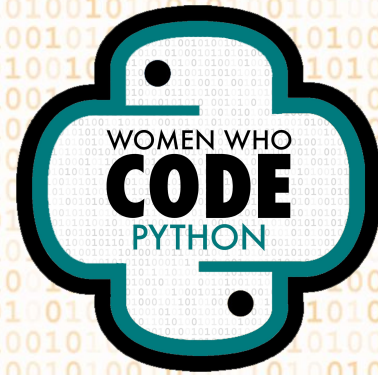


# Women Who Code Python Track

Databases with  
Python: Quickstart

An Overview of SQL: Concepts, Structure and more

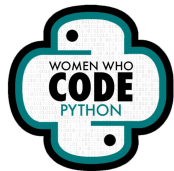


# WELCOME WOMEN WHO CODE





# Hi! We are ...



Hala

WWCode Python  
Volunteer



Ramya

WWCode Python  
Lead



Karen

WWCode Python  
Lead

# OUR MISSION

Inspiring women to  
excel in technology  
careers.

WOMEN WHO  
**CODE**



# OUR VISION

A world where women are representative as technical executives, founders, VCs, board members and software engineers.

WOMEN WHO  
**CODE**





# OUR TARGET

Engineers with two or more years of experience looking for support and resources to strengthen their influence and levelup in their careers.

WOMEN WHO  
**CODE**



# CODE OF CONDUCT

**WWCode is an inclusive community**, dedicated to providing an empowering experience for everyone who participates in or supports our community, regardless of gender, gender identity and expression, sexual orientation, ability, physical appearance, body size, race, ethnicity, age, religion, socioeconomic status, caste, creed, political affiliation, or preferred programming language(s).

Our events are intended to inspire women to excel in technology careers, and anyone who is there for this purpose is welcome. We do not tolerate harassment of members in any form. Our **Code of Conduct** applies to all WWCode events and online communities.

Read the full version and access our incident report form at [womenwhocode.com/codeofconduct](https://womenwhocode.com/codeofconduct)



# 250,000+

## Members

In 95 cities and 122 countries  
with 70 networks,  
10K+ events,  
\$1025 daily Conference tickets,  
\$2M Scholarships and  
Access to [jobs](#) + [resources](#)  
Infinite connections

WOMEN WHO  
**CODE**



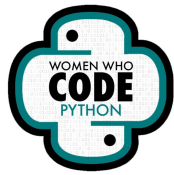


# OUR MOVEMENT

As the world changes, we can be a connecting force that creates a sense of belonging while the world is being asked to isolate.

WOMEN WHO  
**CODE**





# AGENDA

## Database

- What is a database?
- Motivation behind a database
- Roles in a database environment

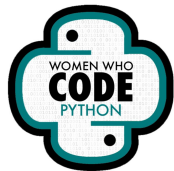
## Relational Database

- What is a data model?
- What makes data model special?
- Designing a relational database: procedure and steps.

## SQL : Structured Query Language + Databases

- What is SQL?
- Examples of SQL Databases



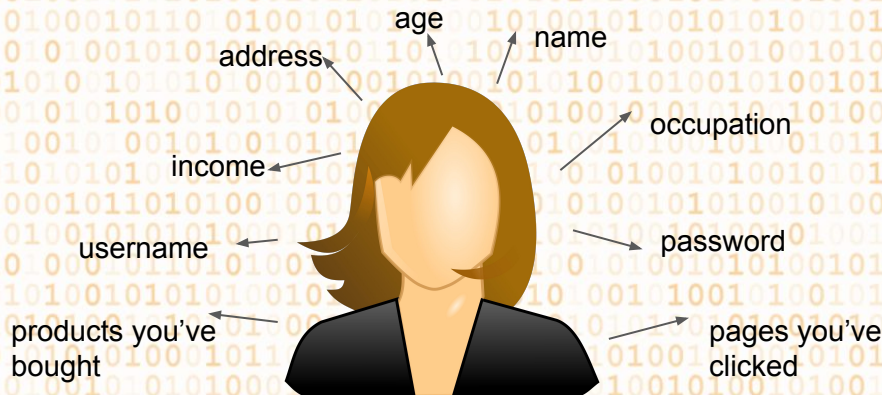


# DATA

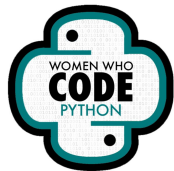
**They say data is the new oil.**

**Before talking about databases, let's discuss:**

**What is data?**







# DATA

For this data to be meaningful and useful, we need a good way to store it.

Common problems in tech:

- Data storage
- Data management
- Data organization

This is where **databases** come in!



An example of messy data



# Database

## What is a database?

- An organized collection of data
- Similar to an information system
- Can save, organize, protect and deliver data
- For example, library is a database of books; folders in a computer



A database icon



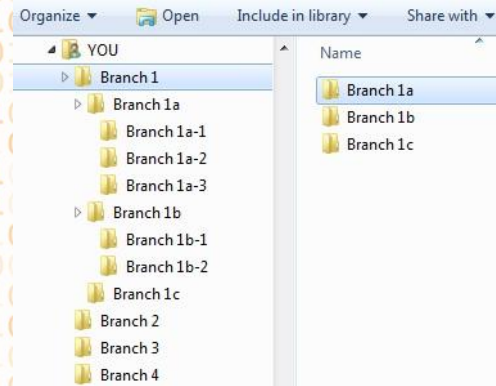
# Database Management Systems

## Motivation behind a DBMS

- File based systems are too slow to search, too messy and not consistent
- Database allows you to easily manage, access and update the information within
- Data security
- Data integrity



VS

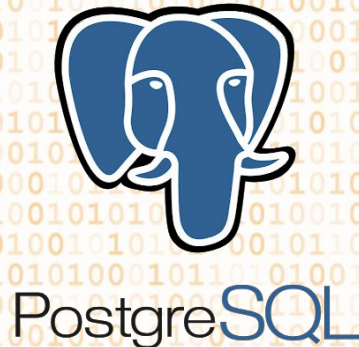




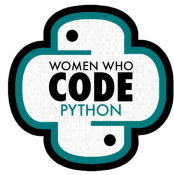
# Database Management Systems

## So what is a DBMS?

- A heavy piece of software meant to store, retrieve, define, and manage data in a database.
- There are multiple popular DBMS adopted by companies worldwide.
- Some examples: MySQL, PostgreSQL, SQLite, Oracle (mainly enterprises)







# Database

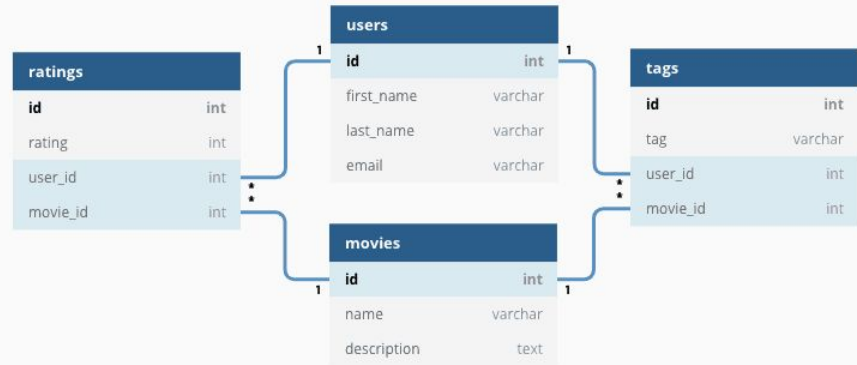
## Roles in a database environment

- *Database designer*
  - Responsible for defining the entire database design details such as tables, indexes, views, constraints, stored procedures and triggers
- *Database administrator*
  - Responsible for managing data resources, physical database design and implementation and data security
- *Developers*
  - Responsible for optimize database systems performance efficiency, troubleshoot database issues, and ensure data quality and integrity
- *End Users*
  - From accessing the database permitted by database administrator, end users usually perform querying, updating and generating reports



# Relational Database

According to Oracle, a **Relational Database** is “a database *structured to recognize relations among stored items of information.*”



an example of a relational database data model. source: [here](#)



# Relational Database

**A Relational Database** is a special kind of database.

- Data is usually in the form of tables.
- These tables have rows of data (records).
- The rows are defined according to the column labels (attributes).

employee_ID	firstname	lastname	address
E43001	John	Smith	New York
E45021	Jane	Doe	Los Angeles



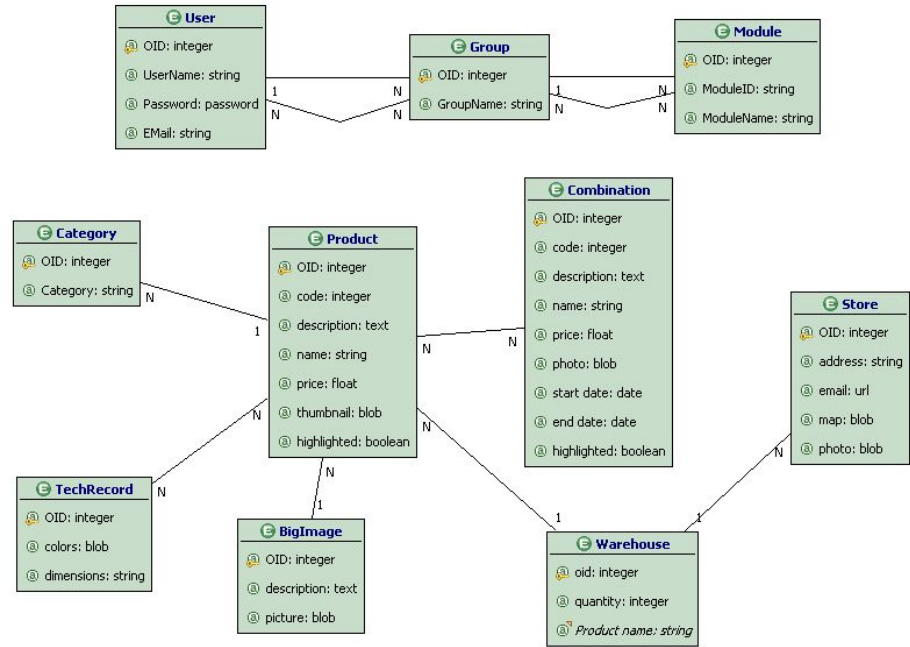
# Relational Databases follow a data model

## What is a data model?

- A data model determines the rules to standardize data and how each table is related to each other
- Describes how data can be represented and accessed

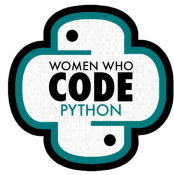
## What makes a data model special?

- Support the development of information systems by providing the definition and format of data



Example of data model mapping





# Structured Query Language (SQL)

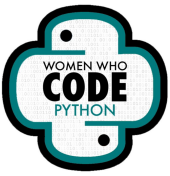
## What is SQL?

- Allows you to access and manipulate data from databases
- Users are able to create, query, update and delete data (with given permissions)
- Basically the language we use to interact with the database.

## To use SQL, we need to understand:

- Its syntax
- How it works with the database





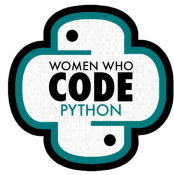
# Structured Query Language (SQL)

## Examples of SQL in Database

- SELECT
- FROM
- WHERE
- ORDER BY

```
SELECT
    attendee_name
FROM
    event_registered
WHERE
    location == "Amsterdam"
ORDER BY
    registered_date ASC
```





# Let's see an example of SQL in action

- **Sample database:**

Chinook database.

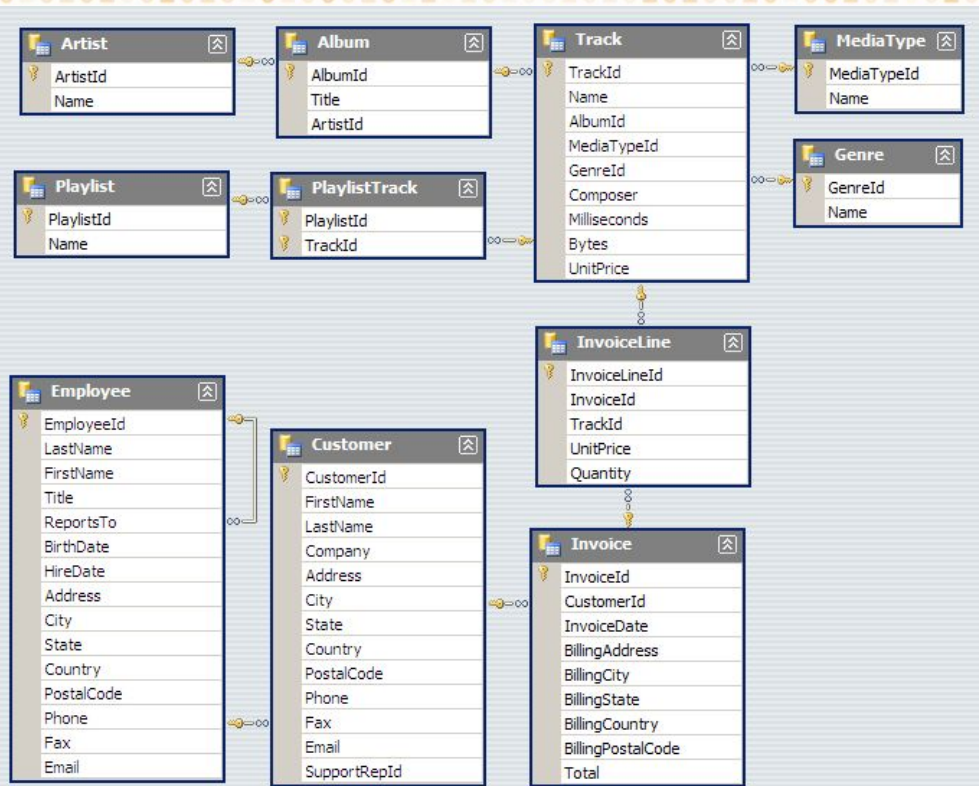
Source:

<https://github.com/lerocha/chinook-database/>

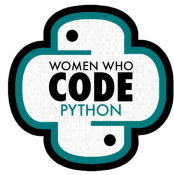




# Schema for the Chinook database







# So, how do we design a relational database?

## Procedure of designing the database

### 1. Think about your data.

What objects do I have? What do I know about those objects? How are these objects related to each other?

### 1. List these objects and their attributes.

Think about the attribute types while you're at it.

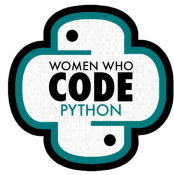
### 3. Define your tables.

Start setting up sample data.

### 3. Draw your schema.

You can get away without doing this for a small project, but it's good practice to get used to the formalism.





# Let's design a database for a bank.

## 1. Think about your data.

What objects do I have? What do I know about those objects? How are these objects related to each other?

**Our objects:** Client, bank account, credit card, employee that handled the client.

**What do I know about these objects?**

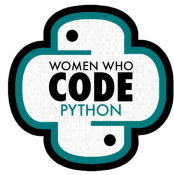
Client: first name, last name, age, employee that handled the client, bank accounts

Bank account: client, cards associated with the bank account, account number

Credit card: card number, account to which the credit card is associated

Employee: first name, last name, age.





# Let's design a database for a bank.

## 2. List these objects and their attributes.

Think about the attribute types while you're at it.

**Client:** `clientID`, firstname, lastname, age, employeeID (**bank accounts**)

**Bank account:** `accountID`, clientID (**cards**)

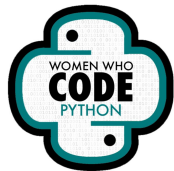
**Credit card:** `cardID`, accountID

**Employee:** `employeeID`, firstname, lastname, age.

In teal are the **primary keys**.

**Primary keys** are unique identifiers for a specific record.





# Let's design a database for a bank.

## 3. Define your tables.

Start setting up sample data.

**Client**

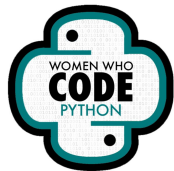
clientID	firstname	lastname	age	employeeID
24001	John	Smith	43	301
24002	Jane	Doe	45	301
24334	Mary	Jane	32	302

**Foreign  
key**

**Employee**

employeeID	firstname	lastname	age
301	Karen	W	50
302	Hala	S	55
303	Ramya	N	32

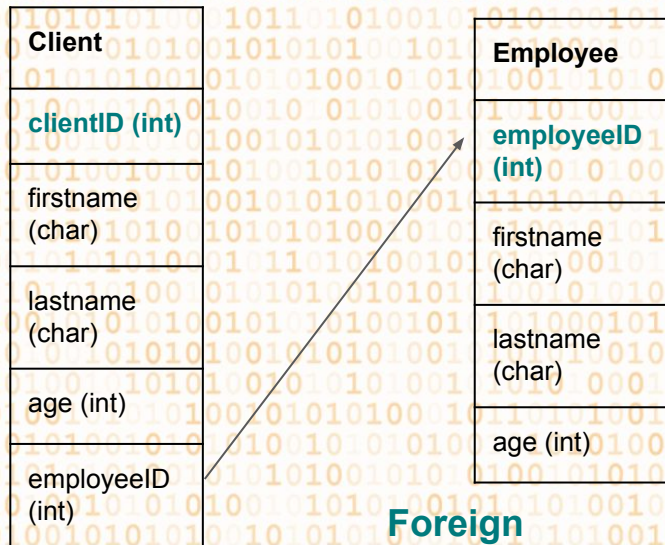




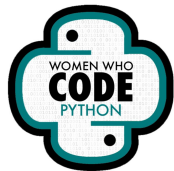
# Let's design a database for a bank.

## 4. Draw your schema.

You can get away without doing this for a small project, but it's good practice to get used to the formalism.







# Let's design a database for a bank.

Let's continue with the rest of our data.

## 3. Define your tables.

Start setting up sample data.

Client

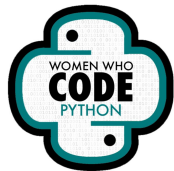
clientID	firstname	lastname	age	employeeID
24001	John	Smith	43	301
24002	Jane	Doe	45	301
24334	Mary	Jane	32	302

Foreign  
key

Employee

employeeID	firstname	lastname	age
301	Karen	W	50
302	Hala	S	55
303	Ramya	N	32





# Let's design a database for a bank.

Let's continue with the rest of our data.

## 3. Define your tables.

Start setting up sample data.

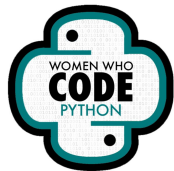
**Client**

clientID	firstname	lastname	age	employeeID
24001	John	Smith	43	301
24002	Jane	Doe	45	301
24334	Mary	Jane	32	302

**Bank Account**

accountID	clientID
552	24002
553	24334
554	24334





# Let's design a database for a bank.

Let's continue with the rest of our data.

## 3. Define your tables.

Start setting up sample data.

**Credit Card**

cardID	accountID
BZGH	552
FGHJ	552
ASDF	554

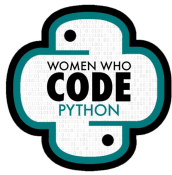
**Bank Account**

accountID	clientID
552	24002
553	24334
554	24334

**Client**

clientID	firstname	lastname	age	employeeID
24001	John	Smith	43	301
24002	Jane	Doe	45	301
24334	Mary	Jane	32	302





# Let's design a database for a bank.

**Credit Card**

cardID	accountID
BZGH	552
FGHJ	552
ASDF	554

**Bank Account**

accountID	clientID
552	24002
553	24334
554	24334

**Client**

clientID	firstname	lastname	age	employeeID
24001	John	Smith	43	301
24002	Jane	Doe	45	301
24334	Mary	Jane	32	302

**Employee**

employeeID	firstname	lastname	age
301	Karen	W	50
302	Hala	S	55
303	Ramya	N	32



# Let's design a database for a bank.

## 4. Draw your schema. *Final Design*

You can get away without doing this for a small project, but it's good practice to get used to the formalism.

CreditCard
cardID (int)
accountID (int)

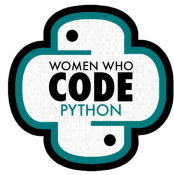
BankAccount
accountID (int)
clientID (int)

Client
clientID (int)
firstname (char)
lastname (char)
age (int)
employeeID (int)

Employee
employeeID (int)
firstname (char)
lastname (char)
age (int)







# Wrap Up!

## Database

- What is a database?
- Motivation behind a database
- Roles in a database environment

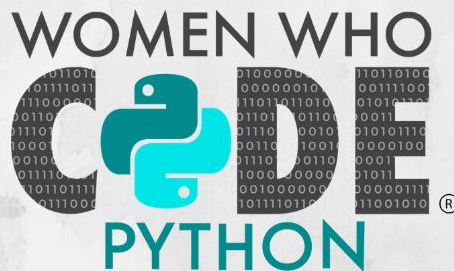
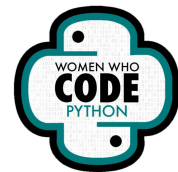
## Relational Database

- What is a data model?
- What makes data model special?

## SQL : Structured Query Language + Databases

- What is SQL?
- Examples of SQL Databases



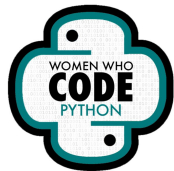


JOIN US ON SOCIAL MEDIA!



@WWCODEPYTHON

[WOMENWHOCODE.COM/PYTHON](https://womenwhocode.com/python)



# Upcoming Events

UPCOMING PAST

WED  
27  
JAN



**Databases with Python: Quickstart : An overview of SQL - Concepts and Structure**



*Featured*

6:00 PM - 8:00 PM (EST) | 📍 Zoom

Register

SUN  
07  
FEB



**Beginner Python Series Extra 🌟: Special Speaker Event: Ahmad**

**Mahmoud** *Featured*

11:00 AM - 12:00 PM (EST) | 📍 Zoom

Register

WED  
17  
FEB



**Databases with Python: Quickstart: SQL and Python**  *Featured*

6:00 PM - 8:00 PM (EST) | 📍 Zoom

Register





Thank you

Databases with Python: Quickstart

Session#2 - **SQL and Python** on 17th Feb 2021

