

COM5961 DATA DRIVEN PRODUCTS & SERVICES DESIGN: LESSON 3 - INTRO TO RELATIONAL DATABASE & SQL

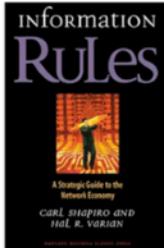
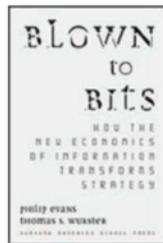
Bernard Suen
Center for Entrepreneurship
Chinese University of Hong Kong

Today's agenda.

1. How to submit your assignment?
2. Introduction to Python functions, modules, packages, and libraries.
3. Introduction to the **MVC** Framework.
4. Storing Data: CSV file vs. Relational (SQL) Database.
5. Representing relational database schema in ERD (Entity Relationship Diagram) visually.
6. Learning relational database in **DB Browser for SQLite3.**
7. What is **SQL** (Structured Query Language)?

Assignment Submission

All Journals



Rethink Digital Transformation

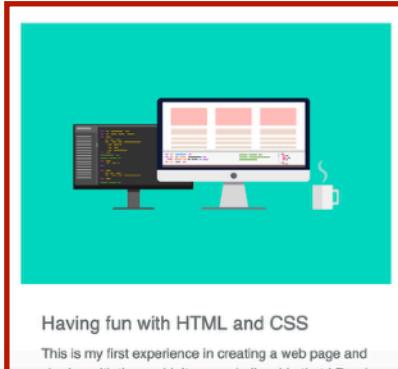
No doubt that "Digital Transformation" is a hot topic. The spread of COVID-19 has accelerated the concern as social distancing [Read more](#)

No-code/Low-code: the Spreadsheet Revolution Replayed?

The no-code/low-code movement, which has captured the attention of the venture investment community in recent years, can be viewed as [Read more](#)

Making Remote Works Collaborative and Accountable

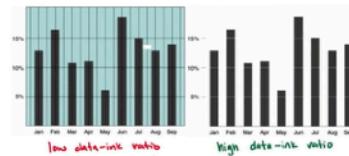
Design thinking and digital transformation are two buzzwords receiving a lot of attention these days. They also give business consultants [Read more](#)



Having fun with HTML and CSS
This is my first experience in creating a web page and sharing with the world. It was really difficult that I Read [Read more](#)

Data-Ink Ratio: Tufte principle of Data Visualisation

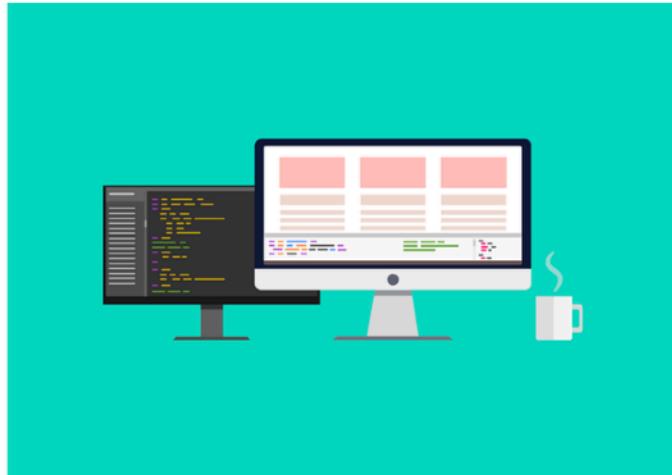
The interior decoration of graphics generates a lot of ink that does not tell the viewer anything new. The purpose [Read more](#)



COVID-19: The Great Accelerator of Work and Learning

In an article Thomas Friedman wrote for New York Times, he said we should focus on [Read more](#)

Having fun with HTML and CSS



September 26, 2022 David Doe

This is my first experience in creating a web page and sharing with the world. It was unbelievable that I can do that as a person coming from a non-technical background. After all, the learning hasn't been as difficult as I thought. Hopefully, I can do more interesting things with the newly acquired skills.

My current design for the first assignment is based on one main idea: How can I make my creative works more accessible and inspiring for other people? That's why I have chosen a simple horizontal navigation bar with the current navigation option highlighted and the paragraph heading displayed in H1 tag. You can access my website [here.](#)

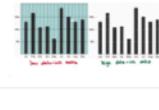
Simplicity is important but amusement, the entertaining impact my works create, is equally so. I still have to work harder on both. They are design attributes that can delight people and make them willing to see my works more.

My Dashboard

[New Post](#) | [Edit Author Account](#) | [My Messages](#)

David Doe's Dashboard

You have created 4 (Posts)

Featured Image	Title	Status	Options
	Having fun with HTML and...	Live	 
	Making Remote Works Collaborative and...	Live	 
	Rethink Digital Transformation	Live	 
	Data-Ink Ratio: Tufte principle of...	Live	 

Author Info

David Doe

I received an undergraduate degree in Communication. My interests include writing, table-tennis, photography and movies. Without a technical background, I'm going to put in extra effort to strengthen my coding and design skills. My goal is to become proficient in HTML/CSS/JS, Python, and SQL programming through my study in the Master of New Media programme at CUHK. I welcome your thoughts on how we can collaborate to learn. You can access my website here.

Edit Post

Post Title *

Having fun with HTML and CSS

Category *

HTML and CSS

Select a category for your post

Post description *

Insert Photo

<p>This is my first experience in creating a web page and sharing with the world. It was unbelievable that I can do that as a person coming from a non-technical background. After all, the learning hasn't been as difficult as I thought. Hopefully, I can do more interesting things with the newly acquired skills.</p>

<p>My current design for the first assignment is based on one main idea: How can I make my creative works more accessible and inspiring for other people? That's why I have chosen a simple horizontal navigation bar with the current navigation option highlighted and the paragraph heading displayed in H1 tag. You can access my website here.</p>

<p>Simplicity is important but amusement, the entertaining impact my works create, is equally so. I still have to work harder on both. They are design attributes that can delight people and make them willing to see my works more.</p>

Write the full description of your Post

Featured Image *



frontend-c3bac

Upload the main image of your post

Excerpt

Free images.

[Photos](#)[Illustrations](#)[Vectors](#)[Videos](#)[Music](#)

...

[Explore](#) ▾[Sign up](#)

Stunning free images & royalty free stock

Over 1.7 million+ high quality stock images and videos shared by our talented community.

[Images ▾](#)

Popular images: easter, wallpaper, coronavirus, nature, corona, love, business, food, spring, money, covid, computer

Free image by mac231

<https://pixabay.com/>



David Doe
Student

My profile

NAME	LAST NAME
David	Doe
BIO	
<p>I received an undergraduate degree in Communication. My interests include writing, table-tennis, photography and movies. Without a technical background, I'm going to put in extra effort to strengthen my coding and design skills.</p> <p>My goal is to become proficient in HTML/CSS/Javascript, Python, and SQL programming through my study in the Master of New Media programme at CUHK.</p> <p>I welcome your thoughts on how we can collaborate to learn. You can access my website here.</p>	
WEBSITE URL	
https://suentze2020.github.io/assignments/	

Change Password

NEW PASSWORD	RE-TYPE NEW PASSWORD
<input type="password"/> Enter your New Password	<input type="password"/> Enter your new password

Assignment 1

Assignment 2

Assignment 3

Assignment 4

Assignment 5

Assignment 1

This is my first assignment...

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque vitae turpis aliquam sem efficitur vestibulum. Phasellus in venenatis est. Maecenas varius, tellus vel congue sodales, lectus ipsum fringilla elit, at elementum mauris tellus in odio. Nulla id blandit massa. Maecenas dictum ipsum justo, vitae iaculis ipsum finibus quis. Integer vel nisl tincidunt enim sodales cursus. Morbi finibus hendrerit orci. Nunc feugiat justo at varius facilisis. Nunc ut faucibus nibh. Etiam nisl nulla, rhoncus id sollicitudin condimentum, hendrerit quis orci. Aenean sed ultrices massa. Integer nulla est, tempus sed elit vel, efficitur vestibulum orci. Etiam auctor sodales nibh in vulputate.

More descriptions can be found in my [journal](#).



Search or jump to...

Pull requests Issues Marketplace Explore



suentze2020 / assignments

Public

Pin

Unwatch 1

Fork 0

Star 0

Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

main

1 branch

0 tags

Go to file

Add file

Code

suentze2020 Assignment 1 demo ...

✓ 3f6b5ac 1 hour ago 2 commits

.vscode

Assignment 1 demo

1 hour ago

.DS_Store

Assignment 1 demo

1 hour ago

.gitattributes

Initial commit

19 hours ago

assign2.html

Assignment 1 demo

1 hour ago

assign3.html

Assignment 1 demo

1 hour ago

assign4.html

Assignment 1 demo

1 hour ago

assign5.html

Assignment 1 demo

1 hour ago

hamburger_menu.png

Assignment 1 demo

1 hour ago

index.html

Assignment 1 demo

1 hour ago

style.css

Assignment 1 demo

1 hour ago

Help people interested in this repository understand your project by adding a README.

Add a README

About



Assignments Repo

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Environments 1

github-pages Active

Languages



[General](#)[Access](#)[Collaborators](#)[Moderation options](#)[Code and automation](#)[Branches](#)[Tags](#)[Actions](#)[Webhooks](#)[Environments](#)[Pages](#)[Security](#)[Code security and analysis](#)[Deploy keys](#)[Secrets](#)[Integrations](#)[GitHub apps](#)[Email notifications](#)

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at <https://suentze2020.github.io/assignments/>

Last deployed by [github-pages](#) 1 hour ago

[Visit site](#)[...](#)

Build and deployment

Source

[Deploy from a branch](#)

Branch

Your GitHub Pages site is currently being built from the main branch. [Learn more](#).

[main](#)[/ \(root\)](#)[Save](#)

Learn how to [add a Jekyll theme](#) to your site.

Your site was last deployed to the [github-pages](#) environment by the [pages build and deployment](#) workflow.

Custom domain

Custom domain

Custom domains allow you to serve your site from a domain other than [suentze2020.github.io](#). [Learn more](#).

[Save](#)[Remove](#)

Combining List & Dictionary

```
student1 = {  
    'Id':'1111',  
    'name':'David Chan',  
    'email':'david@cuhk.edu.hk'  
}  
  
student2 = {  
    'id':'1112',  
    'name':'Peter Lee',  
    'email':'peter@cuhk.edu.hk'  
}  
  
student3 = {  
    'id':'1113',  
    'name':'Mary Fung',  
    'email':'mary@cuhk.edu.hk'  
}
```

```
students = []  
students.append(student1)  
students.append(student2)  
students.append(student3)  
print(students)  
for i in students:  
    print(i)  
key = 0  
for i in students:  
    print("Individual entry:",i)  
    for key in i:  
        print("Key:",key,"Value:",i[key])
```

Python Function

```
def name_of_function(parameter1, parameter2...):
```

[Python variable assignment statements,
mathematical and logical operations]

```
return result
```

Example

```
def calc_avg(x,y):  
    result = x + y  
    return result
```

```
print("Result = ", calc_avg(10,20))
```

Output

Result=30

Exercise

Formula for converting Celcius into Fahrenheit: $(23^{\circ}\text{C} \times 9/5) + 32 = 73.4^{\circ}\text{F}$.

Create a list of temperature in Fahrenheit for the week based on the temperature of Celcius but implement the codes with Python functions. [¶](#)

Sun	Mon	Tue	Wed	Thu	Fri	Sat
23°C	24°C	28°C	19°C	17°C	21°C	25°C

Functions & Classes

```
students = [
    {"name": "Bernard", "score": 80},
    {"name": "John", "score": 90},
    {"name": "Peter", "score": 100},
]

def avg(students):
    total = 0
    count = len(students)
    for student in students:
        print(student['name'], ':', student['score'])
        total = total + student['score']
    return total / count

result = avg(students)
print("Average score =", result)
```

```
class Student:
    studentCount = 0
    studentScore = 0
    avgScore = 0

    def __init__(self, name, score):
        self.name = name
        self.score = score
        Student.studentCount += 1
        Student.studentScore += score

    def displayStudent(self):
        # print("Name : ", self.name, " , Score: ", self.score)
        print(self.name, ":", self.score)

    #def __str__:
    #    return f"Person {self.name} has {self.score} points."

    def __repr__(self):
        return f"<Person('{self.name}'. {self.score})>"

student1 = Student("Bernard", 80)
student2 = Student("John", 90)
student3 = Student("Peter", 100)
student1.displayStudent()
student2.displayStudent()
student3.displayStudent()
avg_score = Student.studentScore/Student.studentCount
print("Average Score = %d" % avg_score)
print(student1)
```

Python Module

Importing a module 'convert_temp.py'

```
from convert_temp import *
total_temp = 0
total_temp = get_daily_celcius()
display_avg(total_temp)
```

convert_temp.py

```
def convert_temp(temp):  
    f_value = 0  
    f_value = (float(temp) * 9/5) + 32  
    return f_value
```

```
def compute_total_temp(temp, total_temp):  
    total_temp = total_temp + float(temp)  
    return total_temp
```

```
def get_daily_celcius():
    total_temp = 0
    week_of_day =['Sun','Mon','Tue','Wed','Thu','Fri','Sat']

    for i in week_of_day:
        temp = input("Enter " + i + "'s temperature in celcius:")
        f_temp = convert_temp(temp)
        print(i + "'s temperature in Fahrenheit:" + str(f_temp))
        total_temp = compute_total_temp(f_temp, total_temp)
    return total_temp
```

```
def display_avg(total_temp):
    avg_temp = round((total_temp/7),2)
    print("\n" + "Average temperature for the week:" + str(avg_temp))
    return
```

Python Package

[Quit](#)[Logout](#)[Files](#) [Running](#) [Clusters](#)

Select items to perform actions on them.

[Upload](#) [New](#)

<input type="checkbox"/> 0	/ Jupyter / Classes / com5961 / mypackage	Name	Last Modified	File size
	..		seconds ago	
	__init__.py		6 minutes ago	0 B

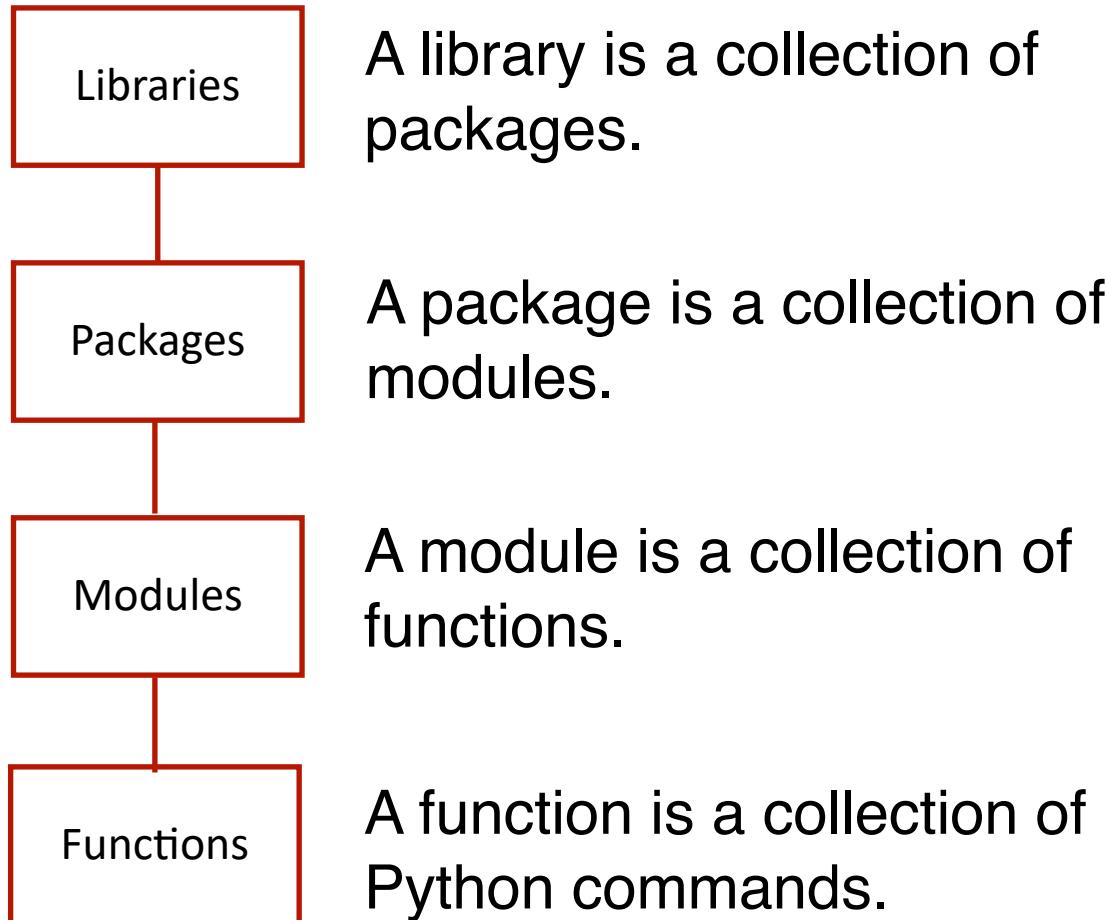
A package is a folder with a
__init__.py file stored in it.

Importing a package (folder) 'temppackage'

```
from temppackage.convert_temp import *
total_temp = 0
total_temp = get_daily_celcius()
display_avg(total_temp)
```

The Pandas Library

(A Collection of Packages)



Save a Pandas Data Frame into CSV file

```
import pandas as pd
scores = [
    {'Student': 'David Chan', 'Jan': 90, 'Feb': 85, 'Mar': 88},
    {'Student': 'Peter Lee', 'Jan': 72, 'Feb': 75, 'Mar': 68},
    {'Student': 'John Lui', 'Jan': 60, 'Feb': 80, 'Mar': 100 }]
df = pd.DataFrame(scores)
df.to_csv('weekly_hours.csv', mode='w', index=False)
df
list = df.to_dict('records')
for entry in list:
    print(entry['Student'], entry['Jan'], entry['Feb'], entry['Mar'])
```

```
import pandas as pd
scores = [
    {'scores': 'David Chan', 'Jan': 90, 'Feb': 85, 'Mar': 88},
    {'scores': 'Peter Lee', 'Jan': 72, 'Feb': 75, 'Mar': 68},
    {'scores': 'John Wong', 'Jan': 60, 'Feb': 80, 'Mar': 2000 }]
df = pd.DataFrame(scores)
df
df.to_csv('students.csv', mode='w', index=False)
df
```

Read a CSV file into a Pandas Data Frame

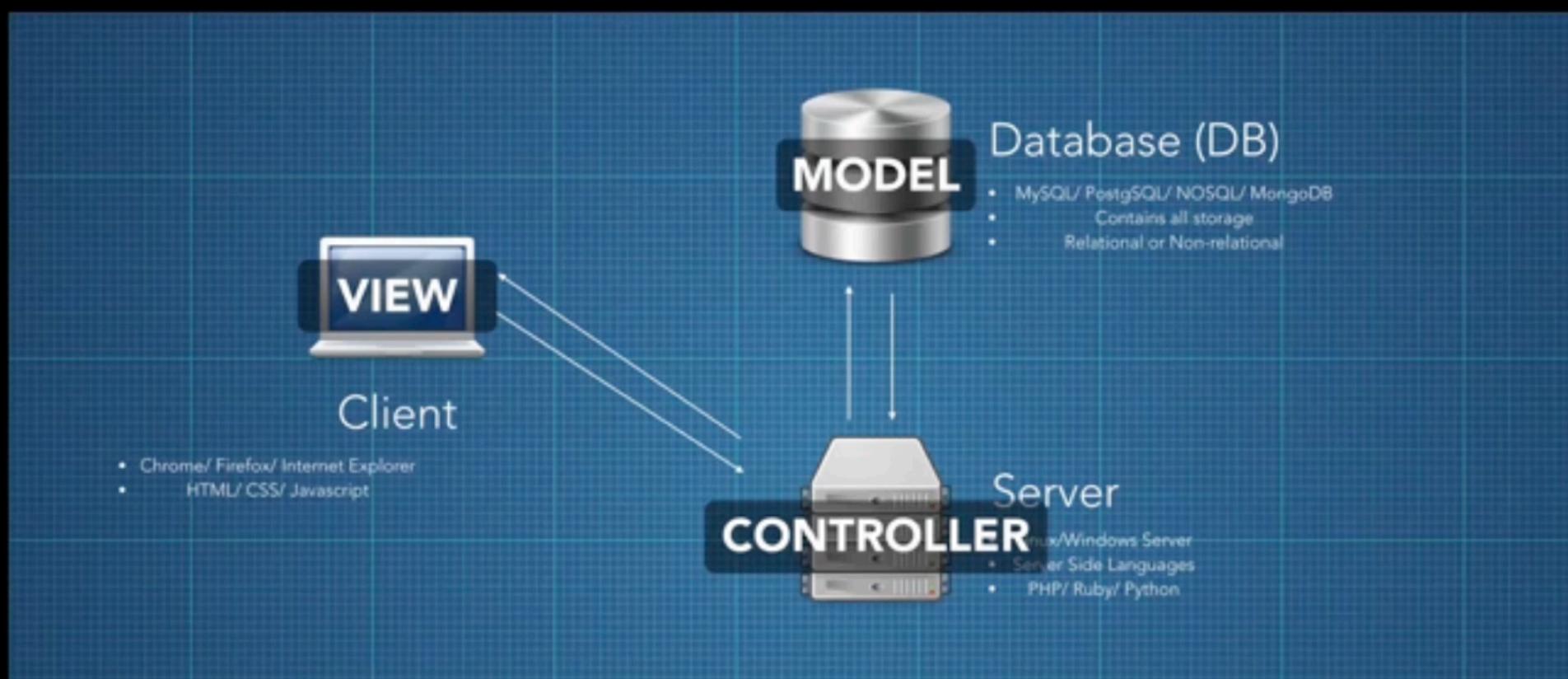
```
import pandas as pd  
df = pd.read_csv("students.csv")  
df
```

MVC Framework

(A Tool for Guiding Data Driven Product Specification)

HOW DOES A WEBSITE WORK?

THE FLOW



<https://www.youtube.com/watch?v=1lsL6g2ixak&t=761s>

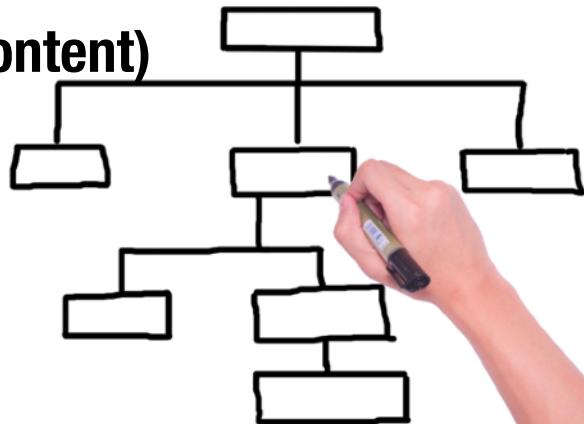
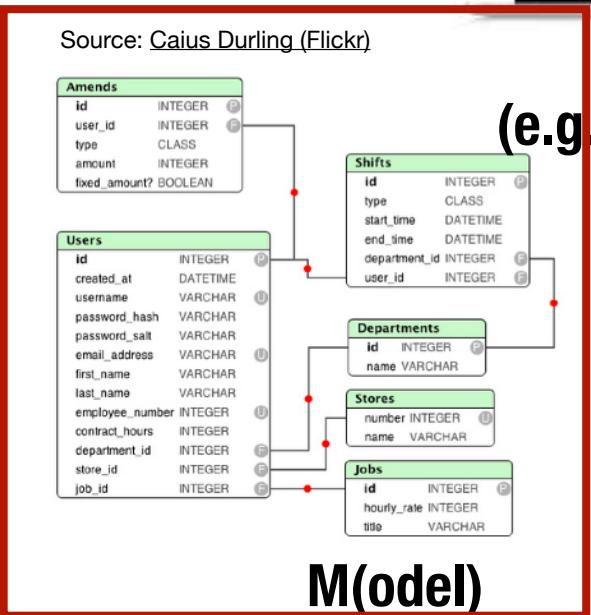
**WHAT
THE HELL IS
MVC?**

Source: commons.wikimedia.org



Source: pexels.com

V(iew) (e.g. user interface, media content)



C(ontroller) (e.g. sitemap, navigation, routes)

Browser/ Mobile



Frontend

Web Server

HTML/CSS/JS



(e.g. Github)



```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title> A Tiny HTML Document </title>
<link href = "styles.css" rel="stylesheet">
<script src= "scripts.js"></script>
</head>

<body>
<p>Let's rock the browser, HTML5 style.</p>
</body>
</html>
```

Backend

Browser/ Mobile

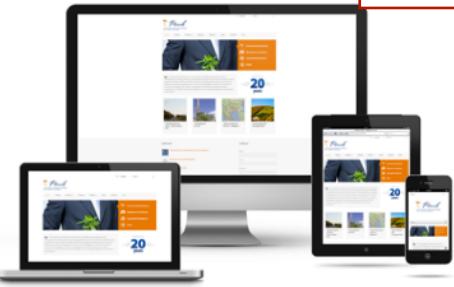


Web Server

HTML/CSS/Javascript



View



App Server

NodeJS/Python

PHP/C#/Java



Controller

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title> A Tiny HTML Document </title>
<link href = "styles.css" rel="stylesheet">
<script src="scripts.js"></script>
</head>

<body>
<p>Let's rock the browser, HTML5 style.</p>
</body>
</html>
```

Database Server

SQL/NoSQL

Data



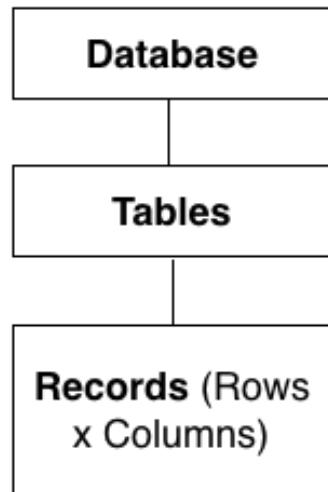
Model

MyTable									
	BROWSE	STRUCTURE	SQL	SEARCH	INSERT	EXPORT	IMPORT	PRIVILEGES	
1	Edit	Copy	Delete	1194023	00994414	preferred_name_00994414	00994414	23.799675	
2	Edit	Copy	Delete	1194021	00994415	popular_name_00994415	00994415	23.799675	
3	Edit	Copy	Delete	1194023	00994416	popular_name_00994416	00994416	23.799675	
4	Edit	Copy	Delete	1194023	00994417	popular_name_00994417	00994417	23.799675	
5	Edit	Copy	Delete	1194023	00994418	popular_name_00994418	00994418	23.799675	
6	Edit	Copy	Delete	1194023	00994419	popular_name_00994419	00994419	23.799675	
7	Edit	Copy	Delete	1194023	00994420	popular_name_00994420	00994420	23.799675	
8	Edit	Copy	Delete	1194021	00994421	popular_name_00994421	00994421	23.799675	
9	Edit	Copy	Delete	1194020	00994422	popular_name_00994422	00994422	23.799675	
10	Edit	Copy	Delete	1194020	00994423	popular_name_00994423	00994423	23.799675	
11	Edit	Copy	Delete	1194020	00994424	popular_name_00994424	00994424	23.799675	
12	Edit	Copy	Delete	1194021	00994425	popular_name_00994425	00994425	23.799675	
13									

Frontend

Backend

What is a “relational database”?



“A **relational database** is made up of a collection of **tables** which relate to each other for storing and managing data entries (**records**), organised by rows and columns.

A database can be used for representing and tracking people, things, events. and transactions.

- People (e.g. students, employees, customers, donors, volunteers)
- Things (e.g. properties, stocks, products, books)
- Events (e.g. campaigns, courses, conferences)
- Transactions (e.g. billings, orders, tasks, donations)

Relational Database vs. Delimited CSV (Comma Separated Values) File

Relational Database

Columns (Fields)

Contacts						
	id	name	age	birthday	email	country_id
	1	Peter Chan	52	1968-05-01	peter@gmail.com	1
	3	Tony Wong	16	2005-07-11	robert@gmail.com	2
	4	Robert Choi	36	1983-03-23	rchoi@cuhk.edu.hk	3
	5	John Li	16	2005-05-23	jli@gmail.com	4

Primary Key (PK)

Foreign Key (FK)

Columns (Fields)

Countries

	country_id	country_name
	1	PRC
	2	USA
	3	France
	4	Japan
	5	Italy
	6	Germany

Primary Key (PK)

Rows
(Records)

CSV File

	id	name	age	birthday	email	country
	1,	"Peter Chan",	52,	"1968-05-01",	peter@gmail.com,	PRC
	3,	"Tony Wong",	16,	"2005-07-11",	robert@gmail.com,	USA
	4,	"Robert Choi",	36,	"1983-03-23",	rchoi@cuhk.edu.hk	France
	5,	"John Li",	16,	"2005-05-23",	jli@gmail.com,	Japan
	6,	"Coco Zhang",	16,	"2005-07-23",	coco@yahoo.com,	People Republic of China
	7,	"	"	"	"	"
	8,	"	"	"	"	PRC
	9,	"	"	"	"	"

country
PRC
USA
France
Japan
People Republic of China
PRC



Advantages and Disadvantages in Using CSV File

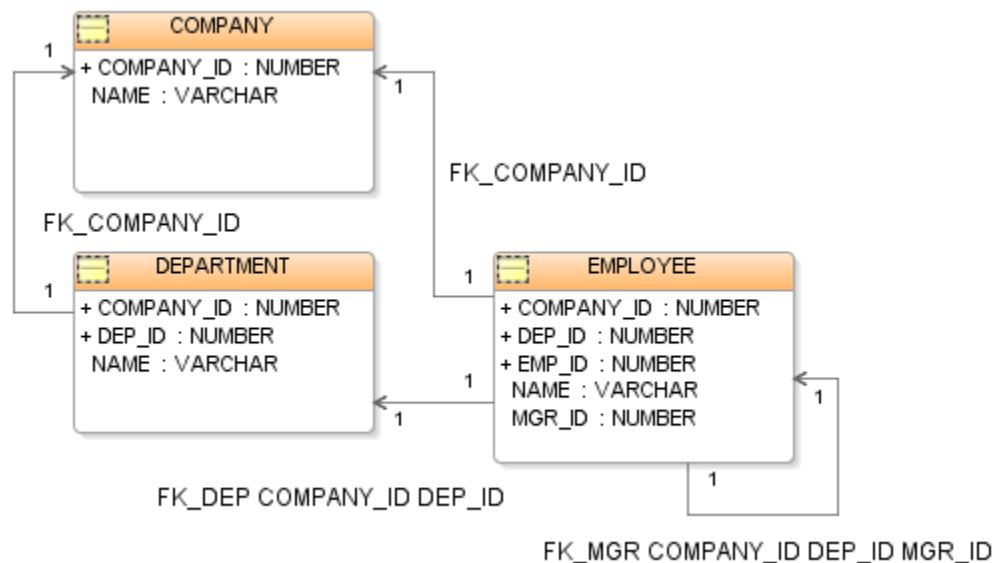
- **Simple to use and build.**
- **Hard to update (when field value changes, all related entries have to be changed manually).**
- **Lend itself to inconsistent entries when different people enter different values representing the same attribute (e.g. country name).**
- **Lead to enormous number of fields (columns) per table when dealing with complex data models.**
- **Make data modelling overly complicated.**
- **CSV file is not as secured as relational database.**

How to Build Data Relationships with Primary Key and Foreign Key

Building relationships between tables by using foreign keys.

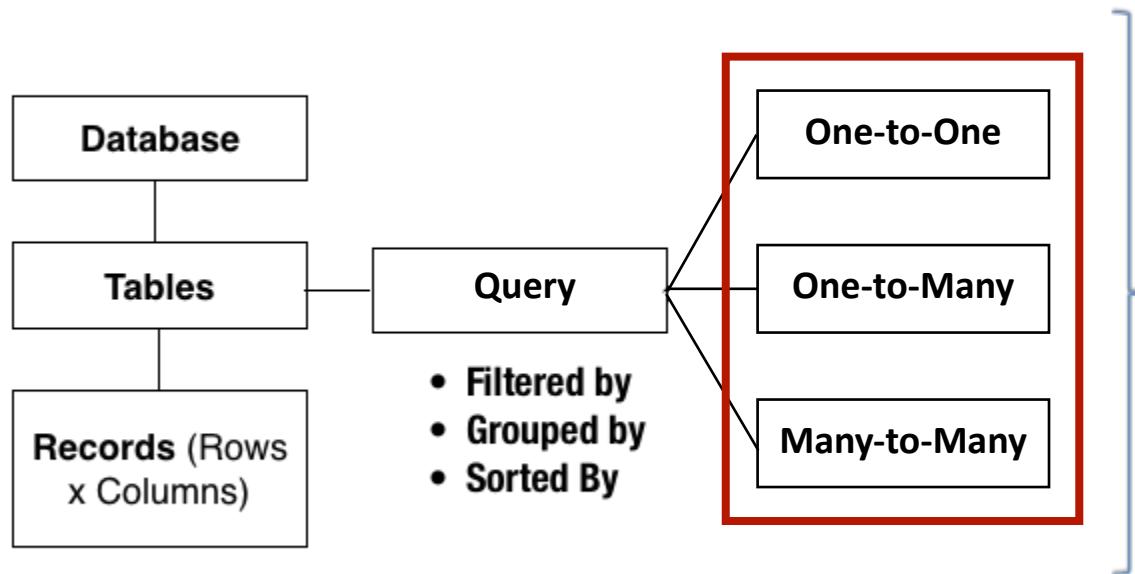
- Every entry in a table should be uniquely defined by a **primary key**.
- Build relationships between tables by making one of the fields in the table as a **foreign key** linked to the **primary key** of another table.

Entity Relationship Diagram (ERD)

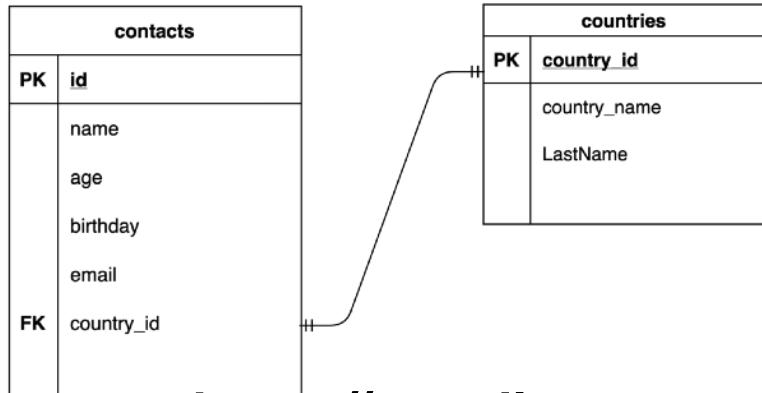


Source: commons.wikimedia.org

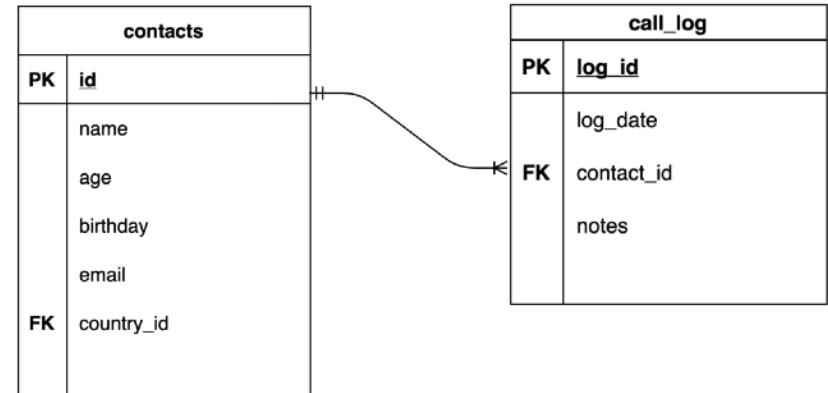
The Three Most Common Types of Relationship in Relational Database



• One-to-One

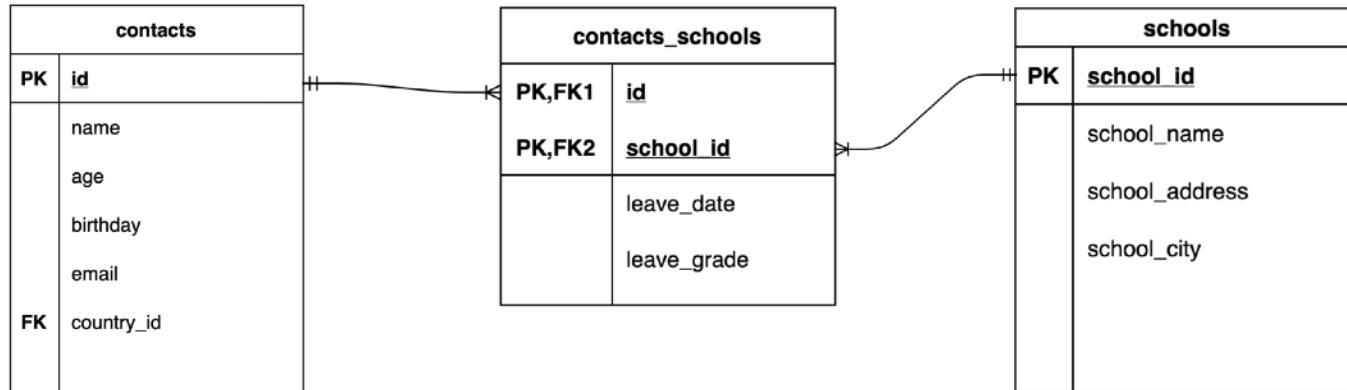


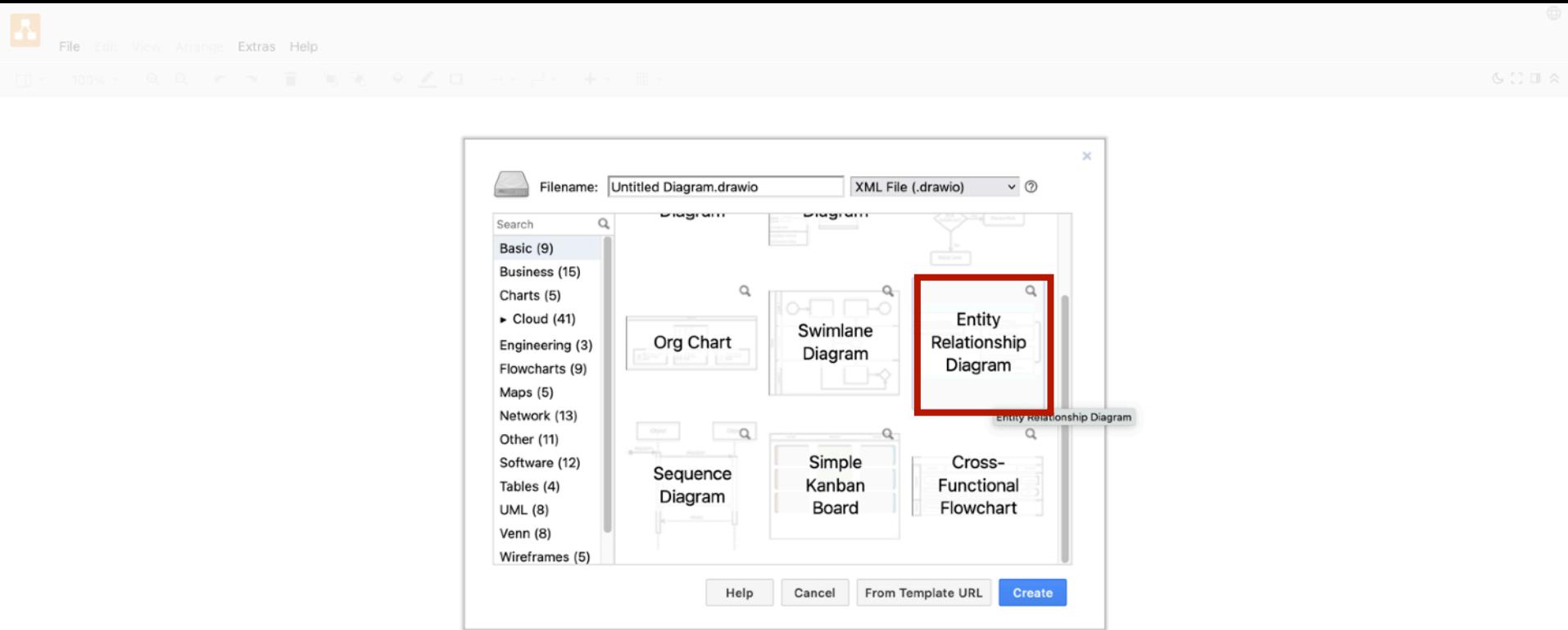
• One-to-Many



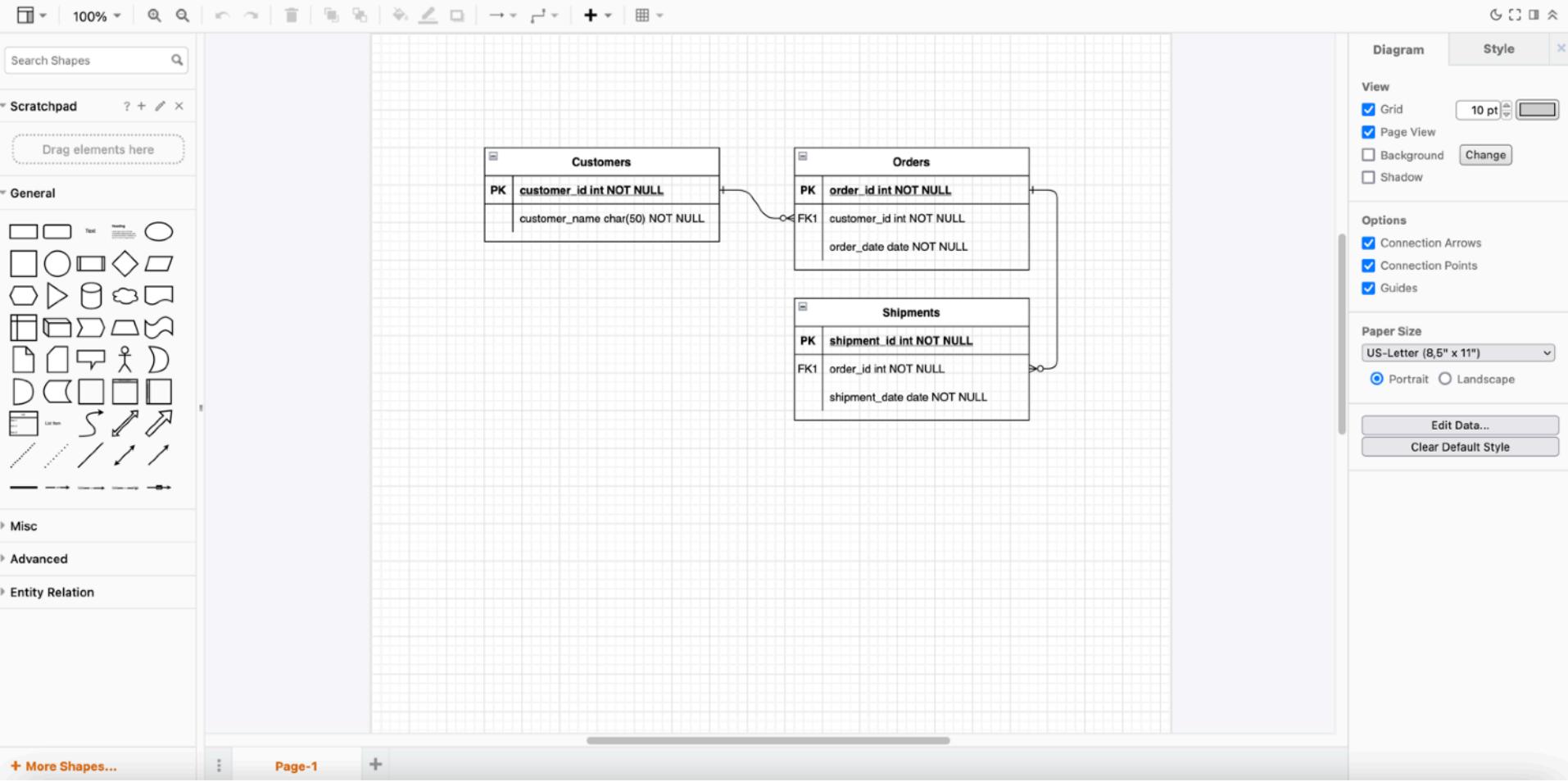
<https://app.diagrams.net/>

• Many-to-Many





<https://app.diagrams.net/>

[More Shapes...](#)

Page-1

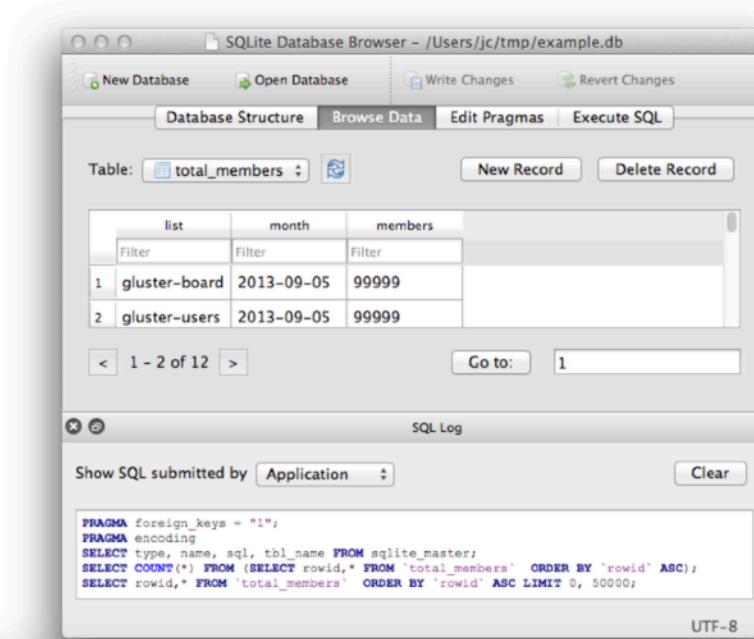
Learning Relational Database with SQLite



DB Browser for SQLite

The Official home of the DB Browser for SQLite

Screenshot



<https://sqlitebrowser.org/>

Downloads

(Please consider sponsoring us on Patreon 😊)

Windows

Our latest release (3.12.2) for Windows:

- [DB Browser for SQLite – Standard installer for 32-bit Windows](#)
- [DB Browser for SQLite – .zip \(no installer\) for 32-bit Windows](#)
- [DB Browser for SQLite – Standard installer for 64-bit Windows](#)
- [DB Browser for SQLite – .zip \(no installer\) for 64-bit Windows](#)

Windows PortableApp

- [DB Browser for SQLite – PortableApp](#)

Note – If for any reason the standard Windows release does not work (e.g. gives an error), try a nightly build ([below](#)).

Nightly builds often fix bugs reported after the last release. 😊

macOS

Our latest release (3.12.2) for macOS:

- [DB Browser for SQLite](#)

<https://sqlitebrowser.org/dl/>



DB Browser for SQLite

The Official home of the DB Browser for SQLite

Screenshot

The screenshot shows the SQLite Database Browser interface. At the top, there's a menu bar with options like 'New Database', 'Open Database', 'Write Changes', and 'Revert Changes'. Below the menu is a toolbar with buttons for 'Database Structure', 'Browse Data' (which is selected), 'Edit Pragmas', and 'Execute SQL'. A dropdown menu labeled 'Table:' shows 'total_members'. There are 'New Record' and 'Delete Record' buttons. The main area displays a table with three columns: 'list', 'month', and 'members'. The data shows two entries:

	list	month	members
1	gluster-board	2013-09-05	99999
2	gluster-users	2013-09-05	99999

Below the table are navigation buttons for pages 1-2 of 12 and a 'Go to:' input field. At the bottom, there's a 'SQL Log' section with a dropdown for 'Show SQL submitted by' (set to 'Application') and a 'Clear' button. The log contains the following SQL code:

```
PRAGMA foreign_keys = "1";
PRAGMA encoding
SELECT type, name, sql, tbl_name FROM sqlite_master;
SELECT COUNT(*) FROM (SELECT rowid,* FROM `total_members` ORDER BY `rowid` ASC);
SELECT rowid,* FROM `total_members` ORDER BY `rowid` ASC LIMIT 0, 50000;
```

The status bar at the bottom right indicates 'UTF-8'.

What is SQL (Structured Query Language)?

Structured Query Language (SQL**)**

“Language (commands) for instructing the database to perform data creation, manipulation, queries, and controls.”

1. **Sort** single or multiple fields as sorting criteria.
2. **Group** entries by a single field or multiple fields to create data segments.
3. **Filter** entries by single or multiple fields as searching conditions.

Use sort, group, and filter to search for information in a table.

Retrieve Record

SELECT FROM table name **WHERE** condition
GROUP BY field name
ORDER BY field name **ASC | DESC**

Retrieve Records by Aggregate/Statistic Functions

SELECT COUNT (field name)
FROM table name GROUP BY field name WHERE condition

SELECT SUM (field name)
FROM table name GROUP BY field name WHERE condition

SELECT AVG (field name)
FROM table name GROUP BY field name WHERE condition

SELECT MAX I MIN (field name)
FROM table name GROUP BY field name WHERE condition

Case Study: Use of the *Northwind* dataset

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](#)[SQL Wildcards](#)[SQL In](#)[SQL Between](#)[SQL Aliases](#)[SQL Joins](#)[SQL Inner Join](#)[SQL Left Join](#)

Demo Database

Below is a selection from the "Customers" table in the Northwind sample database:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

SELECT Column Example

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



graphql-compose / graphql-compose-examples

Public

Sponsor

Notifications

Fork 283

Star 172

Code

Issues 2

Pull requests 9

Actions

Projects

Security

Insights

master

graphql-compose-examples / examples / northwind / data / csv /

Go to file



nodkz Preparation for big example: added sample data from Northwind.

6ab72eb on 20 Aug 2016

History

..

README.md

Preparation for big example: added sample data from Northwind.

6 years ago

categories.csv

Preparation for big example: added sample data from Northwind.

6 years ago

customers.csv

Preparation for big example: added sample data from Northwind.

6 years ago

employee_territories.csv

Preparation for big example: added sample data from Northwind.

6 years ago

employees.csv

Preparation for big example: added sample data from Northwind.

6 years ago

order_details.csv

Preparation for big example: added sample data from Northwind.

6 years ago

orders.csv

Preparation for big example: added sample data from Northwind.

6 years ago

products.csv

Preparation for big example: added sample data from Northwind.

6 years ago

regions.csv

Preparation for big example: added sample data from Northwind.

6 years ago

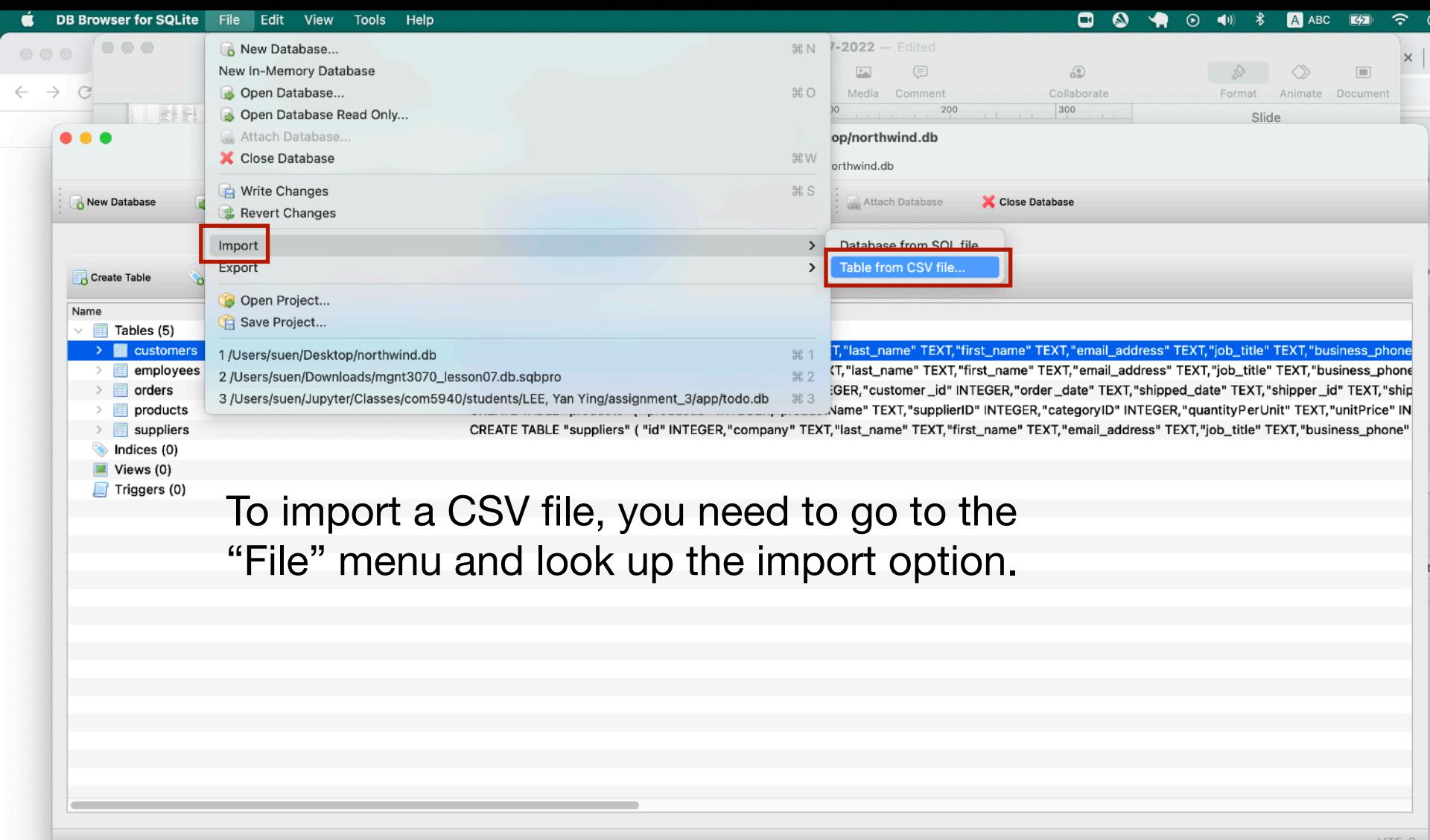
shippers.csv

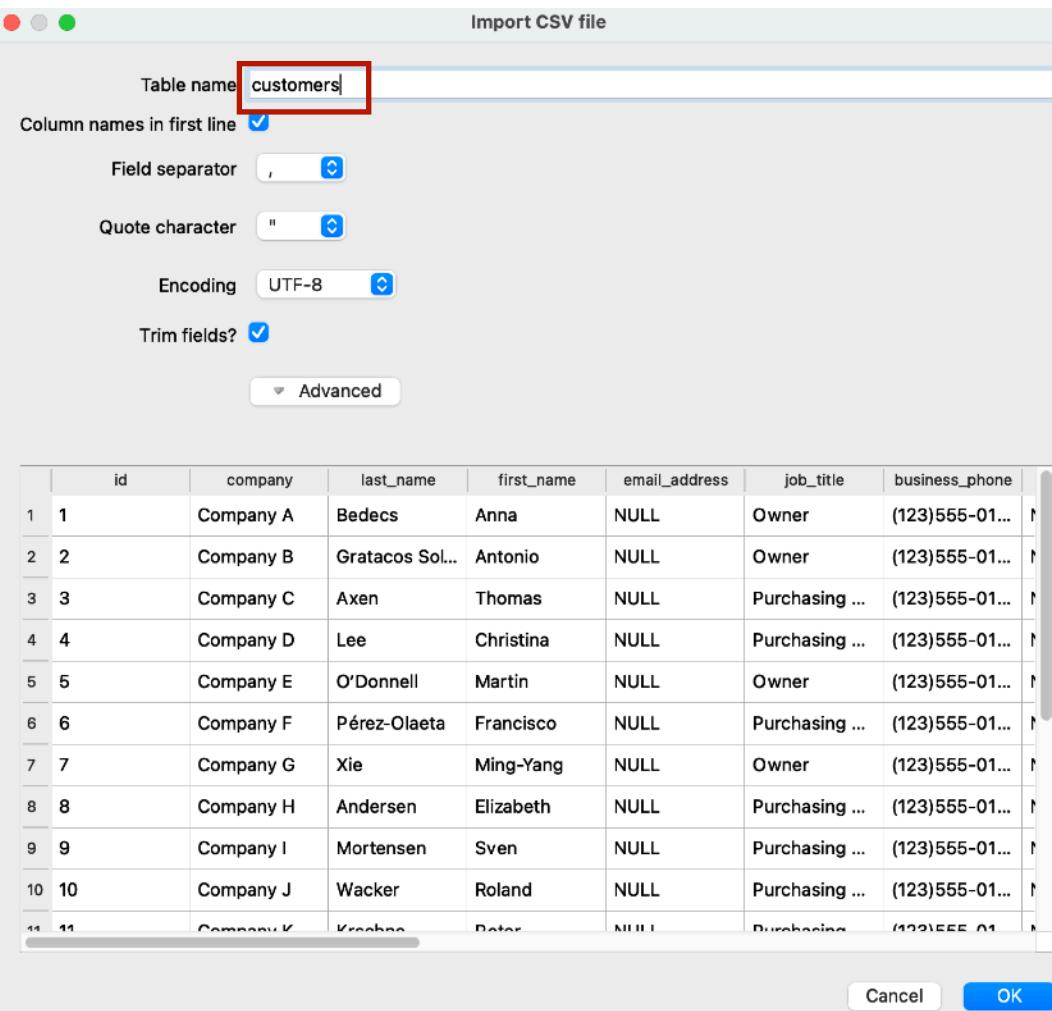
Preparation for big example: added sample data from Northwind.

6 years ago

<https://github.com/graphql-compose/graphql-compose-examples/tree/master/examples/northwind/data/csv>

Import CSV file into DB Browser for SQLite





db

base Close Database

SQL

```
TEXT,"first_name" TEXT,"email_address" TEXT,"job_title" TEXT,"business_phone"
_id" INTEGER,"order_date" TEXT,"shipped_date" TEXT,"shipper_id" TEXT,"ship
plierID" INTEGER,"categoryID" INTEGER,"quantityPerUnit" TEXT,"unitPrice" IN
EXT,"first_name" TEXT,"email_address" TEXT,"job_title" TEXT,"business_phone"
```

Beware of the differences in field names and content between the W3C School example and the downloaded dataset. You need to modify the SQL statements to ensure compatibility when executing the command. For example:

In W3C School, the command is: **SELECT CustomerID, CustomerName** from Customers.

The screenshot shows the W3C School SQL tutorial page. The left sidebar contains a navigation menu with various SQL topics. The main content area is titled "SELECT Syntax" and shows two code snippets: one for selecting specific columns and another for selecting all columns. Below this, a section titled "Demo Database" displays a table from the Northwind sample database with columns: CustomerID, CustomerName, ContactName, Address, City, PostalCode, and Country. The table has five rows of data.

In other words, the table names may be the same, but the structure may differ with different field names.

In SQLite, it becomes: **SELECT ID, Company** from Customers

The screenshot shows an SQLite database interface. On the left, a query window displays the SQL command: "SELECT ID, Company FROM Customers". On the right, a table named "customers" is shown with columns: id, company, last_name, first_name, email_address, job_title, business_phone, home_phone, and mobile_phone. The "id" and "company" columns are highlighted with a red box.

SQL Statement:

```
SELECT CustomerName,City FROM Customers;
```

Edit the SQL Statement, a

Run SQL »

Result:

Number of Records: 91

CustomerName
Alfreds Futterkiste
Ana Trujillo Emparedados
Antonio Moreno Taquería
Around the Horn
Berglunds snabbköp
Blauer See Delikatessen
Blondel père et fils
Bólido Comidas preparadas
Bon app'

CustomerName

City

Execute SQL

DB Browser for SQLite - /Users/suen/Desktop/northwind.db

New Database Open Database Open Project Attach Database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1 SQL 2

1 SELECT company, city FROM customers

	company	city
1	Company A	Seattle
2	Company B	Boston
3	Company C	Los Angeles
4	Company D	New York
5	Company E	Minneapolis
6	Company F	Milwaukee
7	Company G	Boise

Result: 29 rows returned in 11ms
At line 1:
SELECT company, city FROM customers

The screenshot shows the DB Browser for SQLite interface. In the top-left corner, there's a sidebar with a list of customer names. On the right, there's a large central window displaying the results of a SQL query. The query 'SELECT company, city FROM customers' has been run, and the results are shown in a table. The table has three columns: an index (1 to 7), 'company' (Company A through Company G), and 'city' (Seattle, Boston, Los Angeles, New York, Minneapolis, Milwaukee, Boise). Below the table, the status bar indicates 'Result: 29 rows returned in 11ms' and 'At line 1: SELECT company, city FROM customers'.

City
Berlin
México D.F.
México D.F.
London
Luleå
Mannheim
Strasbourg
Madrid
Marseille

SELECT “Distinct” command.

X SQL 1

```
1 SELECT city FROM customers
```

	city
1	Seattle
2	Boston
3	Los Angeles
4	New York
5	Minneapolis
6	Milwaukee
7	Boise
8	Portland
9	Salt Lake City
10	Chicago
11	Miami
12	Las Vegas
13	Memphis
14	Denver
15	Honolulu
16	San Francisco
17	Seattle
18	Boston

Duplicates appear.

Result: 29 rows returned in 6ms
At line 1:
SELECT city FROM customers

X SQL 1

```
1 SELECT DISTINCT city FROM customers
```

	city
1	Seattle
2	Boston
3	Los Angeles
4	New York
5	Minneapolis
6	Milwaukee
7	Boise
8	Portland
9	Salt Lake City
10	Chicago
11	Miami
12	Las Vegas
13	Memphis
14	Denver
15	Honolulu
16	San Francisco

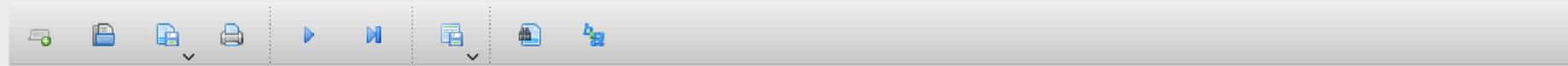
DISTINCT removes the duplicates from display.

Result: 16 rows returned in 15ms

At line 1:
SELECT DISTINCT city FROM customers

Filter Query with “Where” clause.

Database Structure | Browse Data | Edit Pragmas | Execute SQL



× SQL 1 | × SQL 2 | × SQL 3

1 SELECT company, city FROM customers WHERE city = 'Boston'

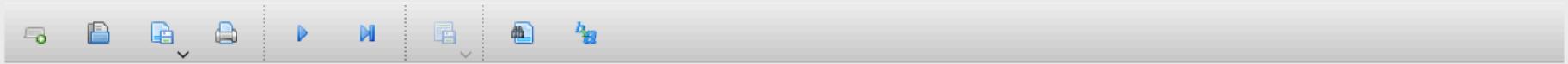
	company	city
1	Company B	Boston
2	Company R	Boston

Result: 2 rows returned in 17ms

At line 1:

SELECT company, city FROM customers WHERE city = 'Boston'

Database Structure | Browse Data | Edit Pragmas **Execute SQL**



× SQL 1 | × SQL 2 | × SQL 3 | **× SQL 4**

1 `SELECT company, city FROM customers WHERE city = 'Boston' AND company = 'Company B'`

	company	city
1	Company B	Boston

Result: 1 rows returned in 3ms

At line 1:

`SELECT company, city FROM customers WHERE city = 'Boston' AND company = 'Company B'`

Use of wildcards using the “Like“ clause.

[]	Represents any single character within the brackets	h[oa]t finds hot and hat, but not hit
^	Represents any character not in the brackets	h[^oa]t finds hit, but not hot and hat
-	Represents any single character within the specified range	c[a-b]t finds cat and cbt

All the wildcards can also be used in combinations!

"Like' is used as wildcard for filling in the blank.

LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	Finds any values that starts with "a"
WHERE CustomerName LIKE '%a'	Finds any values that ends with "a"
WHERE CustomerName LIKE '%or%	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a__%'	Finds any values that starts with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that starts with "a" and ends with "o"

Demo Database

The table below shows the complete "Customers" table from the Northwind sample database:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico



Get started



Play Game

ADVERTISEMENT



Database Structure | Browse Data | Edit Pragmas | Execute SQL

SQL 1 | SQL 2 | SQL 3 | SQL 4 | SQL 5

1 SELECT company, city FROM customers WHERE city LIKE 'Bos%'

	company	city
1	Company B	Boston
2	Company R	Boston

Result: 2 rows returned in 19ms
At line 1:
SELECT company, city FROM customers WHERE city LIKE 'Bos%'

Database Structure | Browse Data | Edit Pragmas | Execute SQL

SQL 1 | SQL 2 | SQL 3 | SQL 4 | SQL 5 | SQL 6

1 SELECT company, city FROM customers WHERE city LIKE '%ton'

	company	city
1	Company B	Boston
2	Company R	Boston

Result: 2 rows returned in 18ms
At line 1:
SELECT company, city FROM customers WHERE city LIKE '%ton'

Database Structure | Browse Data | Edit Pragmas | Execute SQL

SQL 1 | SQL 2 | SQL 3 | SQL 4 | SQL 5 | SQL 6 | SQL 7

1 SELECT company, city FROM customers WHERE city LIKE '%os%'

	company	city
1	Company B	Boston
2	Company C	Los Angelas
3	Company R	Boston
4	Company S	Los Angelas

Result: 4 rows returned in 19ms
At line 1:
SELECT company, city FROM customers WHERE city LIKE '%os%'

Use of the “In” clause.

Database Structure | Browse Data | Edit Pragmas | **Execute SQL**

SQL 1 | SQL 2 | SQL 3 | SQL 4 | SQL 5 | SQL 6 | SQL 7 | **SQL 8**

```
1 SELECT id, company, city, address FROM Customers
2 WHERE City IN ('Boston', 'Los Angelas', 'San Francisco');
```

	id	company	city	address
1	2	Company B	Boston	123 2nd Street
2	3	Company C	Los Angelas	123 3rd Street
3	16	Company P	San Francisco	456 16th Street
4	18	Company R	Boston	456 18th Street
5	19	Company S	Los Angelas	789 19th Street

Result: 5 rows returned in 23ms
At line 1:
SELECT id, company, city, address FROM Customers
WHERE City IN ('Boston', 'Los Angelas', 'San Francisco');

Use of the “Between” clause.

Database Structure | Browse Data | Edit Pragmas **Execute SQL**

X SQL 1 | X SQL 2 | X SQL 3 | X SQL 4 | X SQL 5 | X SQL 6 | X SQL 7 | X SQL 8 | X SQL 9 | X SQL 10 | X SQL 11

1 `SELECT * FROM products WHERE unitPrice BETWEEN 19 AND 20`

	productID	productName	supplierID	categoryID	quantityPerUnit	unitPrice	unitsInStock	unitsOnOrder	reorderLevel	discontinued
1	2	Chang	1	1	24 - 12 oz bottles	19	17	40	25	0
2	36	Inlagd Sill	17	8	24 - 250 g jars	19	112	0	20	0
3	44	Gula Malacca	20	2	20 - 2 kg bags	19.45	27	0	15	0
4	49	Maxilaku	23	3	24 - 50 g pkgs.	20	10	60	15	0
5	57	Ravioli Angelo	26	5	24 - 250 g pkgs.	19.5	36	0	20	0

Result: 5 rows returned in 22ms
At line 1:
`SELECT * FROM products WHERE unitPrice BETWEEN 19 AND 20`

Select of Select Subset

Database Structure | Browse Data | Edit Pragmas | **Execute SQL**

X SQL 1 | X SQL 2 | X SQL 3 | X SQL 4 | X SQL 5 | X SQL 6 | X SQL 7 | X SQL 8 | X SQL 11 | X SQL 12

```
1 SELECT *
2 FROM (SELECT id, Company, city, address FROM customers ORDER BY Company)
3 WHERE city = "Boston"
```

	id	Company	city	address	
1	2	Company B	Boston	123 2nd Street	
2	18	Company R	Boston	456 18th Street	

Result: 2 rows returned in 22ms
At line 1:
SELECT *
FROM (SELECT id, Company, city, address FROM customers ORDER BY Company)
WHERE city = "Boston"

Aggregate Functions

(AVG, MIN, MAX, COUNT, SUM)

DB Browser for SQLite - /Users/suen/Desktop/northwind.db

DB Browser for SQLite - /Users/suen/Desktop/northwind.db

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1 SQL 3

1 SELECT AVG(unitPrice) , productName FROM products

	AVG(unitPrice)	productName
1	28.8663636363636	Chai

Result: 1 rows returned in 18ms
At line 1:
SELECT AVG(unitPrice) , productName FROM products

UTF-8

DB Browser for SQLite - /Users/suen/Desktop/northwind.db

DB Browser for SQLite - /Users/suen/Desktop/northwind.db

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1 SQL 3

1 SELECT ROUND(AVG(unitPrice),2) as "Average Value", productName as "Product Name" FROM products

Average Value	Product Name
28.87	Chai

Result: 1 rows returned in 16ms
At line 1:
SELECT ROUND(AVG(unitPrice),2) as "Average Value", productName as "Product Name" FROM products

Relating Tables with “Join”

(Next Lesson)

Thank you for your time!