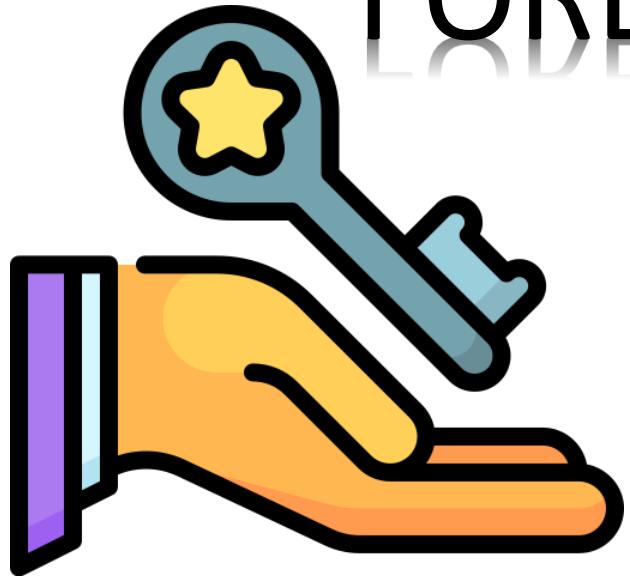


# FOREIGN KEY Constraint



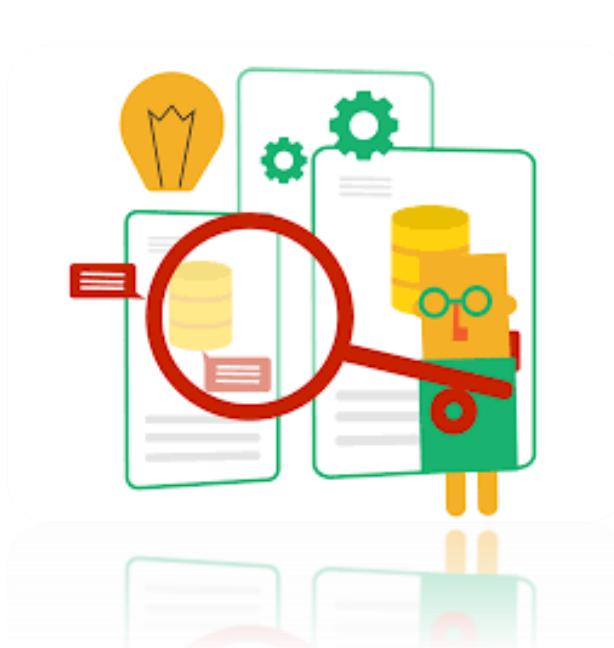
Establishing relationships between tables by enforcing referential integrity using the FOREIGN KEY constraint.

By Dipal Paneri.

# What is Foreign Key ?

Foreign keys create an association between two tables by linking them together on the database layer and help to ensure referential integrity. A foreign key always points to the primary keys of another table. Foreign keys are optional and are only used when we need to associate two tables on the database layer.

A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.



## What is a Primary Key?

A primary key is a column in a table that is used as a unique identifier for each row. It functions a bit like the row's address, and is used for the table's primary index.

# What is a foreign key constraint?



A Foreign Key (FK) Constraint in SQL is a way to enforce referential integrity between two tables. It ensures that the values in a column (or columns) of one table match the values in another table's primary key or unique constraint.

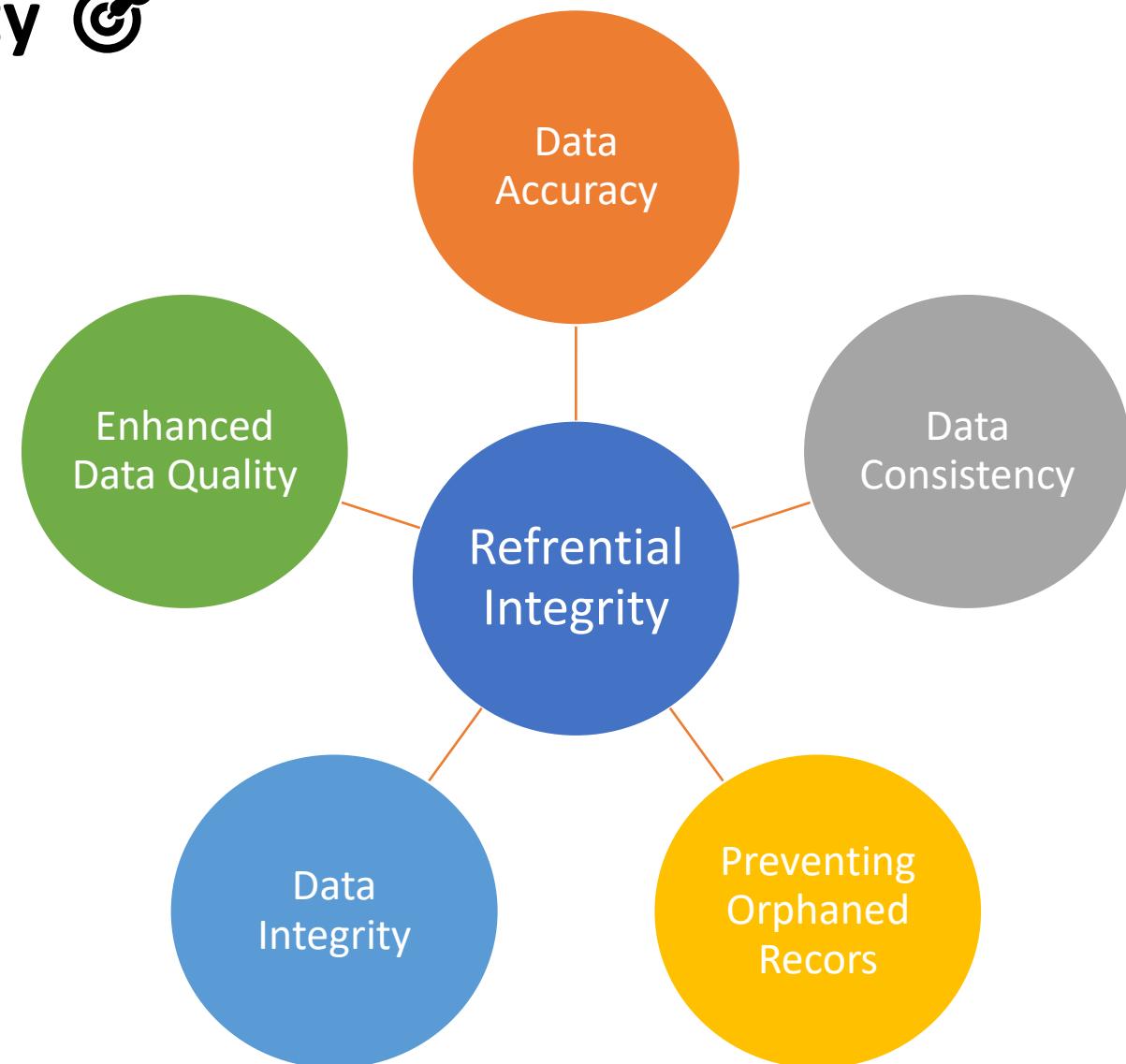
The Syntax of A Foreign Key (FK) Constraint in SQL with Parent Child Relationship in tables.

```
CREATE TABLE parent (
    id INT PRIMARY KEY
);
```

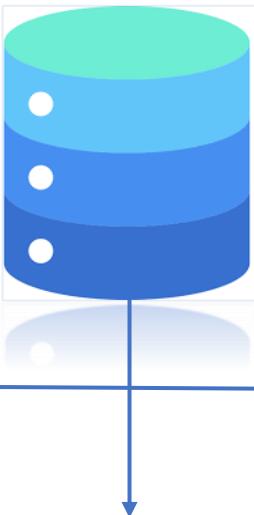
```
CREATE TABLE child (
    pid INT,
    FOREIGN KEY (pid) REFERENCES parent(id),
    id INT,
    PRIMARY KEY (pid, id)
) INTERLEAVE IN PARENT parent(pid);
```

# ⌚ Purpose of Referential Integrity ⌚

1. **Data Accuracy:** Ensures that relationships between tables remain valid, preventing the insertion of inconsistent or invalid data.
2. **Data Consistency:** Guarantees that changes made to data in one table are accurately reflected in related tables, maintaining consistency across the database.
3. **Prevents Orphaned Records:** Ensures that every foreign key value in one table matches a primary key value in another table, preventing the creation of orphaned records.
4. **Maintains Data Integrity:** Prevents the creation of relationships that could result in inconsistent or invalid data, thus maintaining the overall integrity of the database.
5. **Simplified Data Management:** Helps in maintaining and managing data more effectively by automatically handling the integrity of the database, reducing the risk of errors.
6. **Enhanced Data Quality:** Improves the overall quality of the data by ensuring that only valid and consistent data is stored in the database.





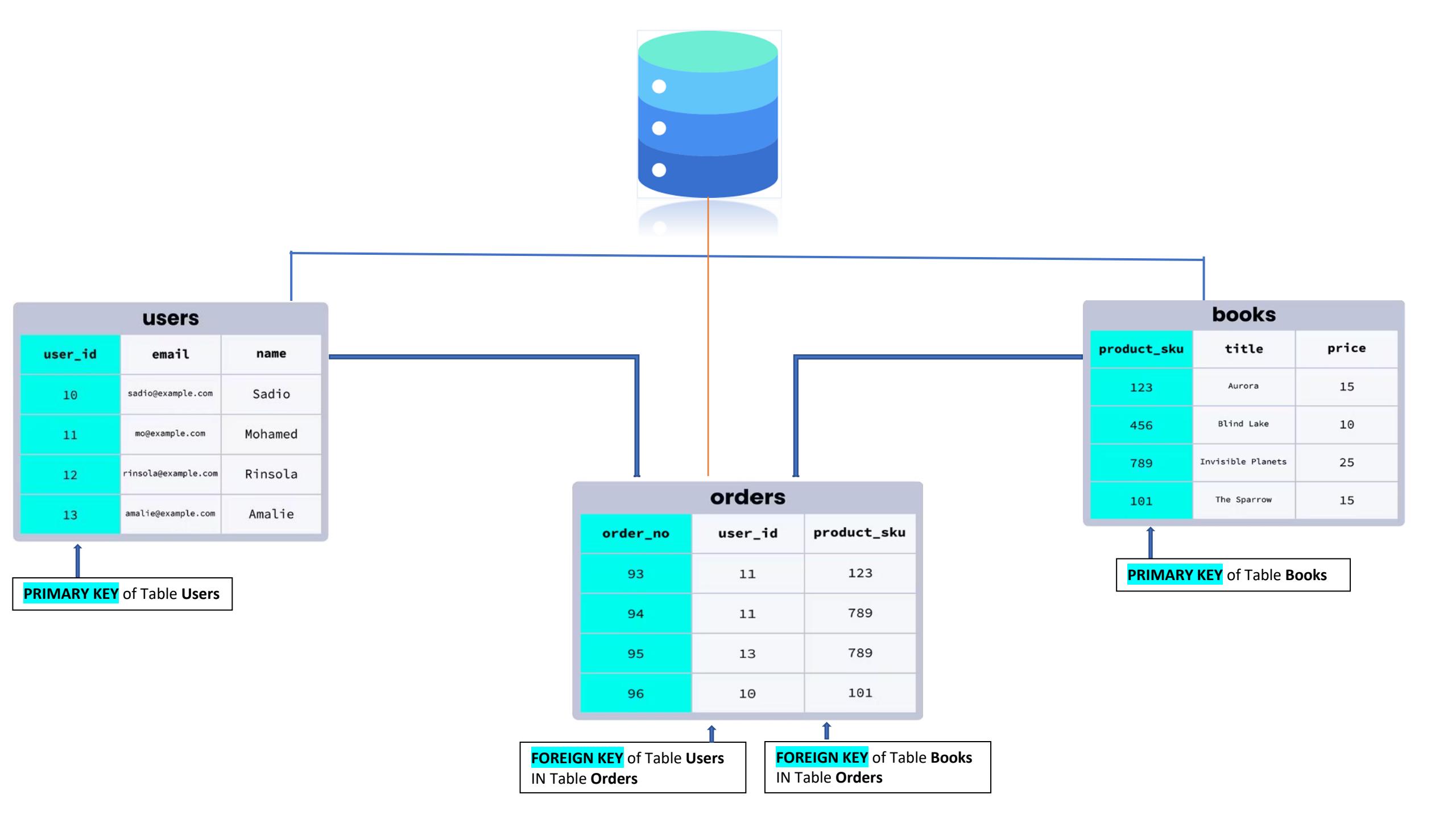


The diagram illustrates a database schema with three tables: **users**, **orders**, and **books**. Arrows point from each table to the central database icon, indicating they are stored within it.

users		
user_id	email	name
10	sadio@example.com	Sadio
11	mo@example.com	Mohamed
12	rinsola@example.com	Rinsola
13	amalie@example.com	Amalie

orders		
order_no	user_id	product_sku
93	11	123
94	11	789
95	13	789
96	10	101

books		
product_sku	title	price
123	Aurora	15
456	Blind Lake	10
789	Invisible Planets	25
101	The Sparrow	15



users			orders		
user_id	email	name	order_no	user_id	product_sku
10	sadio@example.com	Sadio	93	11	123
11	mo@example.com	Mohamed	94	11	789
12	rinsola@example.com	Rinsola	95	13	789
13	amalie@example.com	Amalie	96	10	101

A row can only be added or updated in the Orders table if the value in **Orders.user\_id** matches on the existing **user ID** in the users table.

This type of **database rule** is called a **FOREIGN KEY CONSTRAINT**

users		
user_id	email	name
10	sadio@example.com	Sadio
11	mo@example.com	Mohamed
12	rinsola@example.com	Rinsola
13	amalie@example.com	Amalie

order_no	user_id	product_sku
93	11	123
94	11	789
95	13	789
96	10	101

books		
product_sku	title	price
123	Aurora	15
456	Blind Lake	10
789	Invisible Planets	25
101	The Sparrow	15

97	NULL	101
----	------	-----

```
-- User Table  
CREATE TABLE Users (  
    user_ID INT PRIMARY KEY,  
    name VARCHAR(50) NOT NULL,  
    email VARCHAR(100) NOT NULL,  
);  
  
-- Book Table  
CREATE TABLE Books (  
    product_sku INT PRIMARY KEY,  
    title VARCHAR(100) NOT NULL,  
    price DECIMAL(10, 2) NOT NULL,  
);  
  
-- Order Table  
CREATE TABLE Orders (  
    order_no INT PRIMARY KEY,  
    user_ID INT,  
    product_sku INT,  
    FOREIGN KEY (user_ID) REFERENCES Users(user_ID),  
    FOREIGN KEY (product_sku) REFERENCES Books(product_sku)  
);
```

**PRIMARY KEY** of Users  
Where user\_ID is Integer and assigned as Primary key where Entity Integrity Constraint is applicable.

**PRIMARY KEY** of Books  
table Where product\_sku is Integer and assigned as Primary key where Entity Integrity Constraint is applicable.

**PRIMARY KEY** of Books  
table Where product\_sku is Integer and assigned as Primary key where Entity Integrity Constraint is applicable.

**FOREIGN KEY** of Table Users  
IN Table Orders

**FOREIGN KEY** of Table Books  
IN Table Orders

# Thank You

By Dipal Paneri

Topic From @Mentorness