Introduction to Databases And Querying

What is a database?

A database is an organized collection of data that can be accessed, managed, and updated.

Some examples you have probably encountered:

- A shopping site (like Amazon) that has data representing products, users, purchases, and so on.
- A video site (like Netflix or YouTube) that might keep track of what people watch, let them create favorites or lists to watch, and maybe even recommend things based on how they rate what they've watched.
- Something like a Wordpress blog or even Facebook that would store data for posts, comments, and so on.

A Few More Tidbits...

- Storing data so it can be used later is called **persisting** data.

- Most databases provide intelligent ways to work with data to provide reports of all sorts and add and modify data as needed.

Some Reasons to Use a Database

- Easily support storing large number of records
- Central storage
- Support structured query syntax to retrieve data
- Enforces consistency and integrity of data
- Guarantee data type

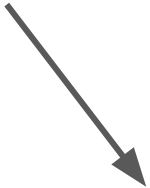
Relational Databases

- **Relational databases** allow data to be accessed and reassembled in many different ways by **Relating** the data rather than reassembling it.

- Real world problems are modeled into **Entities** such as users, posts comments, products, etc.

TABLE (Entity)

COLUMN (Attribute)



Product

ROW (Record or Instance)

ID	BRAND	MODEL	DESCRIPTION	PRICE
1453672	DELL	Vostro 5590	Great laptop.	679.00
1453673	APPLE	MacBook Pro	13” 2020 2 Thunderbold 3 Ports	1299.00

Database Management Systems (DBMS)

A DBMS helps manage a database. We will be working with a Relational Database Management System (RDBMS).

Provides for 4 basic functions:

- Data definition
- Data storage
- Data retrieval
- Administration

Common Database Engines

- Oracle
- SQL Server
- MS Access
- PostgreSQL
- DB2
- MySQL

Database Columns

- Each column in a database has a data type.

- ANSI SQL (Structured Query Language):
  The standard for SQL that is defined by ANSI (American National Standards Institute).

  https://en.wikipedia.org/wiki/SQL#SQL_data_types

Database Columns

ANSI SQL defines many common data types:.

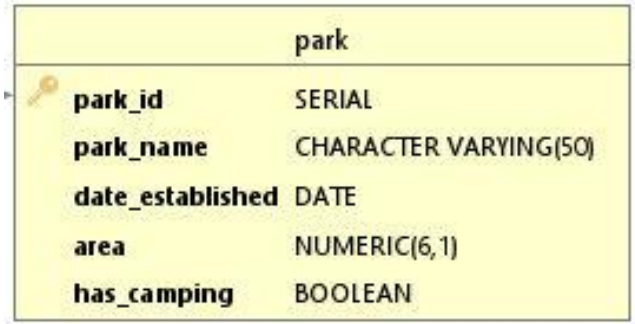
Examples:

- char, varchar, nvarchar
- int, decimal, bigint
- boolean/bit
- Datetime
- More info: https://www.w3resource.com/sql/data-type.php
- Postgres Data Types: https://www.postgresql.org/docs/12/datatype-character.html

Database Table Example

Each table column has a name and data type.

*	park_id	park_name	date_established	area	has_camping
1	1	Acadia	1919-02-26	198.6	true
2	2	American Samoa	1988-10-31	33.4	false
3	3	Arches	1971-11-12	310.3	true
4	4	Badlands	1978-11-10	982.4	true
5	5	Big Bend	1944-06-12	3242.2	false
6	6	Biscayne	1980-06-28	700.0	true
7	7	Black Canyon of the Gunnison	1999-10-21	124.6	false
8	8	Bryce Canyon	1928-02-25	145.0	true
9	9	Canyonlands	1964-09-12	1366.2	true
10	10	Capitol Reef	1971-12-18	979.0	true
11	11	Carlsbad Caverns	1930-05-14	189.3	true
12	12	Channel Islands	1980-03-05	1009.9	true



Intro to Queries

Intro To Queries

SQL stands for **structured query language** and is a **declarative programming language** to retrieve and update records from a database.

SQL consists of:

- **Data definition language** to define the data structures
- **Data manipulation language** to query and modify the data in a database
- **Data control language** to define access to a particular database

Intro To Queries

- The **SELECT** clause indicates what columns to get from a database table.

- The **FROM** clause indicates which table(s) to retrieve the data from.

```
SELECT city_name FROM city;
```

- If you want get all the columns in a table you can use * for the column.

```
SELECT * FROM city;
```

Intro To Queries

- The **SELECT** clause indicates what columns to get from a database table.
- The **FROM** clause indicates which table(s) to retrieve the data from.
- The **WHERE clause** is used to filter the result set using one or more criteria rules.

```
SELECT city_name FROM city WHERE
state_abbreviation='DE';
```

Intro To Queries

- Conditional clauses in the **WHERE clause** can include
  - `=, <>, !=, >, >=, <, <=`
  - `IN(values), NOT IN(values)`
  - `BETWEEN value AND value`
  - `IS NULL, IS NOT NULL`
  - `LIKE (with wildcard character)`

Intro To Queries

- The **DISTINCT clause** indicates that duplicate values should not be included

- The **AS clause** establishes an alias for a particular column name

```
SELECT DISTINCT state_abbreviation AS state
FROM city;
```

Let's Write Some Queries!!!