



Introduction to Basic SQL

Section Leader: Will Calandra

09.17.24

PART 01

Data lives in a database

Assumed “status quo”

- Let's just use .csv files for everything, store on our file system
 - Pros: easy to use, data will stay the same, tree structure
 - Cons: no forced data types, difficult to read/write concurrently, typically one 2D table at a time
 - Any other problems?
- Doesn't scale. What if we have big data? Complex relationships? Work together?
- We need **databases** -> *relational databases* (RDBMS)

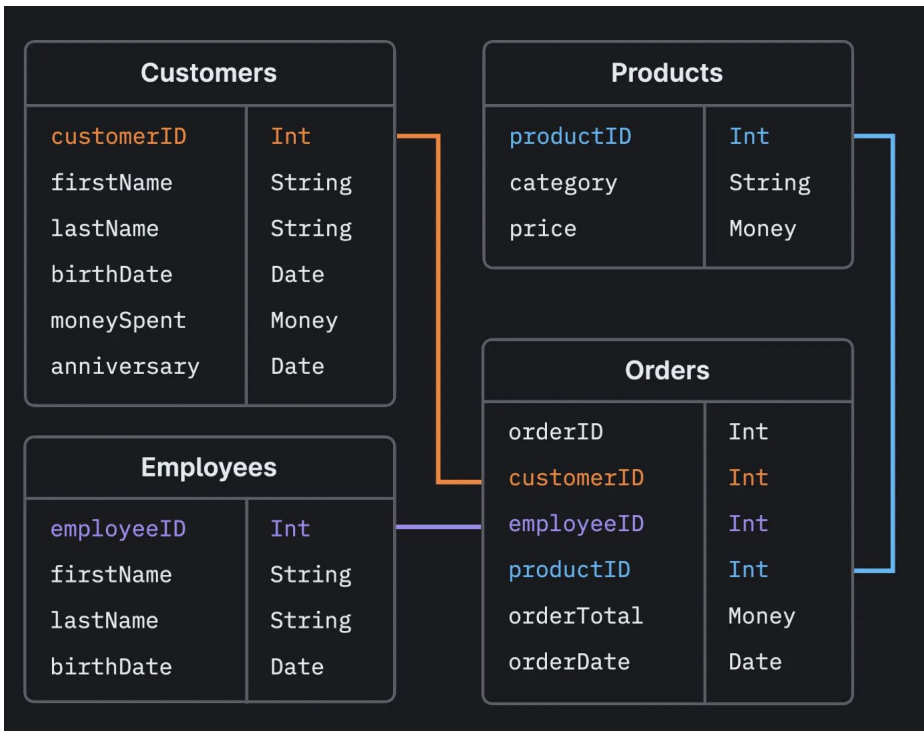
The industry way

- “Data lives in a database”
- Databases ensure a few things, if done well...
 - **Atomicity:** operations are all-or-nothing (something breaks, it won't update!)
 - **Consistency:** db is always in a valid state (no corruption)
 - **Isolation:** concurrent operations do not depend on order of execution (locking, independence)
 - **Durability:** completed transactions are permanent (version controls, recover valid states)
- We'll work with *relational databases*
 - **Tables:** 2D, rows and columns, one entry in each cell
 - **Rows:** represent a unique record/tuple from a relation
 - **Columns:** represent a field of the record/attribute from a relation

Constraints

- Big idea: induce some constraints on our data to make our programming lives easier
 - Don't have to explicitly program/check things that db will automate
- We constrain our data through a **schema**
 - Gives us valid rows of data that must fit the schema, or else reject. Helps with data typing and expected inputs
 - Excel fans: “1”, random date casting
- We search data using **keys**
 - **Primary key**: unique id for each row in a table
 - **Foreign key**: column that links to a primary key in another table

Schema

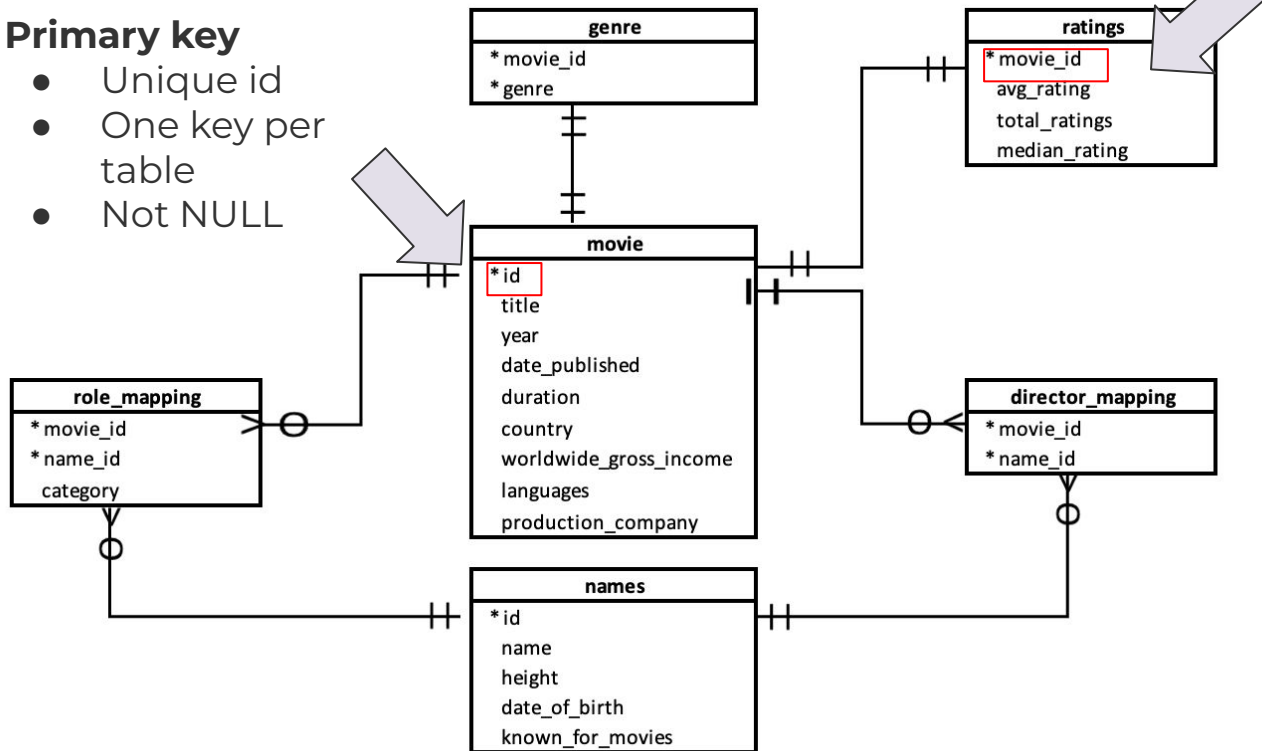


Keys

Data source for today

Primary key

- Unique id
- One key per table
- Not NULL



Foreign key

- Not unique
- Can have multiple in a table
- Can be NULL

PART 01

What's the big deal with SQL?

SQL (Structured Query Language)

- **SQL** is how we are able to access and manipulate relational databases
 - **Declarative:** we tell it what to do, not how to do it (not instruction-based like Python or C++)
- We'll see it in a Pythonic way, and we'll use **SQLite**
 - Integrate it into a pipeline, SQLite comes with Python
 - Serverless, don't have to configure/maintain it, but stored locally
 - US Library of Congress trusts it for their digitization. Would you?
- There exist many flavors of SQL (MySQL, PostgreSQL, etc.), but the syntax is mostly the same
- **Industry standard:** not going anywhere!

PART 02

Our host: Spyder IDE

Spyder IDE

- [Installation \(try with Anaconda\)](#)
- Gemini: 2.7-6.1% of population has arachnophobia. Perhaps you can improve marketing
- **IDE** (Integrated Development Environment) made for data science. “Written in Python, for Python, designed by and for scientists, engineers, and data analysts”
 - A way for developers to code in one GUI (interface)
- Choice of IDE is personal preference
 - Explore extensions (highlighting, dare I say AI)
 - This won't solve your problems: Microsoft Word anyone?
- **Spyder**: balance of software/scientific scripting
 - Try to stray away from Jupyter notebooks

Spyder IDE

The screenshot displays the Spyder IDE interface with the following components:

- Editor:** Contains a Python script named `sample.py` with the following code:


```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 # from https://www.geeksforgeeks.org/graph-plotting-in-python-set-1/
4 import matplotlib.pyplot as plt
5
6 # x-coordinates of left sides of bars
7 left = [1, 2, 3, 4, 5]
8
9 # heights of bars
10 height = [10, 24, 36, 40, 5]
11
12 # labels for bars
13 tick_label = ['one', 'two', 'three', 'four', 'five']
14
15 # plotting a bar chart
16 plt.bar(left, height, tick_label = tick_label,
17        width = 0.8, color = ['red', 'green'])
18
19 # naming the x-axis
20 plt.xlabel('x - axis')
21 # naming the y-axis
22 plt.ylabel('y - axis')
23 # plot title
24 plt.title('My bar chart!')
25
26 # function to show the plot
27 plt.show()
28
```
- Variable Explorer:** Displays the variables defined in the script:

Name	Type	Size	Value
height	list	5	[10, 24, 36, 40, 5]
left	list	5	[1, 2, 3, 4, 5]
tick_label	list	5	['one', 'two', 'three', 'four', 'five']
- Console:** Shows the execution of the script:


```
Python 3.11.9 | packaged by conda-forge | (main, Apr 19 2024, 18:34:54) [Clang 16.0.6 ]
Type "copyright", "credits" or "license()" for more information.
IPython 8.27.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: %runfile '/Users/wilcalandra/Documents/Teaching/IDS SQL Lab/sample.py' --wdir

Important
Figures are displayed in the Plots pane by default. To make them also appear inline in the console, you
need to uncheck "Mute inline plotting" under the options menu of Plots.

In [2]:
```

The status bar at the bottom indicates: `Inline Conda: spyder-runtime (Python 3.11.9) LSP: Python Line 6, Col 39 UTF-8 LF RW Mem 84%`.

Spyder IDE



The screenshot shows the Spyder IDE interface with the File Explorer panel open. The panel displays a list of files and their modification dates. The files are: import_data.py, main.py, movies.csv, query.py, ratings.csv, and sample.py. The modification dates are: 9/17/24 10:10 AM, 9/17/24 10:11 AM, 9/16/24 10:23 PM, 9/17/24 9:50 AM, 9/17/24 10:10 AM, and 9/17/24 11:43 AM respectively. The bottom of the interface shows tabs for "Help", "Variable Explorer", "Debugger", "Plots", and "Files", with "Files" currently selected.

Name	Date Modified
import_data.py	9/17/24 10:10 AM
main.py	9/17/24 10:11 AM
movies.csv	9/16/24 10:23 PM
query.py	9/17/24 9:50 AM
ratings.csv	9/17/24 10:10 AM
sample.py	9/17/24 11:43 AM

PART 03

SQL demo

SQL order of operations

Writing

1. SELECT
2. FROM
3. JOIN, ON
4. WHERE
5. GROUP BY
6. HAVING
7. ORDER BY
8. LIMIT

Execution

1. FROM
2. JOIN, ON
3. WHERE
4. GROUP BY
5. HAVING
6. SELECT
7. ORDER BY
8. LIMIT

SQL resources

- [SQLite Docs](#)
- [W3schools](#)
- [DataLemur](#)
- [LeetCode Interview Qs](#)
- [O'Reilly SQL Books \(click O'Reilly link w/ NYU creds\)](#)
- [Modern Application: Pinterest Text-to-SQL](#)
- CDS students: SQL for Interviewing Workshop next week