A Crash Course in SQL
Part 3

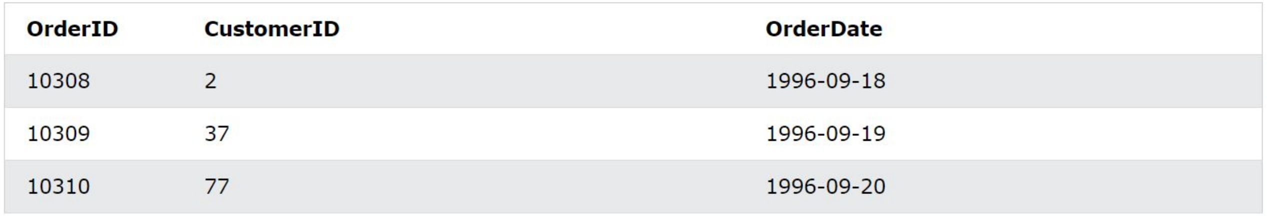
SQL Covered Today:

- Joining Tables
- Working with Dates

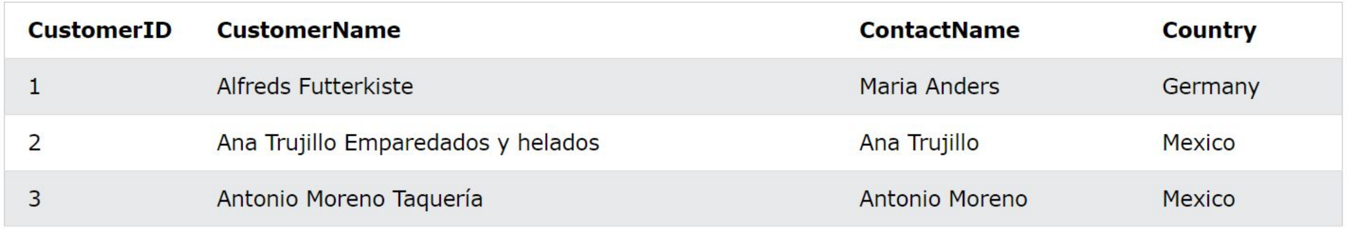
Joining Tables

MySQL Joining Tables

To access information contained in more than one table, we need to *join* the tables. A `JOIN` clause is used to combine rows from two or more tables, based on a related column between them. Let's look at a selection from the "Orders" and “Customers” tables:



OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10309	37	1996-09-19
10310	77	1996-09-20



CustomerID	CustomerName	ContactName	Country
1	Alfreds Futterkiste	Maria Anders	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mexico

Notice that the "CustomerID" column in the "Orders" table refers to the "CustomerID" in the "Customers" table. The relationship between the two tables above is the "CustomerID" column

MySQL Joining Tables

OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10309	37	1996-09-19
10310	77	1996-09-20

CustomerID	CustomerName	ContactName	Country
1	Alfreds Futterkiste	Maria Anders	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mexico

SQL statement contains an `INNER JOIN` thatselects records that have matching values in both tables:

```
SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate
FROM Orders
INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;
```

MySQL Joining Tables

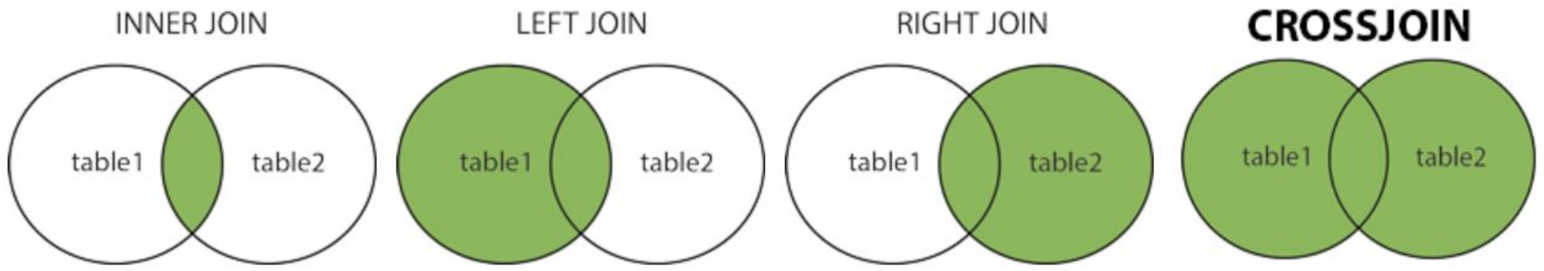
```
SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate
FROM Orders
INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;
```

Produces the following result:

OrderID	CustomerName	OrderDate
10308	Ana Trujillo Emparedados y helados	9/18/1996
10365	Antonio Moreno Taquería	11/27/1996
10383	Around the Horn	12/16/1996
10355	Around the Horn	11/15/1996
10278	Berglunds snabbköp	8/12/1996

Supported Types of Joins in MySQL

- `INNER JOIN`: Returns records that have matching values in both tables
- `LEFT JOIN`: Returns all records from the left table, and the matched records from the right table
- `RIGHT JOIN`: Returns all records from the right table, and the matched records from the left table
- `CROSS JOIN`: Returns all records from both tables



MySQL Dates

MySQL Dates

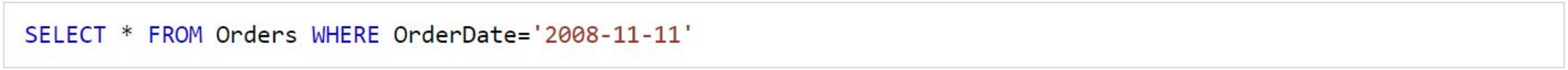
MySQL comes with the following data types for storing a date or a date/time value in the database:

- `DATE` - format YYYY-MM-DD
- `DATETIME` - format: YYYY-MM-DD HH:MI:SS
- `TIMESTAMP` - format: YYYY-MM-DD HH:MI:SS
- `YEAR` - format YYYY or YY

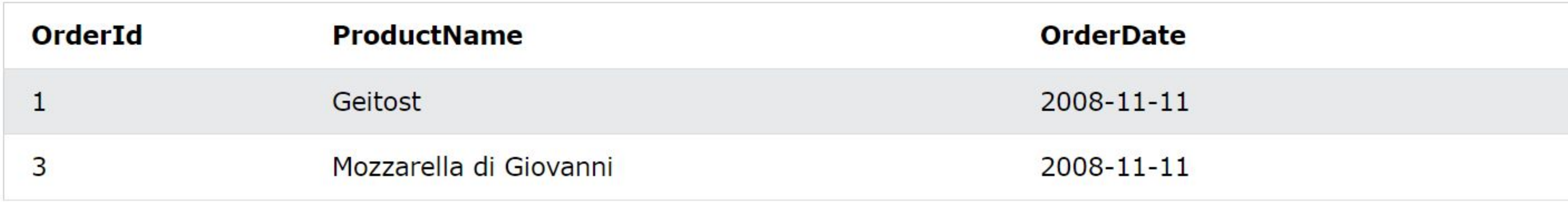
```
CREATE TABLE Orders(

      OrderID int NOT NULL,

      ProductName varchar(255) NOT NULL,

      OrderDate DATE NOT NULL,

      PRIMARY KEY …

);
```

Working with Dates

- Selecting dates with a specific date (once you use a consistent date format, stick with ‘YYYY-MM-DD’):



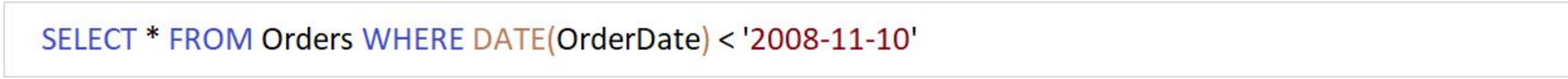
```sql
SELECT * FROM Orders WHERE OrderDate='2008-11-11'
```



OrderId	ProductName	OrderDate
1	Geitost	2008-11-11
3	Mozzarella di Giovanni	2008-11-11

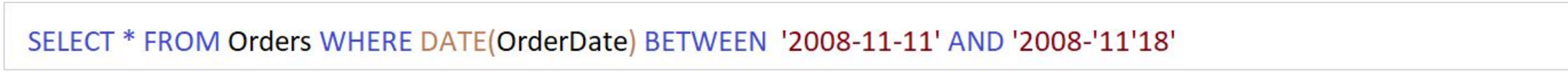
Working with Dates

- Comparing dates using < or >, use the mysql DATE() function



```
SELECT * FROM Orders WHERE DATE(OrderDate) < '2008-11-10'
```

- Similarly, you can use with the DATE() function:
  - BETWEEN <Date 1> AND <Date 2>



```
SELECT * FROM Orders WHERE DATE(OrderDate) BETWEEN  '2008-11-11' AND '2008-'11'18'
```

Working with Today’s Date

- mysql CURDATE() function allows you to work with the current date

```
SELECT * FROM Orders WHERE OrderDate = CURDATE()
```

- The above will return all of today’s orders stored in the Orders table

What we've covered today:

- Joining Tables
- Working with Dates