

SQL Introduction

>

SQL SELECT (I)

>

SQL SELECT (II)

>

SQL JOIN

>

SQL DATABASE & TABLE

>

SQL Insert, Update and Delete

>

SQL Constraints

⌵

✓ SQL Constraints

⌵

✓ SQL Not Null Constraint

⌵

✓ SQL Unique Constraints

⌵

✓ SQL Primary Key

⌵

✓ SQL Foreign Key

⌵

✓ SQL Check

⌵

✓ SQL Default

⌵

✓ SQL Create Index

⌵

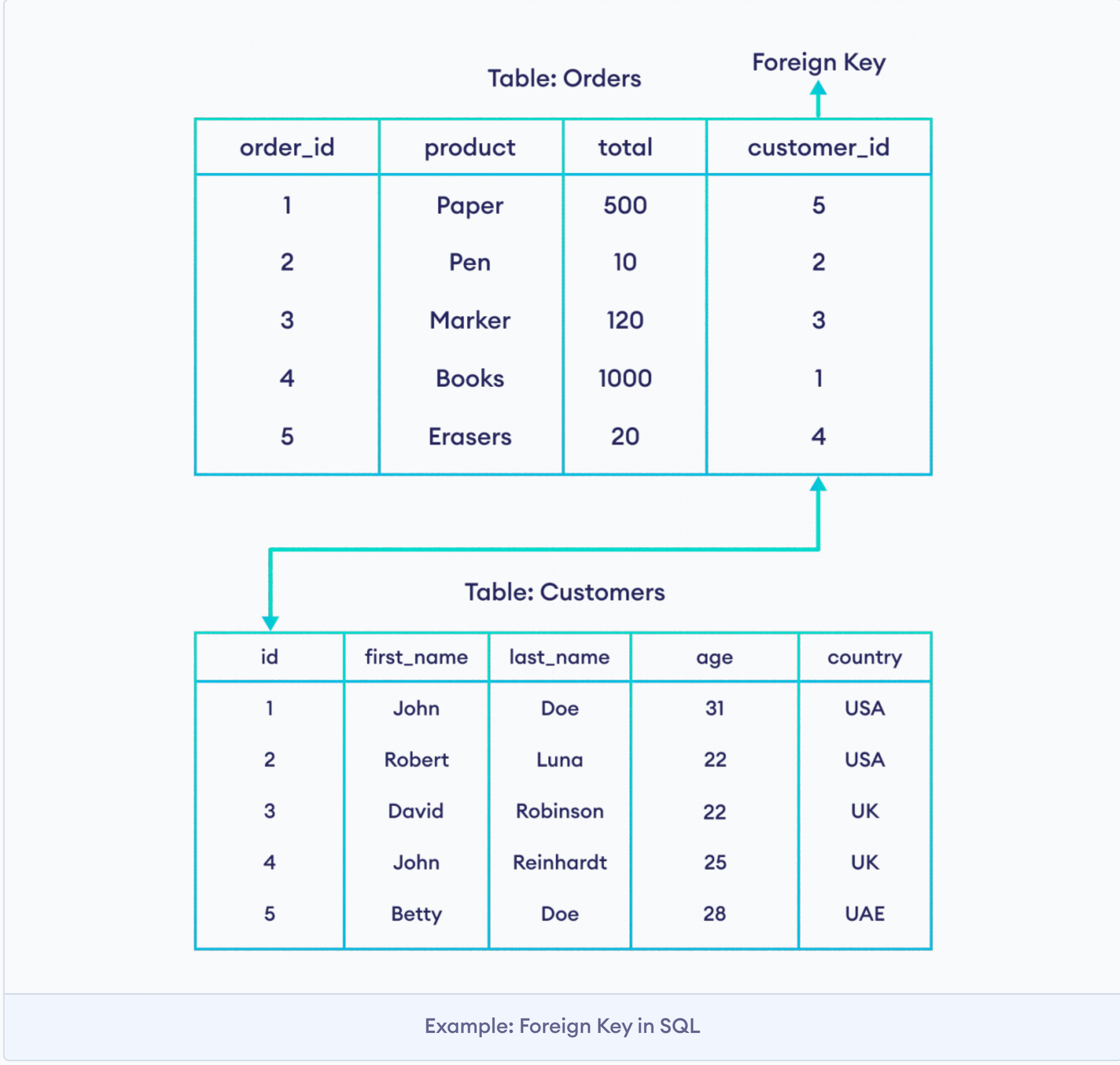
SQL Additional Topics

>

SQL FOREIGN KEY

In this tutorial, we'll learn about the FOREIGN KEY in SQL and how to use them with the help of examples.

In SQL, we can create a relationship between two tables using the FOREIGN KEY constraint.



Example: Foreign Key in SQL

Here, the customer_id field in the Orders table is FOREIGN KEY which references the id field in the Customers table.

This means that the value of the customer_id (of the Orders table) must be a value from the id column (of the Customers table).

Note: The Foreign key can be referenced to any column in the parent table. However, it is general practice to reference the foreign key to the **primary key** of the parent table.

Creating FOREIGN Key

Now, let's see how we can create foreign key constraints in a database.

```
-- This table doesn't have a foreign key
CREATE TABLE Customers (
  id INT,
  first_name VARCHAR(40),
  last_name VARCHAR(40),
  age INT,
  country VARCHAR(10),
  CONSTRAINT CustomersPK PRIMARY KEY (id)
);

-- Adding foreign key to the customer_id field
-- The foreign key references to the id field of the Customers table
CREATE TABLE Orders (
  order_id INT,
  item VARCHAR(40),
  amount INT,
  customer_id INT REFERENCES Customers(id),
  CONSTRAINT OrdersPK PRIMARY KEY (order_id)
);
```

Run Code >>

Here, the value of the customer_id column in the Orders table references the row in another table named Customers with its id column.

Note: The above code works in all major database systems. However, there may be the alternate syntax to create foreign keys depending on the database. Refer to their respective database documentation for more information.

Inserting Records in Table with Foreign Key

Lets try to insert records in a table with foreign keys.

```
-- Inserting record in table with no foreign key first
INSERT INTO Customers
VALUES
(1, 'John', 'Doe', 31, 'USA'),
(2, 'Robert', 'Luna', 22, 'USA');

-- Insertion Success
INSERT INTO Orders
VALUES
(1, 'Keyboard', 400, 2),
(2, 'Mouse', 300, 2),
(3, 'Monitor', 12000, 1);

-- Insertion Error because customer with id of 7 does not exist
INSERT INTO Orders
VALUES (4, 'Keyboard', 400, 7);
```

Run Code >>

Why use Foreign Key?

To normalize data

The FOREIGN KEY helps us to normalize the data in multiple tables and reduce the redundancy. This means, a database can have multiple tables that are related to each other.

Prevent Wrong Data From Insertion

If two database tables are related through a field (attribute), using FOREIGN KEY makes sure that wrong data is not inserted in that field. This helps to eliminate bugs in the database level.

Recommended Reading: [SQL JOIN](#)

Foreign Key with Alter Table

It is possible to add the FOREIGN KEY constraint to an existing table using the ALTER TABLE command. For example,

```
CREATE TABLE Customers (
  id INT,
  first_name VARCHAR(40),
  last_name VARCHAR(40),
  age INT,
  country VARCHAR(10),
  CONSTRAINT CustomersPK PRIMARY KEY (id)
);

CREATE TABLE Orders (
  order_id INT,
  item VARCHAR(40),
  amount INT,
  customer_id INT,
  CONSTRAINT OrdersPK PRIMARY KEY (order_id)
);

-- Adding foreign key to the customer_id field
-- The foreign key references to the id field of the Customers table
ALTER TABLE Orders
ADD FOREIGN KEY (customer_id) REFERENCES Customers(id);
```

Multiple Foreign Key in a Table

A database table can also have multiple foreign keys.

Lets say, we have to record all transactions where each user is a buyer and a seller.

```
-- This table doesn't have a foreign key
CREATE TABLE Users (
  id INT,
  first_name VARCHAR(40),
  last_name VARCHAR(40),
  age INT,
  country VARCHAR(10),
  CONSTRAINT CustomersPK PRIMARY KEY (id)
);

-- Adding foreign key to the buyer and seller field
-- The foreign key references to the id field of the Users table
CREATE TABLE Transactions (
  transaction_id INT,
  amount INT,
  seller INT REFERENCES Users(id),
  buyer INT REFERENCES Users(id),
  CONSTRAINT TransactionsPK PRIMARY KEY (transaction_id)
);
```

Run Code >>

Here, the SQL command creates two foreign keys (buyer and seller) in the Transactions table.

Recommended Reading: [SQL Constraints](#)

Previous Tutorial:
[SQL Primary Key](#)

Next Tutorial:
[SQL Check](#) →

Did you find this article helpful?

😊

😞

Related Tutorials

- Programming

SQL Constraints
- Programming

SQL NOT NULL Constraint
- Programming

SQL PRIMARY KEY
- Programming

SQL CREATE TABLE Statement

Programiz



Tutorials

- Python 3 Tutorial
- JavaScript Tutorial
- SQL Tutorial
- C Tutorial
- Java Tutorial
- Kotlin Tutorial
- C++ Tutorial
- Swift Tutorial
- C# Tutorial
- Go Tutorial
- DSA Tutorial

Examples

- Python Examples
- JavaScript Examples
- C Examples
- Java Examples
- Kotlin Examples
- C++ Examples

Company

- About
- Advertising
- Privacy Policy
- Terms & Conditions
- Contact
- Youtube

Apps

- Learn Python
- Learn C Programming
- Learn Java