DATA SCIENCE INFINITY

SQL

FOR DATA SCIENCE

AN INTRODUCTION

Let's start at the top...

SQL stands for **Structured Query Language**

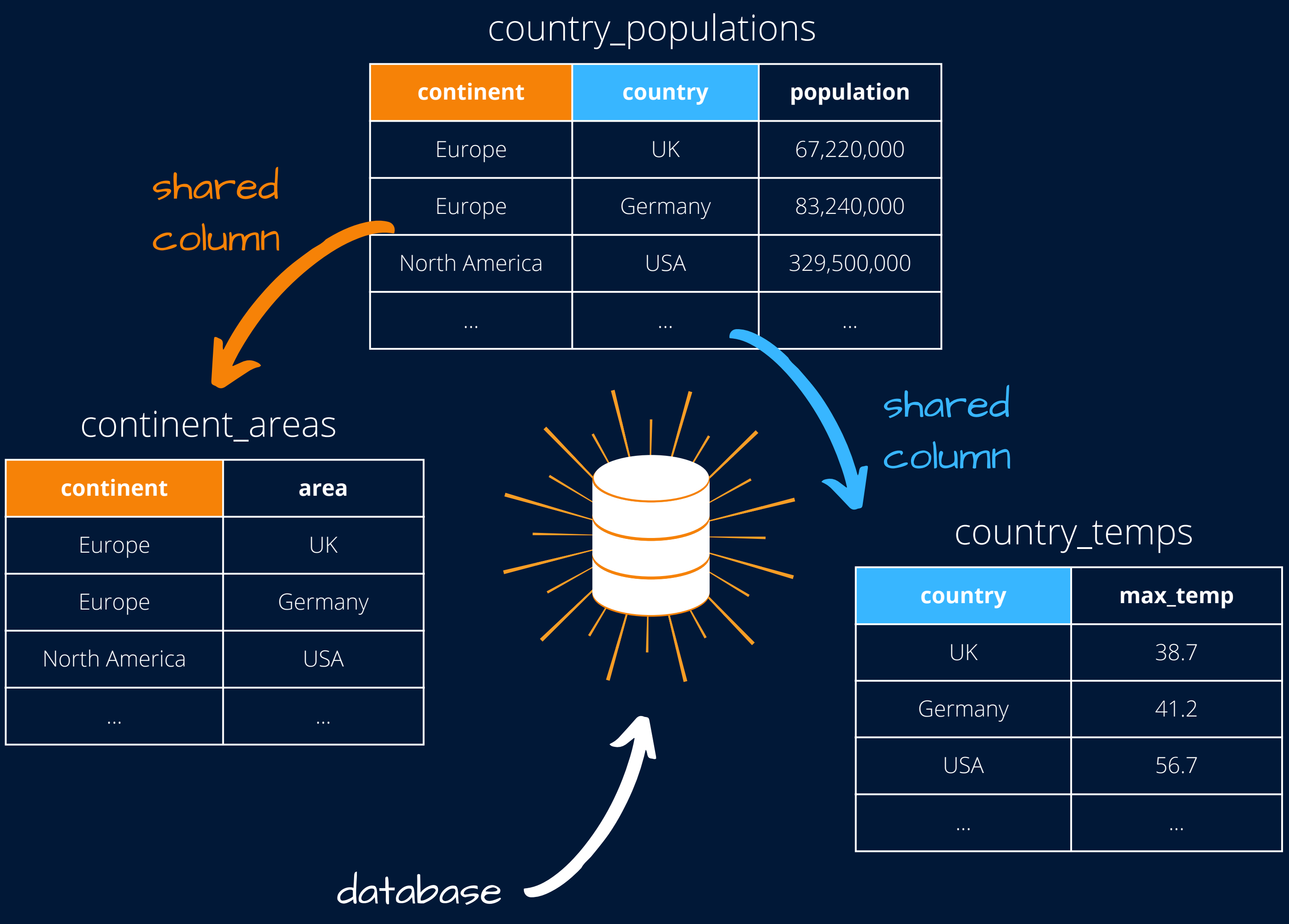
Referred to as "**S-Q-L**" as well as "**Sequel**"

It is known to be the **easiest** programming language to learn & use due to the "common sense" nature of the commands

Used to store, extract & manipulate data in **relational databases**

Relational Database?

A **relational database** is a collection of tabular datasets (*think columns & rows*) that **relate** to each other through **shared** columns



country_populations

continent	country	population
Europe	UK	67,220,000
Europe	Germany	83,240,000
North America	USA	329,500,000
...	...	...

continent_areas

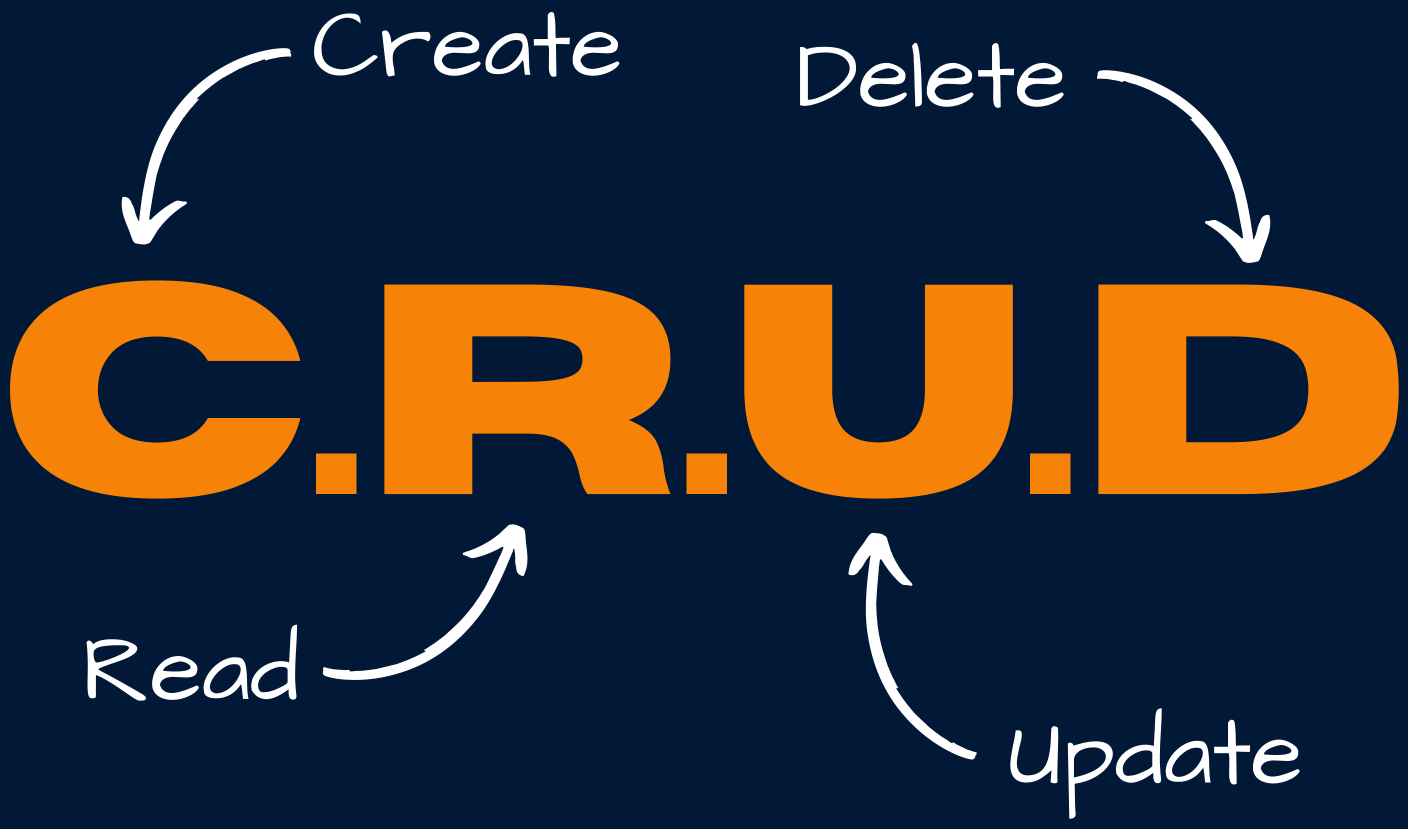
continent	area
Europe	UK
Europe	Germany
North America	USA
...	...

country_temps

country	max_temp
UK	38.7
Germany	41.2
USA	56.7
...	...

What can we do?

A good way to think about what we can using SQL is with the acronym **C.R.U.D**



Create

Delete

C.R.U.D

Read

Update

While this might seem like a slightly informal acronym it’s actually a really good way to describe the core functions or operations that can be performed on a relational database...

Let's take a look!

C

CREATE: We can create **databases**, **schemas** (which are almost like a partitioned area to help keep things organised) and of course we can create **tables** as well!

R

READ: This is mainly about **querying** the data, so essentially **grabbing the relevant rows and columns** from tables that will provide us with the information we need

U

UPDATE: We can **add** more rows & columns to tables that already exist, as well as **modify** records within tables

D

DELETE: This is kinda what you'd expect - we can delete specific **rows and columns**, or we can delete whole **tables**, **schemas** and even **databases**!

SQL in Data Science 1

While all of these C.R.U.D processes can be undertaken using SQL - Data Scientists and Data Analysts will typically spend most of their time in the "Read" area...

C.R.U.D

Read

In a lot of companies the **management** of the databases themselves (so the Create, Update, and Delete functions) are often taken care of by a specific database team, or by Data Engineers.

In saying that however, a **great** Data Scientist or Analyst should have an understanding of how the data they're using is being imported & created as well as how it's being managed and changed over time - so knowing at least the fundamentals of the other functions can be very useful

SQL in Data Science 2

In **Data Science** - common tasks that use **SQL** will be...

✓ Querying & exploring data to extract **useful business insights**

✓ Gathering & aggregating data for business **reporting**

✓ Selecting data for a **specific treatment**, e.g. selecting customers to receive a targeted promotion

✓ Extracting data for **Machine Learning** tasks or other predictive modelling

A simple code example...

We are the owner of **Rolex**, and we're looking for a new spokesperson for our very elite range of watches.

player_details

first_name	last_name	sport	net_worth
Roger	Federer	Tennis	$900m
Novak	Djokovic	Tennis	$220m
Sachin	Tendulkar	Cricket	$170m
Yao	Ming	Basketball	$120m
LeBron	James	Basketball	$500m
Lewis	Hamilton	Motorsport	$280m

For our simple example, we a single table of data called **player_details** that contains 6 famous sports people.

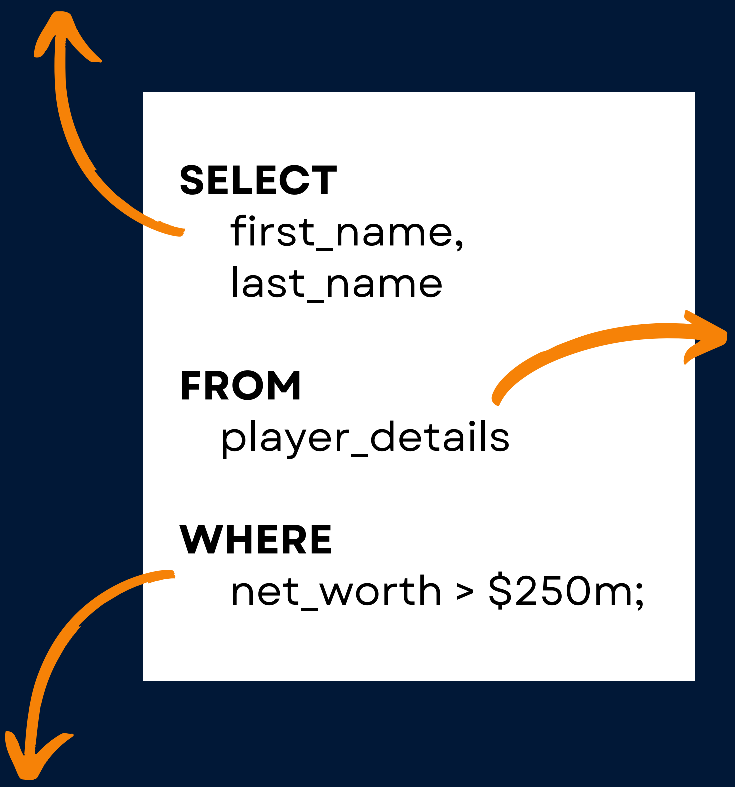
We want to create a shortlist of **names** who are worth **over $250m dollars** - we only want the wealthiest of athletes representing our product of course!

What would the SQL query for this look like?

DATA SCIENCE INFINITY

A simple code example...

We use the **SELECT** statement to specify which **columns** from the original dataset we want returned. We only needed the names, so we've listed those columns with a comma seperating them



```
SELECT
   first_name,
   last_name

FROM
  player_details

WHERE
   net_worth > $250m;
```

We use the **FROM** statement to specify the name of the table that this information resides in

The **WHERE** statement is used to apply any row level filters. Our only requirement was to limited the results to sportspeople worth over $250m - so this is where we apply that rule!

A simple code example...

player_details

first_name	last_name	sport	net_worth
Roger	Federer	Tennis	$900m
Novak	Djokovic	Tennis	$220m
Sachin	Tendulkar	Cricket	$170m
Yao	Ming	Basketball	$120m
LeBron	James	Basketball	$500m
Lewis	Hamilton	Motorsport	$280m

```
SELECT
    first_name,
    last_name

FROM
   player_details

WHERE
    net_worth > $250m;
```

first_name	last_name
Roger	Federer
LeBron	James
Lewis	Hamilton

Voila! Our shortlist of potential spokespeople for our new range of watches!

What else can we do?

Our example covered a **very simple** query - there is much, much more flexibility with SQL that means we can do a whole lot more in terms of processing and manipulating data, such as...

Task	SQL Clause
Find Unique Values	DISTINCT
Merge Multiple Tables	JOIN
Aggregation	SUM, MAX, COUNT (+ GROUP BY)
Appending	UNION, UNION ALL
Conditional Logic	CASE WHEN
Apply logic to a set of rows	RANK, NTILE, LAG, LEAD (Window Functions)