Sql-3NF

What is 3NF?

Third Normal Form (3NF) is a normalization rule in database design that aims to reduce data redundancy and improve data integrity. It was first introduced by Edgar F. Codd, the father of relational databases.

3NF Rules:

A relation (table) is said to be in 3NF if it satisfies the following conditions:

1. **First Normal Form (1NF)**: Each cell in the table must contain a single value.
2. **Second Normal Form (2NF)**: All non-key attributes must depend on the entire primary key, not just part of it.
3. **Third Normal Form (3NF)**: If a relation is in 2NF and there's a non-key attribute that depends on another non-key attribute, then it should be moved to a separate table.

Example:

Suppose we have an `Orders` table with the following columns:

OrderID	CustomerName	OrderDate	ProductID	Quantity
1	John Smith	2022-01-01	101	2
1	John Smith	2022-01-01	102	3
2	Jane Doe	2022-01-15	103	4

This table is not in 3NF because the `CustomerName` column depends on another non-key attribute (`OrderID`) to identify which customer it belongs to. To normalize this table into 3NF, we need to create two separate tables:

Customers table:

CustomerID	CustomerName
1	John Smith
2	Jane Doe

Orders table (renamed `OrderDetails`):

OrderID	ProductID	Quantity	CustomerID
1	101	2	1
1	102	3	1
2	103	4	2

Now, each table has a clear primary key (e.g., `CustomerID` in the `Customers` table) and no non-key attribute depends on another non-key attribute. This design adheres to the 3NF rules.

Benefits of 3NF:

- Reduced data redundancy: By separating related but distinct information into separate tables, we avoid storing duplicate data.
- Improved data integrity: If a customer's name changes, it only needs to be updated in one place (the `Customers` table), rather than multiple places throughout the database.

Curated by Brajesh Kumar