SEO Tech Developer

Databases and SQL

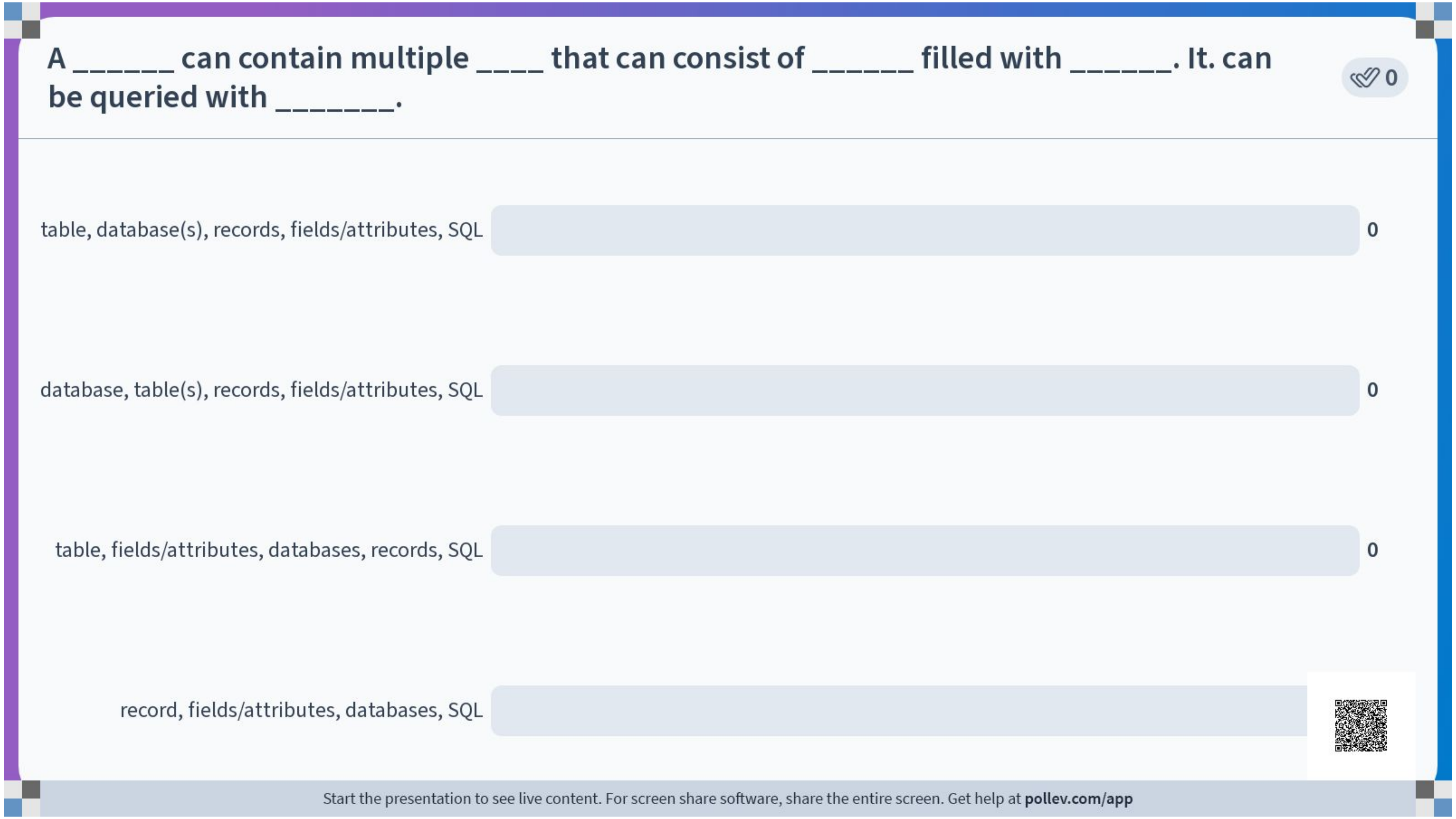
What you will be able to do:

- Describe what a relational database is using the correct vocabulary
- Create and navigate simple SQL databases
- Load data from a python dataframe into a SQL database
- Write SQL queries
- Update existing database

What definitions do you know?

- Database
- SQL
- Table
- Record
- Field/Attribute

polleverywhere



A ______ can contain multiple ____ that can consist of ______ filled with ______. It. can be queried with _______.

table, database(s), records, fields/attributes, SQL 0

database, table(s), records, fields/attributes, SQL 0

table, fields/attributes, databases, records, SQL 0

record, fields/attributes, databases, SQL

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

SEO Tech Developer

Definitions

Database – A set of data tables

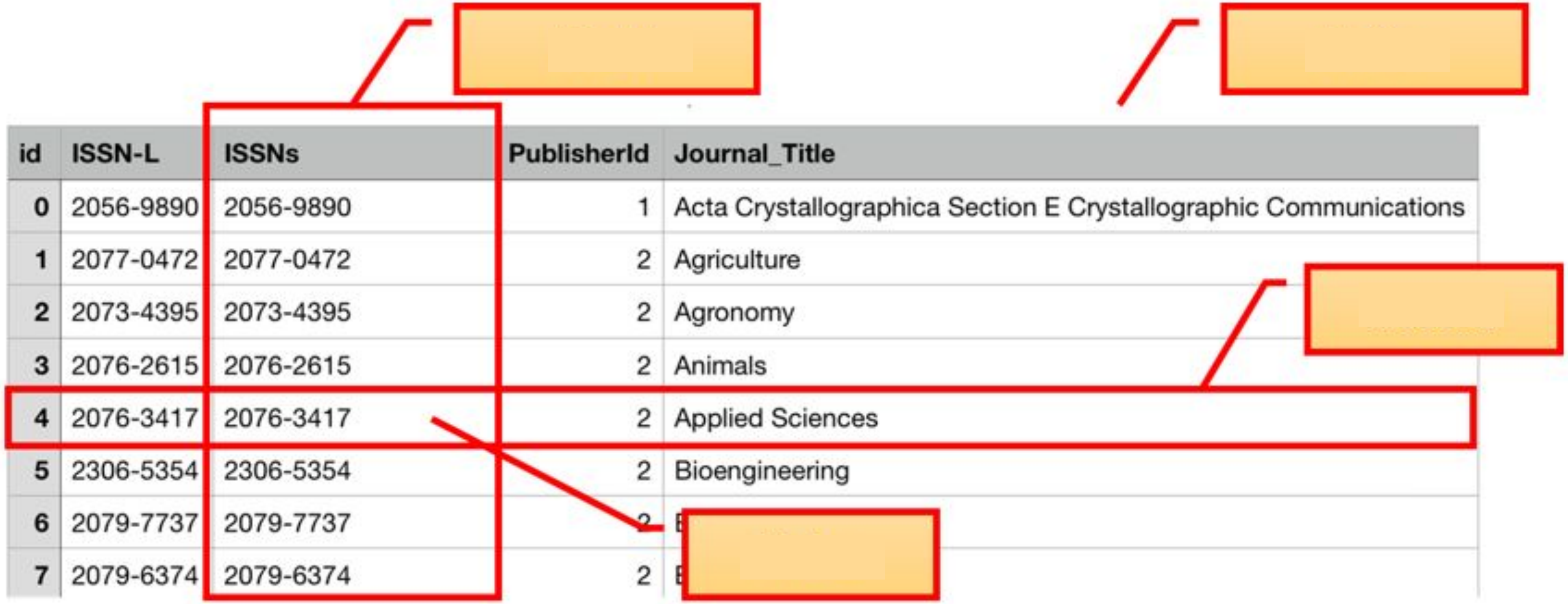
Structured Query Language (SQL) – language for Relational Database Management Systems (RDBMS)

Table – A single set of rows and columns

Record – One data entry – usually represented as a row

Field/Attribute - variables associated with a record – represented as columns

Database Table Diagram

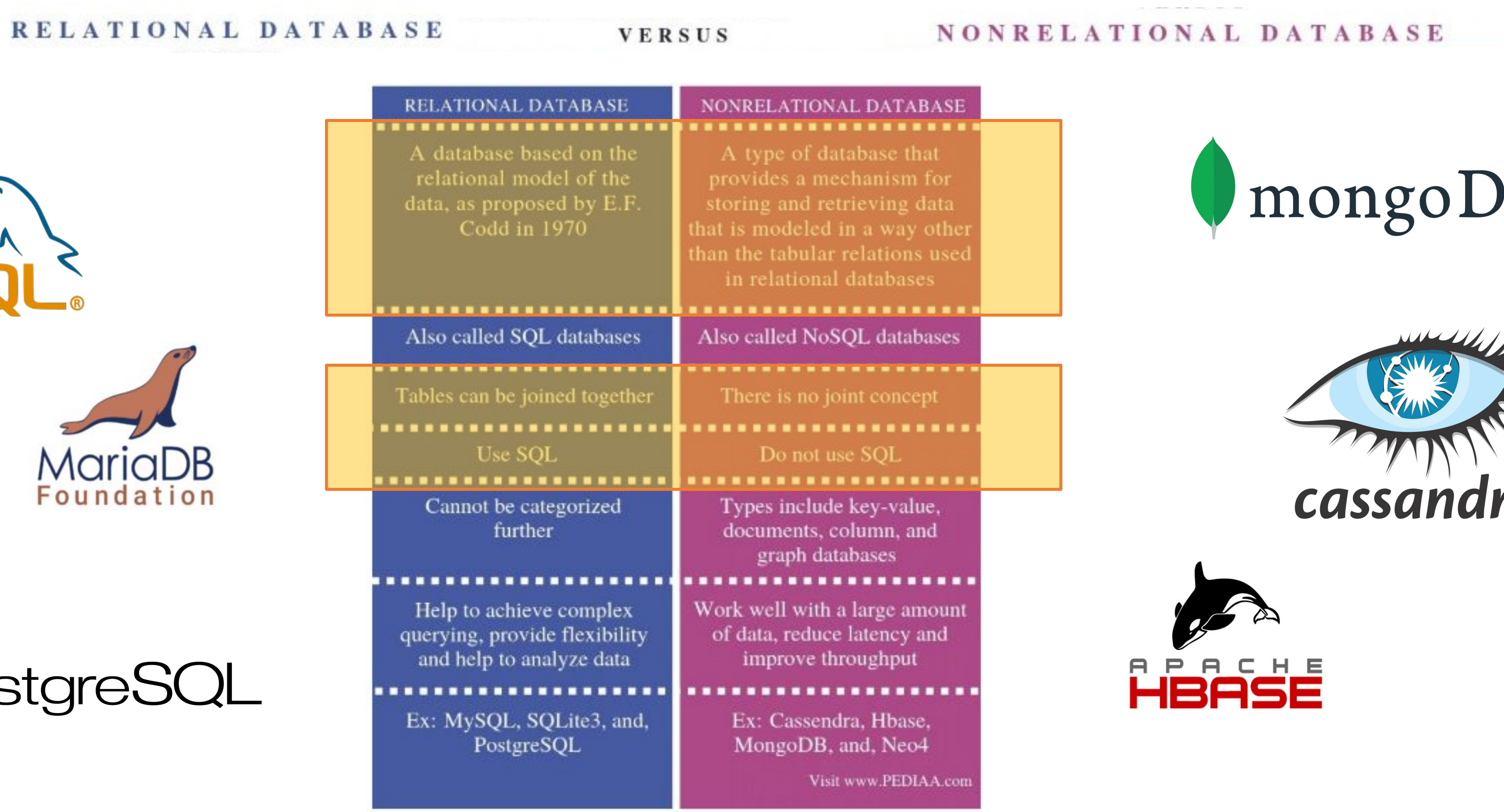


id	ISSN-L	ISSNs	PublisherId	Journal_Title
0	2056-9890	2056-9890	1	Acta Crystallographica Section E Crystallographic Communications
1	2077-0472	2077-0472	2	Agriculture
2	2073-4395	2073-4395	2	Agronomy
3	2076-2615	2076-2615	2	Animals
4	2076-3417	2076-3417	2	Applied Sciences
5	2306-5354	2306-5354	2	Bioengineering
6	2079-7737	2079-7737	2	B
7	2079-6374	2079-6374	2	B

source: /www.fmstudies.org

SEO Tech Developer

Types of Databases



RELATIONAL DATABASE VERSUS NONRELATIONAL DATABASE

RELATIONAL DATABASE	NONRELATIONAL DATABASE
A database based on the relational model of the data, as proposed by E.F. Codd in 1970	A type of database that provides a mechanism for storing and retrieving data that is modeled in a way other than the tabular relations used in relational databases
Also called SQL databases	Also called NoSQL databases
Tables can be joined together	There is no joint concept
Use SQL	Do not use SQL
Cannot be categorized further	Types include key-value, documents, column, and graph databases
Help to achieve complex querying, provide flexibility and help to analyze data	Work well with a large amount of data, reduce latency and improve throughput
Ex: MySQL, SQLite3, and, PostgreSQL	Ex: Cassendra, Hbase, MongoDB, and, Neo4

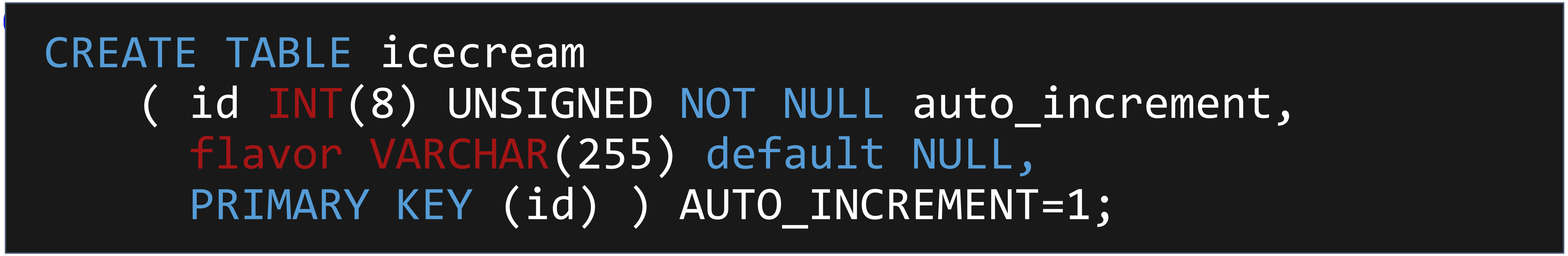
Visit www.PEDIAA.com

source:https://miro.medium.com/

Creating Databases, Tables and Records

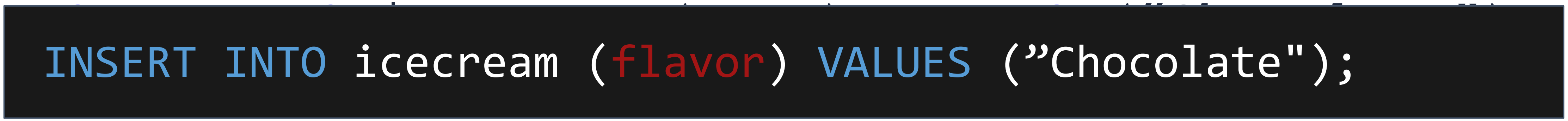
- `CREATE DATABASE database_name;` - create a database

- `CREATE TABLE table_name (field types);`- create a table in the selected database



```
CREATE TABLE icecream
    ( id INT(8) UNSIGNED NOT NULL auto_increment,
      flavor VARCHAR(255) default NULL,
      PRIMARY KEY (id) ) AUTO_INCREMENT=1;
```

- `INSERT INTO table_name (fields) VALUES (values);` - insert a record into the specified table



```
INSERT INTO icecream (flavor) VALUES (”Chocolate");
```

Altering Tables

- ALTER TABLE can perform 3 types of operations:

1. ADD - adds columns to an existing table

```
ALTER TABLE table ADD field type;
```

2. CHANGE - updates the name of an existing table

```
ALTER TABLE table
   CHANGE field_old_name field_new_name type;
```

3. DROP - deletes or removes columns (and their data)

SQL Query

- Selecting Columns

```
    SELECT column1, column2 FROM table_name;
```

- Sorting

```
    ORDER BY column_name ASC
    ORDER BY column_name DESC
    GROUP BY column_name
```

- Filtering with WHERE

```
   SELECT * FROM table_name
    WHERE column OPERATOR data;
```

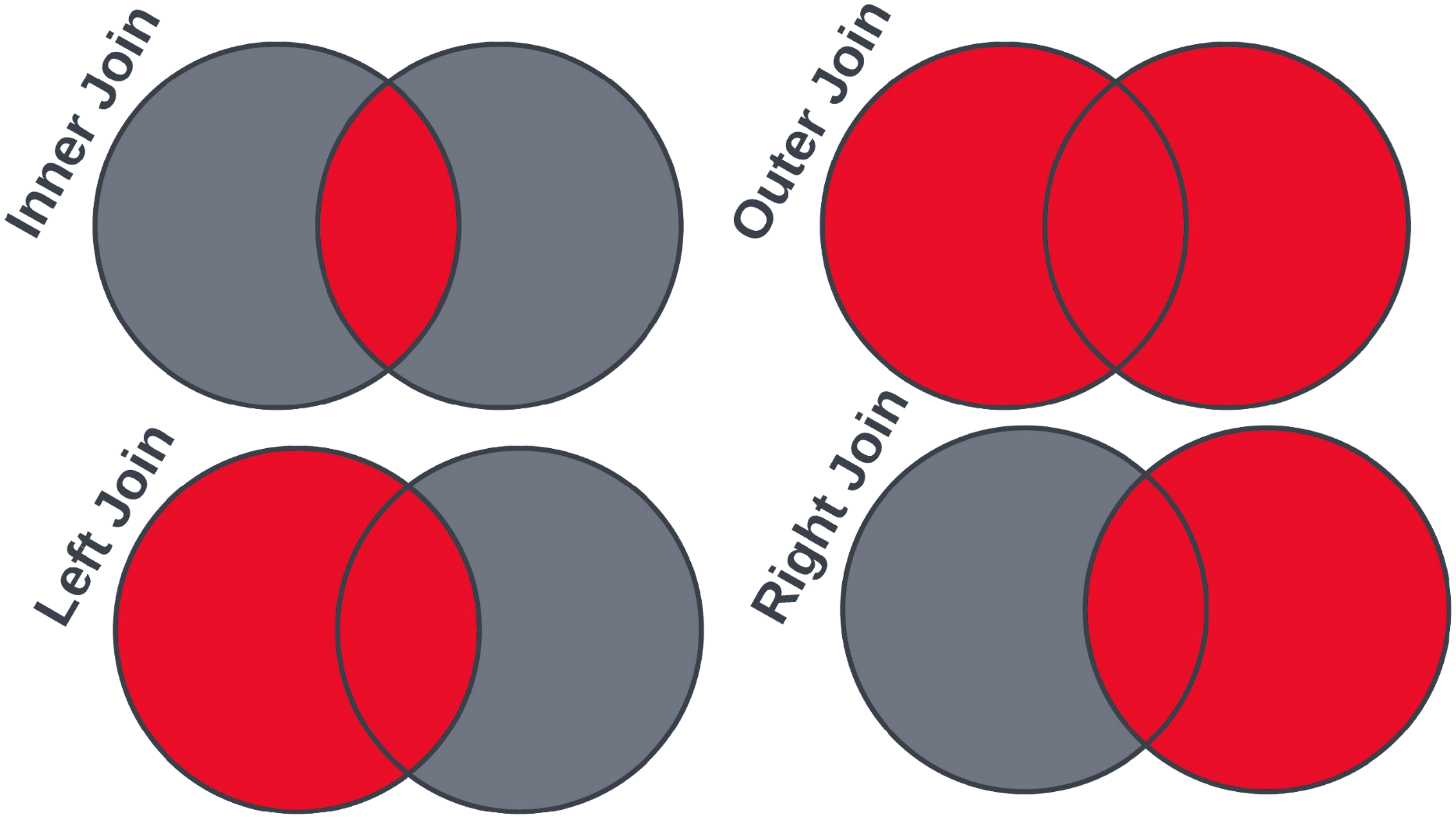
Other Basic SQL Commands

- `SHOW DATABASES;` - view list of databases
- `USE database_name;` - select a database to use
- `SHOW TABLES;` - view list of tables in selected database
- `SELECT * FROM table_name;` - view records in table

Joining Tables

```
SELECT column_from_a
FROM table_A
JOIN TYPE table_ B
ON column_from_b =
column_from_a;
```

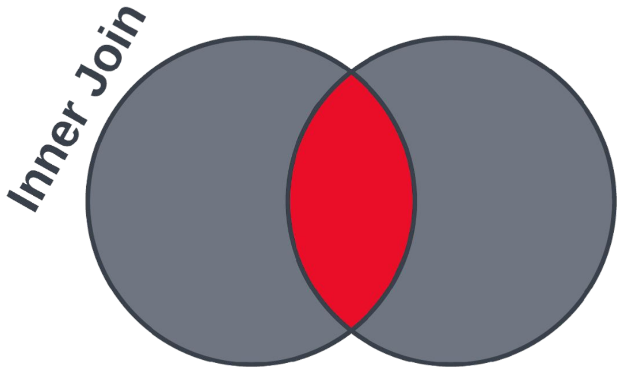
Join Types



SEO Tech Developer

Inner Join

```
SELECT Customers.customer_id,
Customers.first_name, Orders.amount
FROM Customers
INNER JOIN Orders
ON Customers.customer_id =
Orders.customer;
```



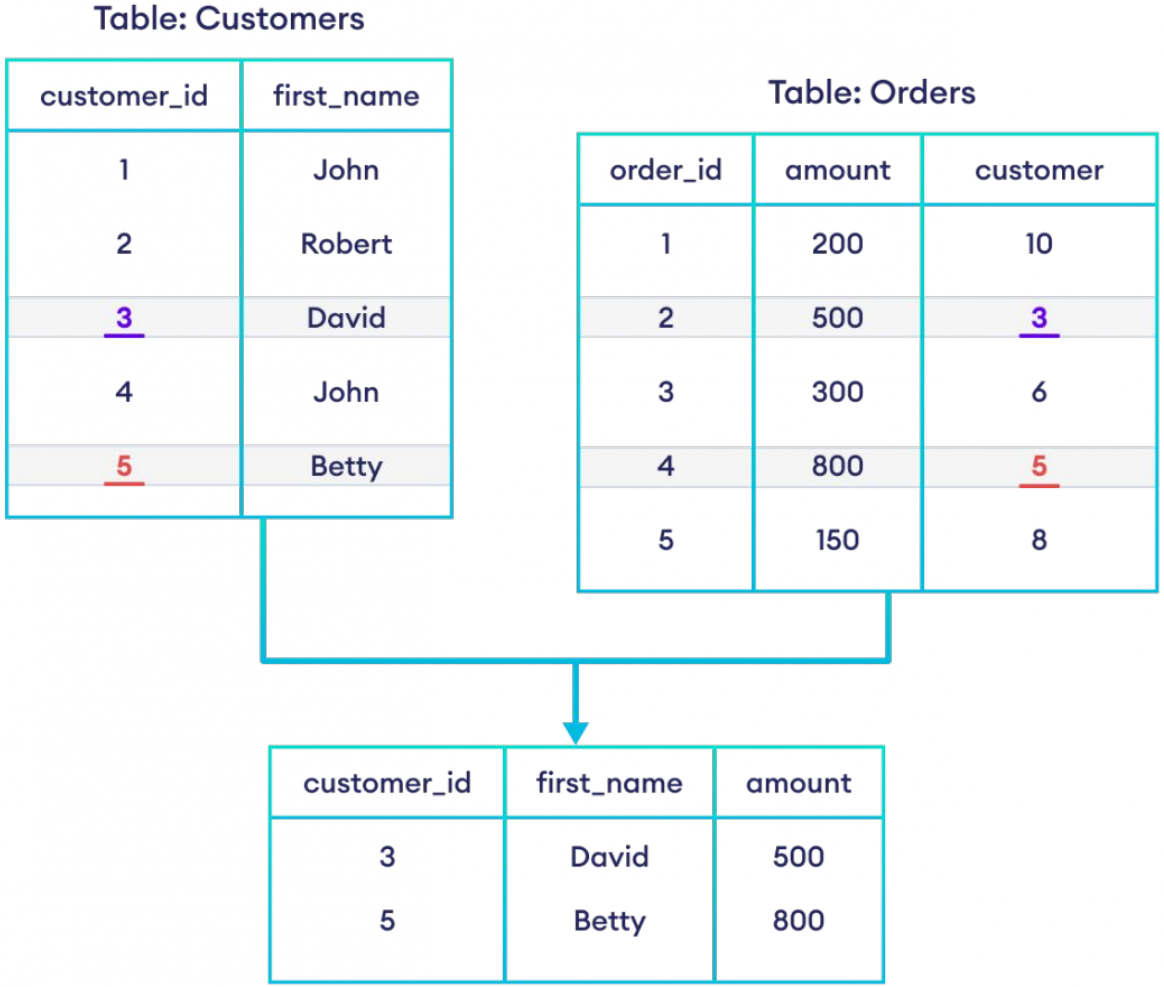


Table: Customers

customer_id	first_name
1	John
2	Robert
3	David
4	John
5	Betty

Table: Orders

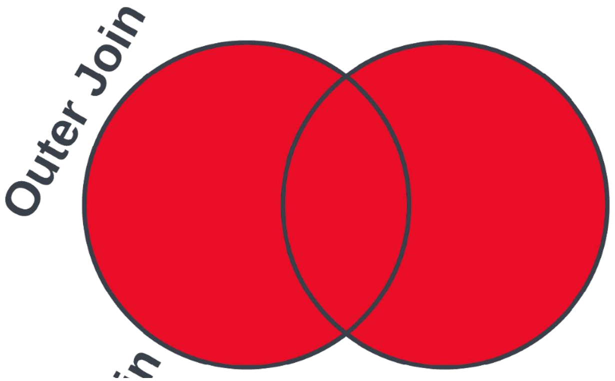
order_id	amount	customer
1	200	10
2	500	3
3	300	6
4	800	5
5	150	8

customer_id	first_name	amount
3	David	500
5	Betty	800

source: https://www.programiz.com/sql/inner-join

Outer Join

```
SELECT Customers.customer_id,
Customers.first_name, Orders.amount
FROM Customers
FULL OUTER JOIN Orders
ON Customers.customer_id =
Orders.customer;
```



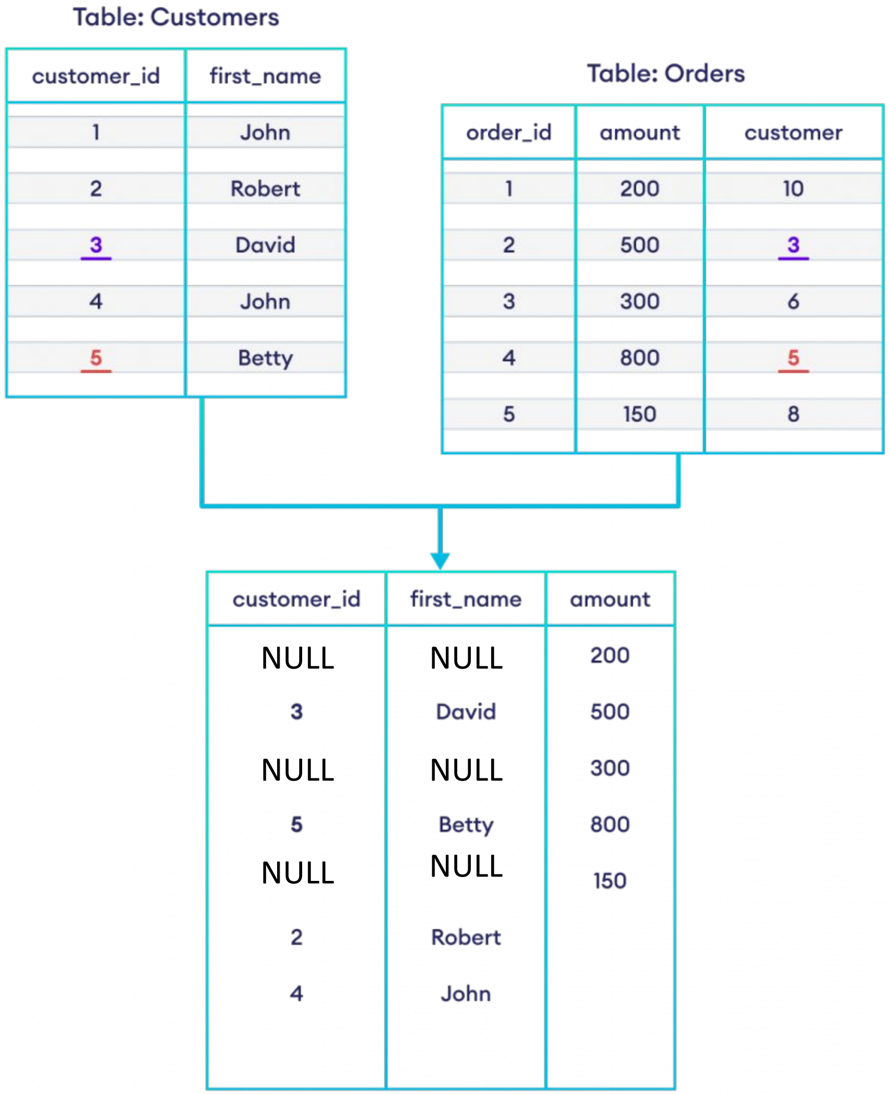


Table: Customers

customer_id	first_name
1	John
2	Robert
3	David
4	John
5	Betty

Table: Orders

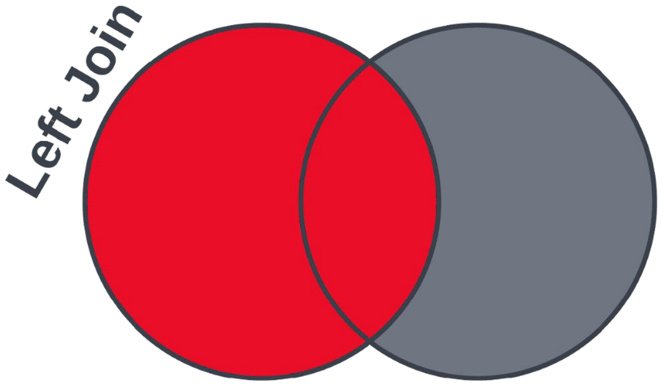
order_id	amount	customer
1	200	10
2	500	3
3	300	6
4	800	5
5	150	8

customer_id	first_name	amount
NULL	NULL	200
3	David	500
NULL	NULL	300
5	Betty	800
NULL	NULL	150
2	Robert	
4	John	

source: https://www.programiz.com/sql/inner-join

Left Join

```
SELECT Customers.customer_id,
Customers.first_name, Orders.amount
FROM Customers
LEFT JOIN Orders
ON Customers.customer_id =
Orders.customer;
```



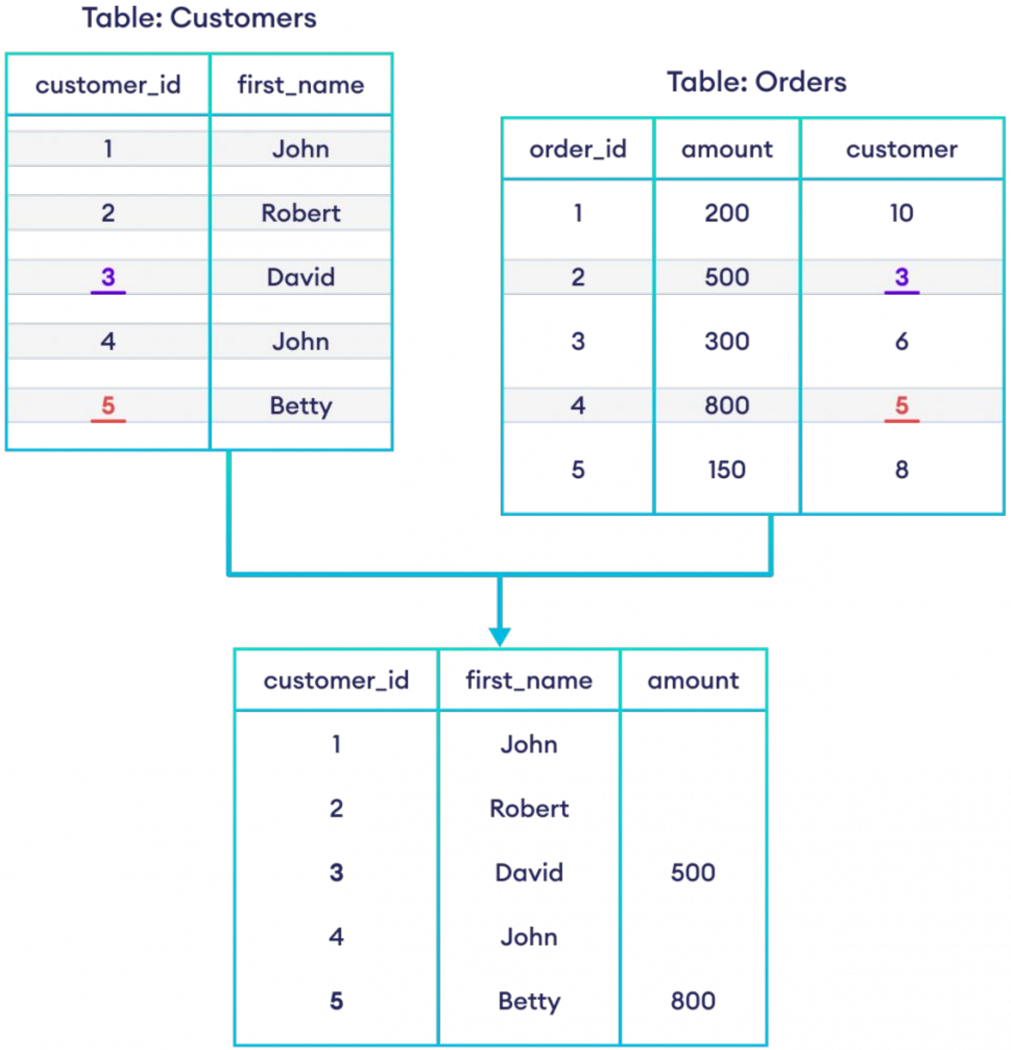


Table: Customers

customer_id	first_name
1	John
2	Robert
3	David
4	John
5	Betty

Table: Orders

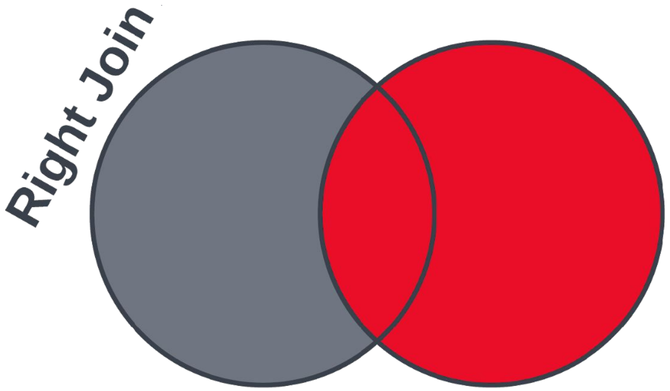
order_id	amount	customer
1	200	10
2	500	3
3	300	6
4	800	5
5	150	8

customer_id	first_name	amount
1	John	
2	Robert	
3	David	500
4	John	
5	Betty	800

source: https://www.programiz.com/sql/left-join

Right Join

```
SELECT Customers.customer_id,
Customers.first_name, Orders.amount
FROM Customers
RIGHT JOIN Orders
ON Customers.customer_id =
Orders.customer;
```



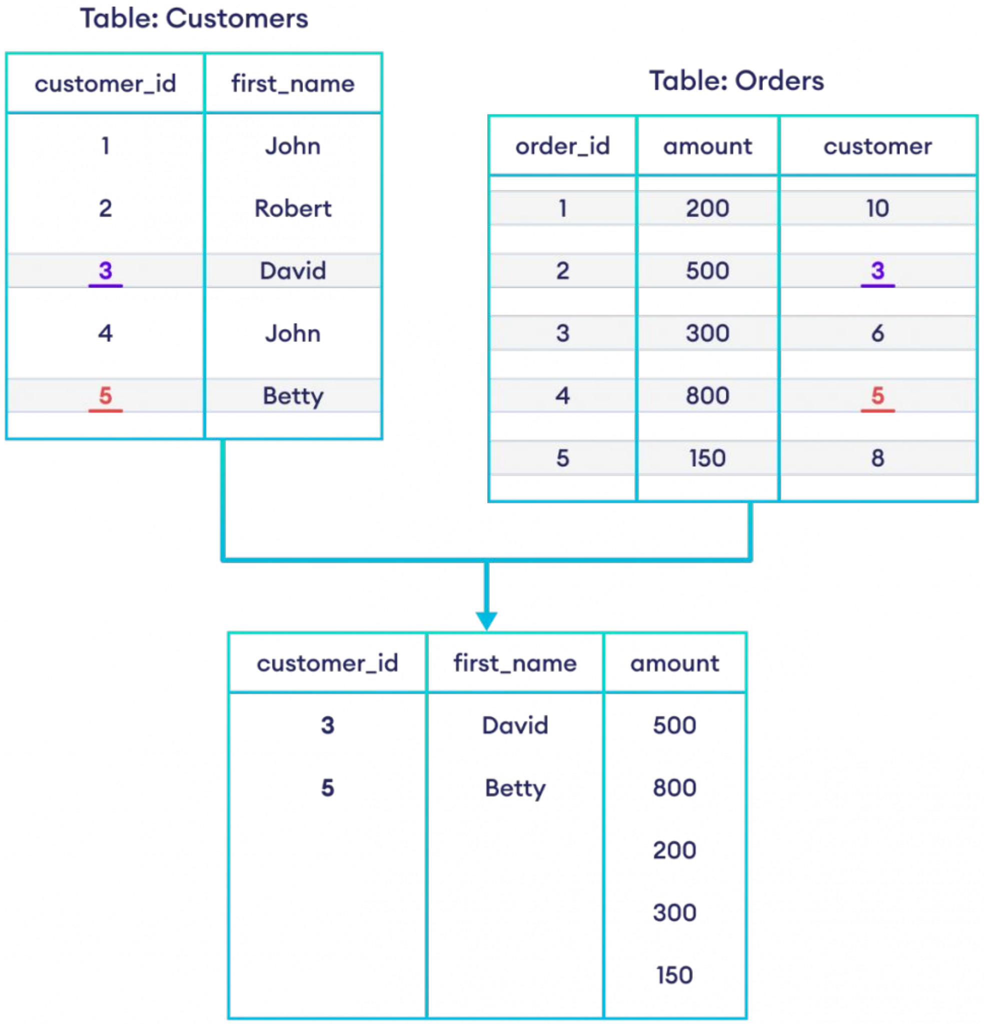


Table: Customers

customer_id	first_name
1	John
2	Robert
3	David
4	John
5	Betty

Table: Orders

order_id	amount	customer
1	200	10
2	500	3
3	300	6
4	800	5
5	150	8

customer_id	first_name	amount
3	David	500
5	Betty	800
		200
		300
		150

source: https://www.programiz.com/sql/right-join

SQLite, SQLAlchemy, and More

Definitions

pandas – an open source Python package that is most widely used for data science/data analysis and machine learning tasks.

SQLAlchemy – a Python SQL toolkit and Object Relational Mapper that makes it easier for developers to use SQL and manipulate databases

sqlite3 – A standalone command-line shell program provided by SQLite, and recently a Python package too!

SQlite - is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine.

SQLite3 Commands

SQLite stores the whole database as a single cross-platform file

- `sqlite3` - start interactive shell
- `sqlite3 file_name.db` - start shell using specified database

Note: if you want to play around with SQLite via command line, you may have to run the commands (codio does this for you):

```
sudo apt update
sudo apt install sqlite3
```

SQL Command line

```
codio@@minutecaravan-miguellabor:~/workspace$ sqlite3 demo1.db
SQLite version 3.37.2 2022-01-06 13:25:41
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> <SOME SQL COMMAND>
sqlite>
```

DEMO

Using sqlite3 | command line

SQLite3 + Python

```
import sqlite3

conn = sqlite3.connect('<DB_NAME>')
c = conn_obj.cursor()

# Run a command
c.execute("<SQL_COMMAND>")

# Example command
# c.execute("SELECT * FROM <DB_NAME>;")
# print(c.fetchall())

# Ensure any changes to table persists
conn.commit()

# Close the connection
conn.close()
```

DEMO

Using sqlite3 | python

SEO Tech Developer

SQL + Python = SQLAlchemy

```
import sqlalchemy as db
import pandas as pd

# Get some data from JSON…

engine = db.create_engine('sqlite:///DB_NAME’)

dataframe_name.to_sql('table_name', con=engine, if_exists='replace', index=False)

with engine.connect() as connection:
query_result = connection.execute( \
             db.text("<SQL COMMAND>")).fetchall()

# Example Query and print
# query_result = connection.execute( \
             db.text(SELECT * FROM table_name;)).fetchall()

# print(pd.DataFrame(query_result))
```

DEMO

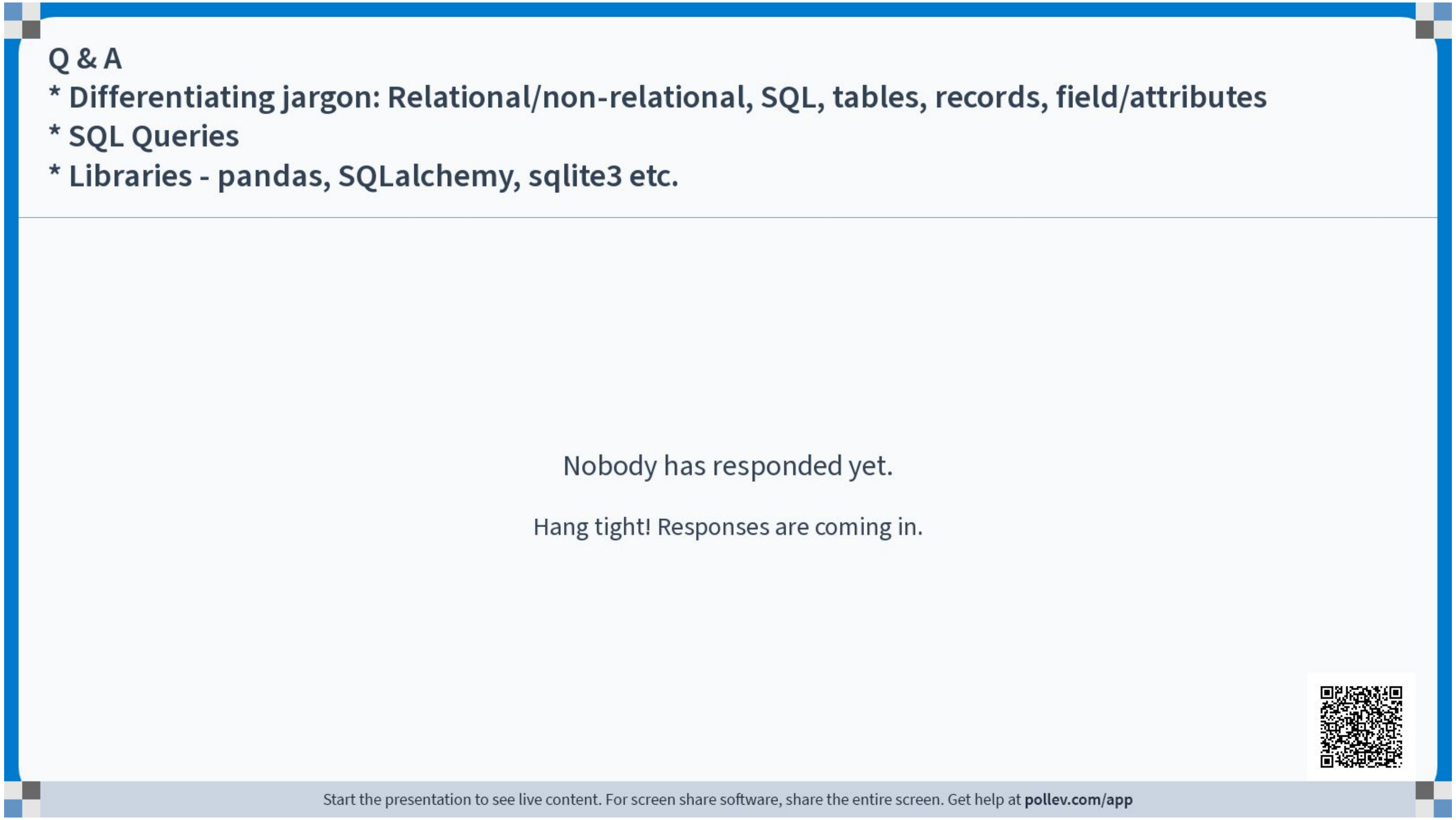
Using sqlalchemy and pandas

SEO Tech Developer

Reminders

- Check your Google Calendars for this week’s events!
- We will be meeting tomorrow!

polleverywhere



Q & A
* Differentiating jargon: Relational/non-relational, SQL, tables, records, field/attributes
* SQL Queries
* Libraries - pandas, SQLalchemy, sqlite3 etc.

Nobody has responded yet.

Hang tight! Responses are coming in.

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

SEO Tech Developer

Thank you!