What is a Database?

- A database (DB) is a systematic collection of data.
- There are two kinds of databases: relational and non-relational.
- You can think about a relational database as an Excel file.
  - Each tab in Excel is a **table** in the database.
  - Each row is a **record**.
  - Each column is a **field**.

CustomerID	CustomerName	ContactName
1	Alfreds Futterkiste	Maria Anders
2	Ana Trujillo Emparedados y helados	Ana Trujillo
3	Antonio Moreno Taquería	Antonio Moreno
4	Around the Horn	Thomas Hardy
5	Berglunds snabbköp	Christina Berglund
6	Blauer See Delikatessen	Hanna Moos
7	Blondel père et fils	Frédérique Citeaux
8	Bólido Comidas preparadas	Martín Sommer
9	Bon app'	Laurence Lebihans
10	Bottom-Dollar Marketse	Elizabeth Lincoln

Sources:
https://www.guru99.com/introduction-to-database-sql.html
https://www.w3schools.com/sql/sql_intro.asp

Non-Relational Database

- Non-relational databases are often called NoSQL databases.
- They store data in a non-tabular format and instead use data structures like documents.
- A document can be highly detailed while containing a range of different types of information in different formats.
- Non-relational databases may perform faster because a query doesn't have to view several tables in order to deliver an answer, as relational datasets often do.
- Non-relational can support rapidly developing applications requiring a dynamic database able to change quickly and to accommodate large amounts of complex, unstructured data.
- Non-relational databases are less common, at least in part because they don't use SQL, so we will focus on relational databases in this course.

Sources:
https://www.mongodb.com/non-relational-database

What is SQL?

- SQL (Structured Query Language) is a standard language for accessing and manipulating databases.
- Although SQL is a standard, there are different versions of the SQL language. But all versions must support at least the major commands (such as SELECT, UPDATE, DELETE, INSERT, WHERE) in a similar manner.

Sources:
https://www.w3schools.com/sql/sql_intro.asp

Commands

Command	Description
BASIC NAVIGATION & QUERYING	
mysql	Start an interactive SQL shell in Codio
SHOW DATABASES;	View list of databases
USE *database_name*;	Select a database to use
SHOW TABLES;	View list of tables (must select a database first)
SELECT * FROM *table_name*;	View records in table
CREATING A DATABASE AND ADDING DATA	
CREATE DATABASE *database_name*;	Create a database
CREATE TABLE *table_name* (*field types*)	Create a table in the selected database. See following slides for more info.
INSERT INTO *table_name* (*fields*) VALUES (*values*);	Insert a record into the specified table. See following slides for more info.

CREATE TABLE

** Make sure you have called USE *database_name;* before you try to create a table **

```
CREATE TABLE Cars (
        id INT(8) UNSIGNED NOT NULL auto_increment,
        model VARCHAR(255) default NULL,
        manufacture_year YEAR(4) default NULL,
        PRIMARY KEY (id) )
 AUTO_INCREMENT=1;
```

- This command identifies the (1) table name, (2) the column names, (3) the type of data to be stored in each column.
- What is the table name?
  - Cars
- How many fields are specified?
  - 3

- What are the types of each field?
  - id → INT(8)
  - model → VARCHAR(255)
  - manufacture_year → YEAR(4)
- What does UNSIGNED NOT NULL auto_increment mean?
  - Only positive integers. NULL value is not allowed. No need to enter this data, instead it gets incremented automatically
- What does default NULL mean?
  - If no value is specified, it will be set to NULL
- What does PRIMARY KEY (id) mean?
  - The primary key value must be unique for each record in the table. This table will use the id as the unique identifier.

Text Data Types (sample)

Text types

Data type	Description
`VARCHAR(size)`	The **variable character field** can contain letters, numbers, and special characters. Max. size is specified in parenthesis. **Can store up to 255 characters**, just enough for a Twitter post. Note: If you put a greater value than 255 it will be converted to a TEXT type
`TEXT`	Holds a string with a maximum length of 65,535 characters
`LONGTEXT`	Holds a string with a maximum length of 4,294,967,295 characters

Numeric Data Types (sample)

Number types:

Data type	Description
`INT(size)`	-2147483648 to 2147483647 normal. 0 to 4294967295 UNSIGNED. The **maximum number of digits may be specified in parenthesis**

The integer (INT) type has an extra option called `UNSIGNED`.

Normally, the integer ranges from a negative to positive value. Adding the `UNSIGNED` attribute will move that range up so it starts at zero instead of a negative number (more on this later).

Date Data Types (sample)

Date types:

Data type	Description
`DATE()`	A date. Format: YYYY-MM-DD **Note:** The supported range is from '1000-01-01' to '9999-12-31'
`YEAR()`	A year in two-digit or four-digit format. **Note**: Values allowed in four-digit format: 1901 to 2155. Values allowed in two-digit format: 70 to 69, representing years from 1970 to 2069

INSERT TABLE

```
INSERT INTO Cars (make, manufacture_year)
VALUES (“Ford", 1960);
```

- You can specify the fields in any order (though left to right is conventional), but the order of the values must match the order of the specified fields.
- If you wanted to use a default value, you could do this:

```
INSERT INTO Cars (make, manufacture_year)
VALUES (“Ford", DEFAULT);
```

** Make sure you have called USE *database_name;* before you try to insert **

What is a Query?

- A **query** is a request for data or information from a **database** table or combination of tables.
- What is an example of a simple query?
  - SELECT column1, column2 FROM table_name;

Demo

Before we can query a database, we have to select a database and a table within the database.

1. We can see the list of databases with: SHOW DATABASES;
2. We can choose a database with: USE *database_name*;
3. We can see the list of tables in the database with: SHOW TABLES;

Now let's select some data!

SELECT modifiers

SQL Command	Description
LIMIT	Limits the number of records returned
ORDER BY column_name ASC	Return the records in ascending order of the given column_name
ORDER BY column_name DESC	Return the records in descending order of the given column_name
WHERE column OPERATOR data	Returns records that satisfy the given condition

Operators

OPERATOR	DESCRIPTION
=	Equal
<>	Not equal. Note: In some versions of SQL this operator may be written as !=
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
BETWEEN	Between an inclusive range
LIKE	Search for a pattern
AND	Displays a record if both the first condition AND the second condition are true
OR	Displays a record if either the first condition OR the second condition is true

SQL Commands to Update Tables

SQL Command	Description
SHOW COLUMNS from table_name;	Show the table's columns and their data types
UPDATE table_name SET column_name = value WHERE column_name operator value;	Update records in *table_name*. Set the value in *column_name* where the given condition matches.
ALTER TABLE table_name CHANGE column_name new_col_name data_type;	Change the name and data type of a column in the table
ALTER TABLE table_name RENAME column_name TO new_col_name;	Change the name only of a column in the table
ALTER TABLE table_name DROP column_name;	Delete a column
ALTER TABLE table_name ADD column_name data_type;	Add a column
RENAME TABLE old_table_name TO new_table_name;	Rename a table
DROP TABLE table_name;	Delete a table
DROP DATABASE database_name;	Delete a database

SLIDE CREDIT: DAMAN OBEROI

Today's Activity

1. You will work with an interactive SQL shell to practice the commands we covered today.
2. You will create a database and load your API data into it.

- You will go into your TA's breakout room and work on the activity.
- Feel free to work with others in your room and ask questions of the group.
- The TA is there to assist you. If you want to ask me a question, come back to the main zoom room.

To Convert Dictionary into DataFrame

- The Codio guide provides the following link. This may or may not prove useful to you, depending on how you implement your program.

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.from_dict.html?highlight=dataframe%20from_dict#pandas.DataFrame.from_dict

- Another way to create a DataFrame from your data is as follows. Note the parts in *italics* that you must update for your program.

```
# creating data frame to add data to
col_names = ['col1', 'col2', 'col3']
df  = pd.DataFrame(columns = col_names)
df.loc[len(df.index)] = [value1, value2, value3]
```