SQL for Data Scientists

Data Science Retreat

September 2020

Antonio Rueda-Toicen

About me

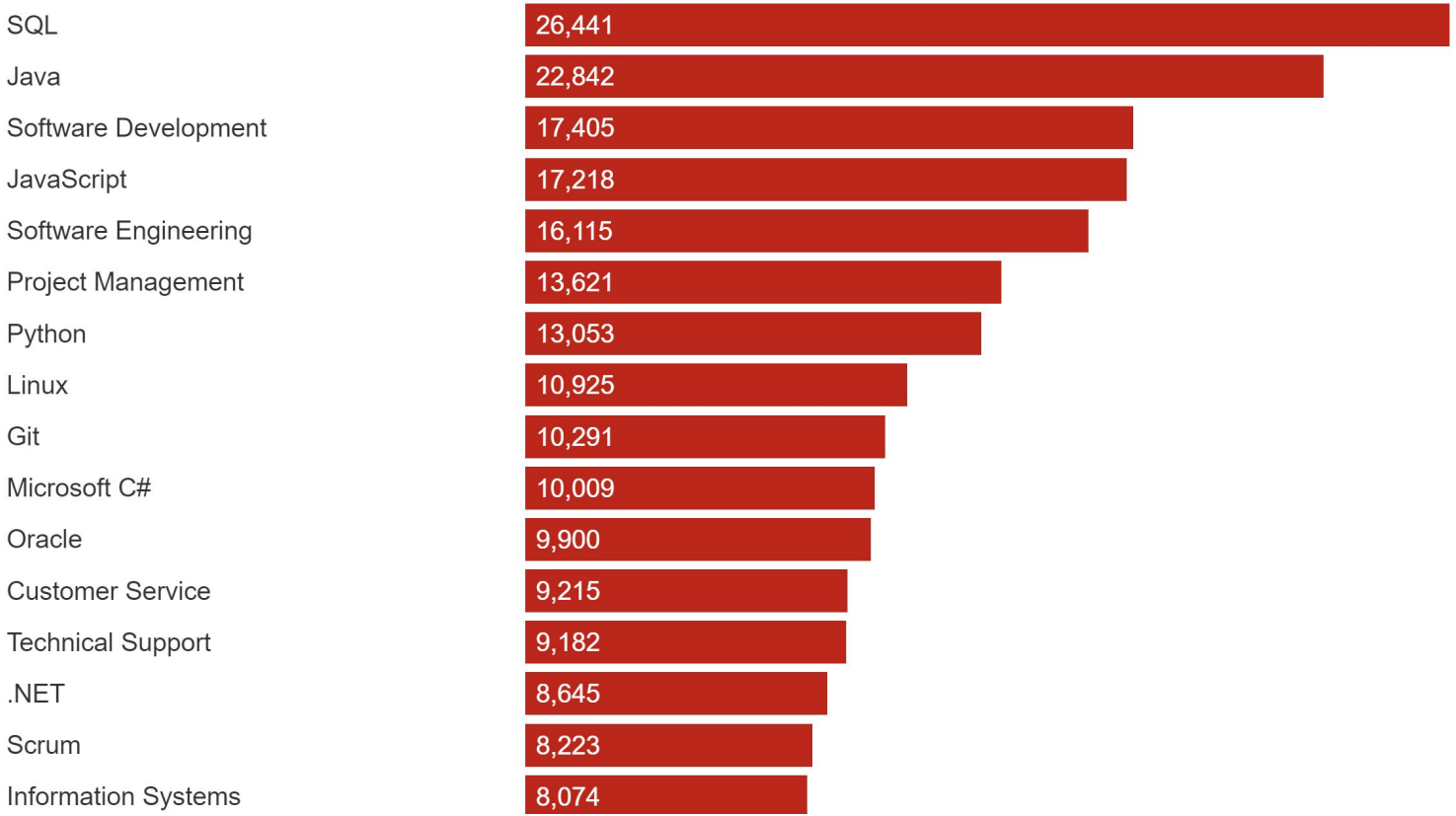
- Senior Data Scientist at Neuraltrain GmbH
  - Background in academia (computer science and bioengineering)
  - Organizer of the Berlin Computer Vision Group
  - Arepa-lover, you can usually find me at Karrecho, near Ostkreuz

Fig 1: Arepas

Agenda

- Intro to SQL
  - What is this and why should we learn it?
  - Relational databases
  - Differences in SQL implementations
  - SQL vs NoSQL
  - Relational databases: schemas and tables
- SQL syntax
  - SQLBolt drills
  - SQL review quiz
  - HackerRank drills
- Intro to Google BigQuery
  - Querying the Google Analytics dataset

Most-Requested Tech Skills, Feb 18 - March 18, 2020



Source: Burning Glass Technologies (https://www.burning-glass.com/)

> *The popularity of SQL is not really surprising given the amount of data which is collected and churned out by thousands of businesses every single day. That said, it's only possible to make sense of big data if you have the right skillset. Data scientists rely heavily on SQL skills which is why SQL as a skill is in such high demand. Interestingly, in the 2019 edition of the Global Technical Hiring and Skills Report, SQL was the #1 IT most developers got tested regardless of their focus.*

Source: Burning Glass Technologies (https://www.burning-glass.com/)

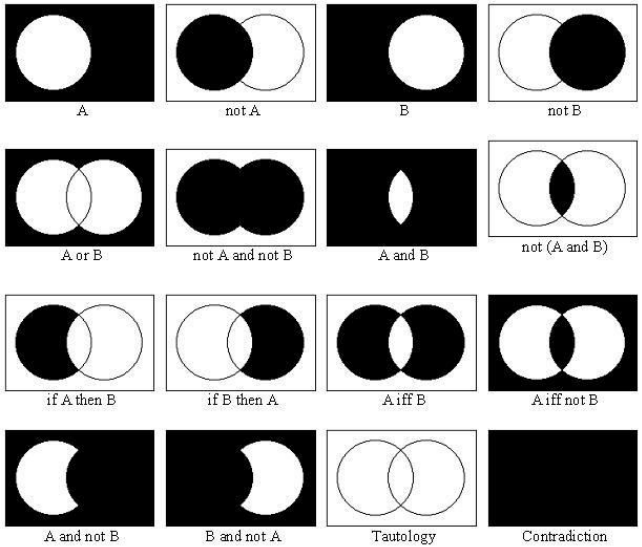
SQL in different job roles

- ***Everyone*** in data uses SQL

- Data engineering, data analysis, machine learning engineering, software engineering, and data science roles all ***require some level of knowledge of SQL***

What is SQL?

- Pronounced “sequel” or “ES-CUE-L” (nobody should frown at either)
- Stands for “Structured Query Language” and is based in set theory
- Has been around since the 1970s (check out “relational algebra” in Wikipedia)



What is a relational database?

bodyweight + dumbell workouts

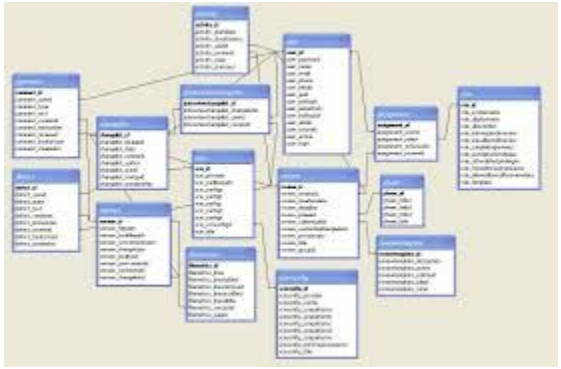
File Edit View Insert Format Data Tools Add-ons Help Last edit was

	A	B	C	D	E
1		pushups	goblet squats	overhead press	squat + overhead press
2	27/04/20	198	85	20	0
3	28/04/20	128	36	0	0
4	29/04/20	72	0	0	0
5	30/04/20	276	0	0	0
6	01/05/20	0	100	0	0
7	02/05/20	0	0	0	0
8	03/05/20	67	25	0	0
9	04/05/20	96	25	0	0
10	05/05/20	108	75	0	0
11	06/05/20	0	0	0	0
12	07/05/20	66	0	0	0
13	08/05/20	622	111	0	0
14	09/05/20	15	0	0	0
15	10/05/20	0	0	0	0

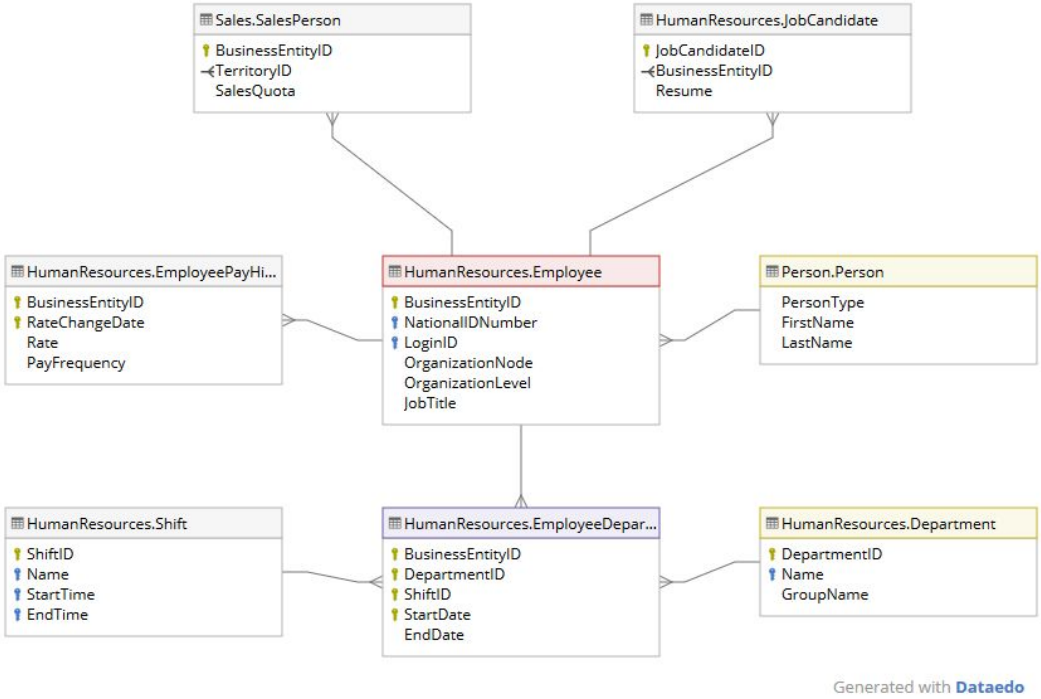
Sheet1 Sheet2

What is a schema?

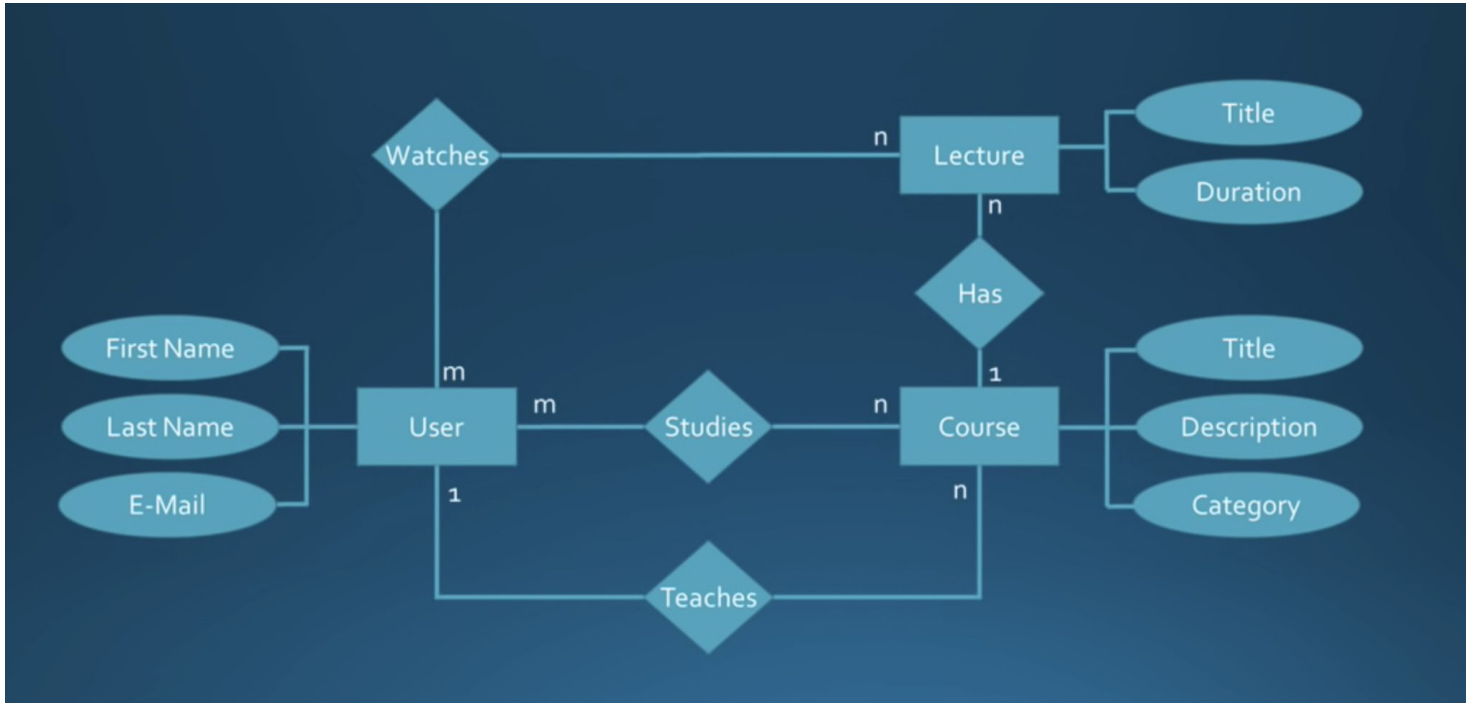
A schema is a "data model", the representation in tables of the **relationships** between **entities** (i.e., *nouns*, tables) and **attributes**(i.e. *adjectives*, features, columns within the tables)



Entity-Relationship diagrams



Entity-Relationship diagrams

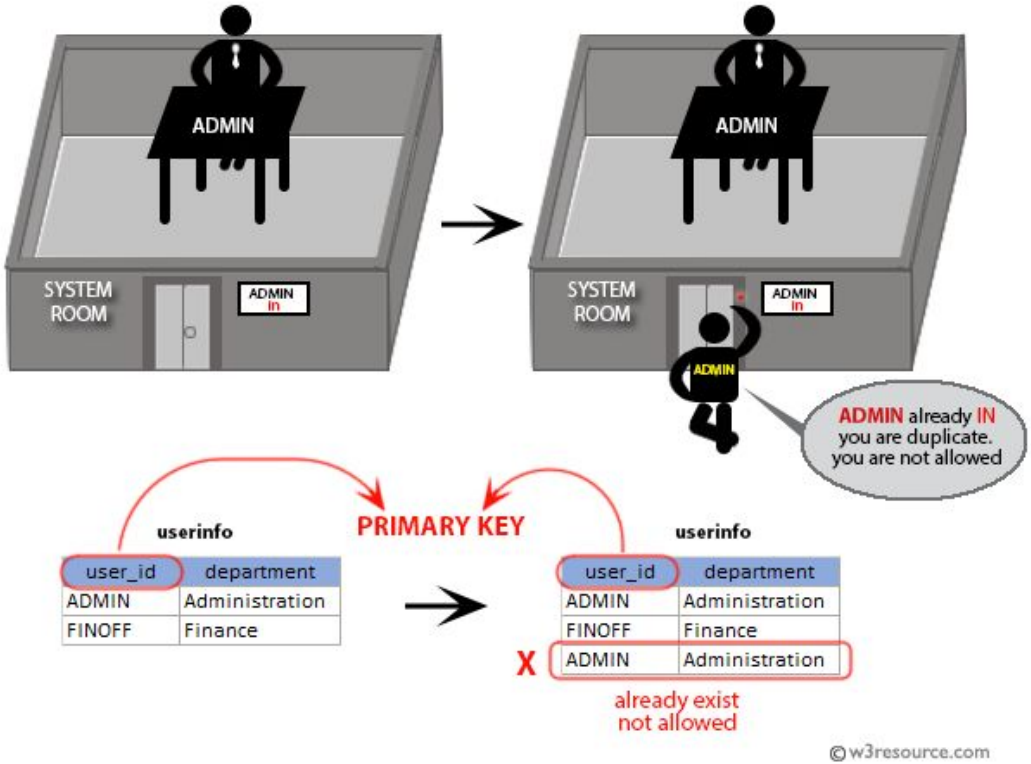


Source: [a brief tutorial on ER diagrams] (recommended watch)

What’s SQL role in an RDBMS?

- SQL was Created for managing data in relational database management systems (RDBMS)
- SQL takes *instances of relations* as input, yields them as output (views, tables)
- RDBMS comprise data models that detail relationships between tables
    - One to one
    - One to many
    - Many to many relationships

Understanding “primary keys”



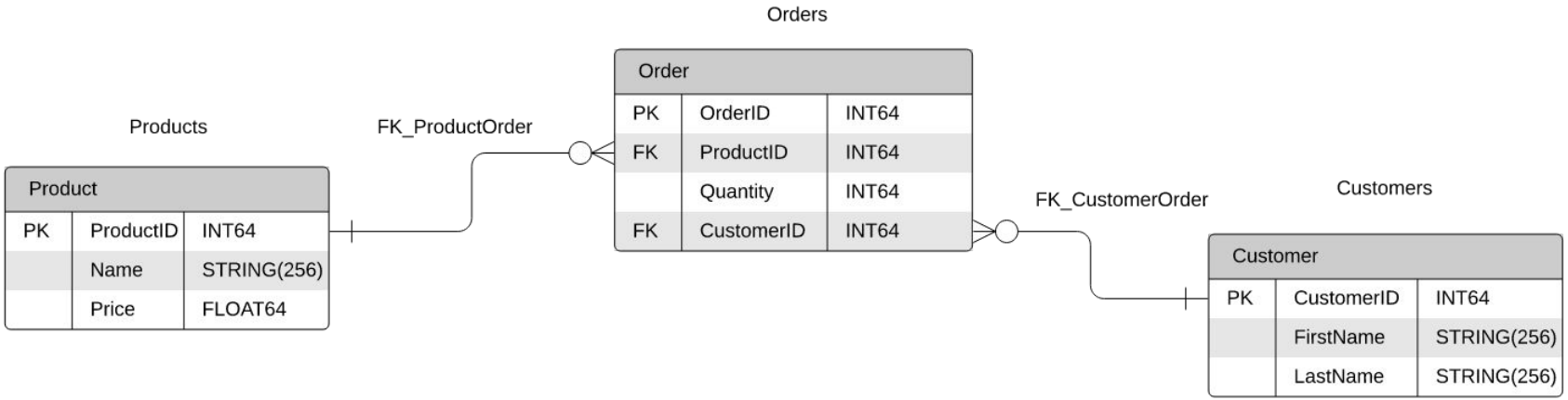
Understanding “primary keys”

They are called “keys” because they should be unique, as if “only one entity should have access to this resource (row)”

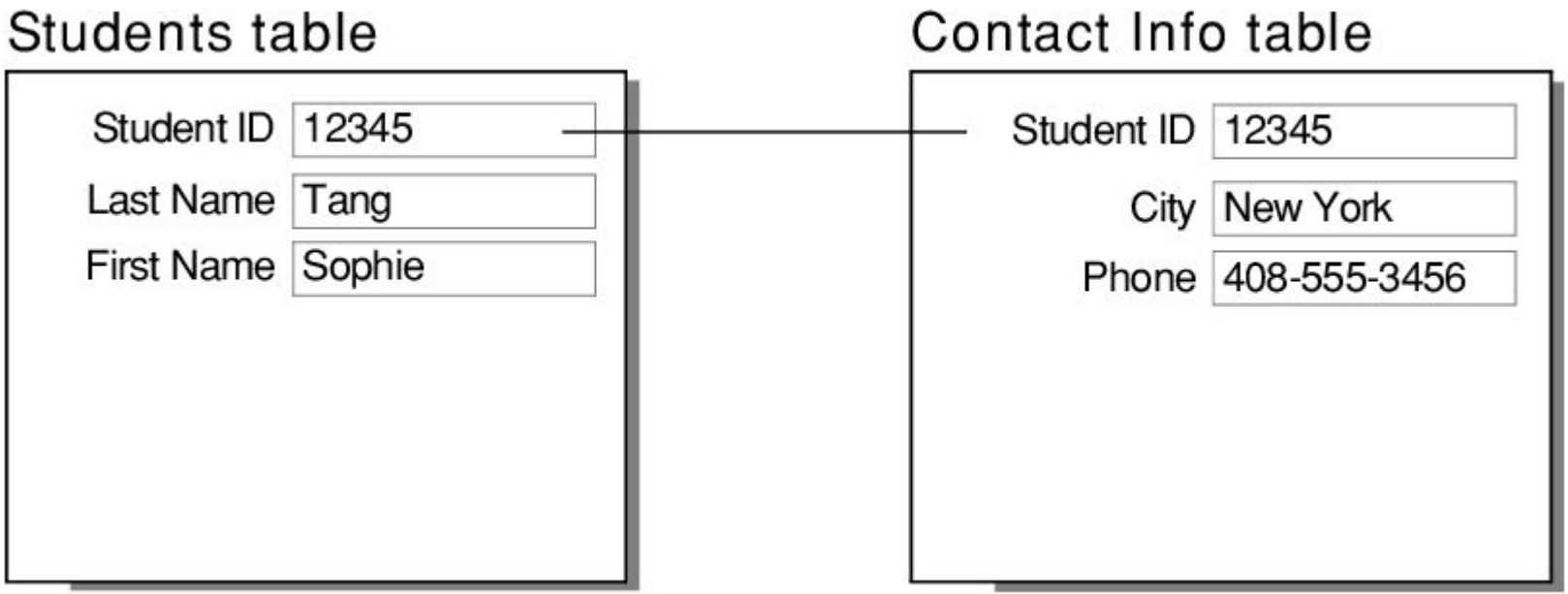
Commonly used primary keys:

- **An automatically generated Universal Unique Identifier (UUID)**
- ~~Social security / tax number.~~
- ~~An email address (but only if two users can't share the same email address)~~
- ~~Vehicle identification number.~~
- ~~Driver licence number~~
- Some other special code that **is** unique to each recod.

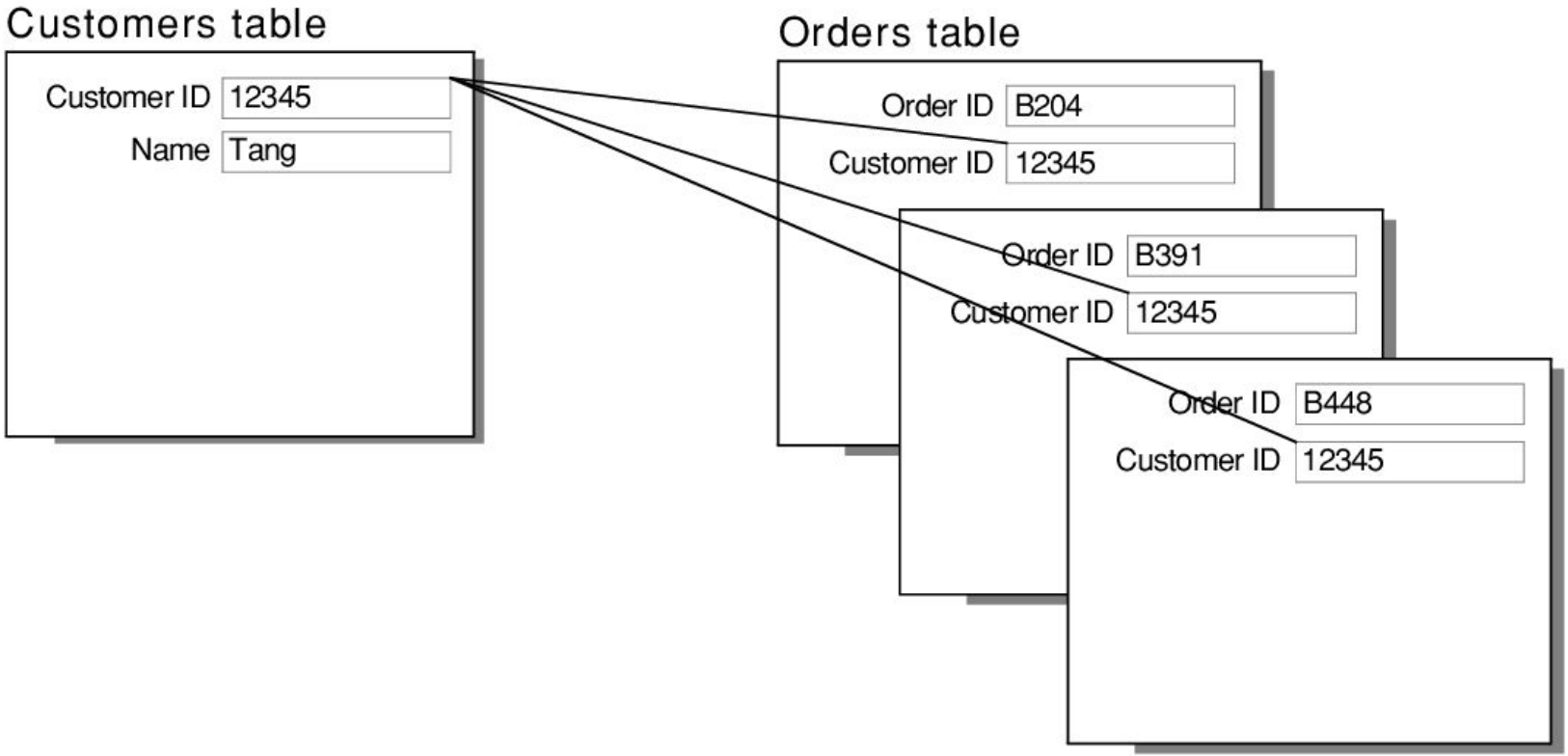
Understanding “foreign keys”



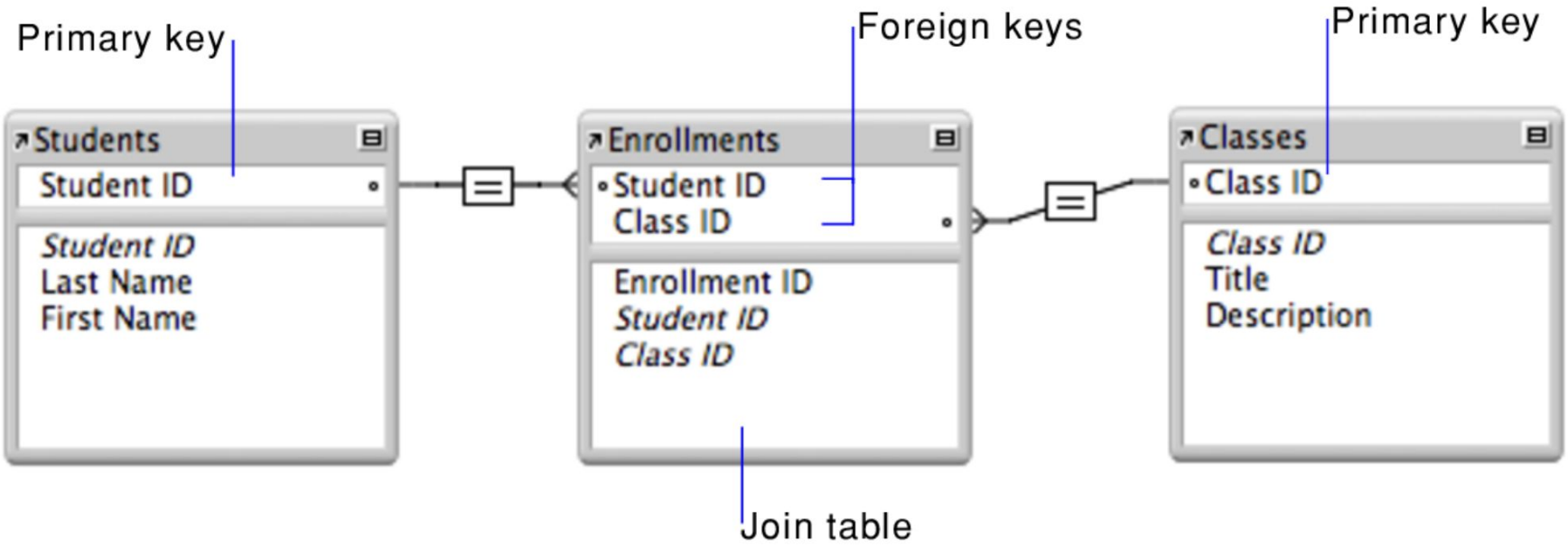
One to one relationships



One to many relationships



Many to many relationships



Challenges of learning (and doing) SQL

- Learning SQL is **boring**
  - **Possible solution: practice logic and syntax, write as many queries as possible**

- Domain knowledge about a particular business is important to make reasonable data models

- We should **not** be learning SQL as if we were database administrators
  - Normalization and index optimization are outside of the scope of our jobs as data scientists

- Minor syntax differences between SQL implementations
  - Not a deal-breaker

The usefulness and beauty of SQLBolt

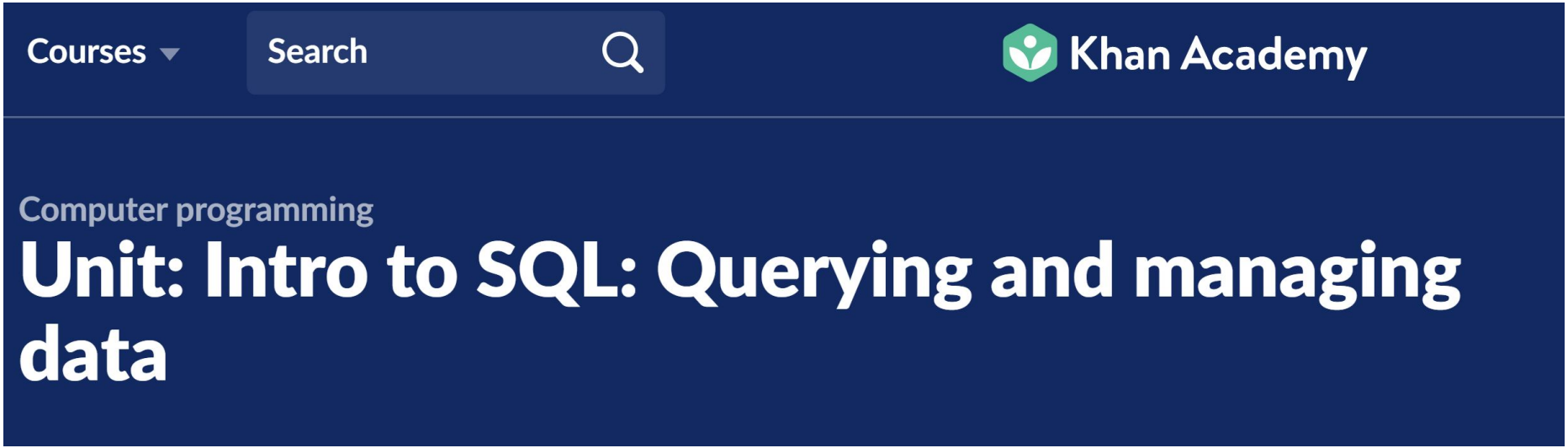
SQLBolt

Learn SQL with simple, interactive exercises.

- We won’t need to install anything
- All answers are in the website
- Syntax is the same as in MySQL, Postgres, and SQLite
- 18 interactive beginner lessons, 2 intermediate / advanced ones on subqueries and set operators

https://sqlbolt.com/

Homework suggestion: Khan Academy's SQL

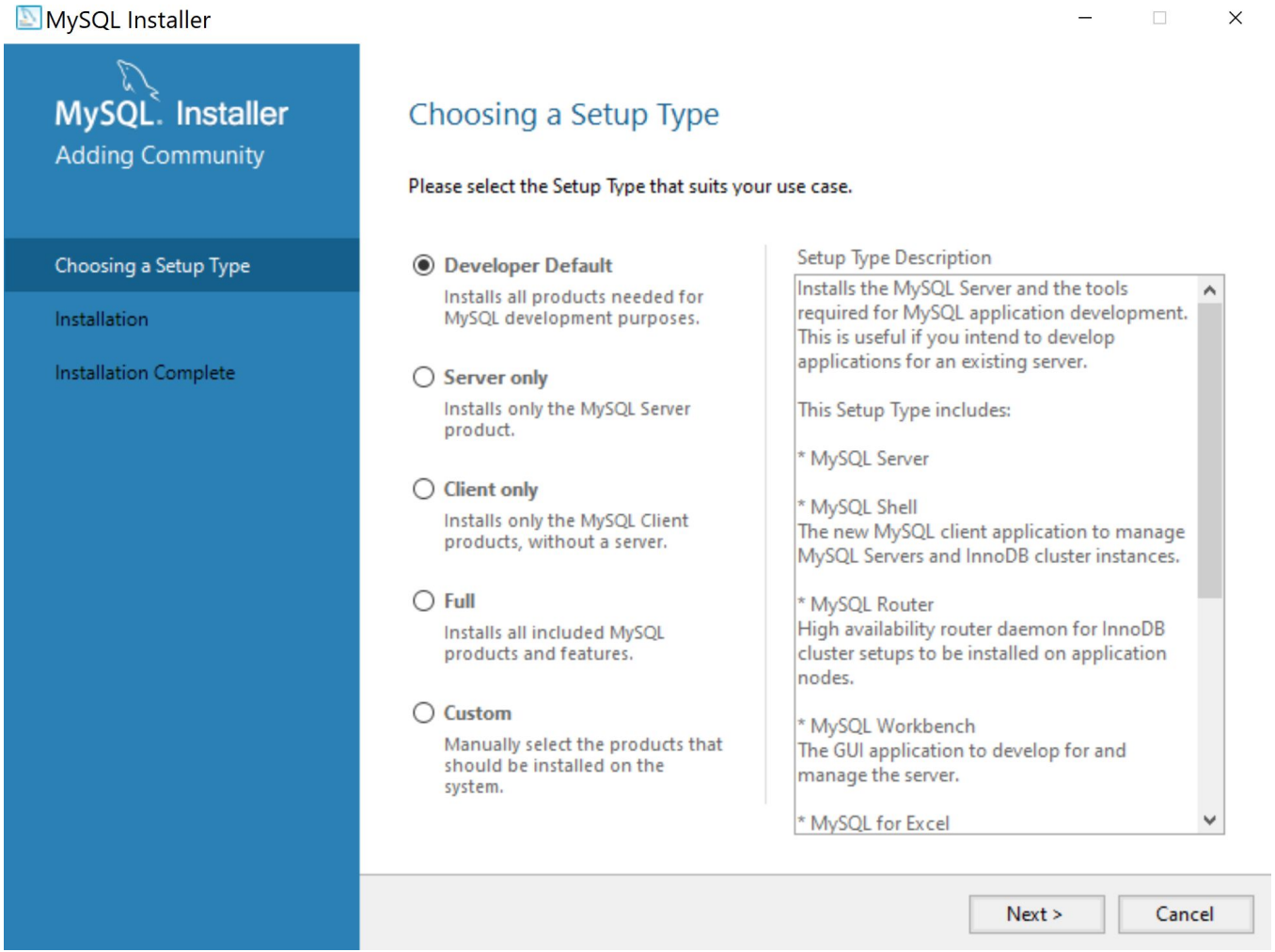


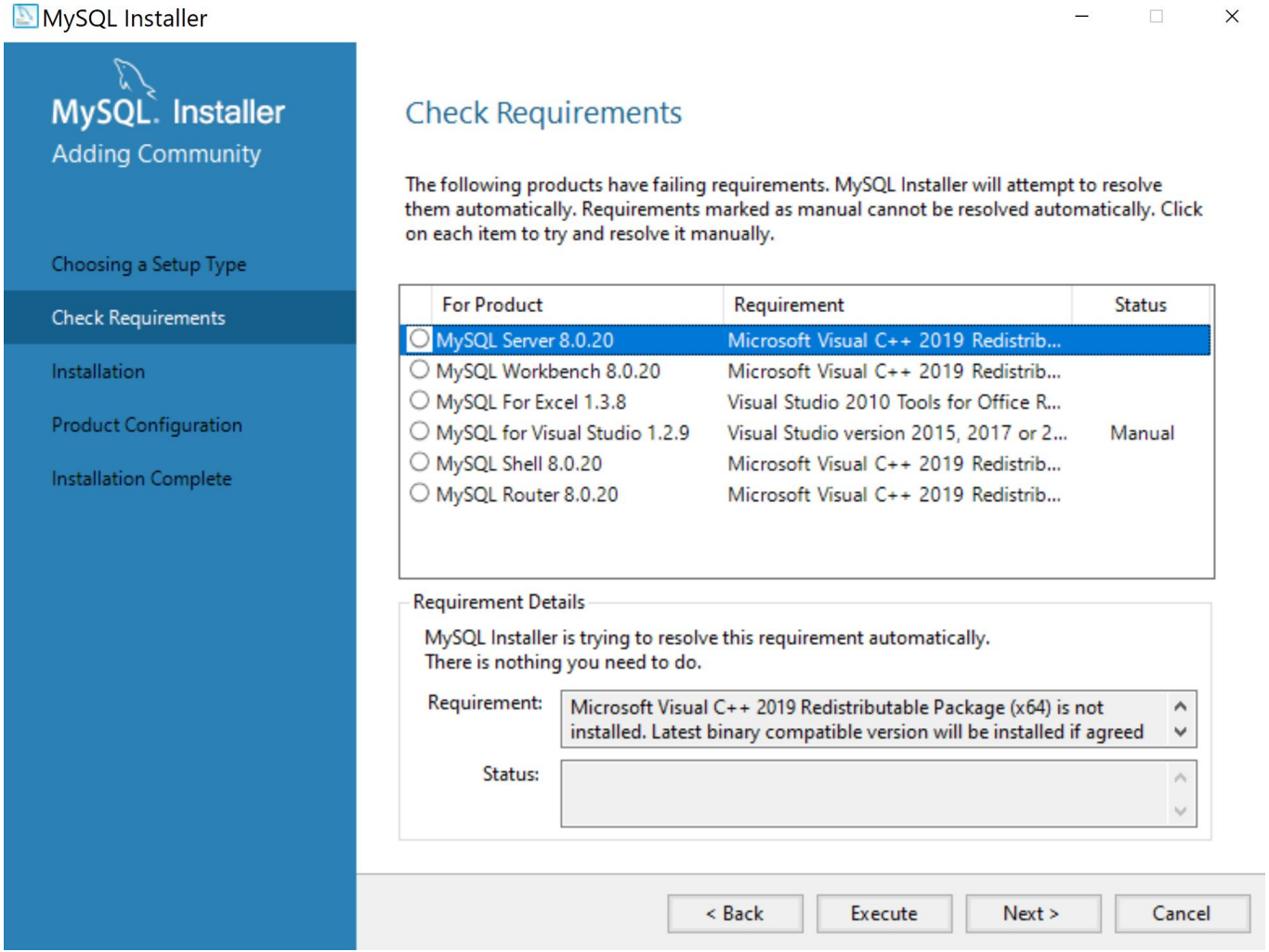
https://www.khanacademy.org/computing/computer-programming/sql

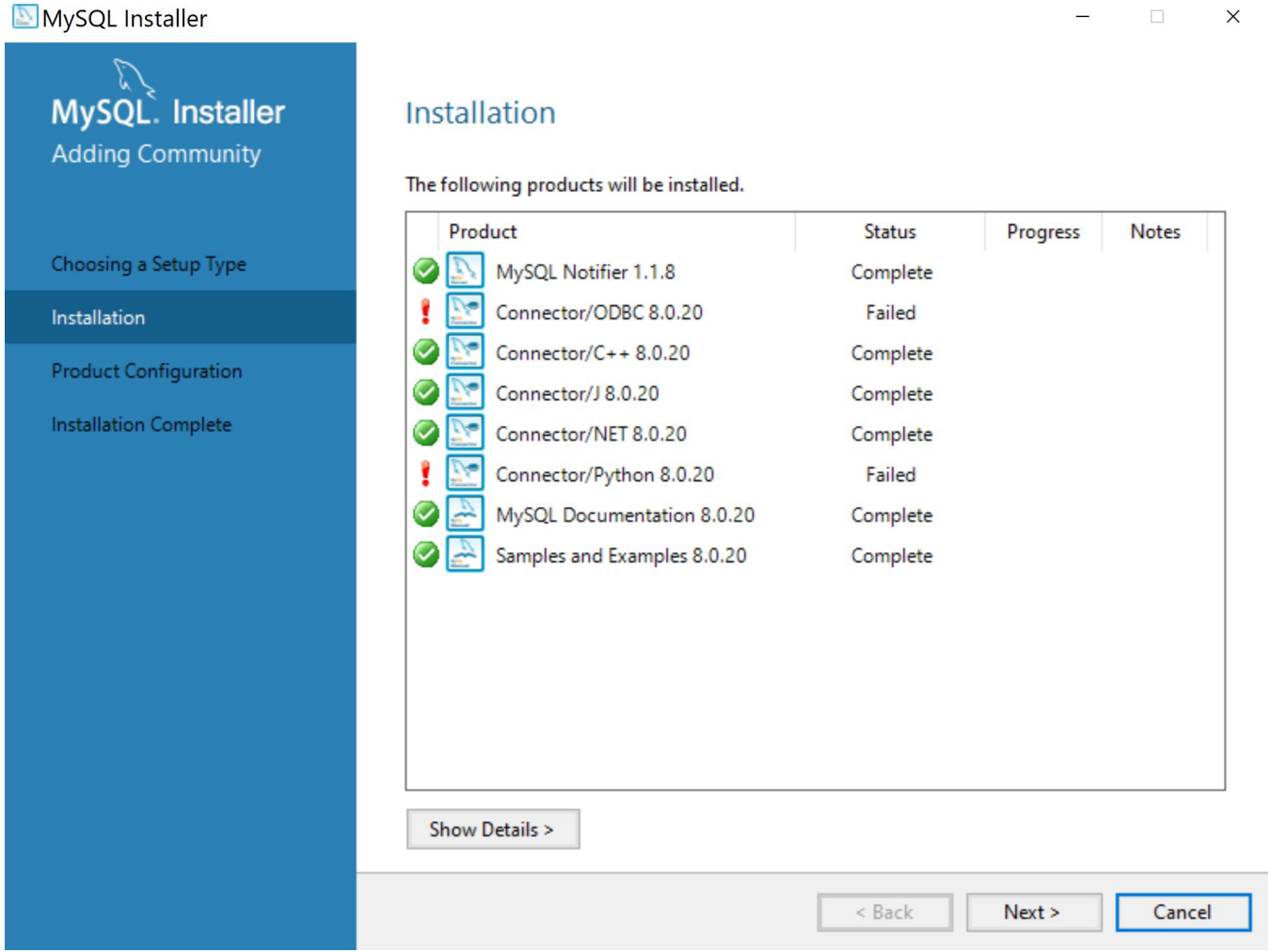
Optional: Install MySQL (Developer Settings)

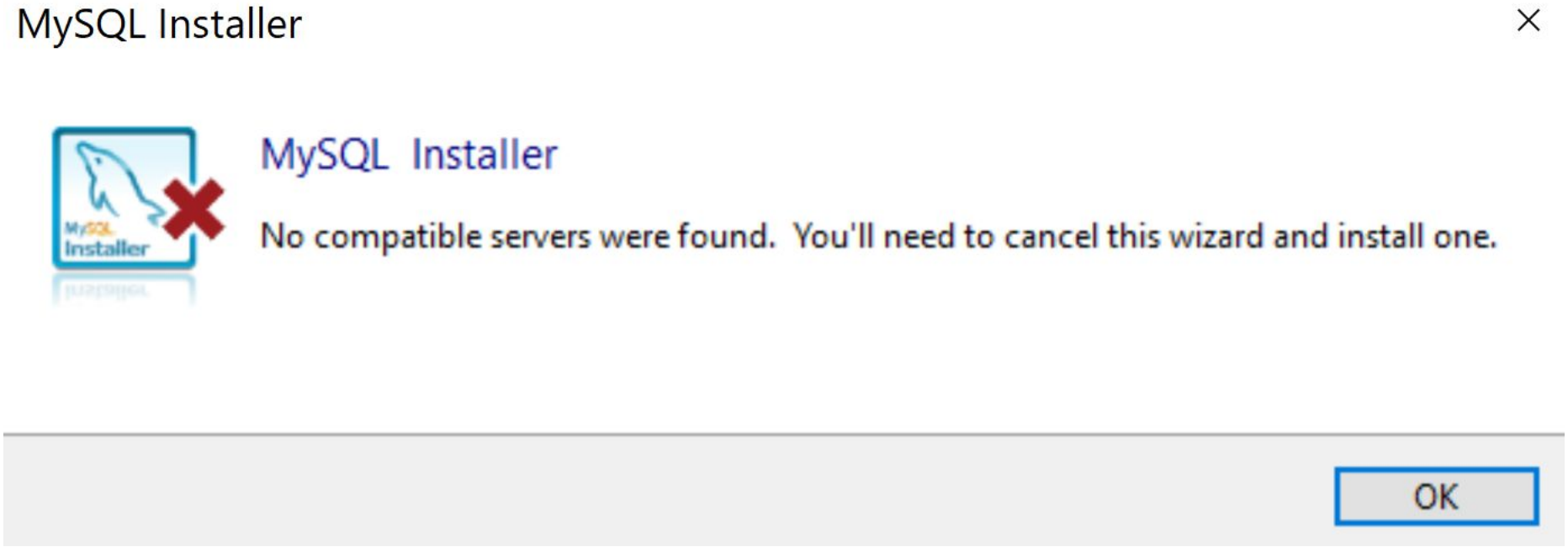
- MySQL is the ***most popular*** relational database system
  - Used by Facebook, Twitter, Youtube, Flickr
  - “LAMP” stacks use it (it’s the ‘M”)
- Now owned by Oracle (many people don’t like this)
- MariaDB is a non-corporate open source fork

https://dev.mysql.com/downloads/


MySQL Installer
MySQL. Installer
Adding Community
Choosing a Setup Type
Installation
Installation Complete
Choosing a Setup Type
Please select the Setup Type that suits your use case.
Developer Default
Installs all products needed for MySQL development purposes.
Server only
Installs only the MySQL Server product.
Client only
Installs only the MySQL Client products, without a server.
Full
Installs all included MySQL products and features.
Custom
Manually select the products that should be installed on the system.
Setup Type Description
Installs the MySQL Server and the tools required for MySQL application development. This is useful if you intend to develop applications for an existing server.
This Setup Type includes:
* MySQL Server
* MySQL Shell
The new MySQL client application to manage MySQL Servers and InnoDB cluster instances.
* MySQL Router
High availability router daemon for InnoDB cluster setups to be installed on application nodes.
* MySQL Workbench
The GUI application to develop for and manage the server.
* MySQL for Excel
Next >
Cancel


MySQL Installer
MySQL. Installer
Adding Community
Choosing a Setup Type
Check Requirements
Installation
Product Configuration
Installation Complete
Check Requirements
The following products have failing requirements. MySQL Installer will attempt to resolve them automatically. Requirements marked as manual cannot be resolved automatically. Click on each item to try and resolve it manually.
For Product
Requirement
Status
MySQL Server 8.0.20
Microsoft Visual C++ 2019 Redistrib...
MySQL Workbench 8.0.20
Microsoft Visual C++ 2019 Redistrib...
MySQL For Excel 1.3.8
Visual Studio 2010 Tools for Office R...
MySQL for Visual Studio 1.2.9
Visual Studio version 2015, 2017 or 2...
Manual
MySQL Shell 8.0.20
Microsoft Visual C++ 2019 Redistrib...
MySQL Router 8.0.20
Microsoft Visual C++ 2019 Redistrib...
Requirement Details
MySQL Installer is trying to resolve this requirement automatically.
There is nothing you need to do.
Requirement:
Microsoft Visual C++ 2019 Redistributable Package (x64) is not installed. Latest binary compatible version will be installed if agreed
Status:
< Back
Execute
Next >
Cancel

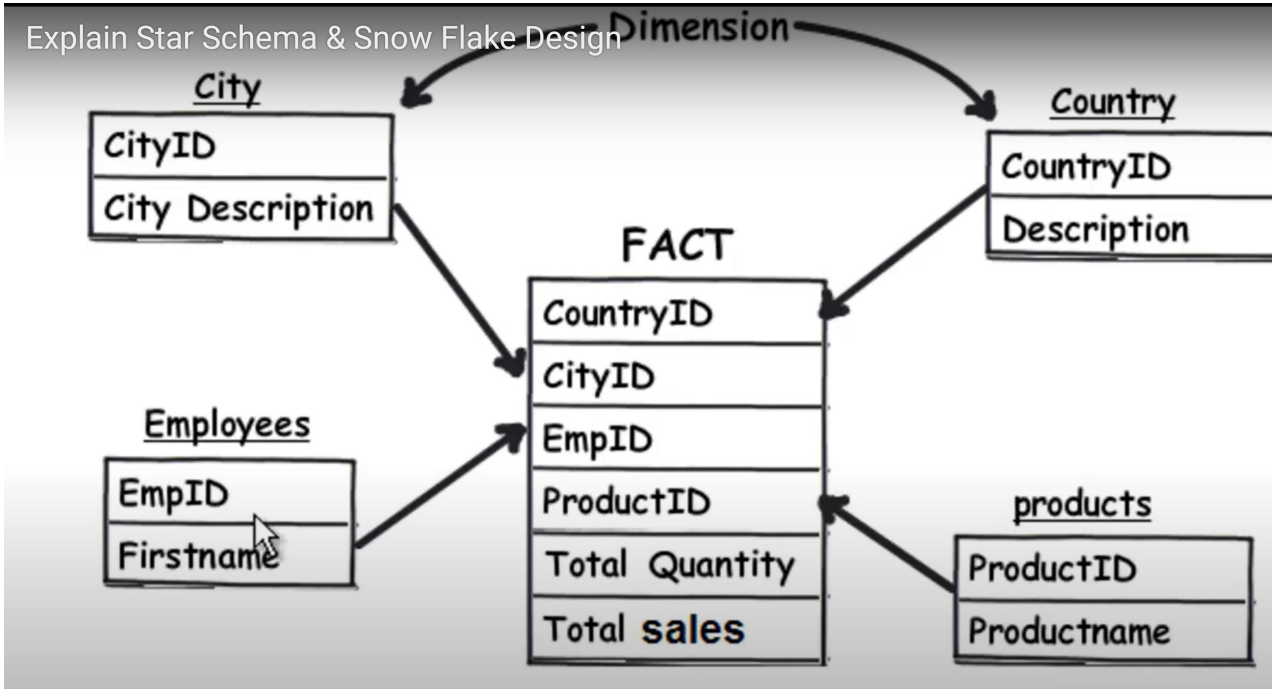

MySQL Installer
MySQL. Installer
Adding Community
Choosing a Setup Type
Installation
Product Configuration
Installation Complete
Installation
The following products will be installed.
Product
Status
Progress
Notes
MySQL Notifier 1.1.8
Complete
Connector/ODBC 8.0.20
Failed
Connector/C++ 8.0.20
Complete
Connector/J 8.0.20
Complete
Connector/NET 8.0.20
Complete
Connector/Python 8.0.20
Failed
MySQL Documentation 8.0.20
Complete
Samples and Examples 8.0.20
Complete
Show Details >
< Back
Next >
Cancel


MySQL Installer
MySQL Installer
No compatible servers were found. You'll need to cancel this wizard and install one.
OK

What is normalization?

- Think of it as **“database cleanup”**
- Normalization **reduces duplicated/redundant/mismatched data**
- Usually **a job for database administrators**, not data scientists
- Curious? https://en.wikipedia.org/wiki/Database_normalization

Recommended viewing: star vs snowflake schema



https://www.youtube.com/watch?v=KUwOcip7Zzc

ACID Principles

- Atomicity
- Consistency
- Isolation
- Durability

https://database.guide/what-is-acid-in-databases/#:~:text=In%20database%20systems%2C%20ACID%20(Atomicity,occur%20while%20processing%20a%20transaction.

On different SQL implementations

‘Standard SQL’ syntax has about 90% overlap between different implementations:

- MySQL/MariaDB
- PostgreSQL
- Oracle
- Microsoft’s SQL Server
- IBM’s Dbase

SQL syntax differences between implementations

Operators	Works in
Arithmetic operators	All databases
Assignment operators	All databases
Bitwise operators	Microsoft SQL Server
Comparison operators	All databases
Logical operators	DB2, Oracle, SQL Server, and PostgreSQL
Unary operators	DB2, Oracle, and SQL Server

Example from https://www.w3resource.com/sql/sql-syntax.php

SQL vs NoSQL

- NoSQL databases are (e.g. MongoDB, DynamoDB, Cassandra) are **write-optimized**
  - Easily mutable
  - Reduced need to update/maintain a schema
- SQL is **read-optimized**, good for querying and understanding data relationships
  - Easily queryable ***for a data analyst***
  - Needs maintenance of a schema

A nice video on SQL vs NoSQL

** End users won't be able to tell the difference between these systems. Only backend engineers and data people care about this*

SQL vs NoSQL

SQL: uses ‘structured’ data (aka tables)

Strengths of SQL

- Read optimized
- Easy querying

NoSQL: uses ‘non-structured’ data (aka not tables, things like JSON)

Strengths of NoSQL

- Write optimized
- Parallel concurrent writing

W3Schools: a doubleplusgood SQL syntax guide

w3schools.com

SQL Tutorial

SQL HOME
SQL Intro
SQL Syntax
SQL Select
SQL Select Distinct
SQL Where
SQL And, Or, Not
SQL Order By
SQL Insert Into
SQL Null Values
SQL Update
SQL Delete
SQL Select Top
SQL Min and Max
SQL Count, Avg, Sum
SQL Like

https://www.w3schools.com/sql/

Going through SQLBolt

- **Rules/order of this honor game**

    1. We read the theory page of the lesson together
    2. Each one of us attempts to write each query individually. If it takes more than 2 minutes, we ask a peer
    3. If two of us are amiss, we look at the solution
    4. We send a Slack message saying that we are done with the lesson
    5. Ask as many questions as you like, but first *try doing the exercise*

How does a SQL query look like? SQLBolt lesson 1

Table: Movies

Id	Title	Director	Year	Length_minutes
1	Toy Story	John Lasseter	1995	81
2	A Bug's Life	John Lasseter	1998	95
3	Toy Story 2	John Lasseter	1999	93
4	Monsters, Inc.	Pete Docter	2001	92
5	Finding Nemo	Andrew Stanton	2003	107
6	The Incredibles	Brad Bird	2004	116
7	Cars	John Lasseter	2006	117
8	Ratatouille	Brad Bird	2007	115
9	WALL-E	Andrew Stanton	2008	104
10	Up	Pete Docter	2009	101

The query to get everything in a table

```
SELECT *
FROM mytable;
```

This query retrieves everything - no restrictions

Selecting columns - SQLBolt lesson 1

```
SELECT column, another_column
FROM mytable;
```

A query is a selection of data. In SQL, anything that starts with a SELECT statement is a query.

Table: Movies

Title	Director
Toy Story	John Lasseter
A Bug's Life	John Lasseter
Toy Story 2	John Lasseter
Monsters, Inc.	Pete Docter
Finding Nemo	Andrew Stanton
The Incredibles	Brad Bird
Cars	John Lasseter
Ratatouille	Brad Bird
WALL-E	Andrew Stanton
Up	Pete Docter

```
SELECT title,director FROM movies;
```

Truth tables for AND, OR, and NOT

A	B	A AND B	A OR B	NOT A
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

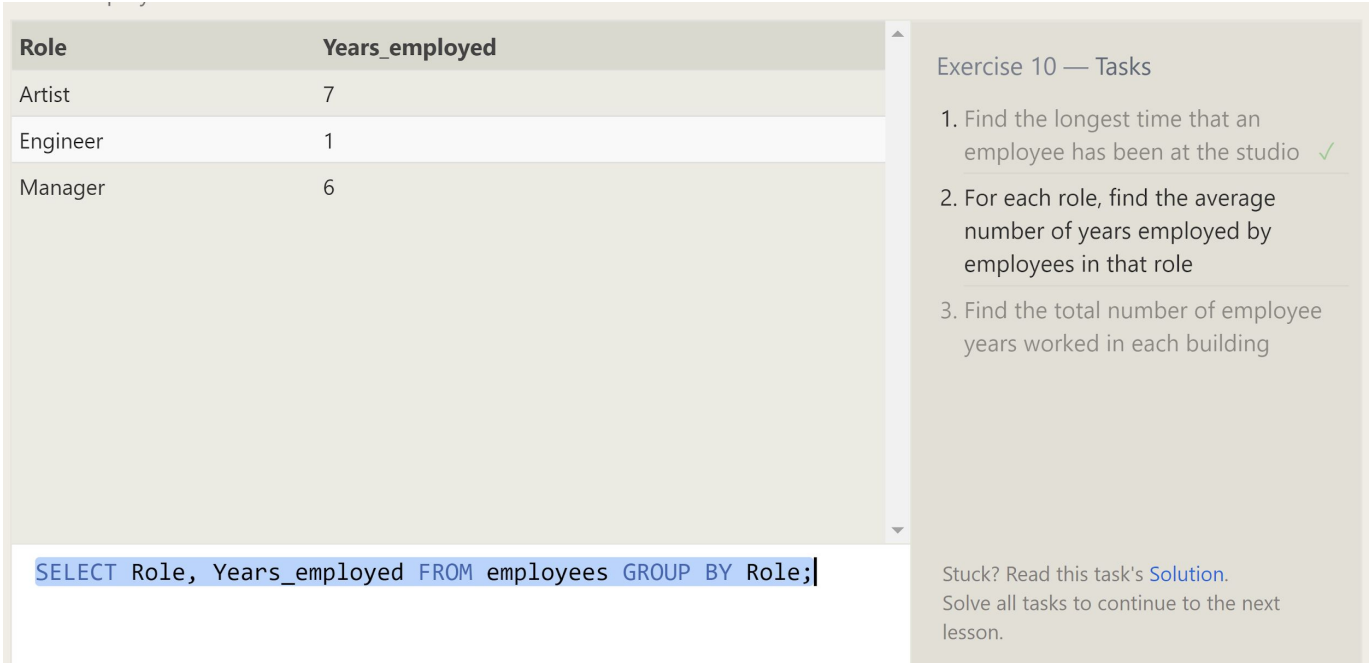
Queries with constraints

Operator	Condition	SQL Example
=, !=, < <=, >, >=	Standard numerical operators	col_name **!=** 4
BETWEEN ... AND ...	Number is within range of two values (inclusive)	col_name **BETWEEN** 1.5 **AND** 10.5
NOT BETWEEN ... AND ...	Number is not within range of two values (inclusive)	col_name **NOT BETWEEN** 1 **AND** 10
IN (...)	Number exists in a list	col_name **IN** (2, 4, 6)
NOT IN (...)	Number does not exist in a list	col_name **NOT IN** (1, 3, 5)

Working with aggregates

Function	Description
COUNT(***), **COUNT(*column***)**	A common function used to counts the number of rows in the group if no column name is specified. Otherwise, count the number of rows in the group with non-NULL values in the specified column.
MIN(*column***)**	Finds the smallest numerical value in the specified column for all rows in the group.
MAX(*column***)**	Finds the largest numerical value in the specified column for all rows in the group.
AVG(*column*)	Finds the average numerical value in the specified column for all rows in the group.
SUM(*column***)**	Finds the sum of all numerical values in the specified column for the rows in the group.

Working with aggregates



Role	Years_employed
Artist	7
Engineer	1
Manager	6

```
SELECT Role, Years_employed FROM employees GROUP BY Role;
```

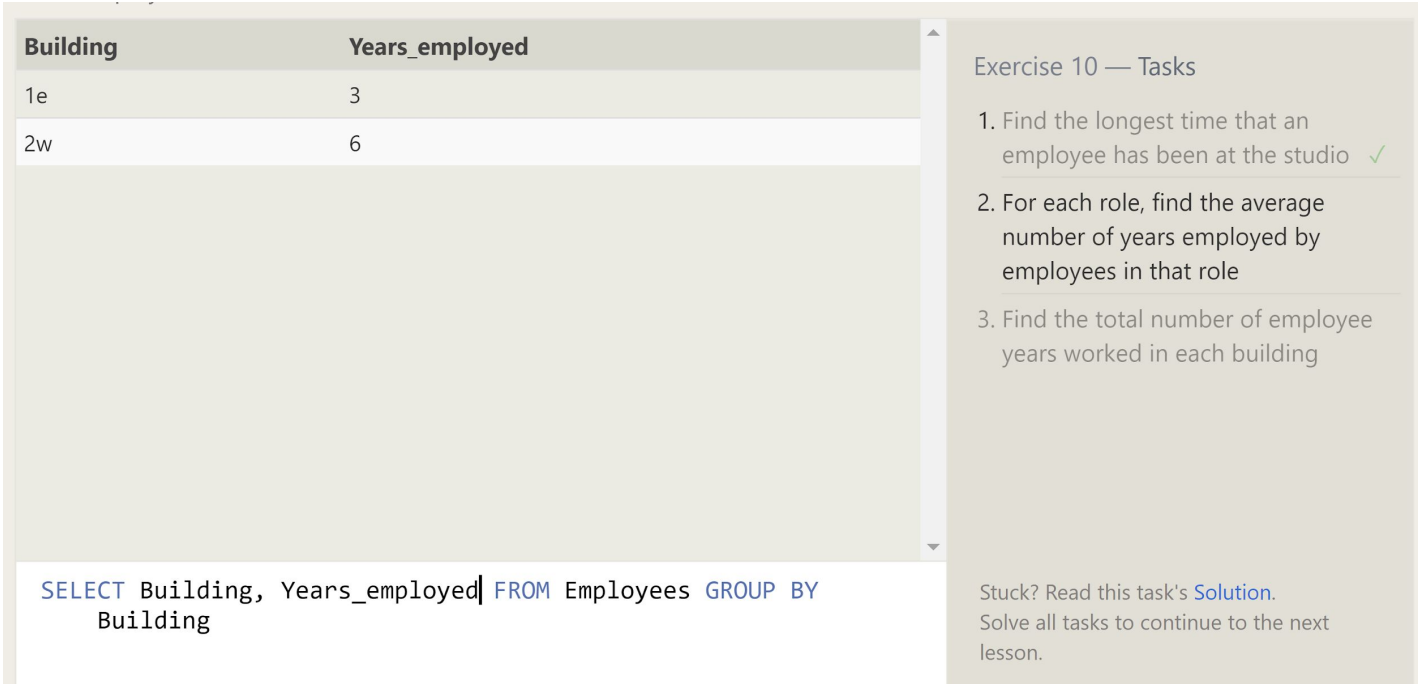
Exercise 10 — Tasks

1. Find the longest time that an employee has been at the studio ✓
2. For each role, find the average number of years employed by employees in that role
3. Find the total number of employee years worked in each building

Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

What's missing in this query?

Working with aggregates



Building	Years_employed
1e	3
2w	6

```
SELECT Building, Years_employed FROM Employees GROUP BY
    Building
```

Exercise 10 — Tasks

1. Find the longest time that an employee has been at the studio ✓
2. For each role, find the average number of years employed by employees in that role
3. Find the total number of employee years worked in each building

Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

What's missing in this query?

Using DISTINCT on aggregate functions

```
SELECT COUNT(DISTINCT Customer_Name)

FROM Customers
```

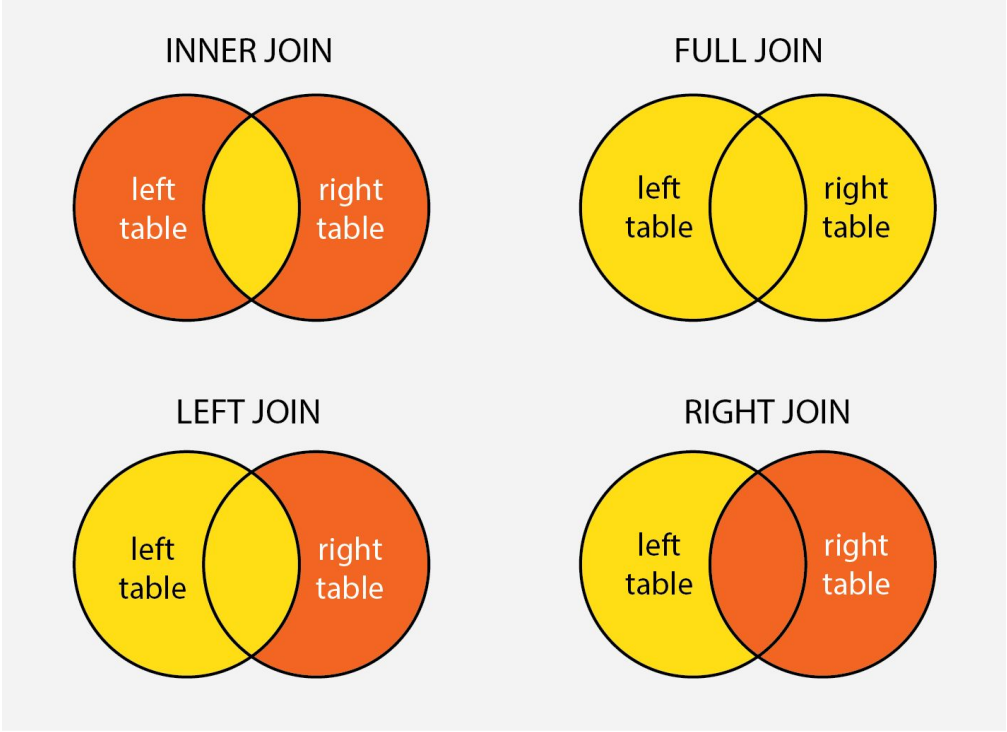
If **DISTINCT is not specified, ALL is assumed*

HAVING as a constraint for aggregates

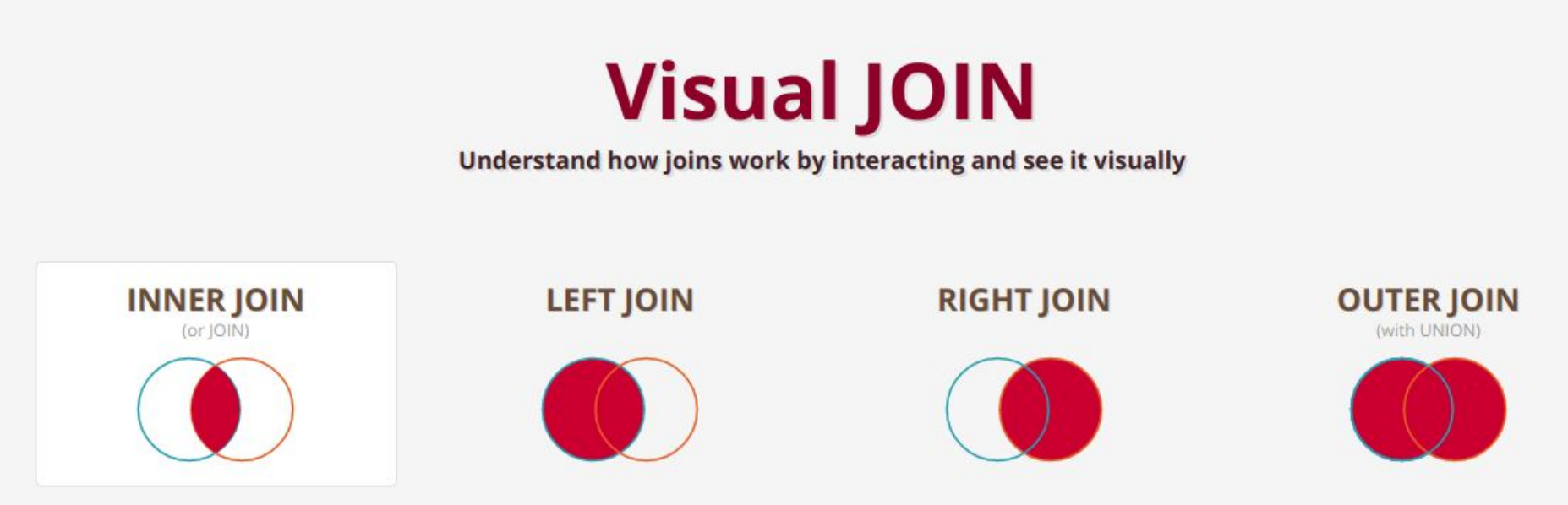
```
SELECT group_by_column, AGG_FUNC(column_expression) AS aggregate
FROM mytable
WHERE condition
GROUP BY column
HAVING group_condition;
```

After filtering with WHERE, we can apply a second filter using HAVING

Types of Joins



Types of Joins



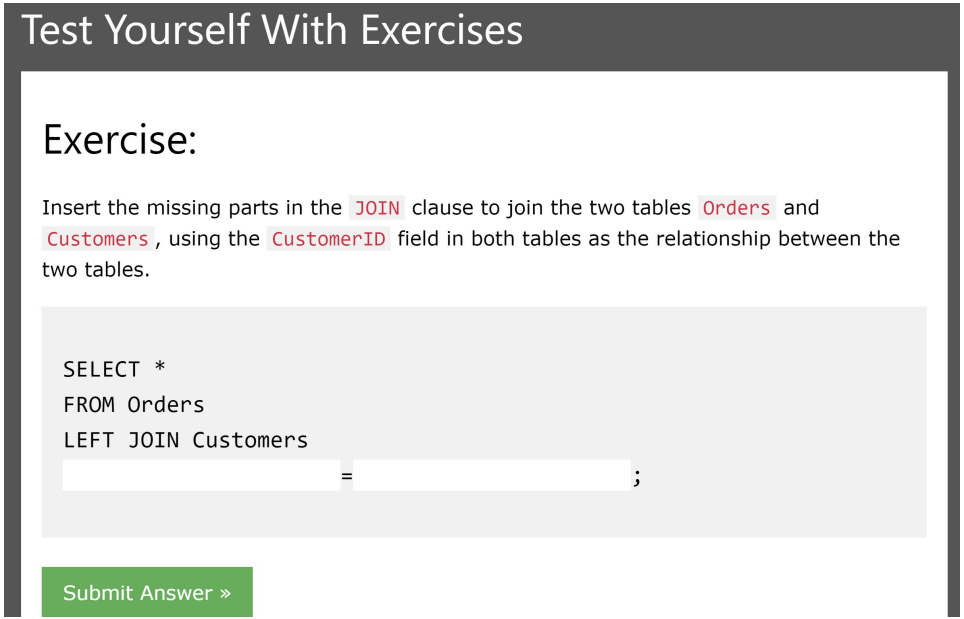
https://joins.spathon.com/

Leetcode - left join example

https://leetcode.com/problems/combine-two-tables/

```
Solution (Highlight text to see):
```

Homework for practicing SQL Joins: W3Schools



https://www.w3schools.com/sql/sql_join.asp

(Basic) SQL Review Quiz

Question 1 of 25:

What does SQL stand for?

- Structured Question Language
- Structured Query Language
- Strong Question Language

Next ›

0

https://www.w3schools.com/quiztest/quiztest.asp?qtest=SQL

Style suggestions: making your SQL readable

- Names must begin with a letter and may not end with an underscore
- Never give a table the same name as one of its columns
- Avoid plurals—use the more natural collective term where possible instead. For example `staff` instead of `employees` or `people` instead of `individuals`.
- Avoid abbreviations and if you have to use them make sure they are commonly understood

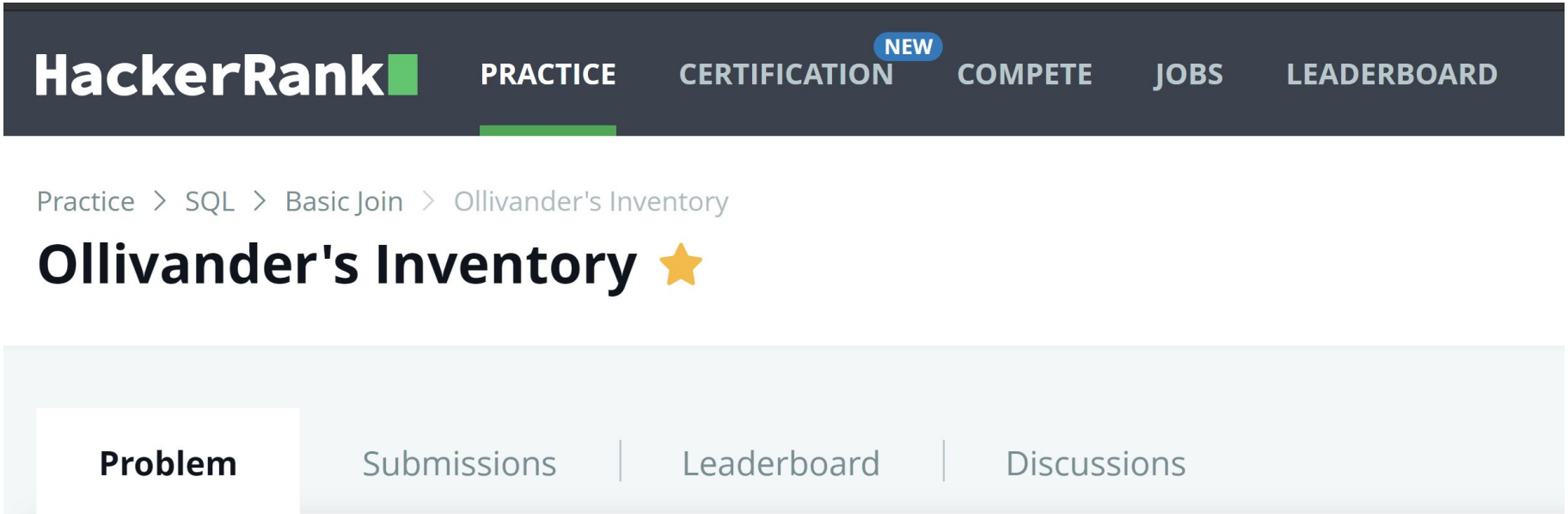
https://www.sqlstyle.guide/

Subqueries

Subqueries are *queries within queries (can easily become unreadable)*

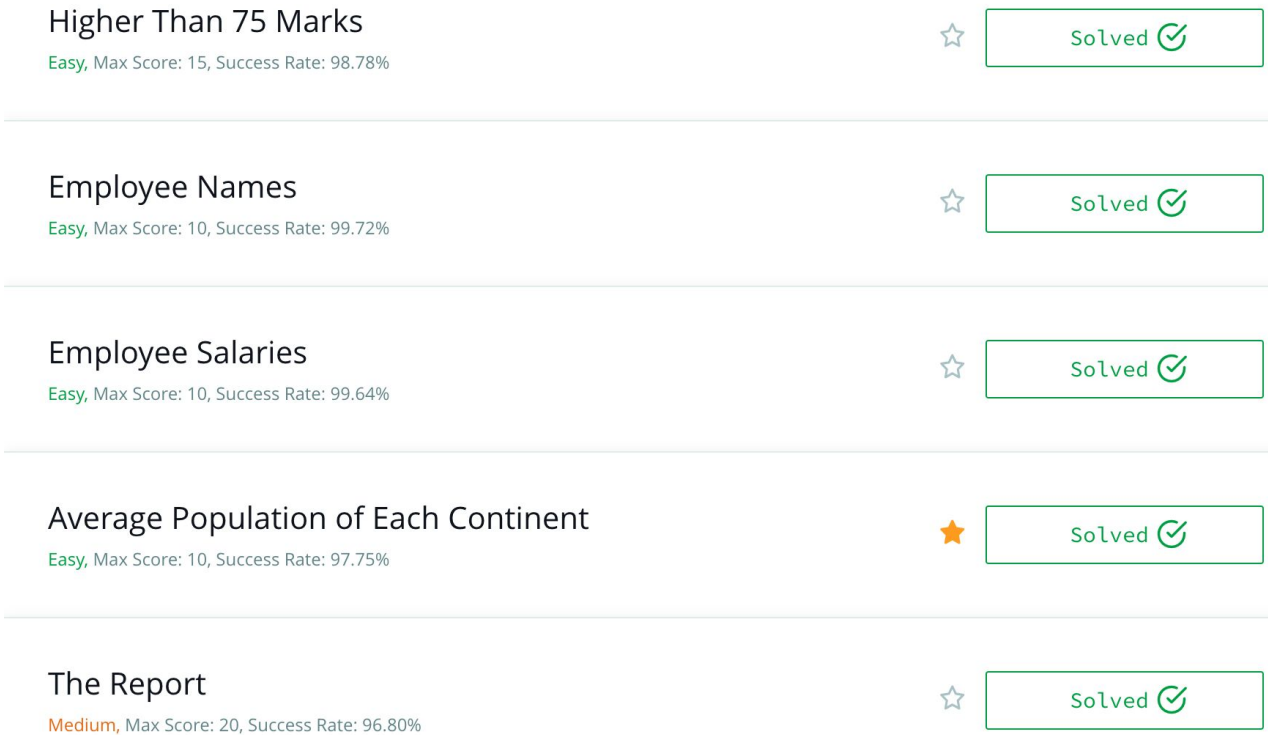
```
SELECT *
FROM sales_associates
WHERE salary >
    (SELECT AVG(revenue_generated)
     FROM sales_associates);
```

Subqueries



https://www.hackerrank.com/challenges/harry-potter-and-wands/problem

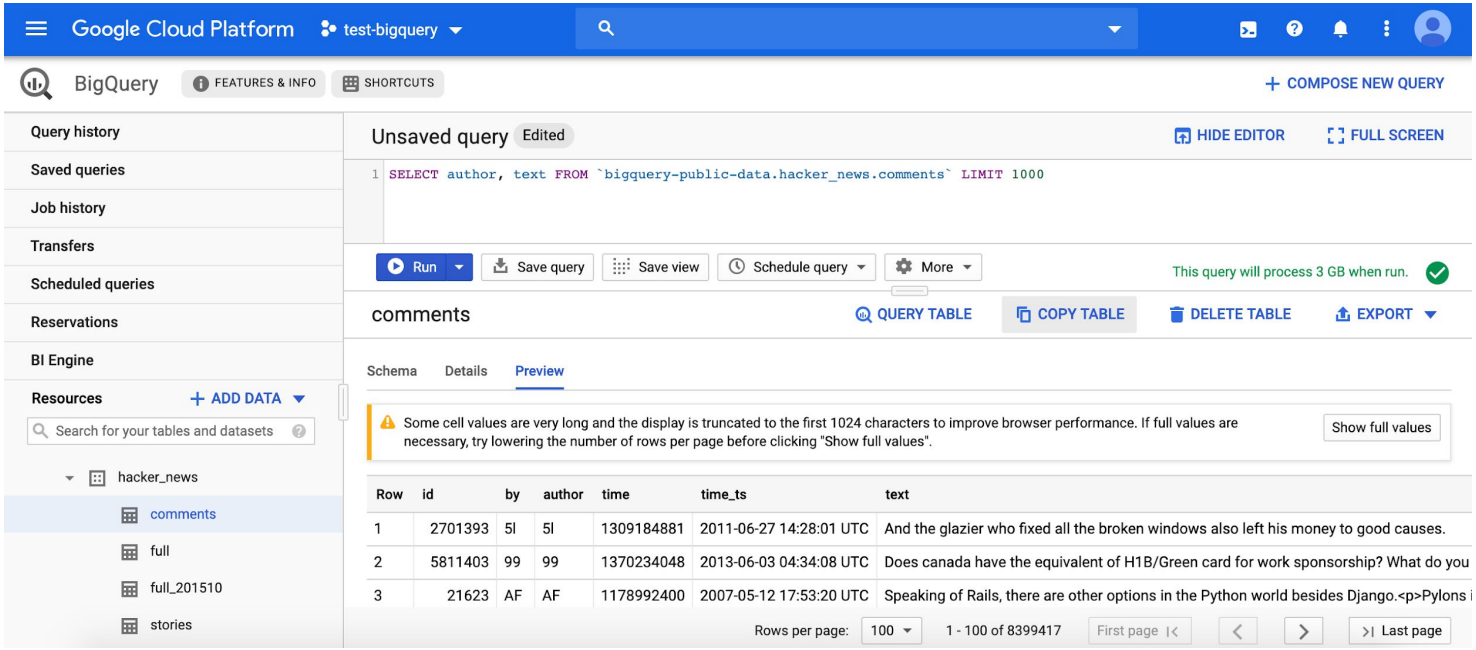
SQL Syntax - Exercises on HackerRank



What is Google Bigquery?

- Uses standard SQL syntax
- Runs queries on petabytes of data in seconds
- No need to create or manage indices to speed up queries
- No need to maintain infrastructure

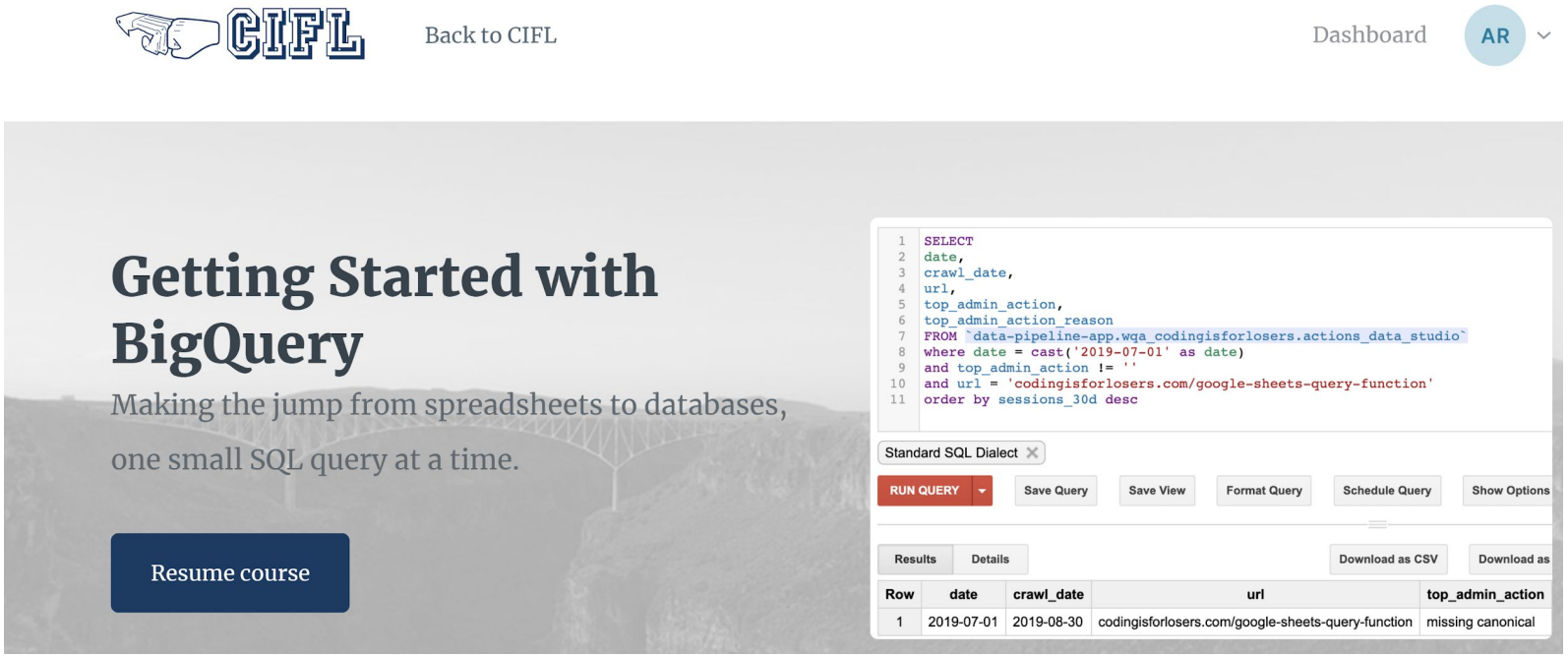
BigQuery sandbox



[Using the BigQuery sandbox]

[Query without a credit card: introducing BigQuery sandbox]

Suggested homework: Google's BigQuery



https://learn.codingisforlosers.com/learn-bigquery-sql

References and further reading

- SQLBolt
  - https://sqlbolt.com/
- W3School
  - https://www.w3schools.com/sql/
- SQL for Data Science - Coursera
  - https://www.coursera.org/learn/sql-for-data-science
- Khan Academy
  - https://www.khanacademy.org/computing/computer-programming/sql
- BigQuery Tutorial - CIFL
  - https://learn.codingisforlosers.com/learn-bigquery-sql
- Understanding database normalization
  - https://medium.com/cracking-the-data-science-interview/an-introduction-to-big-data-data-normalization-b72311f134b7