the Master Course

{CODENATION}

Backend Development

SQL Relationships

{CODENATION}

{CODENATION}

Learning Objectives

To construct diagrams to represent tables in a MySQL database.
Use Logical operators and Case statements in MySQL.
To know the different types of relationships in a MySQL database To use Joins in MySQL when working with multiple tables of related data

What are SQL Relationships?

SQL Relationships are based on common fields that exist in two or more tables.

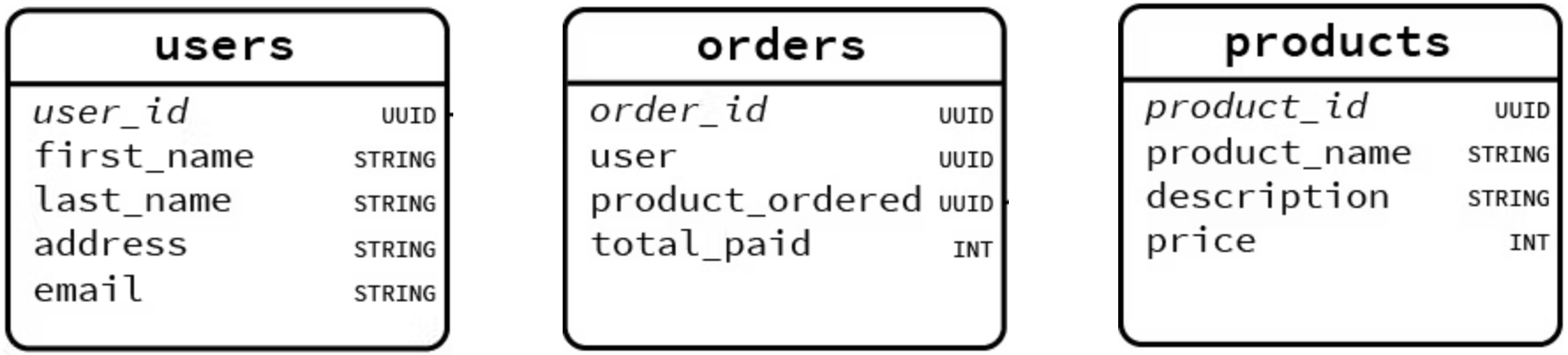
Primary Key  Foreign Key

How did we use **keys** yesterday to link our tables?

Recap ...

Primary Keys are an identifying field in a table. They must be unique and cannot be null.

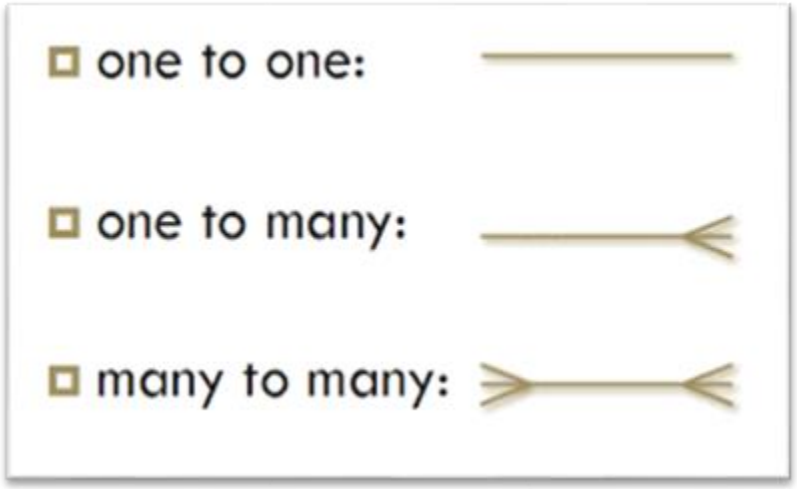
Foreign Keys reference a Primary Key from another table.



How could we view the orders made by a user?

You could query the orders created by a user by searching for the user's primary key in the foreign key field of the orders table.

There are three main types of SQL Relationship:



One To One

For every entry in the primary table, there is only one entry in the foreign table.

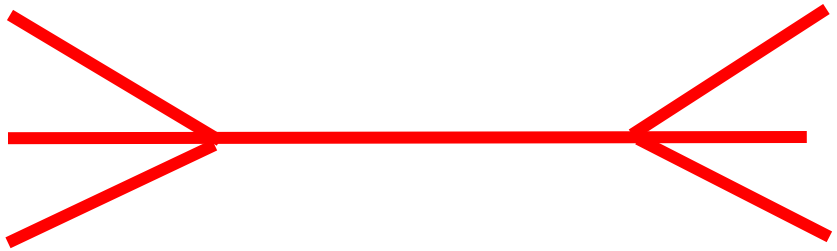
For example, in a school database, each student has only one student ID, and each student ID is assigned to only one person.

One To Many

For every entry in the primary table, there are two or more entries in the foreign table.

For example, If the two entity types are 'Customer' and 'Account,' each 'Customer' can have many 'Accounts,' but each 'Account' can only be owned by one 'Customer. '

Many To Many



SQL

For every entry in the primary table there are many related entries in the foreign table.

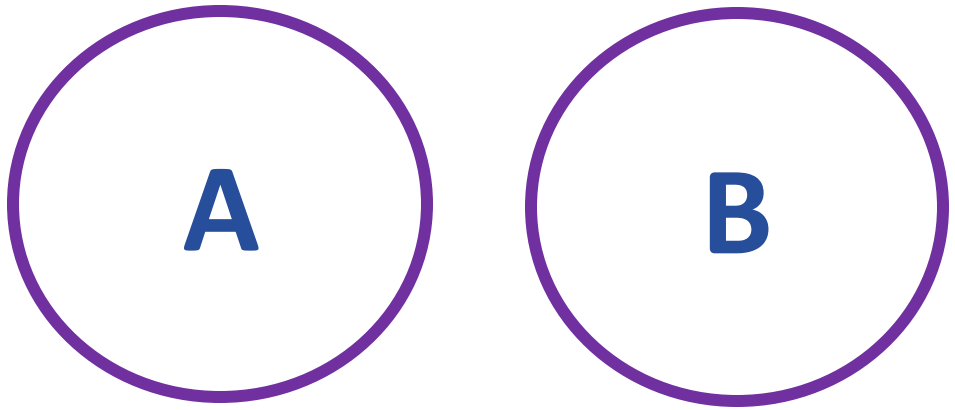
Also, for every entry in the foreign table there are many related entries in the primary table.

For example, products and suppliers: one supplier may deliver one or many products and at the same time, the company may order one product from one or many suppliers.

How do we use these relationships in a query?

This is where we use the JOIN keyword.

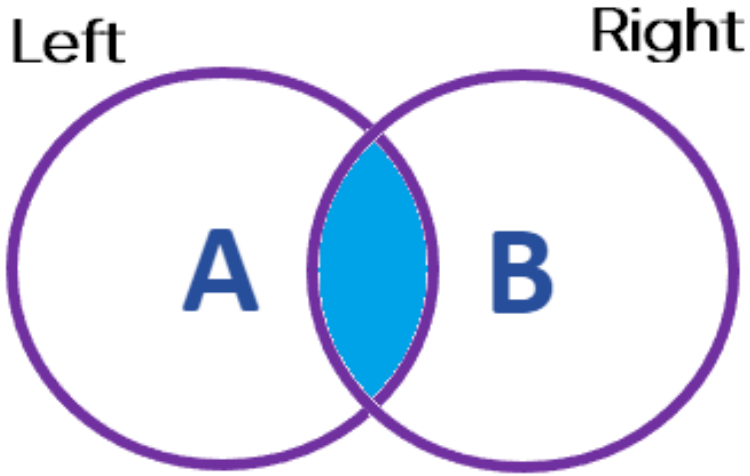
Consider these two data 'tables'
- A and B.



We can join them in different ways...

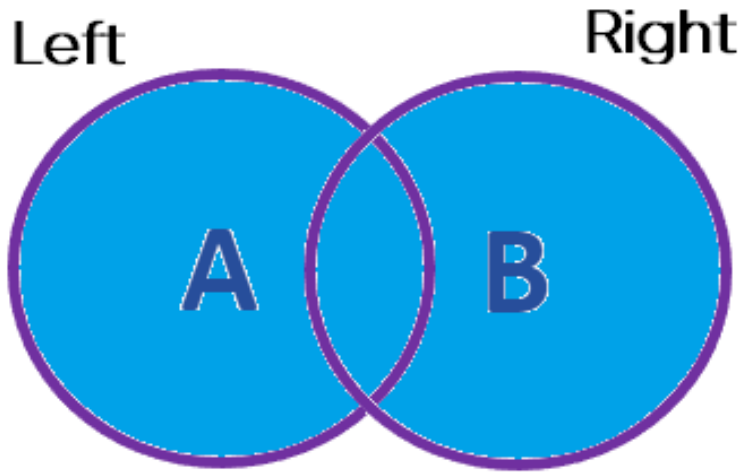
INNER JOIN

Inner Join clause in SQL Server creates a data collection by combining rows that have matching values in
two or more tables.



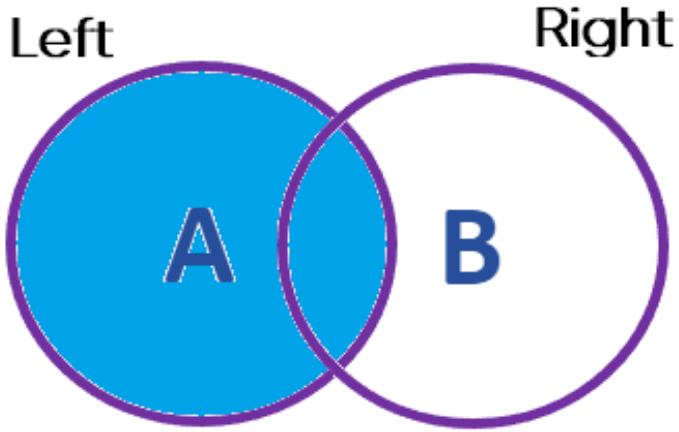
FULL OUTER JOIN

The 'Full Outer Join' clause in SQL creates a data collection which includes all rows from the joined tables
whether the other table
has the matching row,
or not.



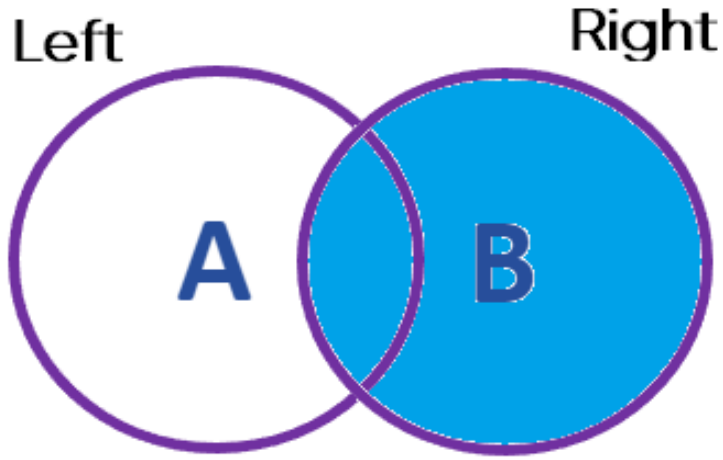
LEFT OUTER JOIN

The 'Left Outer Join' clause in SQL creates a data collection by returning all the records from the right table and those records which satisfy a condition from
the left table.

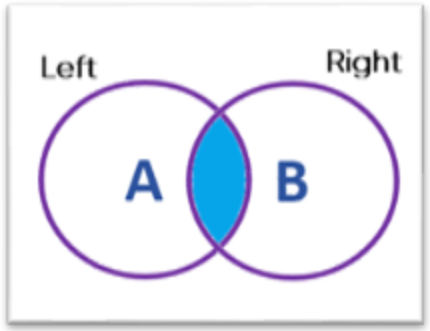


RIGHT OUTER JOIN

Inner Join clause in SQL Server creates a data collection by combining rows that have matching values
in two or more tables.



Let's JOIN ...



title	author	publisher	price	genre	in_stock	Id
David Copperfield	1	Bentleys	8	Traditional	1	1
Esio Trot	2	J Cape	2	Childrens	1	2
Hail Mary	4	The Crown	25	Sci-Fi	1	3
Matilda	2	J Cape	30	Childrens	0	4
Oliver Twist	1	Bentleys	6	Traditional	1	5
Phase Space	5	Doubleday	4	Sci-Fi	1	6
The Long Earth	5	Doubleday	2	Sci-Fi	0	7
The Martian	4	The Crown	4	Sci-Fi	0	8
The Twits	2	J Cape	5	Childrens	1	9
War Horse	3	Kaye & Ward	4	Childrens	1	10

id	first_name	middle_initial	surname
1	Charles	J	Dickens
2	Roald	J	Dahl
3	Michael	J	Morpurgo
4	Andy	J	Weir
5	Steven	J	Baxter
6	Stephen	E	King
7	John	R	Grisham
8	Ian		Fleming
9	Douglas		Adams
10	George	R	Martin

As we saw earlier, this SQL Statement will return the list of all books which match the AuthorId on both tables.

```
SELECT title, first_name, middle_initial, surname, genre FROM books
INNER JOIN authors
ON books.author = authors.id;
```

title	first_name	middle_initial	surname	genre
David Copperfield	Charles	J	Dickens	Traditional
Esio Trot	Roald	J	Dahl	Childrens
Hail Mary	Andy	J	Weir	Sci-Fi
Matilda	Roald	J	Dahl	Childrens
Oliver Twist	Charles	J	Dickens	Traditional
Phase Space	Steven	J	Baxter	Sci-Fi
The Long Earth	Steven	J	Baxter	Sci-Fi
The Martian	Andy	J	Weir	Sci-Fi
The Twits	Roald	J	Dahl	Childrens
War Horse	Michael	J	Morpurgo	Childrens

Any unmatched rows will not be displayed.

CODE ALONG

Activity 1

Experiment with the LEFT JOIN and RIGHT JOIN statements to return different data sets back.

Add relevant comments to your SQL statements.

Stretch

You can add WHERE clause to your JOIN statements. Why would this be useful in certain circumstances? Experiment with the addition of this WHERE clause and report back to your team with an explanation of your findings.

What's Wrong ?

SQL

The following two database tables are correct.

I want to find the following: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Can you take a look at this SQL statement and figure out what is wrong

Activity 2

- What is the total number of employees?
- Who is the highest earning employee and what is their job title?
- How many Senior Developers are there?
- How many developers are in each wage bracket?
- Who is the only employee who isn't a developer and what is their salary?

Stretch

Add multiple tables to your movies database and create join queries.

{CODENATION}

Learning Objectives

To construct diagrams to represent tables in a MySQL database.
Use Logical operators and Case statements in MySQL.
To know the different types of relationships in a MySQL database To use Joins in MySQL when working with multiple tables of related data