FOREIGN KEY CONSTRAINT

The SQL FOREIGN KEY constraint The foreign key constraint ensures that the values in the foreign key column match the values in the primary key column of the referenced table, maintaining referential integrity.

Foreign Key in SQL

A foreign key is a column or a combination of columns in a table that establishes a link between two tables in a relational database. It refers to the primary key in another table, creating a relationship between them.

The table with a foreign key is called a **foreign table/child table/referencing table** and the table with a primary key that is referenced by a foreign key is called a **primary table/parent table /referenced Table**.

Syntax

A foreign key can be created during table creation using **CREATE TABLE** statement or it can be added to a table later using **ALTER TABLE** statement.

SQL FOREIGN KEY on CREATE TABLE

The syntax to create a foreign key in CREATE TABLE statement is:

```
CREATE TABLE table_name (
  column1 datatype,
  column2 datatype,
  …,
  CONSTRAINT fk_constraint_name FOREIGN KEY (column1,
column2, …)
  REFERENCES parent_table (column1, column2, …)
);
```

SQL FOREIGN KEY on ALTER TABLE

The syntax to add a foreign key with ALTER TABLE statement is:

```
ALTER TABLE table_name
ADD CONSTRAINT fk_constraint_name FOREIGN KEY (column1,
column2, …)
REFERENCES parent_table (column1, column2, …);
```

SQL Foreign Key Constraint Example

In this example, we create a foreign key during the table creation.

Query

```
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY,
    CustomerName VARCHAR(50) NOT NULL
);
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    OrderNumber INT NOT NULL,
    CustomerID INT,
    FOREIGN KEY (CustomerID)
REFERENCES Customers(CustomerID)
);
INSERT INTO Customers (CustomerID, CustomerName)
VALUES
    (1, 'John'),
    (2, 'Jane'),
    (3, 'Bob');
INSERT INTO Orders (OrderID, OrderNumber, CustomerID)
VALUES
    (1, 101, 1),
    (2, 102, 2),
    (3, 103, 3);
```

The following query will create the ‘customers’ and ‘orders’ table.

Customers Table

CustomerID (Primary Key)	CustomerName
1	John
2	Jane
3	Bob

OrderID (Primary Key)	OrderNumber	CustomerID (Foreign Key)
1	101	1
2	102	2
3	103	3

Insert Value in Foreign Key Table

If a corresponding value in the foreign table doesn’t exist, a record in the child table cannot be inserted.

Query

```
INSERT INTO Orders (OrderID, OrderNumber, CustomerID)
VALUES
(4, 104, 4);
```

Output

```
Error: FOREIGN KEY constraint failed.
```

Delete a value in Foreign Key Table

When a record in the master table is deleted and the corresponding record in the child table exists, an error message is displayed and prevents the **DELETE** operation from going through.

Query

```
DELETE FROM Customers
WHERE CustomerID = "3";
```

Output

```
Error: FOREIGN KEY constraint failed.
```

SQL DROP FOREIGN KEY

To remove a foreign key from a table, use the **ALTER TABLE with DROP CONSTRAINT command**.

Syntax

Syntax to remove any foreign key from a table:

```
ALTER TABLE table_name
DROP CONSTRAINT fk_name;
```

Important Points About SQL FOREIGN KEY Constraint

- A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.
- The table containing the foreign key is called the child table, and the table containing the candidate key is called the referenced or parent table.
- A table can have multiple FOREIGN KEY constraints.
- When defining a FOREIGN KEY constraint, you can specify what happens when a referenced row in the parent table is deleted or updated. This is done using the ON DELETE and ON UPDATE clauses followed by the CASCADE, SET NULL, or NO ACTION option.

- The FOREIGN KEY constraint in SQL is used to maintain the referential integrity of data within the database.