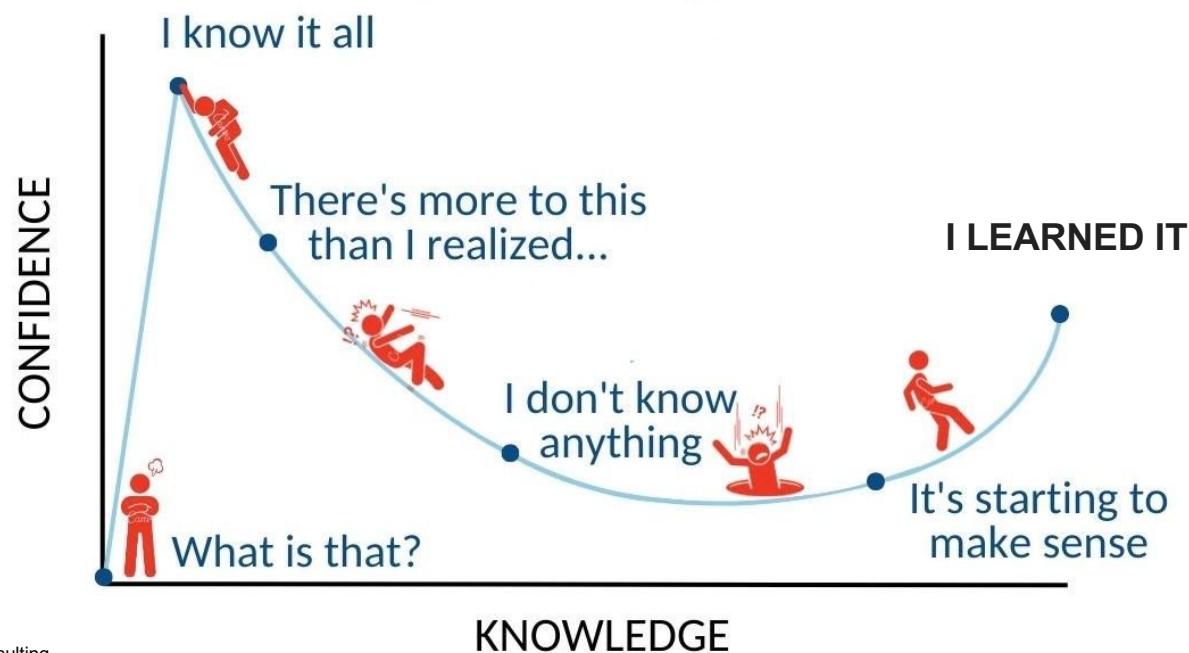


correlation.[•]one
TECH FOR JOBS

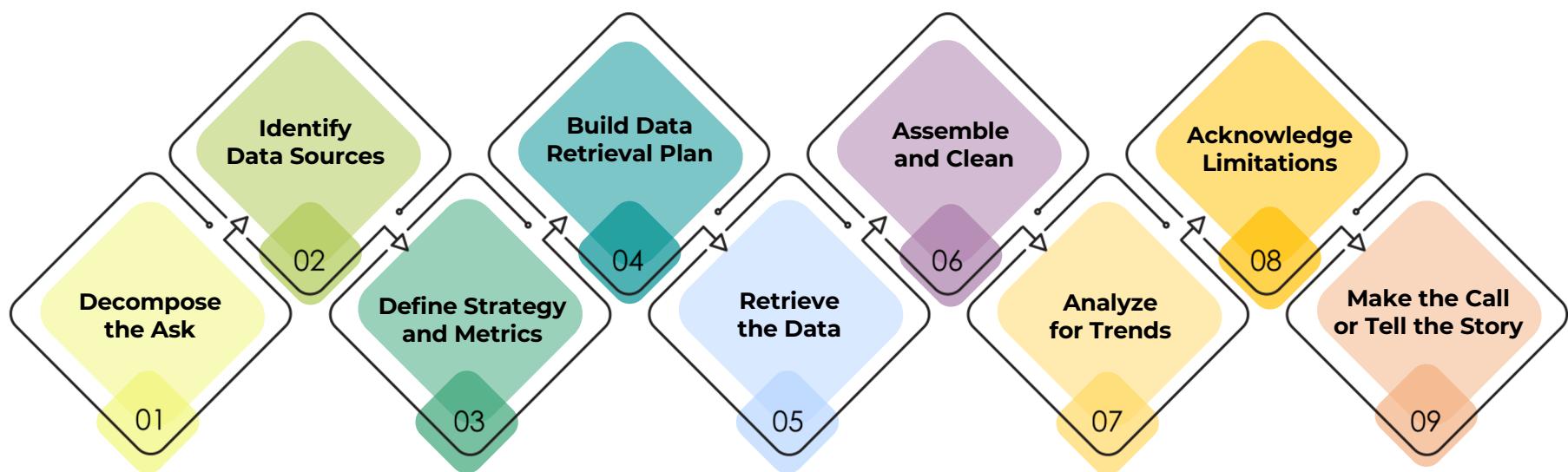
DUNNING-KRUGER Effect



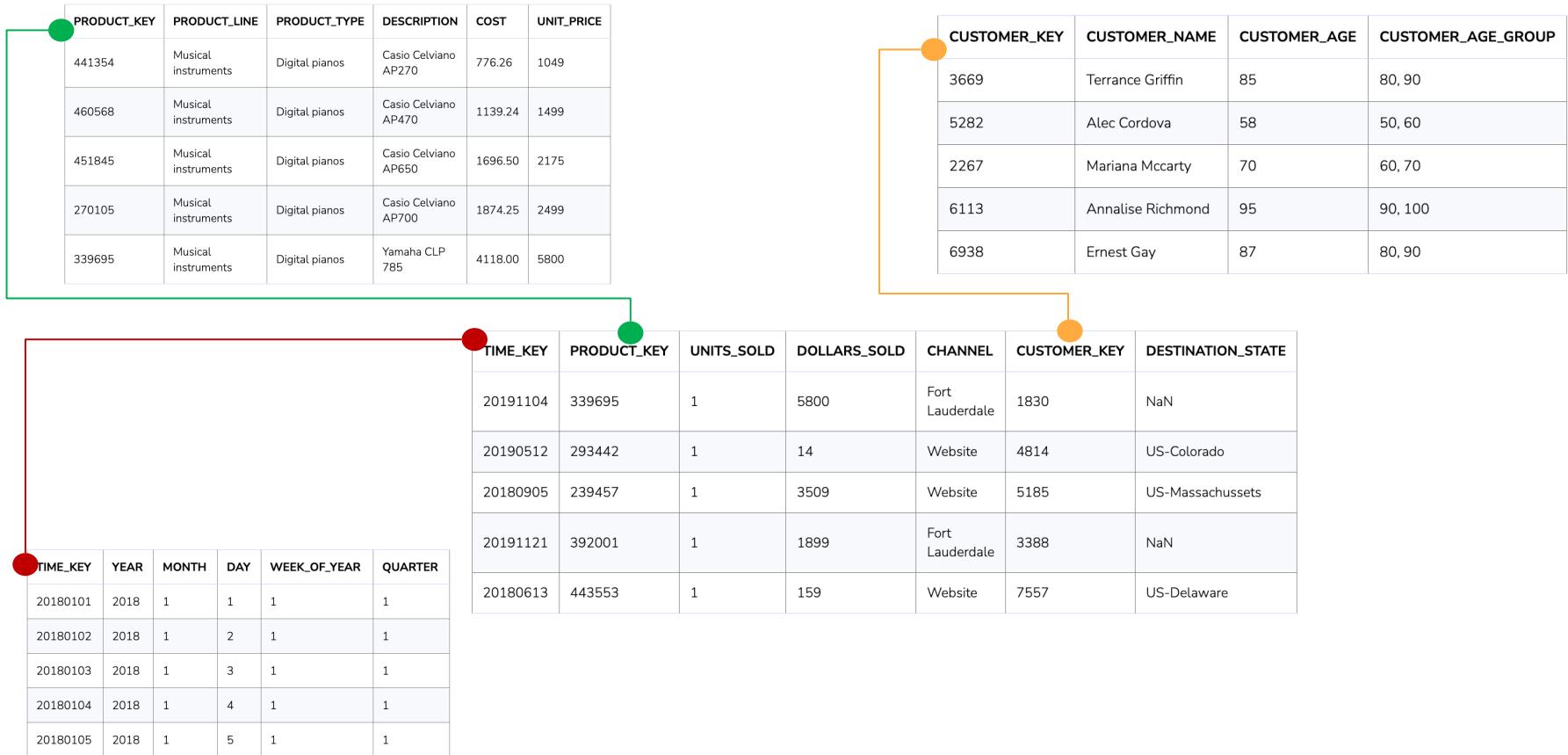
Credit: BVA Nudge Consulting

Analytics Paradigm

Regardless of type or industry, this paradigm provides a repeatable pathway for effective data problem solving.



Simply Music Model



Page Mockup – Bird's Eye View

Company logo

Date picker

Scorecards

Line chart (profit and revenue)

Line chart (sales)

Grouped bar chart (profit and revenue) - by location

Bar chart (sales) - by location

Page Mockup – Detailed View

Company logo

Scorecards

Date picker

Bar chart with drill down (items and sales)

This chart will serve as a page-level filter

Bar chart (items and total profit)

Pie chart (sold items by location)

Bubble map (states, items delivered)

Bar chart (number of customers that buy this item)

Gartner Magic Quadrant

Figure 1: Magic Quadrant for Analytics and Business Intelligence Platforms



Gartner (June 2024)

Gartner

The 2024 MAD Landscape

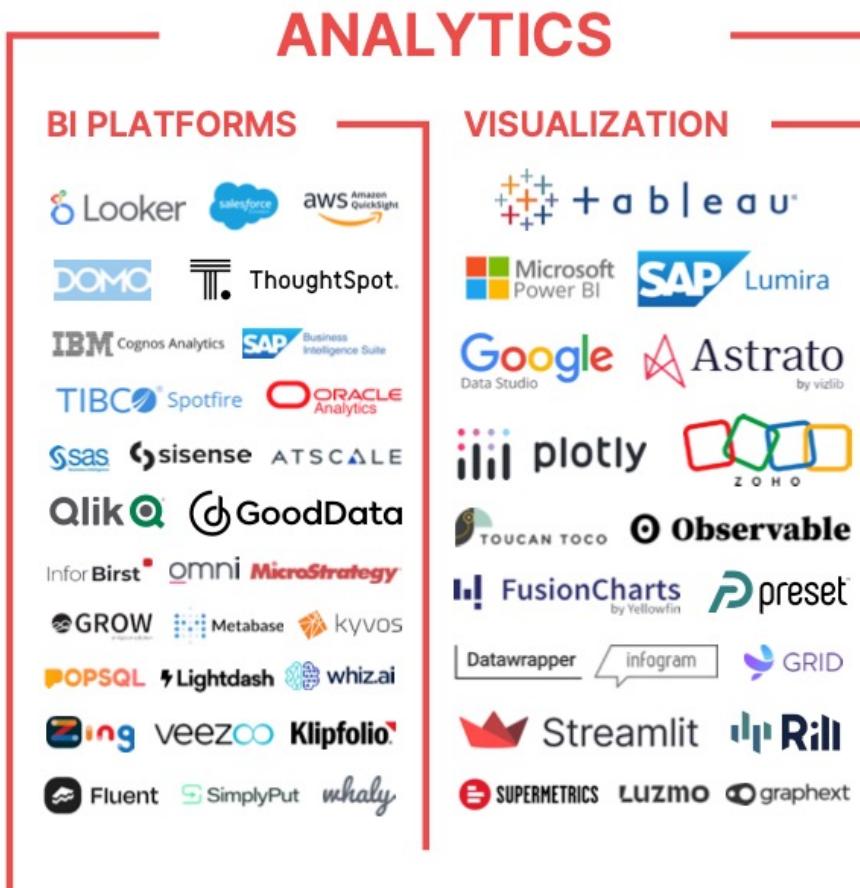
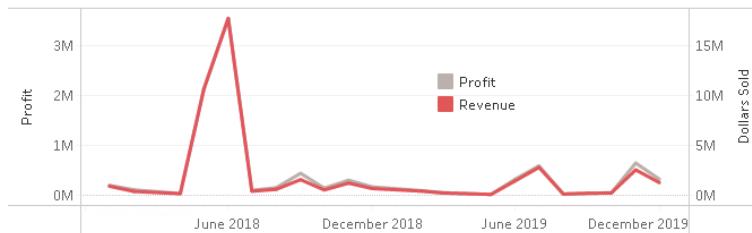


Tableau Bird's Eye View



Revenue vs profit



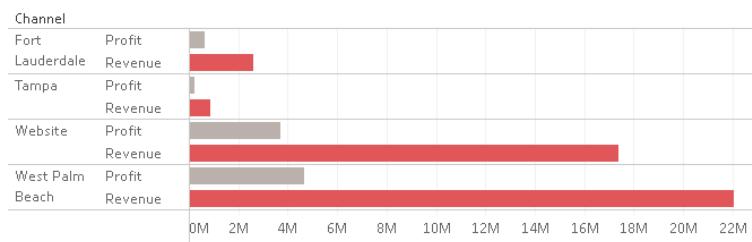
Time Key 1/5/2018 D 12/29/2019

Units sold	Revenue	Profit
6,475	42,889,566	9,235,989

Sales over time



Profit and revenue by channel



Sales by channel

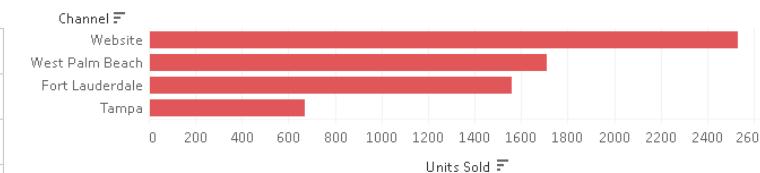
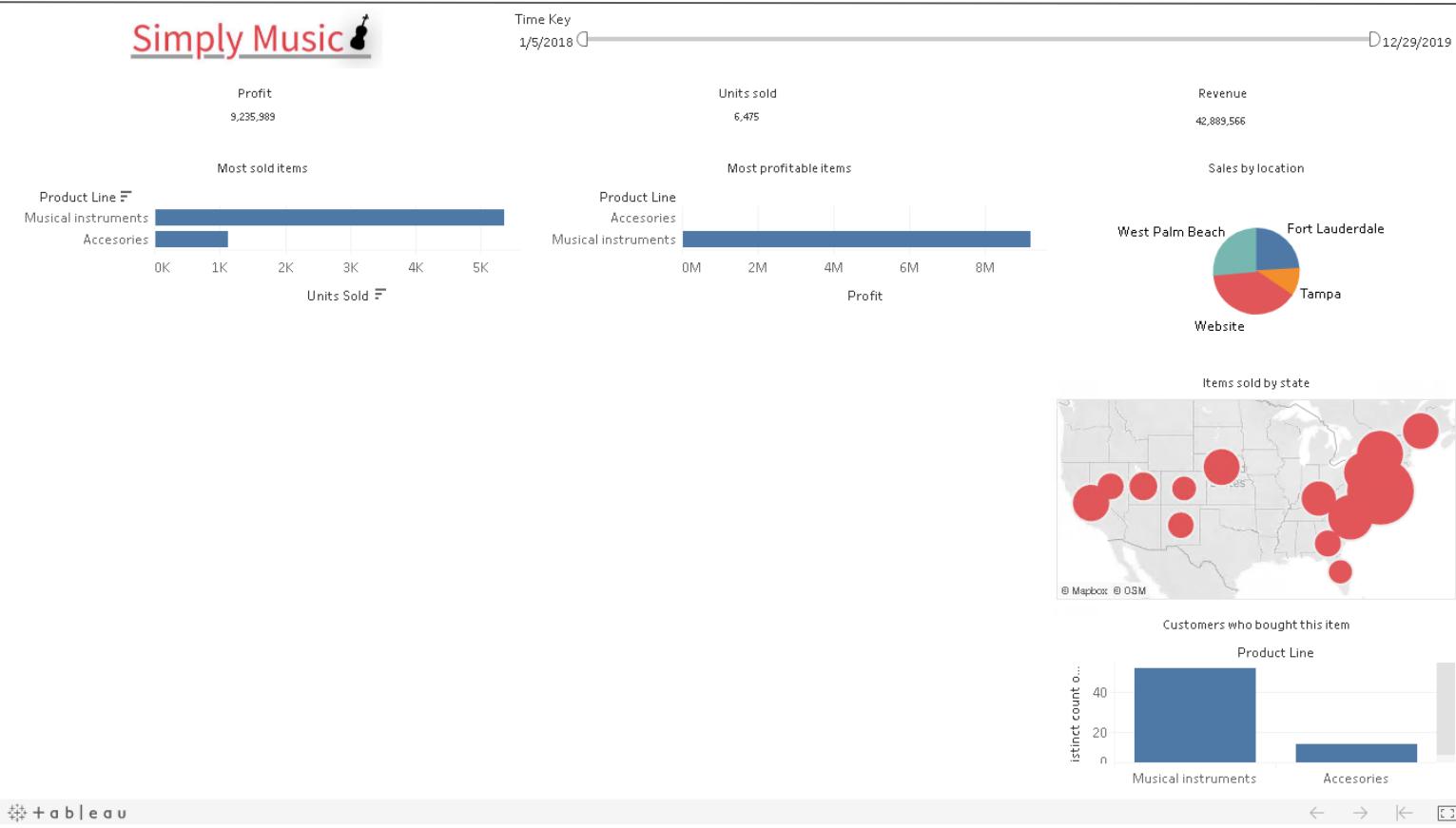


Tableau Detail View



Introduction to SQL

Excel Shortcomings & Limitations

-  **Scalability:** Not suitable for handling very large datasets
-  **Data Integrity:** Limited support for data validations
-  **Concurrency:** Excel does not handle multiple users simultaneously well
-  **Data Security:** Excel has limited security features.
-  **Data Recovery:** Excel lacks sophisticated transaction management and recovery systems
-  **Complex Relations & Queries:** Excel is not designed to handle complex relationships between different sets of data

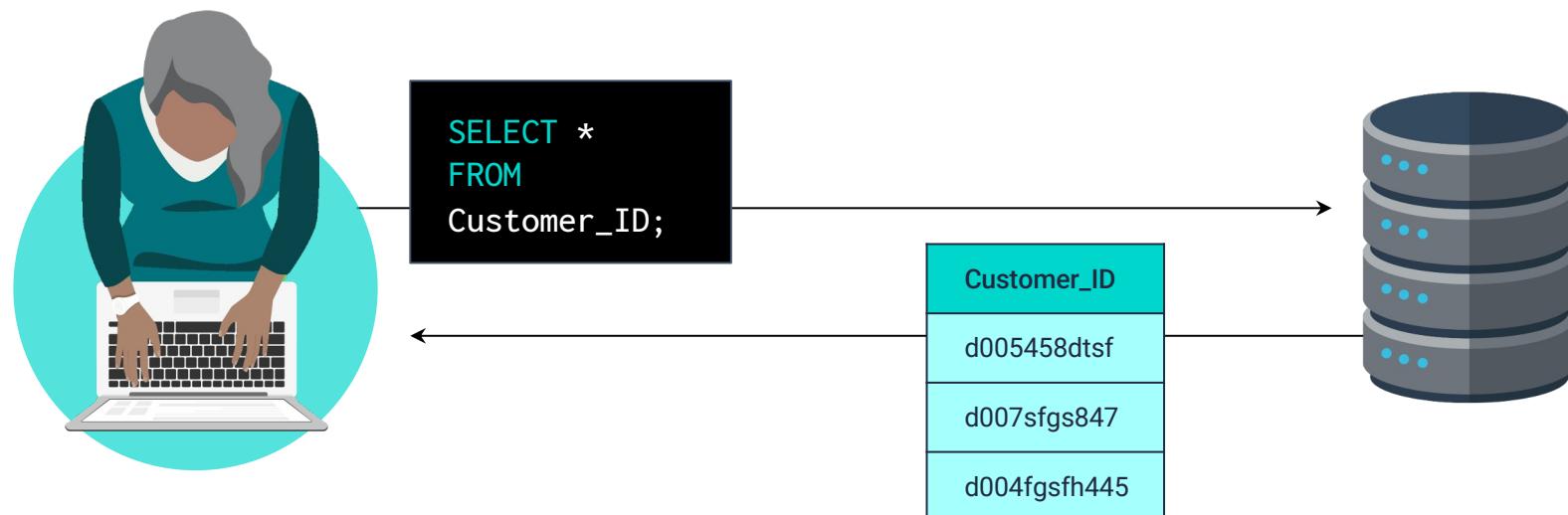
“

A **database** is an organized collection of information or data, stored electronically in a computer system. It is designed to **efficiently** store, retrieve, update, and manage data.

Introduction to SQL

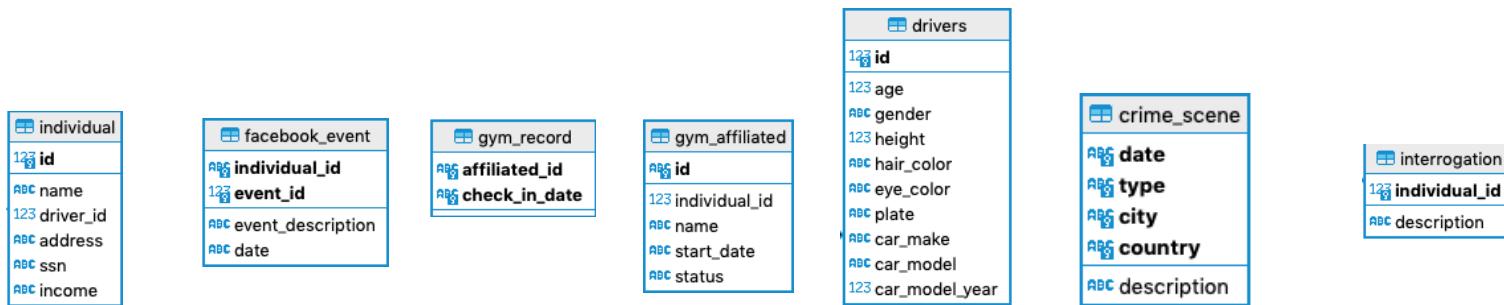
SQL (often pronounced "sequel") stands for Structured Query Language.

It is a powerful tool that enables programmers to create, populate, manipulate, and access databases. It also provides an easy method for dealing with server-side storage.



Relational Databases

A relational database consists of a collection of structured data stored in **Tables**. A Table organizes the data in rows and columns.



A TABLE in a Relational Databases

Primary Key
Columns →

Rows / Tuple

id	name	driver_id	address	ssn	income
1	Herve Lilie	7023162704	9663 Mockingbird Lane	156-14-0067	\$63163.25
2	Jacqueline Colam	4722687668	1396 Westport Circle	733-53-7490	\$94851.22
3	Keenan Treker	9198946080	4 Glendale Junction	380-51-1574	\$58070.55
4	Kimbell Everill	39201465	107 Hovde Circle	415-98-5200	\$40770.06
5	Abramo De Bernardis	7411450154	8251 Maple Wood Park	657-03-1688	\$89059.97

Introduction to SQL

Data using SQL is stored in tables on the server, much like spreadsheets you would create in Microsoft Excel.

This makes the data easy to visualize and search.



Customer_ID	Date_ID
d005458dtsf	6/26/2019
d007sfgs847	8/3/2018
d004fgsfh445	12/3/2018

Order_ID	Customer_ID	Date_ID
10001	d005458dtsf	6/26/2019
10002	d007sfgs847	8/3/2018
10003	d004fgsfh445	12/3/2018

Foreign Keys

Foreign Keys reference the primary key of another table.

Can have a different name. It does not have to be unique.

Primary Key

	A	B
1	family_id	family
2	1	Smiths
3	2	Jones

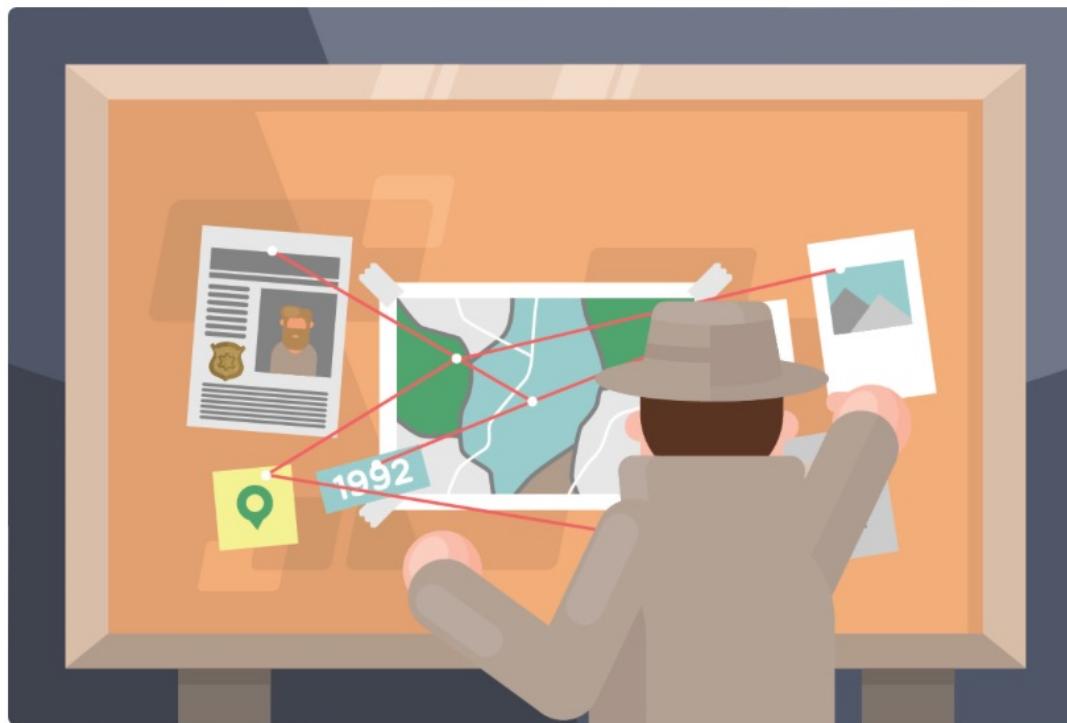
Primary Key

Foreign Key

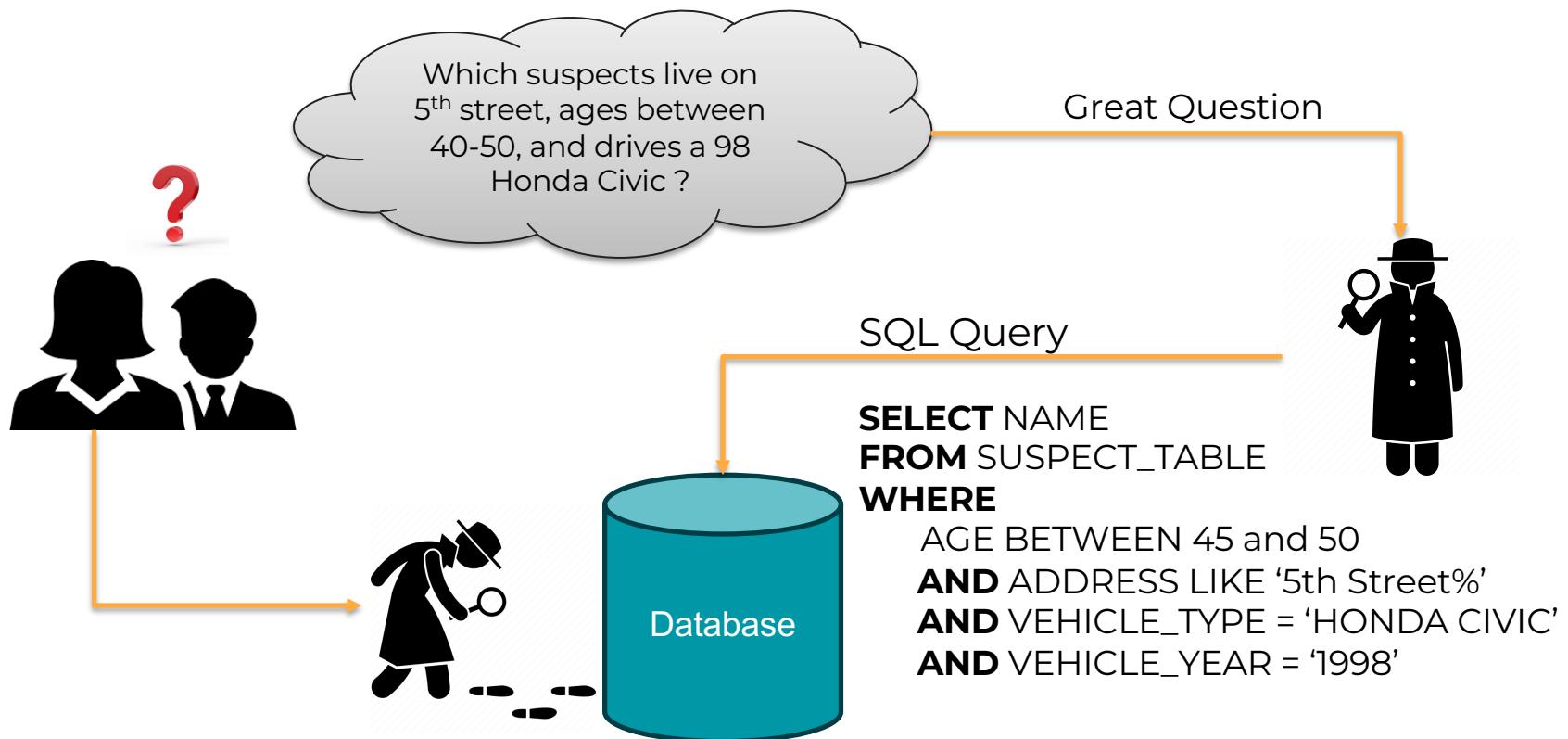
	A	B	C
1	child_id	family_id	children
2	11	1	Chris
3	22	1	Abby
4	33	1	Suzy

Murder in Sequel City

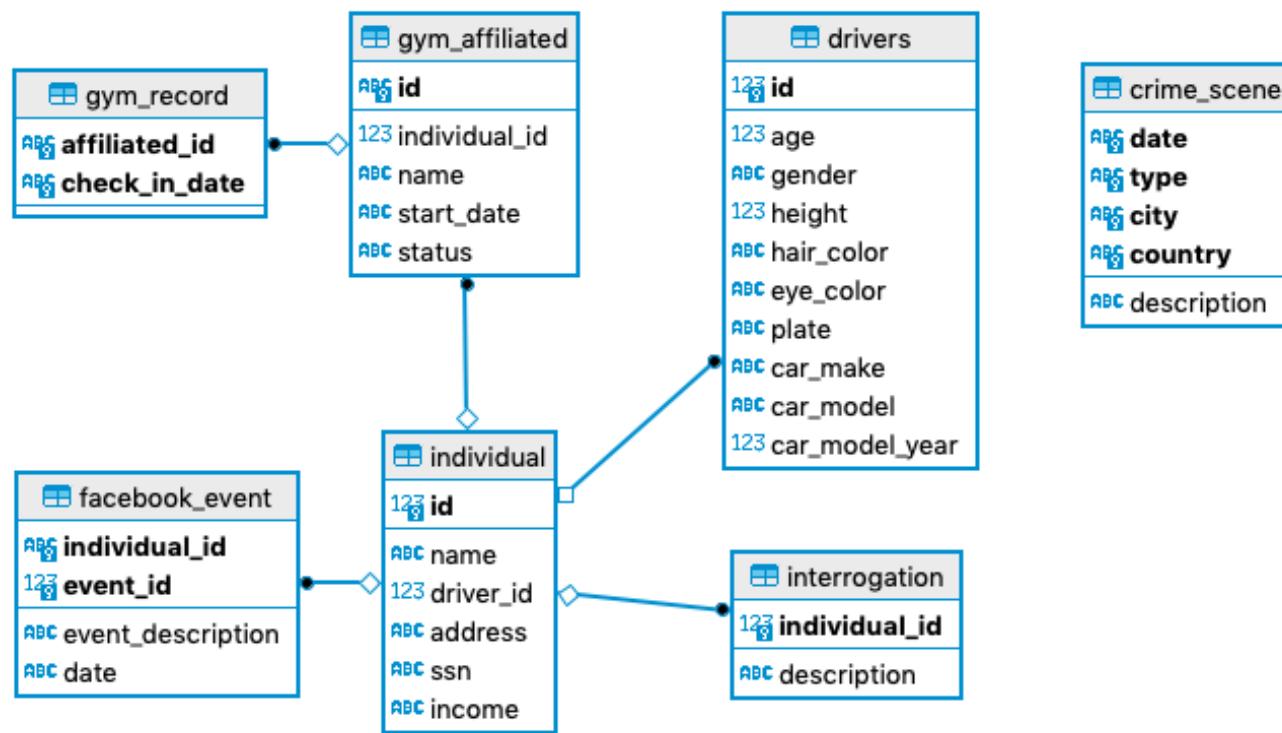
Can you solve the mystery?



Using SQL for your Investigation



Murder Mystery ERD



Query Structure

The `SELECT` clause can specify more than one column.

```
SELECT pet_type, pet_name  
FROM people  
WHERE pet_type = 'dog'  
AND pet_age < 5;
```

Query Structure

Data is filtered by using additional clauses such as WHERE and AND.

```
SELECT pet_type, pet_name  
FROM people  
WHERE pet_type = 'dog'  
AND pet_age < 5;
```

Query Structure

The **WHERE** clause will extract only the data that meets the condition specified.
AND adds a second condition to the original clause, further refining the query.

```
SELECT pet_type, pet_name  
FROM people  
WHERE pet_type = 'dog'  
AND pet_age < 5;
```

Query Structure

Comparing values

```
SELECT pet_type, pet_name  
FROM people  
WHERE pet_type = 'dog'  
AND pet_age < 5;
```

Wildcard: % and _

Use wildcards to substitute zero, one, or multiple characters in a string.
The keyword `LIKE` indicates the use of a wildcard.

```
SELECT *
FROM actor
WHERE last_name LIKE 'Will%';
```

Wildcard: % and _

The % will substitute zero, one, or multiple characters in a query.
In this example, all of the following are matches: Will, Willa, and Willows.

```
SELECT *
FROM actor
WHERE last_name LIKE 'Will%';
```

Wildcard: % and _

The **_** will substitute only **one** character in a query.

_an returns all actors whose first name contains three letters, the second and third of which are **an**.

```
SELECT *
FROM actor
WHERE first_name LIKE '_an';
```

Wildcards: % and _

Wildcards are used to substitute zero, one, or multiple characters in a string. The keyword **LIKE** indicates the use of a wildcard.

```
SELECT *
FROM actor
WHERE last_name LIKE 'Will%';
```

```
SELECT *
FROM actor
WHERE first_name LIKE '_AN';
```

The **%** will substitute **zero, one, or multiple** characters in a query.

For example, all of the following will match: **Will**, **Willa**, and **Willows**.

The **_** will substitute one, and only one, character in a query.

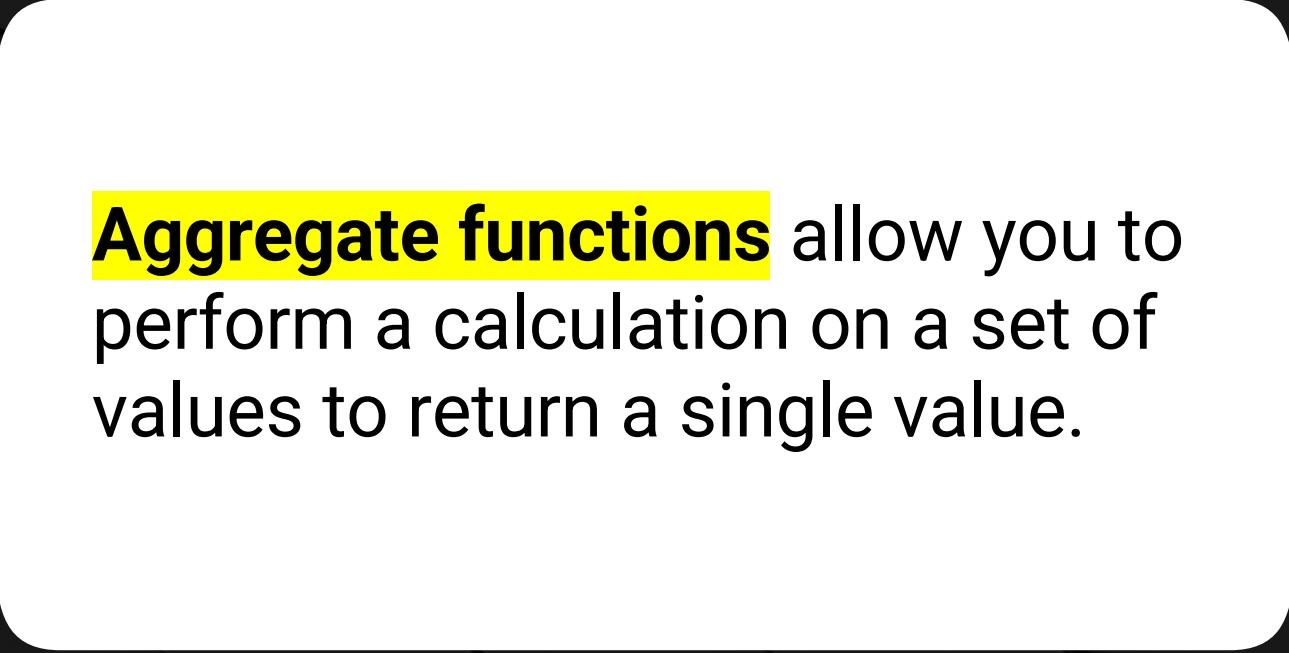
_AN returns all actors whose first name contains three letters, the second and third of which are **AN**.



JOINS

Five Primary Types of Joins used with PostgreSQL

INNER JOIN	Returns records that have matching values in both tables.
LEFT JOIN	Returns all records from the left table and the matched records from the right table.
RIGHT JOIN	Returns all records from the right table and the matched records from the left table.
CROSS JOIN	Returns records that match every row of the left table with every row of the right table. This type of join has the potential to make very large tables.
FULL OUTER JOIN	Places null values within the columns that do not match between the two tables, after an inner join is performed.



Aggregate functions allow you to perform a calculation on a set of values to return a single value.

Aggregate Functions

The most commonly used aggregate functions are:

AVG	Calculates the average of a set of values
COUNT	Counts the rows in a specific table or view
MIN	Returns the minimum value in a set of values
MAX	Returns the maximum value in a set of values
SUM	Calculates the sum of a set of values

Aggregate Functions

Aggregate functions are often used with:

- 01 The **GROUP BY** clause
- 02 The **HAVING** clause
- 03 The **SELECT** statement

Order By Aggregates

The `ORDER BY` function:



Is added towards the end of a query.



Returns in an ascending order by default.



Can also return in a descending order by adding `DESC`.



Can limit the return by adding `LIMIT`.



NOTE: Use the `ROUND` function to round up the number after the decimal.