Sql-FOREIGN KEY

FOREIGN KEY Constraint in SQL

A **FOREIGN KEY** constraint is a type of integrity constraint that ensures the relationship between two tables. It specifies which column(s) in a table (the child table) contains foreign keys that reference the primary key in another table (the parent table).

Key Characteristics:

1. **References**: The FOREIGN KEY constraint references the PRIMARY KEY or UNIQUE KEY constraint of the parent table.
2. **Enforces Relationship**: It enforces the relationship between two tables, ensuring that a value in the child table exists in the parent table.

Example:

Suppose we have two tables: `Customers` and `Orders`.

Customers Table:

CustomerID (PK)	Name	Address
1	John	NY
2	Jane	CA

Orders Table:

OrderID (PK)	CustomerID (FK)	OrderDate
101	1	2020-01-01
102	1	2020-02-01

In this example:

- The `CustomerID` column in the `Customers` table is the PRIMARY KEY (PK).
- The `CustomerID` column in the `Orders` table is the FOREIGN KEY (FK) that references the PRIMARY KEY in the `Customers` table.
- This relationship ensures that each order must be associated with an existing customer.

SQL Syntax:

```
CREATE TABLE Orders (
    OrderID int PRIMARY KEY,
    CustomerID int,  -- REFERENCES Customers(CustomerID)
    OrderDate date
);
```

In this syntax:

- We create a table named `Orders` with columns `OrderID`, `CustomerID`, and `OrderDate`.
- The `CustomerID` column is specified as the FOREIGN KEY that references the PRIMARY KEY in the `Customers` table.

Tips:

- Use FOREIGN KEY constraints to maintain data consistency across related tables.
- Always specify the foreign key column(s) when creating a table with a FOREIGN KEY constraint.
- Be aware of potential issues, such as orphaned records or inconsistent relationships, which can occur if not properly managed.

Curated by Brajesh Kumar