

Foreign Keys

Introduction:

- It's a fundamental concept in relational database management systems.
- They play a critical role in maintaining integrity and consistency of data within databases by establishing relationships between tables
- We will talk about the concept of foreign keys, their usages and how we can use them in SQL.

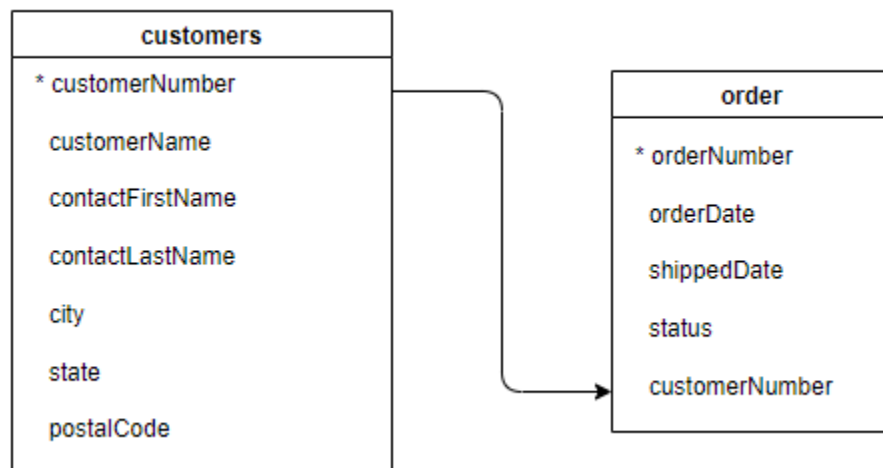
Concept of foreign keys:

- A foreign key is a column or a set of columns in one table that references the primary key of another table.
- Foreign key references are stored within a child table and linked up to a primary key in a separate table.
- The purpose of a foreign key is to ensure referential integrity of the data. It creates a link between the data in two tables, enforcing a rule that the column acting as a

foreign key must have a corresponding value in its linked table. This creates referential integrity.

- We must be careful when we create tables or alter tables (such as inserting or deleting data from the foreign key column) to avoid altering or destroying the linkage between them.

Example of two tables linked together:



The original table is called the **parent table** (customers) or referenced table, and the referencing table with a foreign key is called a **child table (order)**.

The **CUSTOMER_ID** in the order table becomes the **foreign key** that references parent or primary keys in the customer table.

Key Characteristics of Foreign Keys:

- 1) **Cascading Actions:** Actions like ON DELETE CASCADE and ON UPDATE CASCADE can be used to maintain referential integrity automatically when the referenced data is updated or deleted.
- 2) **Referential Integrity:** Ensures that a row in one table corresponds to a valid row in another table.
- 3) **Relationships:** Foreign keys help in defining various types of relationships between tables, such as one-to-one, one-to-many, and many-to-many.

How to use it in SQL:

- 1) **Creating a Foreign Key at Table Creation:**

SQL:

```
CREATE TABLE orders (  
  order_id INT PRIMARY KEY,  
  customer_id INT,
```

```
order_date DATE,  
FOREIGN KEY (customer_id) REFERENCES customers(customer_id)  
);
```

2) Adding a Foreign Key to an Existing Table:

SQL:

```
ALTER TABLE orders  
ADD CONSTRAINT fk_customer  
FOREIGN KEY (customer_id) REFERENCES customers(customer_id);
```

Foreign Key versus Primary Key:

- Unlike a foreign key, a primary key works within a relational database to uniquely identify specific records. Primary keys must be unique to the table and are often used as an absolute reference point for other types of database keys.
- Foreign keys are used more as a link than as a unique identifier of a specific row or record.
- While multiple foreign keys can be utilized within a specific table or relational database, only one primary

key is permitted per database. Duplicate values are permitted for foreign keys, as well as null values.

- Null values are not permitted for primary keys, and references to any primary keys within a table must be deleted before removing a primary key row/record.

Advantages:

Data Integrity: We need foreign keys as they help us to make sure that data is consistent, complete, between both the tables and overall accuracy is maintained.

Query Optimization: Foreign keys optimize the query execution by utilizing query plans more efficiently and improving the relationships between tables. It also helps in fast data retrieval.

Establishing Relationships: The main requirement of foreign keys is the establishment of relationships between tables. It makes sure that data is linked across multiple tables and helps in storing and retrieving data.

Data Security: Foreign keys help in improving the security of data by preventing unauthorized modifications or deletions of important data in the referenced table.

Database Maintenance: Foreign keys are required in database maintenance tasks and help to ensure integrity and consistency of data during these operations.

Best Practices for Using Foreign Keys:

1. Use clear and consistent naming conventions for foreign keys and constraints.
2. Index columns that are used as foreign keys to improve query performance.
3. Be cautious of creating circular references, as they can lead to complex and maintenance challenges.
4. Ensure that the foreign key and the referenced primary key columns have the same data type and size.
5. Clearly document the relationships between tables to aid in database maintenance and understanding.

Conclusion:

Foreign keys are essential for maintaining the relational integrity of a database. By enforcing rules and relationships between tables, they ensure data consistency and facilitate efficient database operations. Proper use of foreign keys, along with cascading actions and best practices, can significantly enhance the reliability and performance of a database system.