index on a user-defined view, it becomes an indexed view. It improves the performance for the queries that aggregate many rows. They are not good where the data is updated frequently.

**Partitioned Views:** A partitioned view joins horizontally partitioned data from a set of member tables across one or more servers.

**System Views:** System views expose metadata in the database. They can be used to get information about the instance of SQL Server or the database objects, e.g. the sys.databases view to return information about the user-defined databases available in the instance.

## Important Points

* Unless indexed, a view does not exist as a stored set of data values in a database.
* Views can be created by using tables or other views in the current or other databases.
* The SQL statements comprising the view are stored in the database and not the resulting data.
* The data from a view is generated dynamically when a view is referenced.
* Views are used as a security mechanism to mask the underlying base tables and permit user access only to the view.

