Databases for data analytics

https://github.com/evidencebp/databases-course/

Introduction

Goals

- Students will learn to use SQL for data science.
- Students will know how to build a database that will fit their business needs.
- Students will know how to evaluate and protect the data integrity.
- Students will learn basic performance for analytics.
- Students will build a recommendation system to improve analysis skills.

Out of scope

- Database administration (e.g., physical layer, architectures), advanced database programming (e.g., transactions,concurrency), NOSQL databases, and much more.

Databases store data

A little hand waving with theoretical definitions

- **Data** (/ˈdeɪtə/ *DAY-tə*, US also /ˈdætə/ *DAT-ə*) are a collection of discrete or continuous values that convey information, describing the quantity, quality, fact, statistics, other basic units of meaning, or simply sequences of symbols that may be further interpreted formally.
- In computer science, **data** (treated as singular, plural, or as a mass noun) is any sequence of one or more symbols; **datum** is a single unit of data. Data requires interpretation to become information.
- **Information** is an abstract concept that refers to something which has the power to inform.

Concrete examples:

- Customer's address
- The year in which a movie was presented
- List of actors in a movie

Deeper considerations regarding data: data characteristics

- Intended uses (e.g., customer address might be needed for operational vs. analytical need)
- Reliability
- Semantics (consider the various meaning of having an address in Tel Aviv).
- Related entities
- Data type

Data types

- Common types
  - Sequence of bits
  - Characters strings
  - Numbers (integer, float)
  - Dates
  - A set of values (a non negative price, cities in Israel)
- The more specific type we use, we gain semantics, protection, and related operations

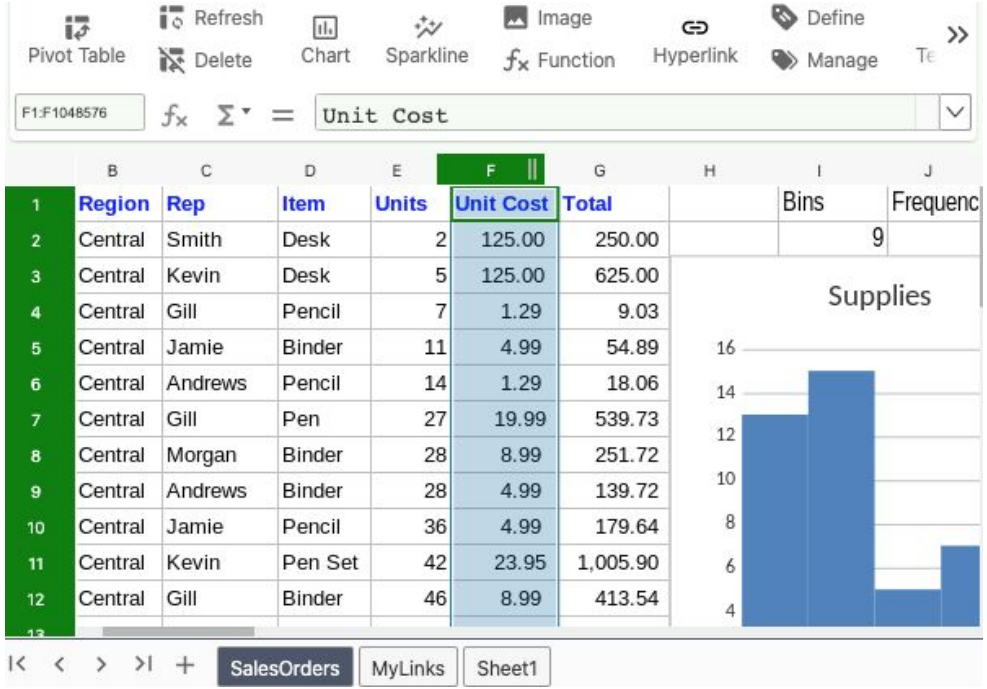
Database Management Systems are big software

- What's the purpose of a DBMS –
  - store data, supports multiple tables
  - provide utilities for - maintaining, accessing, updating/manipulating data
- SQLite, 29K commits
- MySql, 184K commits
- Oracle, 1977 funded, 465.19 billion USD, 159,000 employees company.
- Plenty DBMS exist (E.g. Sql Server, BigQuery, Postgres, Analytics DBs)
- Most DBMS use SQL as interface
- Even those that do not use SQL are called NoSQL DBs (e.g., key-value, graph).

Excel vs. databases

Excel files can

- Have colors
- Have graphs
- Be sent in email
- Various sheets
- No strict structure
- Readable format (csv)



Benefits of Databases - table creation

```
CREATE TABLE movies (
  `id` varchar(100),
  `name` varchar(100),
  `year` varchar(100),
  `rank` varchar(100)
);
```

	id	name	year	rank
▶	1	Superman	BaShana Habaa	10

Benefits of Relational Databases - strict data types

```
CREATE TABLE movies (
  `id` int,
  `name` varchar(100),
  `year` int,
  `rank` float
);
```

	id	name	year	rank
▶	NULL	Superman	2000	10

Benefits of Relational Databases - Constraints

```
CREATE TABLE movies (
  `id` int NOT NULL, # Not allowing nulls
  `name` varchar(100) NOT NULL,
  `year` int,
  `rank` float
);
```

	id	name	year	rank
▶	1	Superman	-300	10

Benefits of Relational Databases - range constraint

```
CREATE TABLE movies (
  `id` int NOT NULL,
  `name` varchar(100) NOT NULL,
  `year` int,
  `rank` float,
  CHECK (year>=1900)
);
```

Benefits of Relational Databases - setting default values

```
CREATE TABLE movies (
  `id` int NOT NULL,
  `name` varchar(100) NOT NULL,
  `year` int,
  `rank` float DEFAULT NULL,
  CHECK (year>=1900)
);
```

Benefits of Relational Databases - using keys

```
CREATE TABLE movies (
  `id` int NOT NULL,
  `name` varchar(100) NOT NULL,
  `year` int,
  `rank` float DEFAULT NULL,
  CHECK (year>=1900),
  PRIMARY KEY (`id`)
);
```

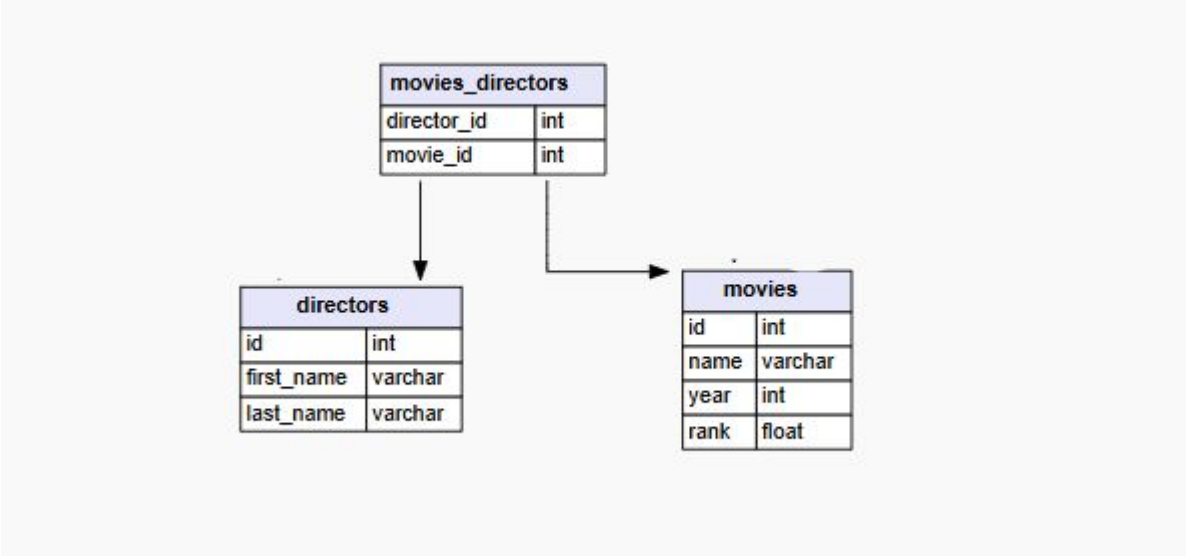
Benefits of Relational Databases - indices

```
CREATE TABLE movies (
  `id` int NOT NULL,
  `name` varchar(100) NOT NULL,
  `year` int,
  `rank` float DEFAULT NULL,
  CHECK (year>=1900),
  PRIMARY KEY (`id`),
  KEY `movies_name` (`name`) # Faster search
);
```

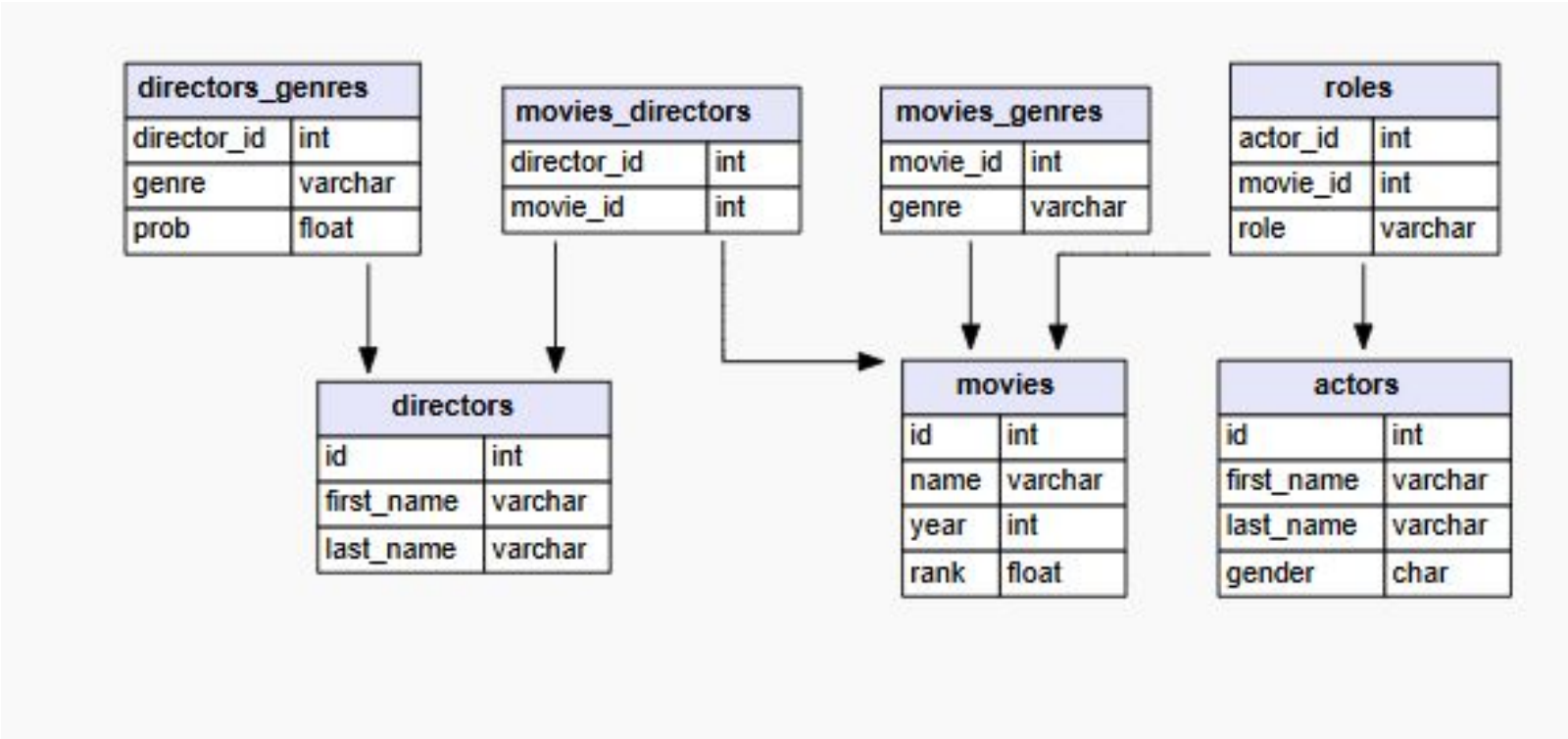
Database benefits - multiple entities in a single table

```
CREATE TABLE movies (
  `id` int NOT NULL,
  `name` varchar(100) NOT NULL,
  `year` int,
  `rank` float DEFAULT NULL,
  `director` varchar(255),
  CHECK (year>=1900),
  PRIMARY KEY (`id`),
  KEY `movies_index` (`name`)
);
```

Benefits of Relational Databases - Directors



Benefits of Relational Databases - Multiple tables



More benefits

- Handling large volumes (e.g., distribution)
- High performance
- Data integrity
- Security
- Multiple users support
- Transaction
- Backup and recovery
- Scaling

The power of SQL

- A common flexible data interface language - easy query and manipulation
- SEQUEL: A Structured English Query Language, 1974
  - C was created in 1972, C++ in 1985, Python in 1991, Java in 1995
- Based on set theory, making it elegant and powerful for analytics

SQL makes it easy to answer complex questions like

- Distribution of movies by the number of actors
- Pairs of movies with at least 3 common actors
- Does a developer have more bugs than his usual in a more buggy project? (environment influence)
- Does an increase in cholesterol increases stroke risk?

Exercise 1

- Install MySql Workbench
- Install IMDB dataset
- Run “select count(*) from imdb_ijs.movies;” and see the you get 388,269
- No need to submit but please do it since it will be required for the next lessons.