

Spring 2024 CSE 380

1-16-2024

Office Hours Update

- Abigail's office hours all Zoom (this week only)
 - Mondays: 10:30-12:00pm (normally in-person)
 - Wednesdays: 10:30-12:00pm (normally in-person)
 - Fridays: 11:30-12:30pm (always Zoom)
 - Abigail's Zoom link is on Piazza
- Professor Mariani's office hours (always 3145 EB)
 - Mondays: 1:30pm-3:00pm
 - Wednesdays: 2:00pm-3:30pm

Quiz Reminder

- We have our first quiz exactly 1 week from today (January 23rd)
- Please remember, no phones, cheatsheets, computers, etc. of any kind
- Make sure you're studying and practicing

Introduction to SQL

- In the past, databases mostly held boring stuff: census data, employee records, bank accounts, ...
- Today, databases hold virtually all of humankind's collected information
 - Everything you've ever read, seen, or done (according to Snowden)
 - Everything on the web Scientific data
 - Medical data
 - Etc.

Introduction to SQL

- Databases use restricted programming languages
 - One of the few places where non-Turing-complete languages are an advantage
 - Easier to optimize
 - Faster/Easier to write
 - This leads to very terse programming and deep query-optimization problems
 - I could probably teach you every SQL command in 2 lectures total
- Concurrency and Transactions are essential
 - Many activities happen simultaneously
 - Think on the scale of YouTube or Google
 - Need to heavily utilize parallel programming techniques
 - Much more powerful than a flat file

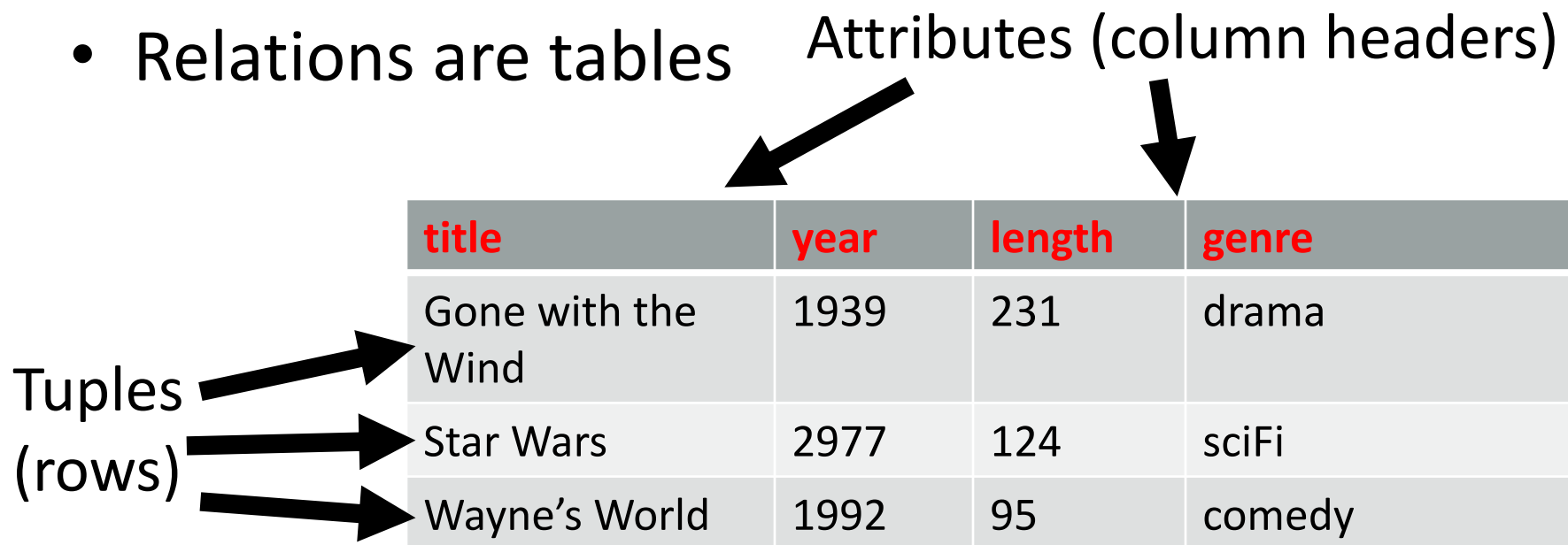
Introduction to SQL

- Data Models – Components
 - Mathematical representation of the data
 - Relational Models (CSV) = tables
 - Semi-structured Models (XML, JSON) = trees/graphs
 - Operations on data (what is allowed, i.e. comparisons, equality, math)
 - Constraints (what types of data are allowed, and where)

Introduction to SQL

- Relations are tables

Attributes (column headers)



| title | year | length | genre |
|--------------------|------|--------|--------|
| Gone with the Wind | 1939 | 231 | drama |
| Star Wars | 2977 | 124 | sciFi |
| Wayne's World | 1992 | 95 | comedy |

Tuples (rows)

Relation
name → **Movies**

Introduction to SQL

- Schemas (sound familiar?)
 - Relation schema = relation name and attribute list
 - Optionally: types of attributes
 - Example: Movies (title, year, length genre) or Movies (title TEXT, year INTEGER, length INTEGER, genre TEXT)
 - Database = collection of relations
 - Database schema = set of all relation schemas in the database

Example Database Schema

```
Movies(  
    title TEXT,  
    year INTEGER,  
    length INTEGER,  
    genre TEXT  
)  
MovieStar(  
    name TEXT,  
    address TEXT,  
    birthdate DATE  
)  
StarsIn(  
    movieTitle TEXT,  
    movieYear INTEGER,  
    starName TEXT  
)  
Studio(  
    name TEXT,  
    address TEXT  
)
```

Relation Name on
its own line

Attribute names
and types comma
separated and each
on new line

Introduction to SQL

- Why Relations (tables)?
 - Very simple model
 - Often (but not always) matches how we think about data
 - Abstract model that underlies SQL, the most important database language today

Introduction to SQL

- Database Schemas in SQL
 - SQL is primarily a query language, for getting information from a database.
 - But SQL also includes a *data-definition* component for describing database schemas.

Basic SQL Querying

- Creating (Declaring) a Relation

- Simplest form:

```
CREATE TABLE <name> (  
    <list of elements>  
);
```

Basic SQL Querying

- Elements of Table Declarations
 - Most basic element: an attribute and its type
 - The SQLite types are:
 - INTEGER
 - REAL
 - TEXT
 - BLOB (Binary Large Object)

Basic SQL Querying

- Example: Create Table

```
CREATE TABLE Movies (  
    title TEXT,  
    year INTEGER,  
    length INTEGER,  
    genre TEXT  
);
```

Basic SQL Querying

- SQL Values
 - Integers and real values (floats) are represented as you would expect
 - Strings are as well, except they require single quotes
 - Two single quotes = one real quote
 - 'Joe''s Bar' = Joe's Bar
 - Any value can be NULL
 - For now, we will learn later how this can be prevented

Basic SQL Querying

- Insertion
 - To insert a single record:
`INSERT INTO <relation>`
`VALUES (<list of values>);`
 - Example: add Star Wars to our Movies relation
`INSERT INTO Movies`
`VALUES ('Star Wars', 1977, 124, 'sciFi');`

Basic SQL Querying

- Select
 - Using our Movies relation, what movies have been released?

SELECT title

FROM Movies;

Column we want to
select


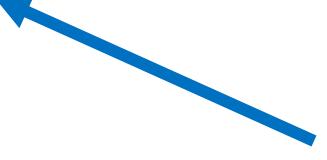


Table we want to
select from



Basic SQL Querying

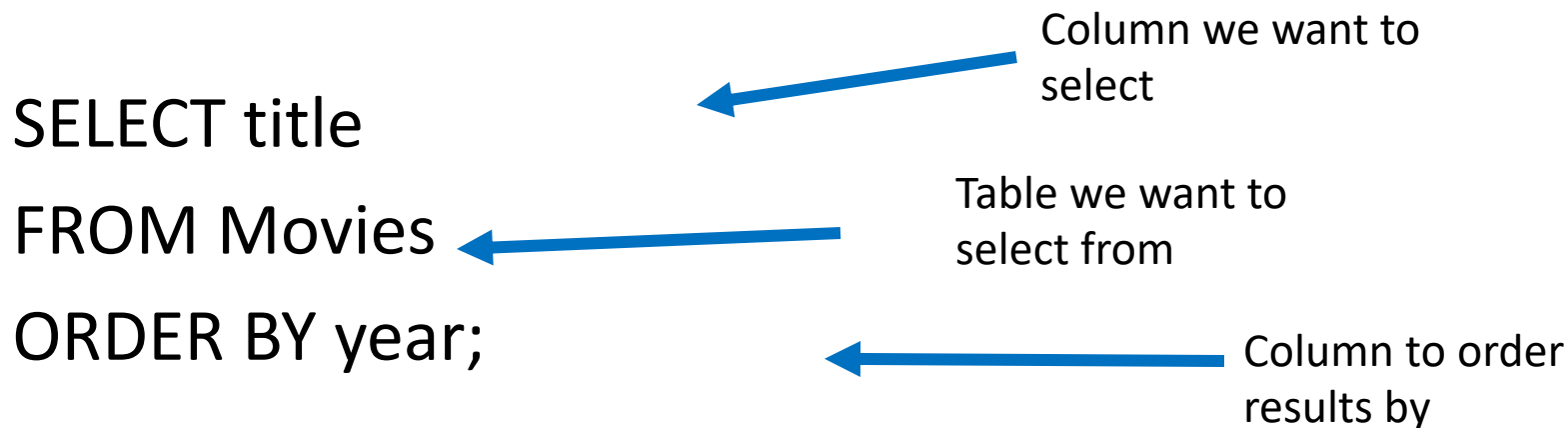
- Select and Order By
 - Using our Movies relation, what movies came out, ordered from oldest to newest

```
SELECT title
FROM Movies
ORDER BY year;
```

Column we want to select

Table we want to select from

Column to order results by

The diagram shows the SQL query 'SELECT title FROM Movies ORDER BY year;' with three blue arrows pointing from descriptive text to specific parts of the query. One arrow points from 'Column we want to select' to 'title'. Another arrow points from 'Table we want to select from' to 'Movies'. A third arrow points from 'Column to order results by' to 'year'.

Basic SQL Querying

- Select and Order By
 - Using our Movies relation, what movies were released, ordered by release date

```
SELECT title  
FROM Movies  
ORDER BY year, length;
```

If same year, break
ties by 'length'
column



Basic SQL Querying

- Select Multiple Attributes
 - Using our Movies relation, what are the names of the movies and the year they were released

```
SELECT title, year  
FROM Movies  
ORDER BY year;
```

Basic SQL Querying

- Select *
 - What are all of the attributions of the relation 'players'?

```
SELECT *
```

```
FROM Movies
```

```
ORDER BY year;
```

Introduction to SQL

- Misc SQL Facts
 - Don't technically need the ';', but you should
 - Don't technically need SQL keywords to be all capital letters, but you should
 - Whitespace generally doesn't matter in SQL (except in strings)

SQL Data Types

- Null
 - NULL is a legal value in any column. It is outside the domain of any data type
 - It normally represents an absence of information or inapplicable value
 - Other uses (bad practice):
 - Nothing (Empty)
 - False
 - 0
 - Example:
 - CREATE TABLE states (name TEXT, admissionToUnion INTEGER);
 - INSERT INTO states ('Puerto Rico', NULL);

SQL Data Types

- Integer
 - Signed integer up to 8 bytes long (64 bits)
 - From -170,141,183,460,469,231,731,687,303,715,884,105,728 to 170,141,183,460,469,231,731,687,303,715,884,105,727
 - Example:
 - 1

SQL Data Types

- Real
 - Signed 64 bit floating point
 - From VERY BIG to VERY SMALL
 - Can be specified as a decimal or exponential notation (using capital E)
 - Examples:
 - 1.0, 4.5, -6.0, .8
 - 5.6E12.4, -0.05E-67

SQL Data Types

- TEXT
 - Text can be of any length, but it must be enclosed in single quotes
 - Single quotes must be escaped with a single quote
 - Example:
 - `"', 'I'm a string'`
 - Many other databases give you access to fixed length TEXT or variable length TEXT below a certain character limit

SQL Data Types

- Date and Time
 - TEXT as ISO8601 strings ("YYYY-MM-DD HH:MM:SS.SSS").
 - REAL as Julian day numbers, the number of days since noon in Greenwich on November 24, 4714 B.C. according to the proleptic Gregorian calendar.
 - INTEGER as Unix Time, the number of seconds since 1970-01-01 00:00:00 UTC.
 - More info https://www.sqlite.org/lang_datefunc.html

SQL Data Types

- Blob
 - Used to hold arbitrary binary data. Often images, video, or other non-text data.
 - No storage limit.
 - Example:
 - X'012e2b041cc23f4a0549'
 - Rarely used in text INSERT statements, more often used in parameterized queries using programming language APIs (like python's sqlite3 module)
 - We'll be avoiding them in this class for simplicity

SQL Data Types

- Non SQLite Data Types
 - Memo – holds 65,536 characters
 - Currency – 15 whole digits and 4 decimal places in base 10
 - Yes/No – Microsoft Access doesn't like scaring people with the word boolean
 - Enum – Restricted text strings
 - You can imitate any of these types with SQLite's default types (or extend your own with wrapper functions).

Practice

Relation Schema Questions

- The relations on the right make up part of a banking database
- Indicate the following:
 - The attributes of each relation
 - The tuples of each relation
 - The schema for each relation
 - The database schema
 - Suitable types of each attribute

| <i>acctNo</i> | <i>type</i> | <i>balance</i> |
|---------------|-------------|----------------|
| 12345 | savings | 12000 |
| 23456 | checking | 1000 |
| 34567 | savings | 25 |

The relation Accounts


| <i>firstName</i> | <i>lastName</i> | <i>idNo</i> | <i>account</i> |
|------------------|-----------------|-------------|----------------|
| Robbie | Banks | 901-222 | 12345 |
| Lena | Hand | 805-333 | 12345 |
| Lena | Hand | 805-333 | 23456 |

The relation Customers

SQLite Demo

- Download sqlite
- Type:
 - `sqlite3 test.db`

If you're starting a new DB, you can name this whatever you want



```
host-262820:~ cse498$ sqlite3 test.db
SQLite version 3.30.0 2019-10-04 15:03:17
Enter ".help" for usage hints.
sqlite> █
```

SQLite Demo

- Create the table

```
host-262820:~ cse498$ sqlite3 test.db
SQLite version 3.30.0 2019-10-04 15:03:17
Enter ".help" for usage hints.
sqlite> CREATE TABLE Movies(title TEXT, year INTEGER, length INTEGER, genre TEXT);
sqlite> █
```

- Insert values into the table

```
host-262820:~ cse498$ sqlite3 test.db
SQLite version 3.30.0 2019-10-04 15:03:17
Enter ".help" for usage hints.
sqlite> CREATE TABLE Movies(title TEXT, year INTEGER, length INTEGER, genre TEXT);
sqlite> INSERT INTO Movies VALUES ('Gone With the Wind', 1939, 231, 'drama');
sqlite> INSERT INTO Movies VALUES ('Star Wars', 1977, 123, 'sciFi');
sqlite> INSERT INTO Movies VALUES ('Wayne's World', 1992, 95, 'comedy');
sqlite> █
```


SQLite Demo

- Let's see what's in the DB

```
[sqlite> SELECT * FROM Movies;  
Gone With the Wind|1939|231|drama  
Star Wars|1977|123|sciFi  
Wayne's World|1992|95|comedy  
sqlite> █
```

- That's Ugly

```
[sqlite> .headers on  
[sqlite> .mode columns
```

SQLite Demo

- One more SELECT

```
sqlite> .headers on
sqlite> .mode columns
sqlite> SELECT * FROM Movies;
title                year          length        genre
-----
Gone With the Wind   1939          231           drama
Star Wars            1977          123           sciFi
Wayne's World        1992          95            comedy
sqlite> █
```

SQLite Info

- Try the queries from the slides
- Try new queries as we learn more
- Explore on your own

