

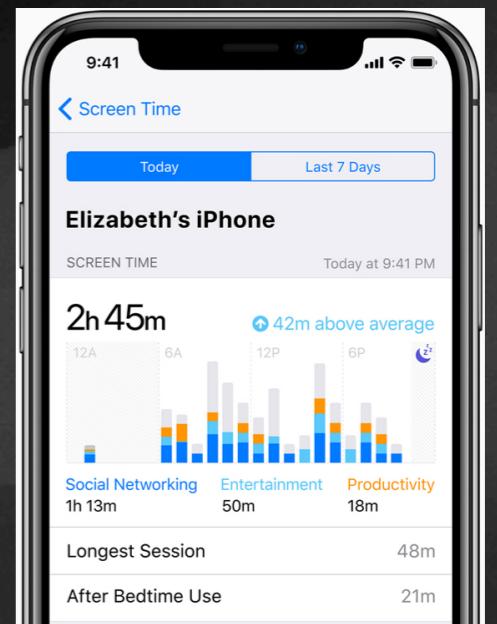
# Scratching SQL for Poké Power

## Quick Intro to Databases and SQL



# What is a Database?

- A system for storing organized data with interfaces to efficiently retrieve and process the data
- Can be as small as a single file and application on your machine
  - “ScreenTime” on Apple devices work by writing your current application to a database file every second. Reports come from this data. (“Sqlite”)
- To a cloud cluster of servers with many files and applications cooperating over a network processing high transaction load with high reliability:
  - FaceBook, Banking systems. (Postgres, Mysql)

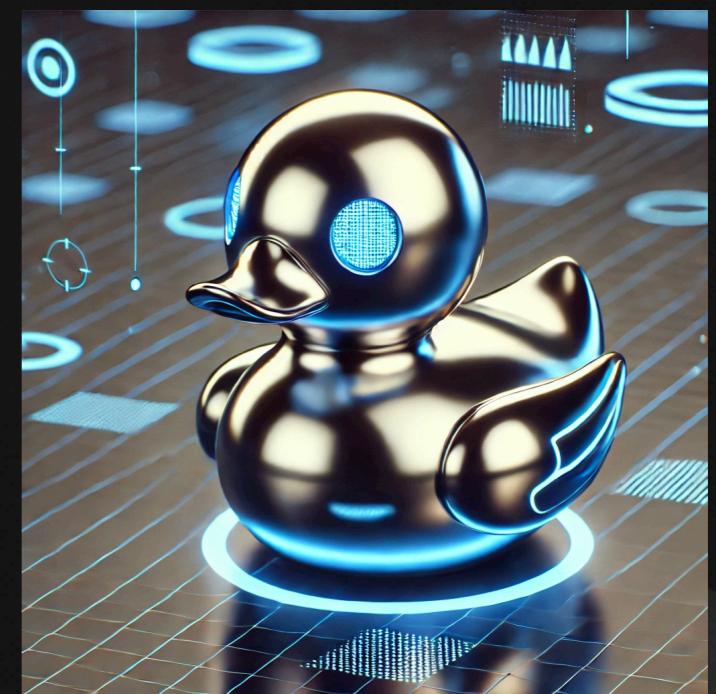


# Data... it is vast...

- Medical records
- Commercial Inventory, Order, Transaction History
- Leaderboards, World State
- Geospatial
- Financial and stock market
- Sensor and telemetry data
- General Documents
- Social Media Relationships
- ...



# Relational Database (RDBMS)



- Data is organized into “Tables”
  - The table has data records stored in “rows”, having a common set of fixed “columns” each with a “type”
  - **Table: “Student”**

ID	Name	Age	House	← Columns / Fields
0	Molly	16	Clark	
1	Kasper	15	Folsom	
2	Fielder	14	Bella	
3	Will	14	Cantor	

INTEGER type → 0 → Molly → 16 → Clark ← Columns / Fields

VARCHAR type → 1 → Kasper → 15 → Folsom ← Rows / Records

→ 2 → Fielder → 14 → Bella ←

→ 3 → Will → 14 → Cantor ←

# Comma Separated Value (CSV)

- *Insanely common file format everywhere.*
  - You can download your bank transactions in it
- *Because it is so simple:*
  - An optional header row with column names
  - Each line is a record, with values separated by a delimiter, most often comma

```
ID,Name,Age,House
0,Molly,16,Clark
1,Kasper,15,Folsom
2,Fielder,14,Bella
3,Will,14,Cantor
```

# Structured Query Language

## (SQL    S-Q-L or Sequel)

- Developed in the 1970's as a way for humans to query and administer databases
- Standardized, but there are many dialects
- Primary operations are:
  - **CREATE** to define a table structure
  - **INSERT** to add rows, **DELETE** to remove rows
  - **SELECT** to search for those rows based on **WHERE** conditions
- Built in functions like **AVG**, **MIN**, **MAX**, **COUNT**

# Some Context for Learning...

# Pokemon Fanaticism

*What is the average  
height across every  
kind of Pokémon?*



# Pokémon Data On Kaggle

The screenshot shows the 'All Pokemon Dataset' page on Kaggle. At the top, there's a profile picture of Mario Hernández, a 'Data by Mario Hernández · UPDATED 4 YEARS AGO', a '67' rating, a 'Code' button, a 'Download' button, and a more options menu. Below the header, the title 'All Pokemon Dataset' is displayed in large bold letters, followed by a subtext 'Data from all Pokemons and their different forms from all 8 generations'. To the right is the iconic 'Pokémon' logo. Below the title, there are navigation links: 'Data Card' (underlined), 'Code (9)', 'Discussion (0)', and 'Suggestions (0)'. The main content area is titled 'About Dataset'. Under 'Context', it says: 'This Dataset contains data from all Pokemons and their forms from the eight generations. It includes their number, name, primary and secondary type, abilities, base stats, performance vs other types, etc. Data was scrapped from [serebii](#) [bulbapedia](#) and [pokemon database](#)'. To the right of this, there are sections for 'Usability' (8.82), 'License' (CC0: Public Domain), and 'Expected update frequency' (Not specified). The entire page has a dark background.

<https://www.kaggle.com/datasets/maca11/all-pokemon-dataset>

```
Number,Name,Type 1,Type 2,Abilities,HP,Att,Def,Spa,Spd,Spe,BST,Mean,Standard Deviation,Generation,Experience type,Experience to level
1,Bulbasaur,Grass,Poison,"['Chlorophyll', 'Overgrow']",45,49,49,65,65,45,318,53.0,8.640987597877146,1.0,Medium Slow,1059860,0.0,45,0,>
2,Ivysaur,Grass,Poison,"['Chlorophyll', 'Overgrow']",60,62,63,80,80,60,405,67.5,8.902246907382429,1.0,Medium Slow,1059860,0.0,45,0.0,>
3,Venusaur,Grass,Poison,"['Chlorophyll', 'Overgrow']",80,82,83,100,100,80,525,87.5,8.902246907382429,1.0,Medium Slow,1059860,1.0,45,0,>
3,Mega Venusaur,Grass,Poison,['Thick Fat'],80,100,123,122,120,80,625,104.16666666666669,18.747592438023133,6.0,Medium Slow,1059860,1.>
4,Charmander,Fire,,,"['Blaze', 'Solar Power']",39,52,43,60,50,65,309,51.5,8.995369179009089,1.0,Medium Slow,1059860,0.0,45,0.0,0.0,0.0,>
5,Charmeleon,Fire,,,"['Blaze', 'Solar Power']",58,64,58,80,65,80,405,67.5,9.233092656309694,1.0,Medium Slow,1059860,0.0,45,0.0,0.0,0.0,>
6,Charizard,Fire,Flying,"['Blaze', 'Solar Power']",78,84,78,109,85,100,534,89.0,11.575836902790224,1.0,Medium Slow,1059860,1.0,45,0.0,>
6,Mega Charizard Y,Fire,Flying,['Drought'],78,104,78,159,115,100,634,105.66666666666669,27.378012264508094,6.0,Medium Slow,1059860,1.>
6,Mega Charizard X,Fire,Dragon,['Tough Claws'],78,130,111,130,85,100,634,105.66666666666669,20.154955277107963,6.0,Medium Slow,1059860,>
7,Squirtle,Water,,,"['Rain Dish', 'Torrent']",44,48,65,50,64,43,314,52.33333333333336,8.919392106839767,1.0,Medium Slow,1059860,0.0,4,>
8,Wartortle,Water,,,"['Rain Dish', 'Torrent']",59,63,80,65,80,58,405,67.5,9.142392101268316,1.0,Medium Slow,1059860,0.0,45,0.0,0.0,0.0,>
9,Blastoise,Water,,,"['Rain Dish', 'Torrent']",79,83,100,85,105,78,530,88.3333333333333,10.386957633921922,1.0,Medium Slow,1059860,1.>
```

# Let's Play...



# Load Pokemon CSV into DuckDB

```
duckdb -ui poke.duckdb
$ duckdb -ui poke.duckdb
```

result  
varchar

UI started at <http://localhost:4213/>

v1.2.2 7c039464e4  
Enter ".help" for usage hints.

D  
D

duckdb -ui poke.duckdb

1 CREATE TABLE all\_pokemon AS  
2 FROM read\_csv('All\_Pokemon.csv');

1 row returned in 46ms

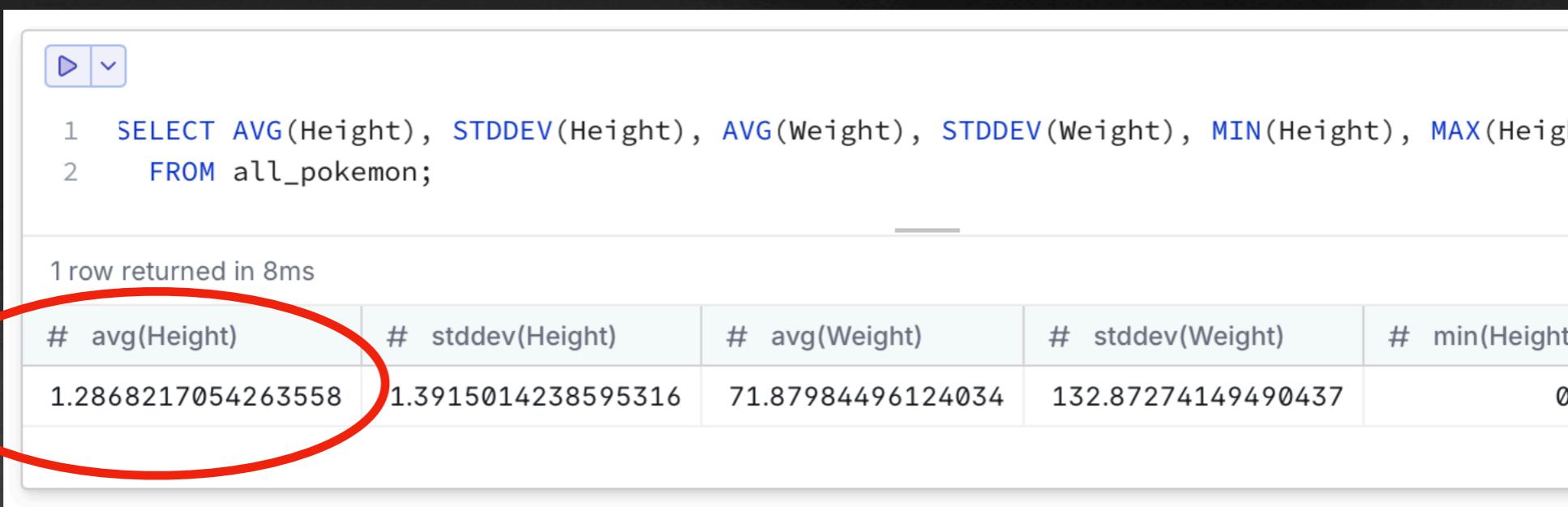
1 SELECT \* FROM all\_pokemon;

1032 rows returned in 18ms

Number	Type 1	Type 2	Abilities	HP	Att	Def	Spa
1	Rattata	Poison	['Chlorophyll', 'Overgrow']	45	49	49	49
2	Rattata	Poison	['Chlorophyll', 'Overgrow']	60	62	63	63
3	Rattata	Poison	['Chlorophyll', 'Overgrow']	80	82	83	83
4	Charmander	Poison	['Thick Fat']	80	100	123	123
5	Charmeleon	Fire	['Blaze', 'Solar Power']	39	52	43	43
6	Charizard	Fire	Flying	58	64	58	58
6	Mega Charizard Y	Fire	Flying	78	84	78	78
6	Mega Charizard X	Fire	Dragon	78	104	78	78
7	Squirtle	Water	['Blaze', 'Solar Power']	44	48	65	65
8	Wartortle	Water	['Blaze', 'Solar Power']	59	63	80	80
9	Blastoise	Water	['Blaze', 'Solar Power']	79	83	100	100
9	Mega Blastoise	Water	['Mega Launcher']	79	103	120	120
10	Caterpie	Bug	['Run Away', 'Shield Dust']	45	30	35	35
11	Metapod	Bug	['Shed Skin']	50	20	55	55

Filter 1,032 Rows

# ANSWER



A screenshot of a SQL query results window. The query is:

```
1 SELECT AVG(Height), STDDEV(Height), AVG(Weight), STDDEV(Weight), MIN(Height), MAX(Height)
2 FROM all_pokemon;
```

The output shows one row returned in 8ms:

# avg(Height)	# stddev(Height)	# avg(Weight)	# stddev(Weight)	# min(Height)	# max(Height)
1.2868217054263558	1.3915014238595316	71.87984496124034	132.87274149490437	0	10.7

~1.29m ≈ 4 feet 3 inches

# So much more...

- Relational Aspect – JOIN ing multiple tables
- Grouping by column, Math across rows/groups
- Time series windowing

**SQL is in many places. But it's not the only way to interact with data...**

**Python Language...**

**DataFrames, Pandas  
Can also call SQL**



[https://medium.com/@Anirudh\\_Singh\\_Chauhan/data-analysis-on-pokemon-dataset-44cdc7d15e56](https://medium.com/@Anirudh_Singh_Chauhan/data-analysis-on-pokemon-dataset-44cdc7d15e56)

# Extra: Some SQL Commands

```
DROP TABLE all_pokemon;
```

```
CREATE TABLE all_pokemon  
AS FROM read_csv('All_Pokemon.csv');
```

```
SELECT * FROM all_pokemon;
```

```
SELECT AVG(Height), STDDEV(Height), AVG(Weight), STDDEV(Weight),  
       MIN(Height), MAX(Height), MIN(Weight), MAX(Weight)  
FROM all_pokemon;
```

```
SELECT "Type 1", COUNT(*)  
FROM all_pokemon  
WHERE Legendary > 0 GROUP BY "Type 1";
```

```
SELECT Name, BMI, *  
FROM all_pokemon ORDER BY BMI DESC;
```