

SQL for Data Science

Module 8: Getting Data from Multiple Tables

- The need for joins
- Different types of joins
- When to use which join?
- Joining tables – Examples
- Self Join
- Self Join - Example
- Subquery
- Subquery - Example

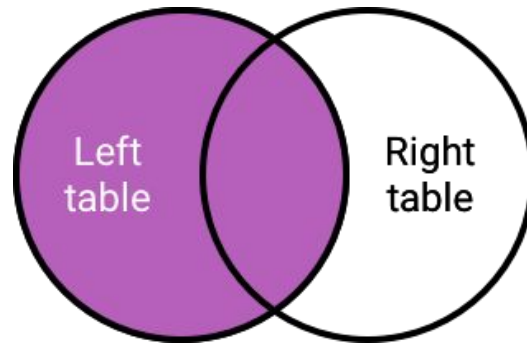


What is a Join?

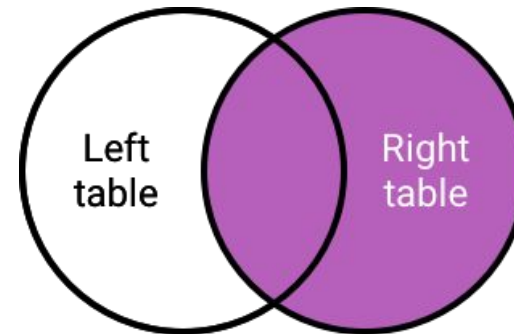
JOIN is just a clause in MySQL which helps us retrieve data from multiple tables.

Types of Joins

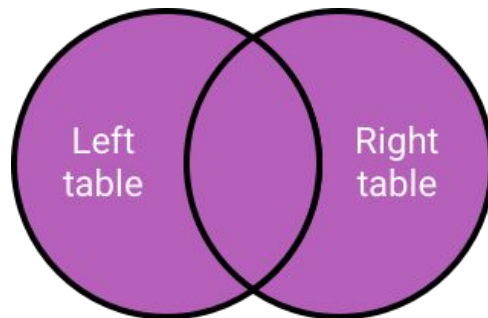
Types of Joins



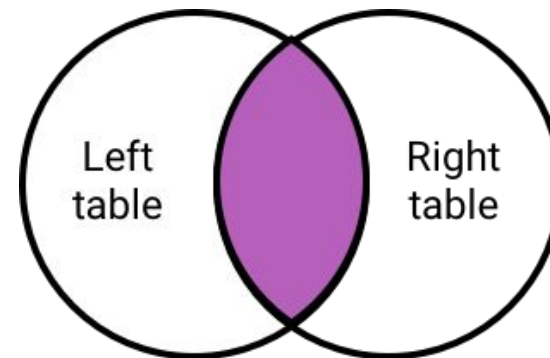
Left Join (90%)



Right Join (< 0.01 %)



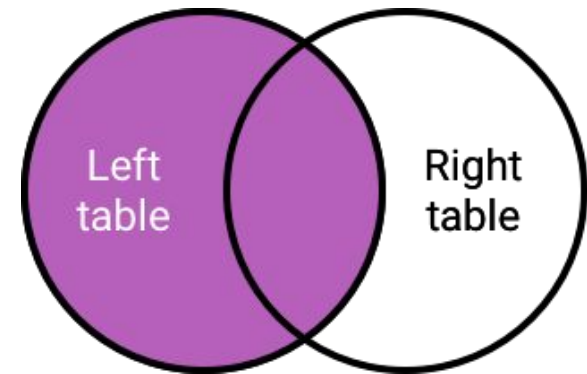
Cartesian Join (< 0.01 %)



Inner Join (9.99%)

Left Join

- Take all the records from the left table and get only the corresponding (matching) records from the right table
- ONLY records matching to the left table are considered
- Matching is done based on a particular column known as the key



Command

```
select <table 1> .a, <table 1> .b, <table 2> .c from <table 1> left join <table 2> on  
    <table1>.<key1> = <table2>.<key2>
```

Left Join – Practical Example

Purchases

- Purchase ID
- Department
- Price
- Item Name
- User ID

Contact Details

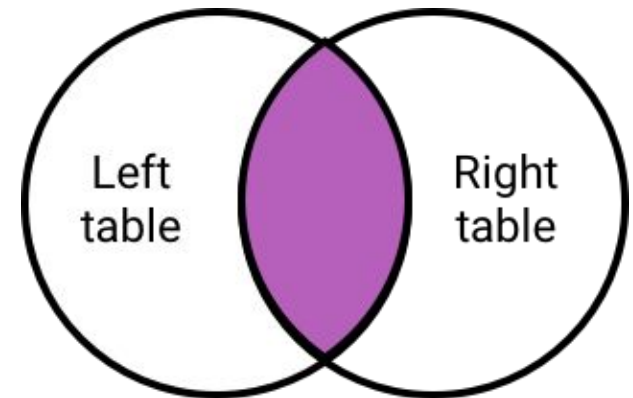
- User ID
- Name
- Phone Number

Query to find the Contacts of top revenue customers

```
select ContactDetails.Name, ContactDetails.PhoneNumber from  
Purchases left join ContactDetails  
on Purchases.UserID = ContactDetails.UserID
```

Inner Join

- The inner join returns ONLY the records that are matching in BOTH the tables
- Can do one inner join after other – there is no limit
- More the number of tables involved – Slower the query

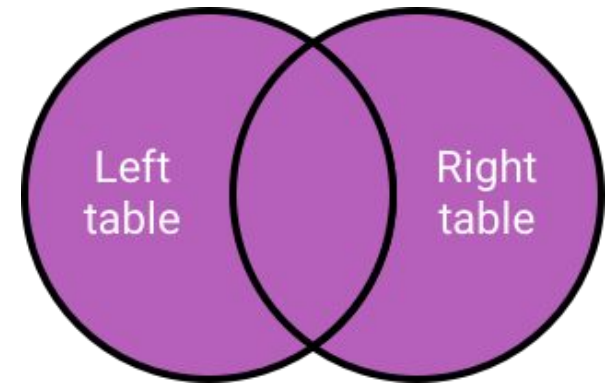


Command

```
select a, b, c from <table 1> inner join <table 2> on <table1>.<key1> = <table2>.<key2>
```

Cartesian Join or Cross Join

- Each row of the first table joins all the rows of second table
- It is computationally taxing
- If 1M records in one table and 1M in another – we need 10^{12} computations
- Not frequently used

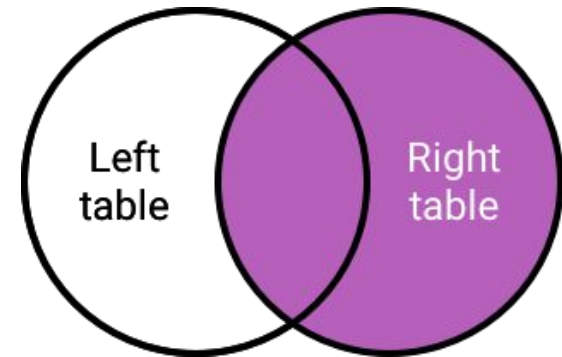


Command

```
select a, b, c from <table 1> cross join <table 2>
```


Right Join

- Take all the records from the right table and get only the corresponding (matching) records from the left table
- ONLY records matching to the right table are considered
- Unlimited number of tables can be joined one after another



Command

```
select a, b, c from <table 1> right join <table 2> on <table1>.<key1> = <table2>.<key2>
```

Self Join

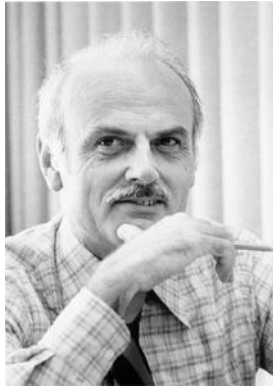
- Self Join is when we join a table to itself
- Why do we do so?
 - Creating pairwise lists
 - No other way left
 - Difficult to debug

Command

```
select U.a, V.b from table as U <left/inner/..> join table as V on U.<key> = V.<key>
```

SQL started with IBM Researcher Edgar Codd's Research on Relational Databases

1972



Edgar Codd

- **Researcher at IBM Research Center**
- **Mathematician trained from Oxford**
- **Researching on Relational Databases**
- **Chamberlin and Boyce come up with SEQUEL (Structured English Query Language to interact with IBM System R database)**

1979



- **Trademark Issue with a Firm**
- **SEQUEL was changed to SQL**

Connecting to MySQL Server

Connecting to MySQL Server is pretty straightforward

Goto Terminal/ Command Prompt and type

```
[anands-MacBook-Pro:~ analytics$ mysql -uroot -p  
[Enter password:
```

Data Definition Language

Commands used to

- Define the schema of database or its objects (like tables and indexes)
- Create and Modify the structure of database objects
- Examples:
 - CREATE
 - DROP
 - ALTER

Data Manipulation Language

Commands used to

- Manipulate and Select data in the database
- Examples:
 - SELECT
 - INSERT
 - UPDATE
 - DELETE

Data Control Language

Commands dealing with

- Rights, permissions and other controls of the database system
- Examples:
 - GRANT
 - REVOKE

Here we explore some simple commands. Note that all commands end with ; or \G in MySQL

Show all databases

```
mysql> show databases;
```

Work with a particular database

```
mysql> use <database_name>;
```

Get help about commands

```
mysql> help;
```

Get topicwise help

```
mysql> help contents;
```

```
mysql> help Data Manipulation;
```


Here we explore some simple commands. Note that all commands end with ; or \G in MySQL

Show all databases

```
mysql> show databases;
```

Work with a particular database

```
mysql> use <database_name>;
```

Get help about commands

```
mysql> help;
```

Get topicwise help

```
mysql> help contents;
```

```
mysql> help Data Manipulation;
```

We have listed the most commonly used datatypes here. There are a lot more, to learn more: Refer to <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

Most Popular

- int(10)
- varchar(255)
- text
- TIMESTAMP
- ENUM ('Choice1', 'Choice2', ...)

Not so common

- FLOAT
- DECIMAL
- BLOB
- TINYBLOB
- MEDIUMBLOB
- BIGINT
- SMALLINT
- TINYINT
- DATE
- TIME
- SET
- DOUBLE
- CHAR

Some fields we can keep optional – Others are Mandatory

Difference between NULL and NOT NULL Columns/ Fields

- A column which has NOT NULL constraint means it is mandatory to put some value for the column while inserting the row
- A column which has NULL constraint means its ok to give NULL value – a special value which means blank
- This is defined in the structure of the table