**SQL FOREIGN KEY: How to Create in SQL Server with Example**

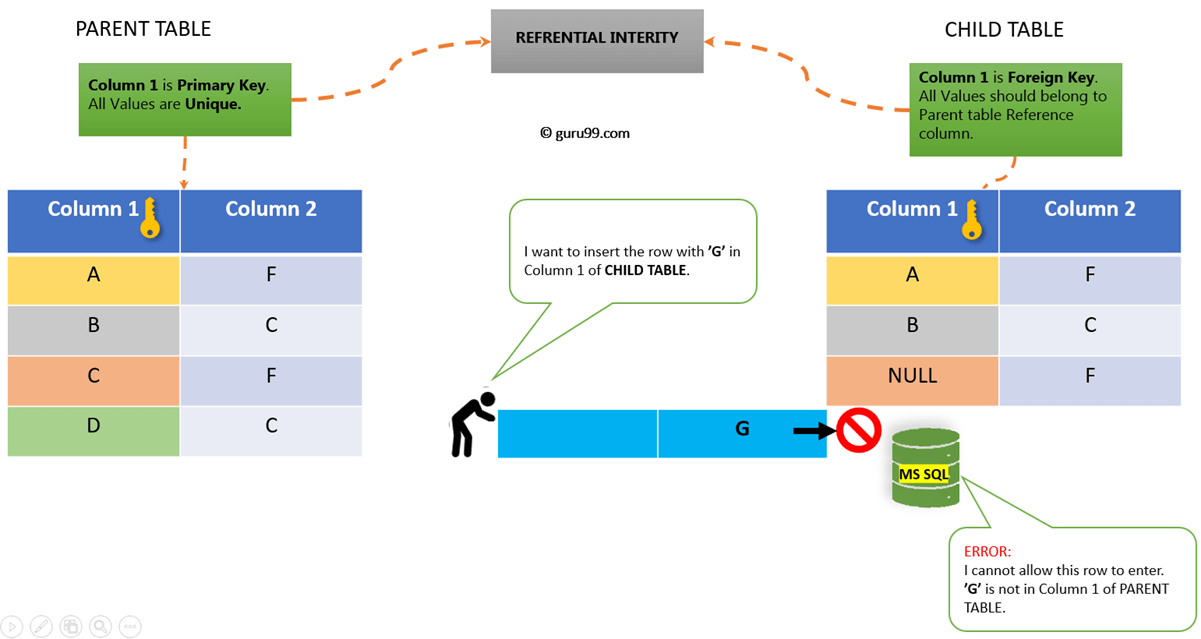
**What is a FOREIGN KEY?**

A **FOREIGN KEY** provides a way of enforcing referential integrity within SQL Server. In simple words, foreign key ensures values in one table must be present in another table.

**Rules for FOREIGN KEY**

* NULL is allowed in SQL Foreign key.
* The table being referenced is called the Parent Table
* The table with the Foreign Key in SQL is called Child Table.
* The SQL Foreign Key in child table references the primary key in the parent table.
* This parent-child relationship enforces the rule which is known as "Referential Integrity."

The Below Foreign Key in SQL example with diagram summarizes all the above points for FOREIGN KEY

How Foreign Key Works

In this tutorial, you will learn

* [How to Create FOREIGN KEY in SQL](https://www.guru99.com/sql-server-foreign-key.html#2)
  + [Using SQL Server Management Studio](https://www.guru99.com/sql-server-foreign-key.html#3)
  + [Using T-SQL: Create Child Table](https://www.guru99.com/sql-server-foreign-key.html#4)
  + [Using ALTER TABLE](https://www.guru99.com/sql-server-foreign-key.html#5)
  + [Example Query FOREIGN KEY](https://www.guru99.com/sql-server-foreign-key.html#6)

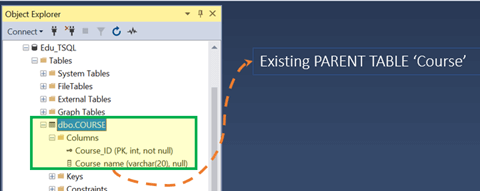
**How to Create FOREIGN KEY in SQL**

We can Create a **Foreign Key** in SQL server in **2 ways:**

1. SQL Server Management Studio
2. T-SQL

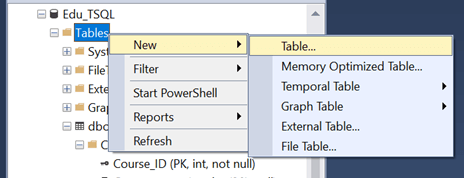
**SQL Server Management Studio**

**Parent Table:** Say, we have an existing Parent table as 'Course.' Course\_ID and Course\_name are two columns with Course\_Id as Primary Key.

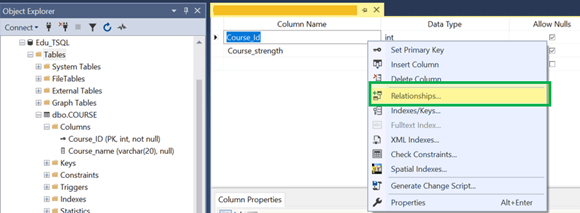


**Child Table:**We need to create the second table as a child table. 'Course\_ID' and 'Course\_Strength' as two columns. However, 'Course\_ID' shall be Foreign Key.

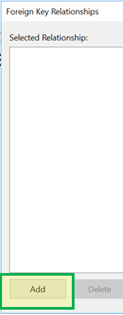
**Step 1)** Right Click on Tables>New> Table…



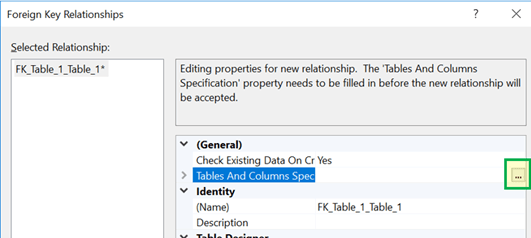
**Step 2)**Enter two column name as 'Course\_ID' and 'Course\_Strength.' Right click on 'Course\_Id' C useolumn. Now click on Relationship.



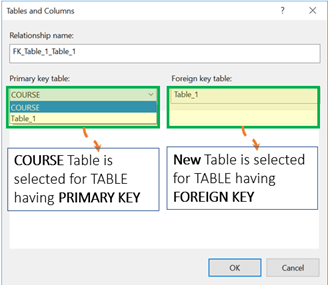
**Step 3)** In **'Foreign Key Relationship**,' Click **'Add'**



**Step 4)** In 'Table and Column Spec' click on**'…' icon**

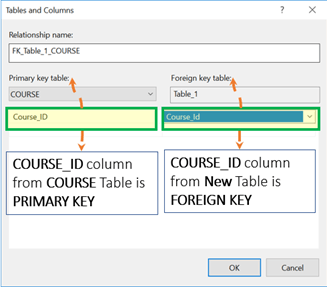


**Step 5)**Select 'Primary Key Table' as 'COURSE' and the new table now being created as 'Foreign Key Table' from the drop down.

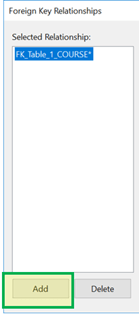


**Step 6)**'Primary Key Table' - Select 'Course\_Id' column as 'Primary Key table' column.

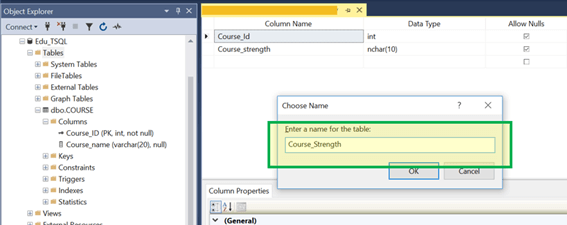
'Foreign Key Table'- Select 'Course\_Id' column as 'Foreign Key table' column. Click OK.



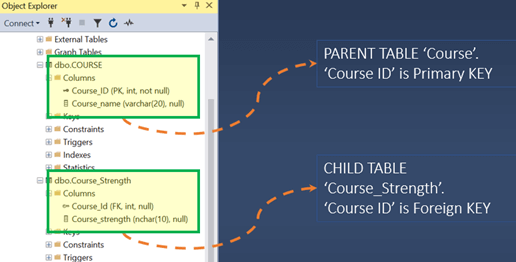
**Step 7)** Click on **Add.**



**Step 8)**Give the Table name as 'Course\_Strength' and click on **OK.**



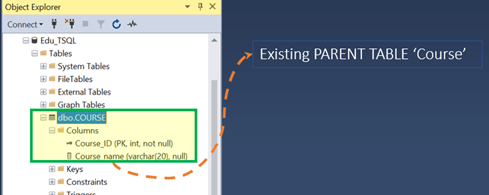
**Result:**We have set Parent-child relationship between **'Course'** and **'Course\_strength.'**



**T-SQL: Create a Parent-child table using T-SQL**

**Parent Table:** Reconsider, we have an existing Parent table with table name as 'Course.'

Course\_ID and Course\_name are two columns with Course\_Id as Primary Key.



**Child Table:**We need to create the second table as the child table with the name as 'Course\_Strength\_TSQL.'

'Course\_ID' and 'Course\_Strength' as two columns for child table Course\_Strength\_TSQL.' However, 'Course\_ID' shall be Foreign Key.

Below is the syntax to create a table with FOREIGN KEY

**Syntax:**

CREATE TABLE childTable

(

column\_1 datatype [ NULL |NOT NULL ],

column\_2 datatype [ NULL |NOT NULL ],

...

CONSTRAINT fkey\_name

FOREIGN KEY (child\_column1, child\_column2, ... child\_column\_n)

REFERENCES parentTable (parent\_column1, parent\_column2, ... parent\_column\_n)

[ ON DELETE { NO ACTION |CASCADE |SET NULL |SET DEFAULT } ]

[ ON UPDATE { NO ACTION |CASCADE |SET NULL |SET DEFAULT } ]

);

Here is a description of the above parameters:

* childTable is the name of the table that is to be created.
* column\_1, column\_2- the columns to be added to the table.
* fkey\_name- the name of the foreign key constraint to be created.
* child\_column1, child\_column2…child\_column\_n- the name of chidTable columns to reference the primary key in parentTable.
* parentTable- the name of parent table whose key is to be referenced in the child table.
* parent\_column1, parent\_column2, ... parent\_column3- the columns making up the primary key of parent table.
* ON DELETE. An optional parameter. It specifies what happens to the child data after deletion of the parent data. Some of the values for this parameter include NO ACTION, SET NULL, CASCADE, or SET DEFAULT.
* ON UPDATE- An optional parameter. It specifies what happens to the child data after update on the parent data. Some of the values for this parameter include NO ACTION, SET NULL, CASCADE, or SET DEFAULT.
* NO ACTION- used together with ON DELETE and ON UPDATE. It means that nothing will happen to the child data after the update or deletion of the parent data.
* CASCADE- used together with ON DELETE and ON UPDATE. The child data will either be deleted or updated after the parent data has been deleted or updated.
* SET NULL- used together with ON DELETE and ON UPDATE. The child will be set to null after the parent data has been updated or deleted.
* SET DEFAULT- used together with ON DELETE and ON UPDATE. The child data will be set to default values after an update or delete on the parent data.

Let's see a Foreign Key in SQL example to create a table with One Column as a FOREIGN KEY:

**Query:**

CREATE TABLE Course\_Strength\_TSQL

(

Course\_ID Int,

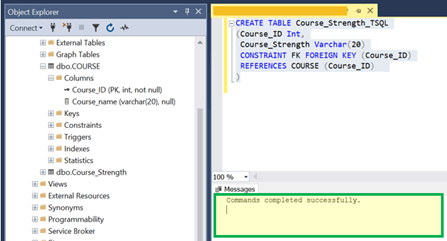
Course\_Strength Varchar(20)

CONSTRAINT FK FOREIGN KEY (Course\_ID)

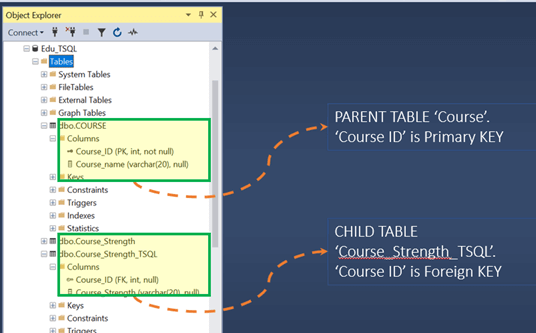
REFERENCES COURSE (Course\_ID)

)

**Step 1)** Run the query by clicking on execute.



**Result:**We have set Parent-child relationship between **'Course'** and **'Course\_strength\_TSQL.'**



**Using ALTER TABLE**

Now we will learn how to use Foreign Key in SQL and add Foreign Key in SQL server using the ALTER TABLE statement, we will use the syntax given below:

ALTER TABLE childTable

ADD CONSTRAINT fkey\_name

FOREIGN KEY (child\_column1, child\_column2, ... child\_column\_n)

REFERENCES parentTable (parent\_column1, parent\_column2, ... parent\_column\_n);

Here is a description of the parameters used above:

* childTable is the name of the table that is to be created.
* column\_1, column\_2- the columns to be added to the table.
* fkey\_name- the name of the foreign key constraint to be created.
* child\_column1, child\_column2…child\_column\_n- the name of chidTable columns to reference the primary key in parentTable.
* parentTable- the name of parent table whose key is to be referenced in the child table.
* parent\_column1, parent\_column2, ... parent\_column3- the columns making up the primary key of parent table.

Alter table add Foreign Key example:

ALTER TABLE department

ADD CONSTRAINT fkey\_student\_admission

FOREIGN KEY (admission)

REFERENCES students (admission);

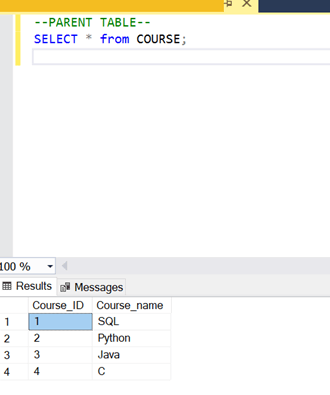
We have created a foreign key named fkey\_student\_admission on the department table. This foreign key references the admission column of the students table.

**Example Query FOREIGN KEY**

First, let's see our Parent Table Data, COURSE.

**Query:**

SELECT \* from COURSE;



Now let's insert some row in Child table: **'Course\_strength\_TSQL.'**

We will try to insert two types of rows

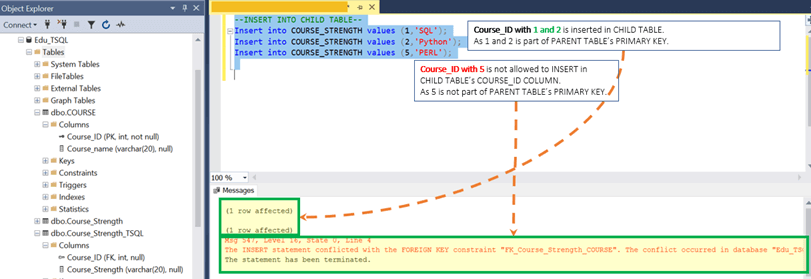
1. The first type, for which Course\_Id in child table will exist in Course\_Id of Parent table. i.e. Course\_Id = 1 and 2
2. The second type, for which Course\_Id in child table doesn't exist in the Course\_Id of Parent table. i.e. Course\_Id = 5

**Query:**

Insert into COURSE\_STRENGTH values (1,'SQL');

Insert into COURSE\_STRENGTH values (2,'Python');

Insert into COURSE\_STRENGTH values (5,'PERL');



**Result:**Let's run the Query together to See our Parent and Child table

Row with Course\_ID 1 and 2 exist in Course\_strength table. Whereas, Course\_ID 5 is an exception.

