iscte

INSTITUTO UNIVERSITÁRIO DE LISBOA

emprego digital

UPskill

Contents

What will this *Code Labs* be about?

- DataBase Management Systems (DBMS)
- Setup working environment in Sqlite
- Northwind Traders
- Databases using Entity Framework Core (EF Core) and Language-Integrated Query (LINQ)

DataBase Management Systems (DBMS)

Information in the Enterprise

Requirements:

- Accuracy
- Correct
- Relevant
- Available
- Readable

Information in the Enterprise

- Good enterprise decisions can only be made if the available information is accurate, correct, relevant, available and readable

    - Data should be arranged, filtered, and grouped in order to be useful
    - If the requirements are not met, the data becomes useless

What is a Database (DB)?

- Related data collection
- Data set that typically describes enterprises’ related activities

What is a Database Management System (DBMS)?

“A DBMS is a software package designed to store and manage databases.”

“A DBMS is software designed to assist in maintaining and utilizing large collections of data.”

“A DBMS is a collection of programs that enables users to create and maintain a database.”

Elmasri & Navathe

What is a Database Management System (DBMS)?

“The DBMS is hence a general-purpose software system that facilitates the process of defining, constructing, manipulating and sharing databases among various users and applications.”

Elmasri & Navathe

A DBMS is a system that

- Stores and manipulates large sets of data
- Specifies data types, structures and restrictions to be applied
- Stores data in a storage medium controlled by the DBMS itself
- Transforms data
- Various users or software modules can access data simultaneously

File Systems vs DBMS

- Information is stored in file systems
  - With time, information tends to grow
  - Information retrieval determines whether or not a DBMS should be used
- Typical file systems can be enough for certain enterprises
- Other may need to use various DBMS

DBMS’ Advantages

- Data independence - global abstract view of data
- Efficient access to data
- Reduced application development time
- Data Integrity and Security
- Data Administration
- Failover and concurrent access

Relational Vs Non Relational Database

- Relational SGBDs:
  - SQLite, SQL Server, MySQL, MariaDB, PostgreSQL
- Non Relational Database:
  - MongoDB, CouchDB, Microsoft Azure Tables, Microsoft Azure DocumentDB
- When to choose one over the other?
  - Scalability, eventual consistency, flexibility

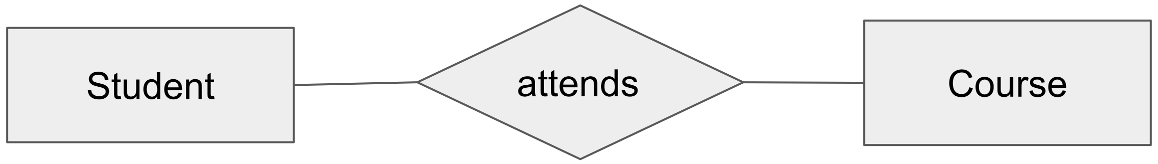
Entity-Relationship Model (ER)

- Proposed by Peter Chen (1976)

- Used to conceptually and graphically **represent data** by modelling the real world using a **predefined set of known patterns**
    - High-level of abstraction
    - Top-down approach
    - Enables communication between project stakeholders

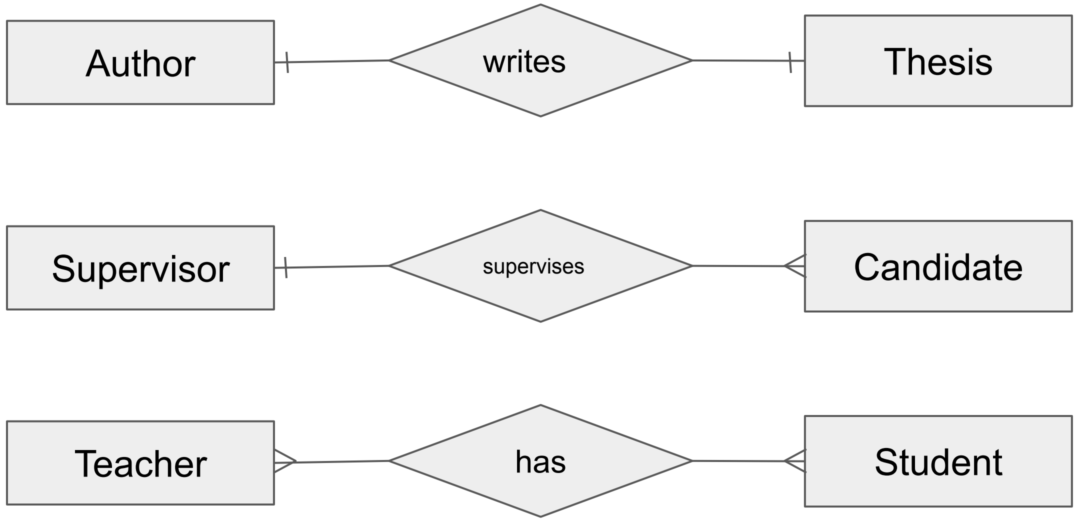
Concepts

- Entity (set of entities)
  - An object or concept that has a known set of characteristics (attributes) that make it different from others
  - Must have a unique identifier
- Relationship (set of relationships)
  - Among entities



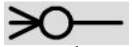
Concepts

- Multiplicity
  - One-to-one
  - One-to-Many
  - Many-to-Many

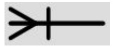


Concepts

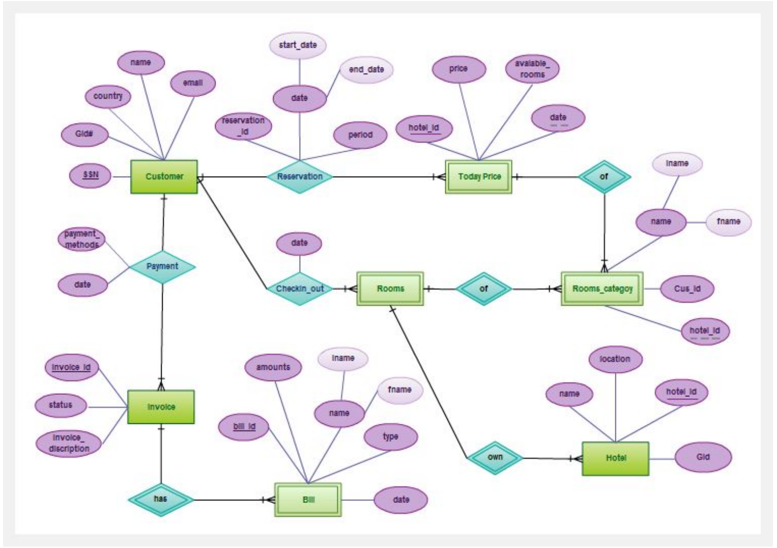
- Connectivity
  - One entity may not be associated with another. Sometimes this association is optional (O sign)



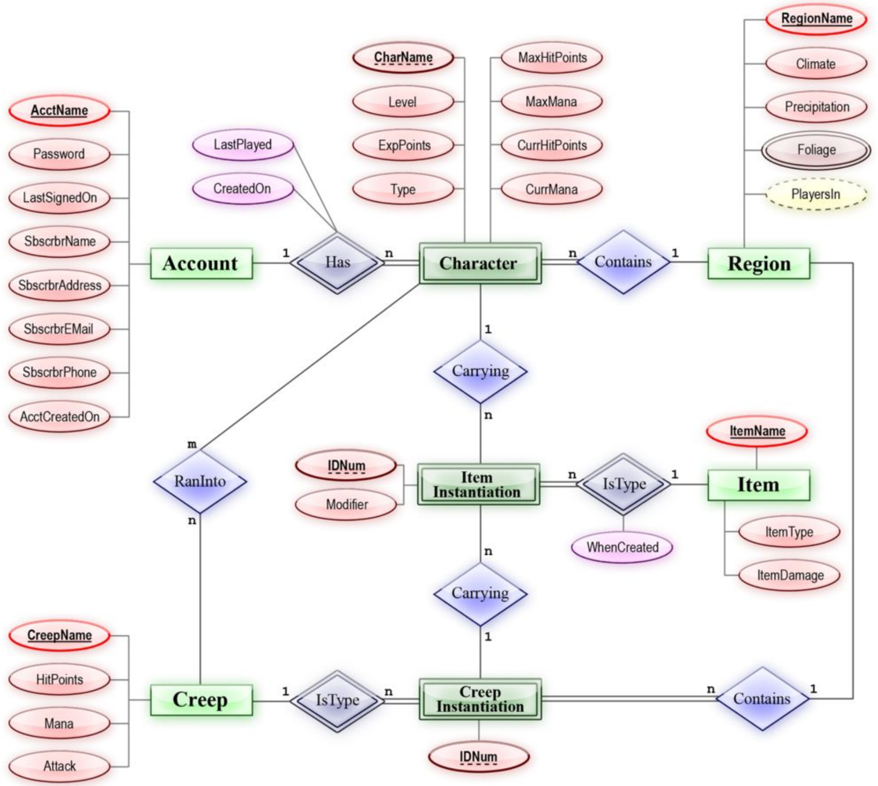
  - If, otherwise, the entity set must have at least one object use the | sign



Notation and Relationship



https://creately.com



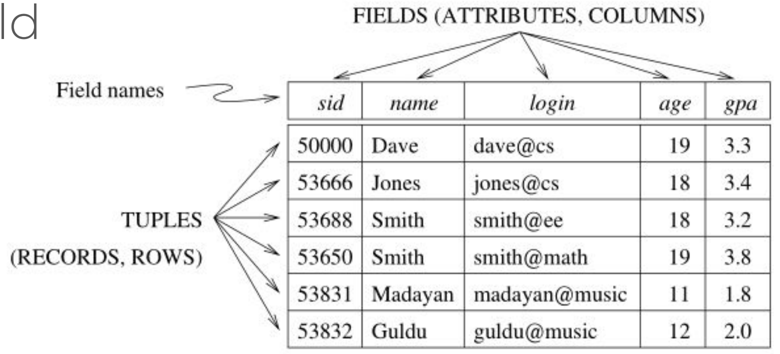
https://wikipedia.com

Relational Model

- Proposed by E.F. Codd (1970)
- Based on
  - Set Theory (has a strong theoretical basis)
  - Abstract concept of relationships
- Defines which operations can be applied to relationships
- Independent of the physical model

Relational Model - main elements

- Fields
- Domain
  - set of possible values for each field
- Relationship



sid	name	login	age	gpa
50000	Dave	dave@cs	19	3.3
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

from [Ramakrishnan, 2002]

Key constraints

- Candidate Key
  - Set of fields that uniquely identify a tuple
  - E.g., Student ID (*sid*)
- Out of all Candidate Keys, identify a Primary Key (PK)

SQLite

- SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine
- Small, cross-platform, self-contained RDBMS that is available in the public domain
- The most common RDBMS for mobile platforms such as iOS (iPhone and iPad) and Android
- SQLite Home Page

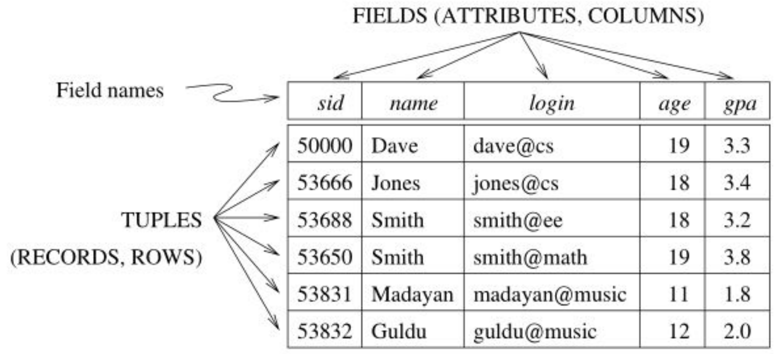
Appropriate Uses For SQLite

- SQLite is not directly comparable to client/server SQL database engines such as MySQL, Oracle, PostgreSQL, or SQL Server
- SQLite strives to provide local data storage for individual applications and devices
- SQLite does not compete with client/server databases
- See: Situations where SQLite works well

Physical Model (SQLite)

- CREATE TABLE Students (sid CHAR(20),
  name CHAR(30),
  login CHAR(20),
  age INTEGER,
  gpa REAL)

- INSERT
  INTO Students (sid, name, login, age, gpa)
  VALUES ('53688', 'Smith', 'smith@ee', 18, 3.2)



sid	*name*	*login*	*age*	*gpa*
50000	Dave	dave@cs	19	3.3
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

from [Ramakrishnan, 2002]

Key constraints

- Sqlite:

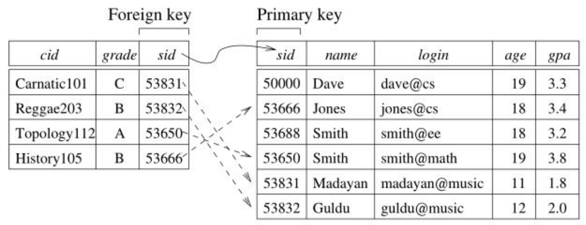
  -
    ```
    CREATE TABLE Students ( sid CHAR(20),
                            name CHAR(30),
                            login CHAR(20),
                            age INTEGER,
                            gpa REAL,
                            UNIQUE (name, age),
                            CONSTRAINT StudentsKey PRIMARY KEY (sid) )
    ```

Key constraints

- Foreign Key
  - Sometimes the information stored in a relation is linked to the information stored in another relation.

- Sqlite:

```
CREATE TABLE Enrolled (   sid CHAR(20),
                          cid CHAR(20),
                          grade CHAR(10),
                          PRIMARY KEY (sid, cid),
                          FOREIGN KEY (sid) REFERENCES Stuc
```



cid	*grade*	*sid*
Carnatic101	C	53831
Reggae203	B	53832
Topology112	A	53650
History105	B	53666

sid	*name*	*login*	*age*	*gpa*
50000	Dave	dave@cs	19	3.3
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

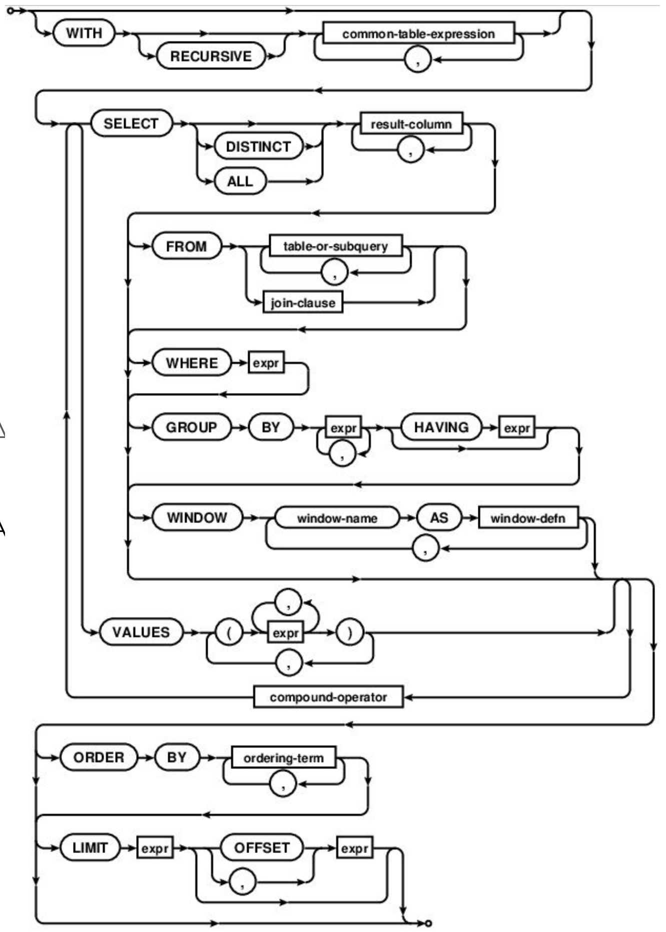
Enrolled (Referencing relation)

Students (Referenced relation)

SELECT SQL statement

- SELECT * FROM Students
- SELECT NAME, AGE FROM Students WHERE AGE > 15
- SELECT * FROM Students WHERE AGE > 15 GROUP BY GPA
- SELECT * FROM Students WHERE AGE > 15 ORDER BY GPA

https://www.sqlite.org/lang_select.html#overview

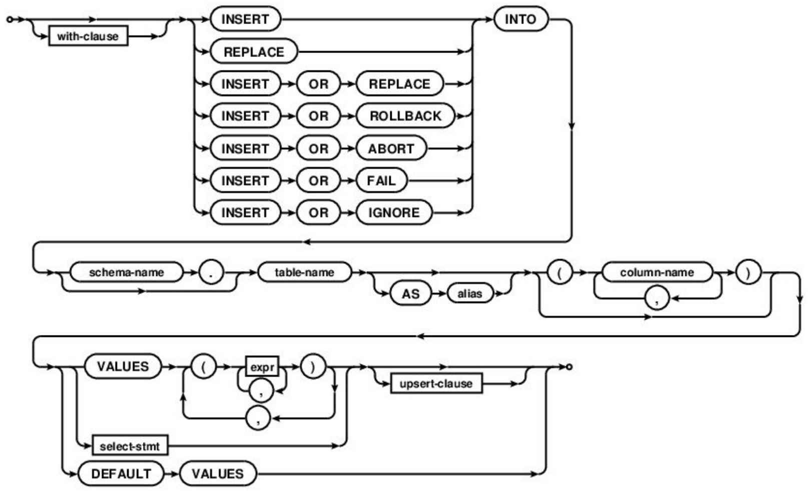


INSERT SQL statement

- INSERT

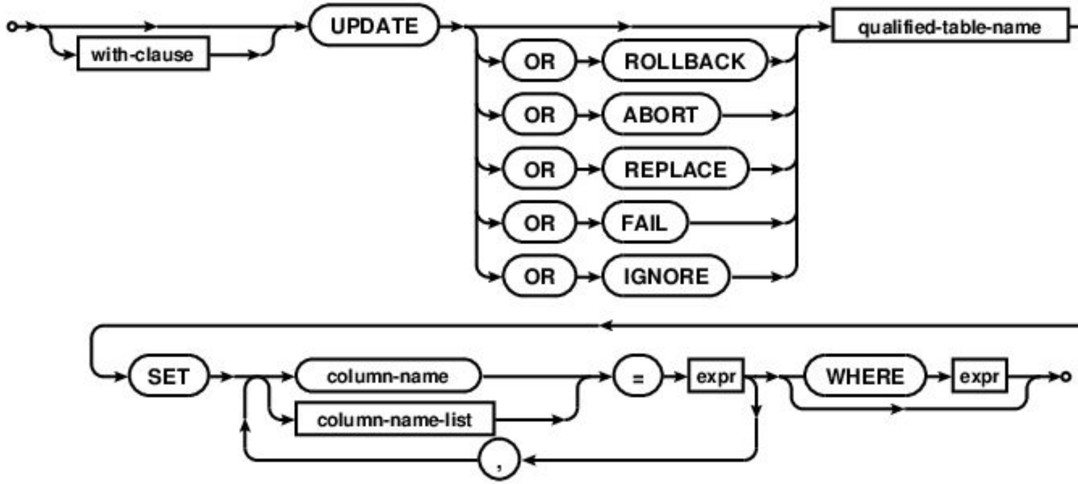
  INTO Students (sid, name, login, age, gpa)

  VALUES ('53688', 'Smith', 'smith@ee', 18, 3.2)



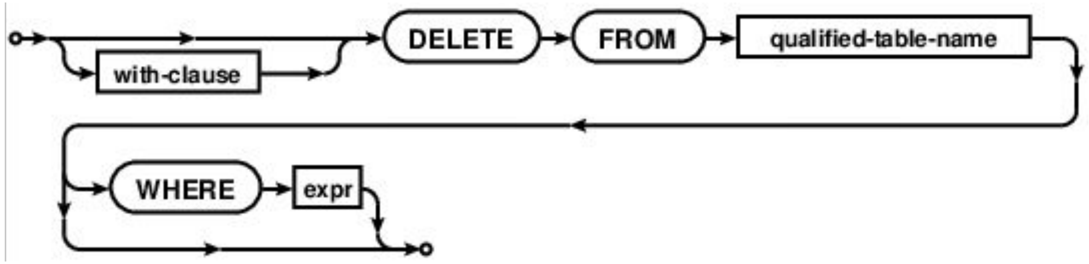
UPDATE SQL statement

- UPDATE STUDENTS SET GPA = 3 WHERE SID = 53688



DELETE SQL statement

- DELETE FROM STUDENTS WHERE SID = 53688

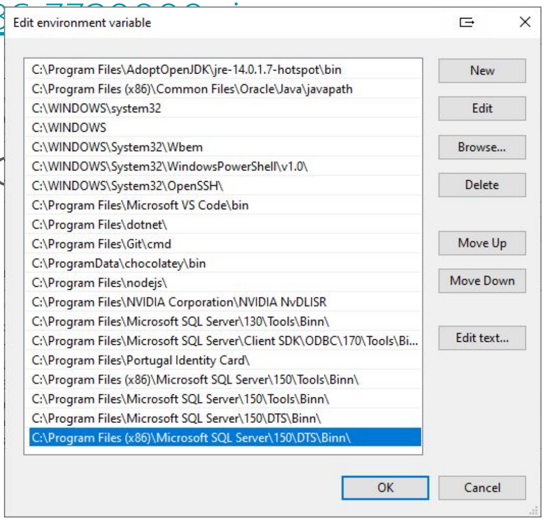


Setup working environment

Sqlite and DBBrowser

Sqlite command line tools

- Download https://www.sqlite.org/2020/sqlite-tools-win32-x8
- Unzip to a known location
- (Optionally) Avoid changing directories by c
  environment variable
  - Win+pause
  - Advanced System Settings
  - Environment Variables
  - Path



```
C:\Program Files\AdoptOpenJDK\jre-14.0.1.7-hotspot\bin
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
C:\WINDOWS\system32
C:\WINDOWS
C:\WINDOWS\System32\Wbem
C:\WINDOWS\System32\WindowsPowerShell\v1.0\
C:\WINDOWS\System32\OpenSSH\
C:\Program Files\Microsoft VS Code\bin
C:\Program Files\dotnet\
C:\Program Files\Git\cmd
C:\ProgramData\chocolatey\bin
C:\Program Files\nodejs\
C:\Program Files\NVIDIA Corporation\NVIDIA NvDLISR
C:\Program Files\Microsoft SQL Server\130\Tools\Binn\
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\170\Tools\Bi...
C:\Program Files\Portugal Identity Card\
C:\Program Files (x86)\Microsoft SQL Server\150\Tools\Binn\
C:\Program Files\Microsoft SQL Server\150\Tools\Binn\
C:\Program Files\Microsoft SQL Server\150\DTS\Binn\
C:\Program Files (x86)\Microsoft SQL Server\150\DTS\Binn\
```

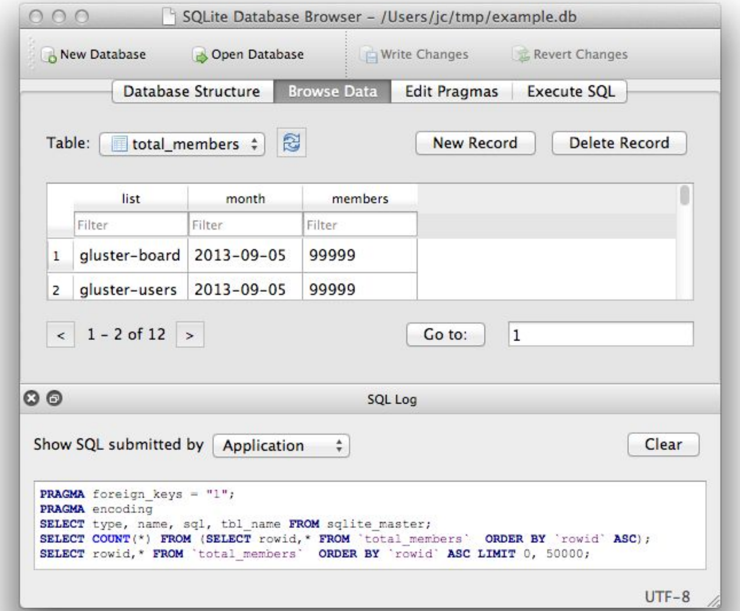
Install DBBrowser

https://download.sqlitebrowser.org/

DB Browser for SQLite

The Official home of the DB Browser for SQLite

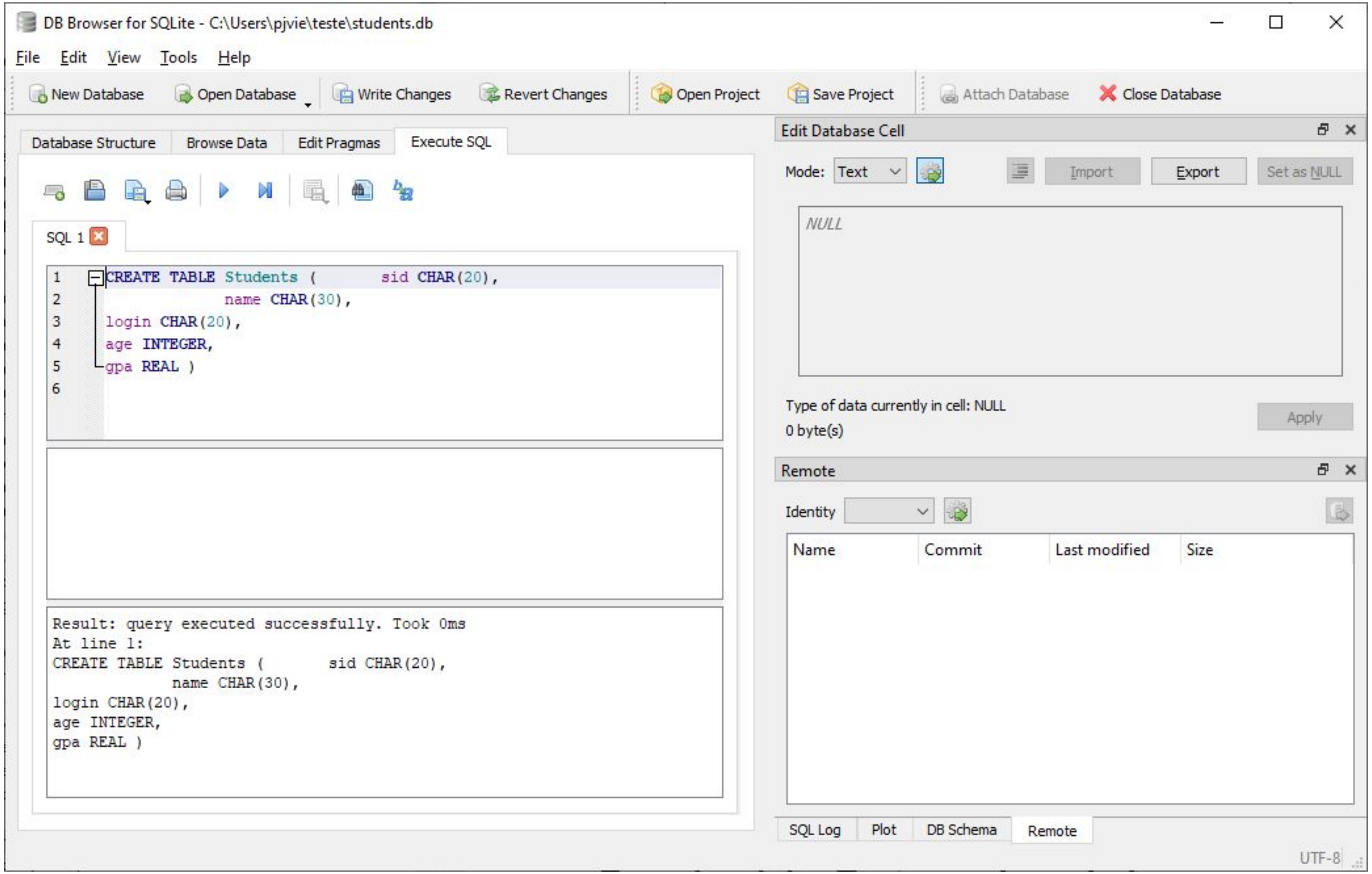
Screenshot



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

< 1 - 2 of 12 >   Go to: 1

```
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;
```



```
CREATE TABLE Students (        sid CHAR(20),
                       name CHAR(30),
login CHAR(20),
age INTEGER,
gpa REAL )
```

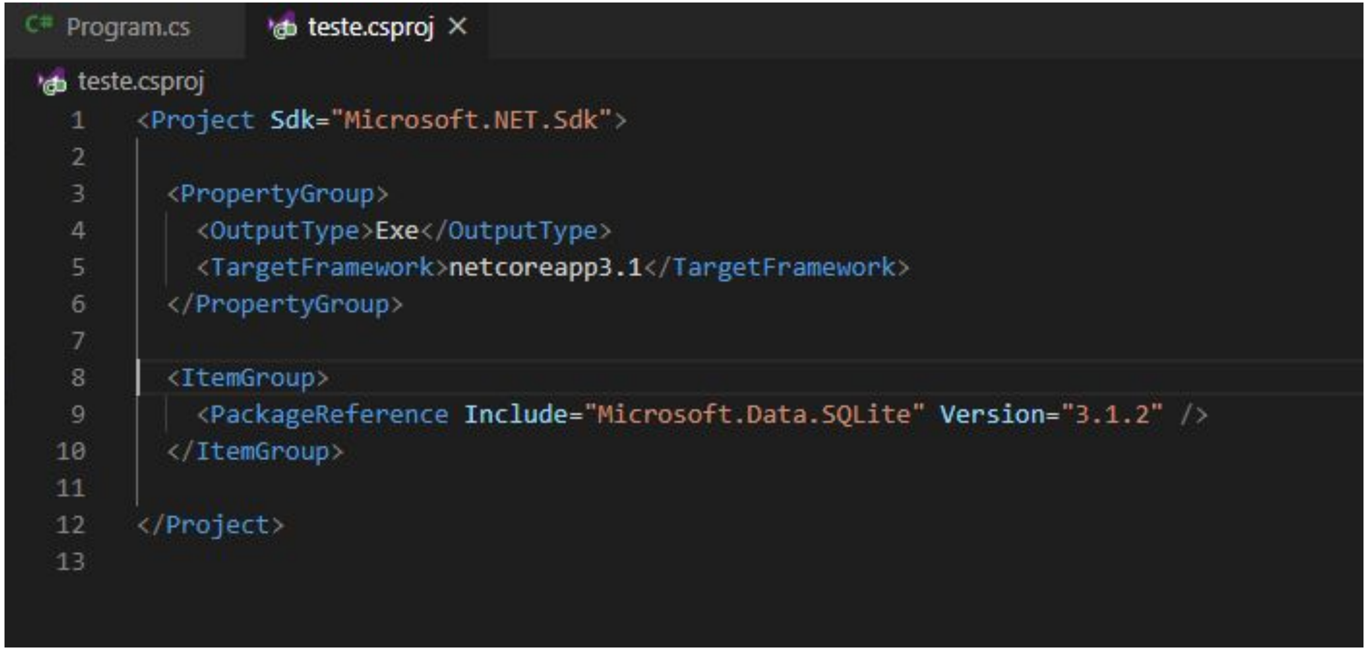
Type of data currently in cell: NULL
0 byte(s)

```
Result: query executed successfully. Took 0ms
At line 1:
CREATE TABLE Students (        sid CHAR(20),
             name CHAR(30),
login CHAR(20),
age INTEGER,
gpa REAL )
```

Microsoft.Data.Sqlite

https://www.nuget.org/packages/Microsoft.Data.Sqlite/

- dotnet add package Microsoft.Data.Sqlite



```xml
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>netcoreapp3.1</TargetFramework>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference Include="Microsoft.Data.SQLite" Version="3.1.2" />
  </ItemGroup>

</Project>

```

Exemplo: ligação BD

```
using System;

using Microsoft.Data.Sqlite;

namespace teste{

    class Program    {

        static void Main(string[] args){

            String dbName = "students.db";

            using (SqliteConnection db = new SqliteConnection($"Filename={dbName};")){

                db.Open();

                SqliteCommand selectCommand = new SqliteCommand("SELECT name from Students", db);

                SqliteDataReader query = selectCommand.ExecuteReader();

                while (query.Read()){

                    Console.WriteLine($"Hello {query.GetString(0)}!");
```

Northwind Traders

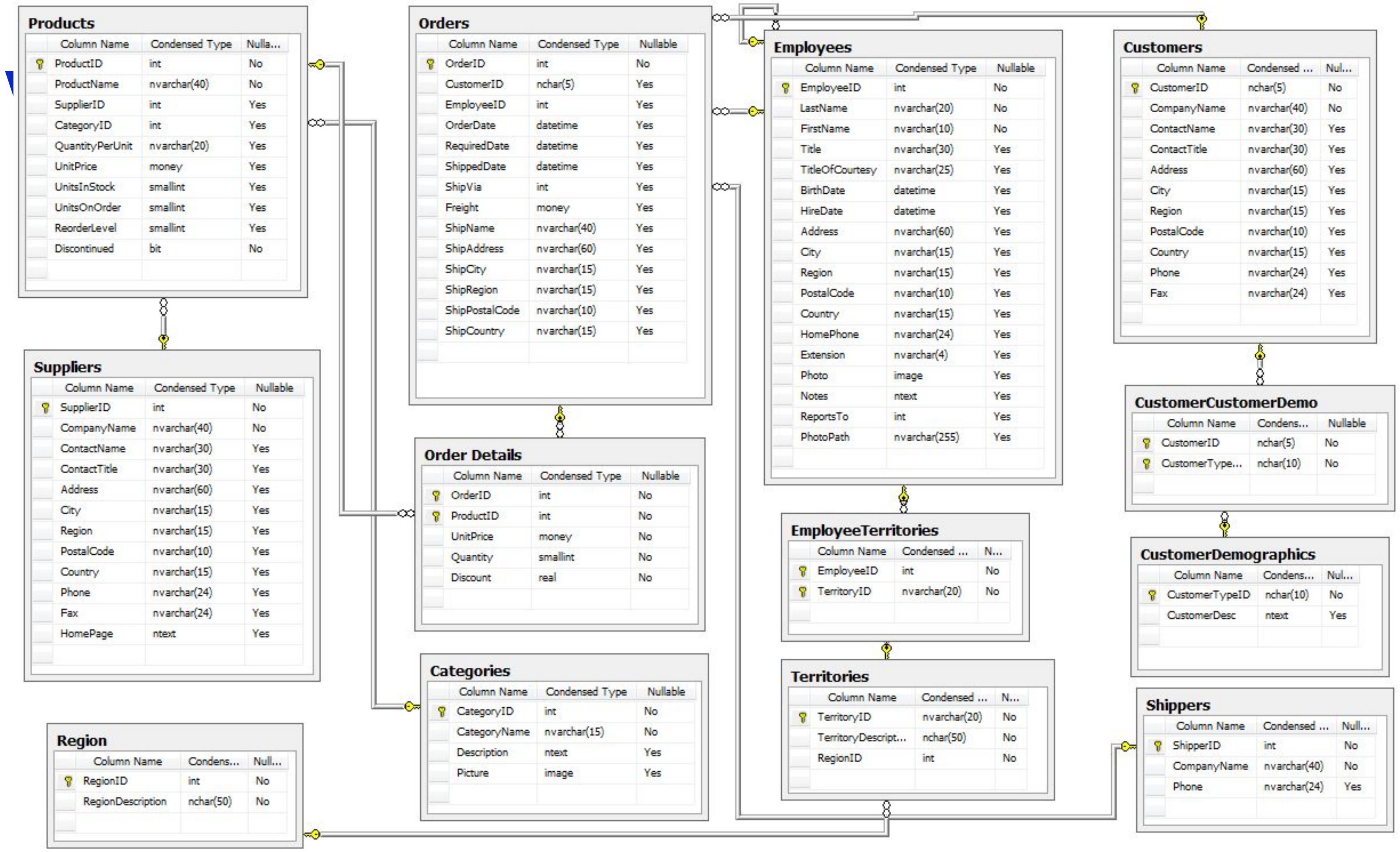
Relational data sample

Working with Northwind

- Microsoft Access launched with Northwind Traders as its main sample database to showcase the power of Access
- Great teaching tool because it has a very simple and relatable schema

NORTHWIND
TRADERS

Northwind Traders relational data sample



Products

Column Name	Condensed Type	Nulla...
ProductID	int	No
ProductName	nvarchar(40)	No
SupplierID	int	Yes
CategoryID	int	Yes
QuantityPerUnit	nvarchar(20)	Yes
UnitPrice	money	Yes
UnitsInStock	smallint	Yes
UnitsOnOrder	smallint	Yes
ReorderLevel	smallint	Yes
Discontinued	bit	No

Orders

Column Name	Condensed Type	Nullable
OrderID	int	No
CustomerID	nchar(5)	Yes
EmployeeID	int	Yes
OrderDate	datetime	Yes
RequiredDate	datetime	Yes
ShippedDate	datetime	Yes
ShipVia	int	Yes
Freight	money	Yes
ShipName	nvarchar(40)	Yes
ShipAddress	nvarchar(60)	Yes
ShipCity	nvarchar(15)	Yes
ShipRegion	nvarchar(15)	Yes
ShipPostalCode	nvarchar(10)	Yes
ShipCountry	nvarchar(15)	Yes

Employees

Column Name	Condensed Type	Nullable
EmployeeID	int	No
LastName	nvarchar(20)	No
FirstName	nvarchar(10)	No
Title	nvarchar(30)	Yes
TitleOfCourtesy	nvarchar(25)	Yes
BirthDate	datetime	Yes
HireDate	datetime	Yes
Address	nvarchar(60)	Yes
City	nvarchar(15)	Yes
Region	nvarchar(15)	Yes
PostalCode	nvarchar(10)	Yes
Country	nvarchar(15)	Yes
HomePhone	nvarchar(24)	Yes
Extension	nvarchar(4)	Yes
Photo	image	Yes
Notes	ntext	Yes
ReportsTo	int	Yes
PhotoPath	nvarchar(255)	Yes

Customers

Column Name	Condensed ...	Nul...
CustomerID	nchar(5)	No
CompanyName	nvarchar(40)	No
ContactName	nvarchar(30)	Yes
ContactTitle	nvarchar(30)	Yes
Address	nvarchar(60)	Yes
City	nvarchar(15)	Yes
Region	nvarchar(15)	Yes
PostalCode	nvarchar(10)	Yes
Country	nvarchar(15)	Yes
Phone	nvarchar(24)	Yes
Fax	nvarchar(24)	Yes

Suppliers

Column Name	Condensed Type	Nullable
SupplierID	int	No
CompanyName	nvarchar(40)	No
ContactName	nvarchar(30)	Yes
ContactTitle	nvarchar(30)	Yes
Address	nvarchar(60)	Yes
City	nvarchar(15)	Yes
Region	nvarchar(15)	Yes
PostalCode	nvarchar(10)	Yes
Country	nvarchar(15)	Yes
Phone	nvarchar(24)	Yes
Fax	nvarchar(24)	Yes
HomePage	ntext	Yes

Order Details

Column Name	Condensed Type	Nullable
OrderID	int	No
ProductID	int	No
UnitPrice	money	No
Quantity	smallint	No
Discount	real	No

CustomerCustomerDemo

Column Name	Condens...	Nullable
CustomerID	nchar(5)	No
CustomerType...	nchar(10)	No

EmployeeTerritories

Column Name	Condensed ...	N...
EmployeeID	int	No
TerritoryID	nvarchar(20)	No

CustomerDemographics

Column Name	Condens...	Nul...
CustomerTypeID	nchar(10)	No
CustomerDesc	ntext	Yes

Categories

Column Name	Condensed Type	Nullable
CategoryID	int	No
CategoryName	nvarchar(15)	No
Description	ntext	Yes
Picture	image	Yes

Territories

Column Name	Condensed ...	N...
TerritoryID	nvarchar(20)	No
TerritoryDescript...	nchar(50)	No
RegionID	int	No

Shippers

Column Name	Condensed ...	Null...
ShipperID	int	No
CompanyName	nvarchar(40)	No
Phone	nvarchar(24)	Yes

Region

Column Name	Condens...	Null...
RegionID	int	No
RegionDescription	nchar(50)	No

Try it

- Open “Northwind_small.sqlite” in DBBrowser
- Get a testing copy, to play around
  - sqlite3 MyTestDB < Northwind.Sqlite3.create.sql
- Get a report detailing size and storage efficiency
  - sqlite3_analyzer MyTestDB.sqlite > report.txt
- Output SQL text that would transform DB1 into DB2
  - sqldiff MyTestDB.sqlite Northwind_small.sqlite > differences.txt

Questions about Northwind

1. How many tables?
2. How many customers?
    a. How many customers are from Berlin?
3. How many regions and which?
4. Which is Laura’s Surname and title of courtesy, knowing that Laura works for Northwind?
5. How old is Northwind’s sales manager?
6. Retrieve all columns in the Region table

SQLite: datetime Function

Questions about Northwind

7. Select the FirstName and LastName columns from the Employees table

8. Select the FirstName and LastName columns from the Employees table. Sort by LastName

9. Create a report showing Northwind's orders sorted by Freight from most expensive to cheapest. Show OrderID, OrderDate, ShippedDate, CustomerID, and Freight

10. Create a report showing the title and the first and last name of all sales representatives

Entity Framework Core

Entity Framework Core

- Object-relational mapping (ORM) technology that is designed to work with data stored in relational databases such as SQLite, Oracle, and Microsoft SQL Server

- Supports modern cloud-based, nonrelational, schema-less data stores, such as Microsoft Azure Cosmos DB and MongoDB, sometimes with third-party providers.

Entity Framework Core

- EF Core uses a combination of conventions, annotation attributes, and Fluent API statements to build an entity model at runtime
    - any actions performed on the classes are automatically translated into actions performed on the actual database

EF Core conventions

- The name of a table is assumed to match the name of a DbSet<T> property in the DbContext class
- The names of the columns are assumed to match the names of properties in the class
- The string .NET type is assumed to be a nvarchar type in the database
- The int .NET type is assumed to be an int type in the database

Querying Data

- Entity Framework Core uses Language Integrated Query (LINQ) to query data from the database
- LINQ allows you to use C# to write strongly typed queries
    - Uses your derived context and entity classes to reference database objects
    - EF Core passes a representation of the LINQ query to the database provider
    - Database providers in turn translate it to database-specific query language (for example, SQL for a relational database)

https://docs.microsoft.com/en-us/ef/core/querying/

Explore LINQ

- Basic LINQ Query Operations (C#)
    - Walkthrough: Writing Queries in C# (LINQ)

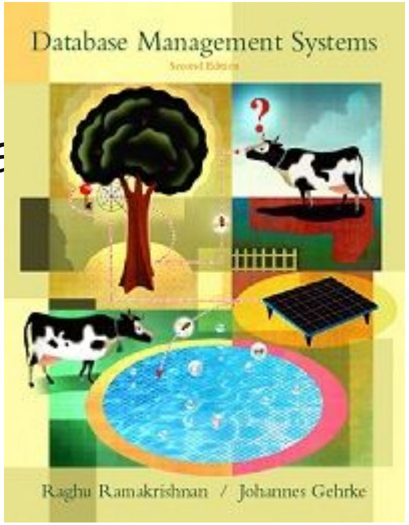
- LINQPad - The .NET Programmer's Playground

References

- Ramakrishnan, R., & Gehrke, J. (2002). *Database ma*
  systems. Boston: McGraw-Hill.

- https://www.sqlite.org/docs.html

- https://docs.microsoft.com/en-us/ef/core/


Database Management Systems
Second Edition
Raghu Ramakrishnan / Johannes Gehrke