SQL For Data Science: A Beginner Guide!

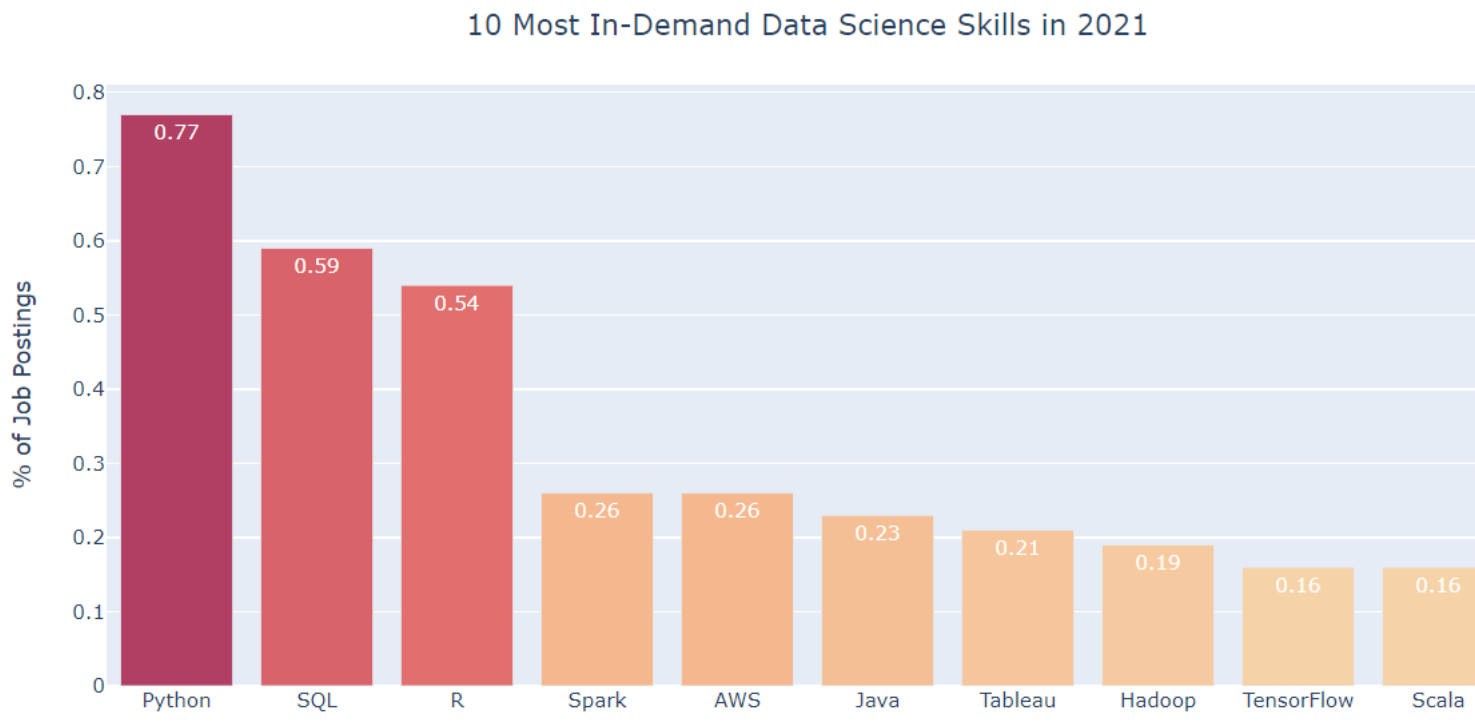
BEGINNER DATA ENGINEERING PYTHON SQL

Introduction

Data Science is a most emerging field with numerous job opportunities. We all must have been heard about the topmost Data Science skills. To start with, the easiest, as well as an essential skill that every data science aspirant should acquire, is SQL.

Nowadays, most companies are going towards being data-driven. These data are stored in a database and are managed and processed through a Database Management system. DBMS makes our work so easy and organized. Hence, it is essential to integrate the most popular programming language with the incredible DBMS tool.

SQL is the most widely used programming language while working with databases and supported by various relational database systems, like MySQL, SQL Server, and Oracle. However, the SQL standard has some features that are implemented differently in different database systems. Thus, SQL becomes one of the most important concepts to be learned in this field of Data Science.



This article was published as a part of the Data Science Blogathon

Need of SQL in Data Science

SQL (Structured Query Language) is used for performing various operations on the data stored in the databases like updating records, deleting records, creating and modifying tables, views, etc. SQL is also the standard for the current big data platforms that use SQL as their key API for their relational databases.

Data Science is the all-around study of data. To work with data, we need to extract it from the database. This is where SQL comes into the picture. Relational Database Management is a crucial part of Data Science. A Data Scientist can control, define, manipulate, create, and query the database using SQL commands.

Many modern industries have equipped their products data management with NoSQL technology but, SQL remains the ideal choice for many business intelligence tools and in-office operations.

Many of the Database platforms are modeled after SQL. This is why it has become a standard for many database systems. Modern big data systems like Hadoop, Spark also make use of SQL only for maintaining the relational database systems and processing structured data.

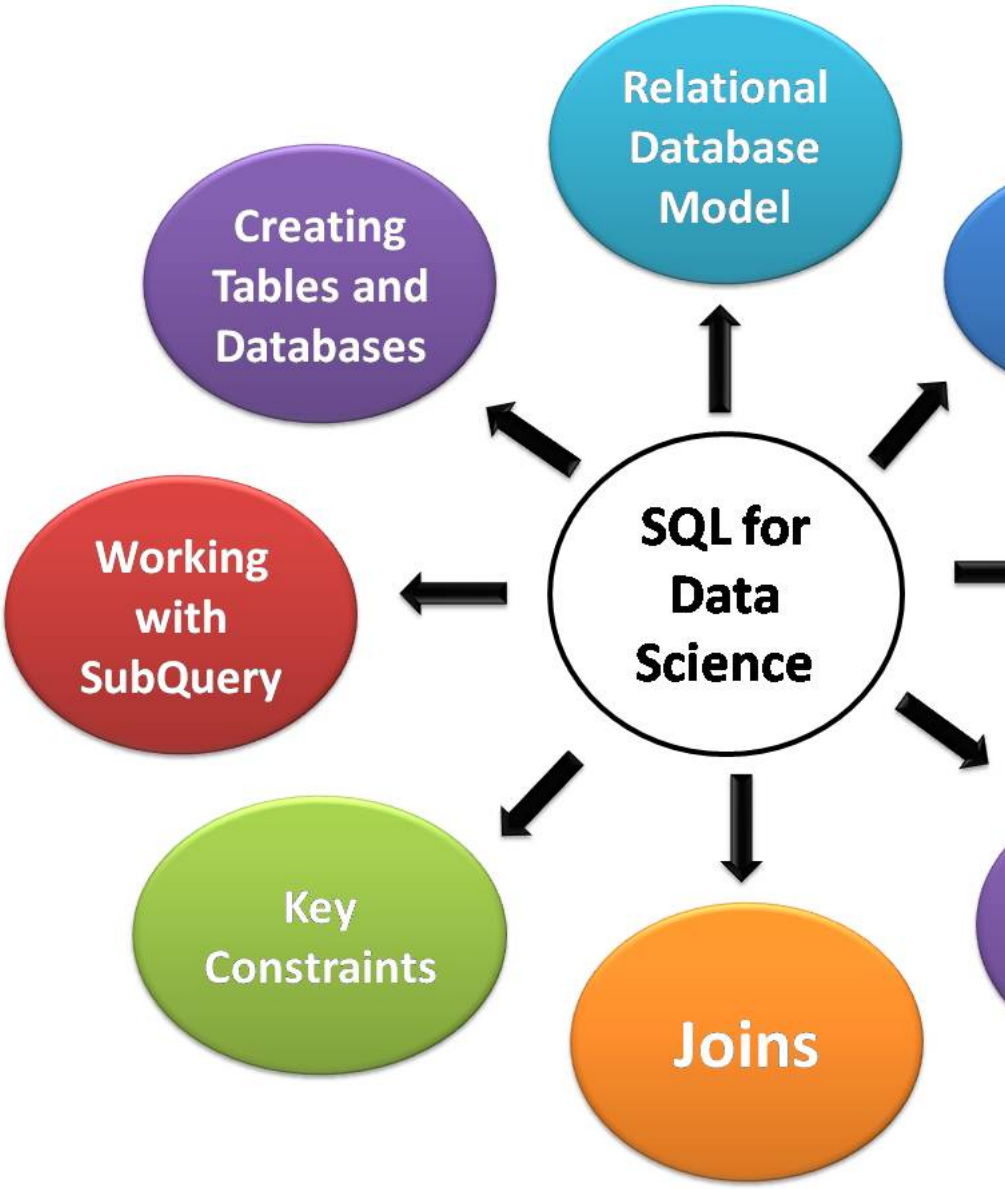
We can say that:

- A **Data Scientist** needs SQL to handle structured data. As the structured data is stored in relational databases. Therefore, to query these databases, a data scientist must have a good knowledge of SQL commands.
- Big Data Platforms like Hadoop and Spark provide an extension for querying using SQL commands for manipulating.
- SQL is the standard tool to experiment with data through the creation of test environments.
- To perform analytics operations with the data that is stored in relational databases like Oracle, Microsoft SQL, MySQL, we need SQL.
- SQL is also an essential tool for data wrangling and preparation. Therefore, while dealing with various Big Data tools, we make use of SQL.

Also Read: ***Comprehensive SQL Guide: From Basics to Advanced Levels***

Key elements of SQL for Data Science

Following are the key aspects of SQL which are most useful for Data Science. Every aspiring Data Scientists must know these necessary SQL skills and features.



Introduction to SQL with Python

As we all know that SQL is the most used Database Management Tool and Python is the most popular Data Science Language for its flexibility and wide range of libraries. There are various ways to use SQL with Python. Python provides multiple libraries that are developed and can utilize for this purpose. SQLite, PostgreSQL, and MySQL are examples of these libraries.

Why use SQL with Python

There are many use cases for when Data Scientists want to connect Python to SQL. Data Scientists need to connect a SQL database so that data coming from the web application can be stored. It also helps to communicate between different data sources.

There is no need to switch between different programming languages for data management. It makes Data scientists' work more convenient. They will be able to use your Python skills to manipulate data stored in a SQL database. They don't need a CSV file.

MySQL with Python

MySQL is a server-based database management system. One MySQL server can have multiple databases. A MySQL database consist two-step process for creating a database:

- Make a connection to a MySQL server.
- Execute separate queries to create the database and process data.

Let's get started with MySQL with python

First, we will create a connection between the MySQL server and MySQL DB. For this, we will define a function that will establish a connection to the MySQL database server and will return the connection object:

```
!pip install mysql-connector-python import mysql.connector from mysql.connector import Error def create_connection(host_name, user_name, user_password):  connection = None  try:  connection = mysql.connector.connect(  host=host_name,  user=user_name,  passwd=user_password  )  print("Connection to MySQL DB successful")  except Error as e:  print(f"The error '{e}' occurred")  return connection connection = create_connection("localhost", "root", "")
```

In the above code, we have defined a function create_connection() that accepts following three parameters:

- host_name
- user_name
- user_password

The mysql.connector is a Python SQL module that contains a method .connect() that is used to connect to a MySQL database server. When the connection is established, the connection object created will be returned to the calling function.

So far the connection is established successfully, now let's create a database.

```
#we have created a function to create database that contions two parameters #connection and query def create_database(connection, query): #now we are creating an object cursor to execute SQL queries cursor = connection.cursor() try: #query to be executed will be passed in cursor.execute() in string form cursor.execute(query) print("Database created successfully") except Error as e: print(f"The error '{e}' occurred") #now we are creating a database named example_app create_database_query = "CREATE DATABASE example_app" create_database(connection, create_database_query) #now will create database example_app on database server #and also cretae connection between database and server def create_connection(host_name, user_name, user_password, db_name): connection = None try: connection = mysql.connector.connect( host=host_name, user=user_name, passwd=user_password, database=db_name ) print("Connection to MySQL DB successful") except Error as e: print(f"The error '{e}' occurred") return connection #calling the create_connection() and connects to the example_app database. connection = create_connection("localhost", "root", "", "example_app")
```

SQLite

SQLite is probably the most uncomplicated database we can connect to a Python application since it's a built-in module we don't need to install any external Python SQL modules. By default, Python installation contains a Python SQL library named sqlite3 that can be used to interact with an SQLite database.

SQLite is a serverless database. It reads and writes data to a file. That means we don't even need to install and run an SQLite server to perform database operations like MySQL and PostgreSQL!

Let’s use sqlite3 to connect to an SQLite database in Python:

```
import sqlite3 from sqlite3 import Error def create_connection(path): connection = None try: connection = sqlite3.connect(path) print("Connection to SQLite DB successful") except Error as e: print(f"The error '{e}' occurred") return connection
```

In the above code, we have imported sqlite3 and the module’s Error class. Then define a function called .create_connection() that will accept the path to the SQLite database. Then .connect() from the sqlite3 module will take the SQLite database path as a parameter. If the database exists at the path specified in .connect, then a connection to the database will be established. Otherwise, a new database is created at the specified path, and then a connection is established.

sqlite3.connect(path) will return a connection object, which was also returned by create_connection(). This connection object will be used to execute SQL queries on an SQLite database. The following line of code will create a connection to the SQLite database:

```
connection = create_connection("E:\example_app.sqlite")
```

Once the connection is established we can see the database file is created in the root directory and if we want, we can also change the location of the file.

In this article, we discuss how SQL is essential for Data Science and also how we can work with SQL using python. Thanks for reading. Do let me know your comments and feedback in the comment section.

Conclusion

SQL is extremely important for data science. It allows you to work with structured data stored in databases. As a data scientist, you need SQL to extract, manipulate, and analyze data from these databases. Big data tools like Hadoop and Spark also use SQL for processing structured data. Learning SQL skills like querying, joining tables, filtering, and aggregating data is crucial. Combining SQL with Python makes data management and analysis convenient without switching between languages. Whether dealing with relational databases or working with big data, mastering SQL is essential for any aspiring data scientist. It empowers you to wrangle, prepare, and gain insights from data effectively.

Frequently Asked Question?

Q1. Is SQL used in data science?

A. Yes, SQL (Structured Query Language) is commonly used in data science for data manipulation, querying databases, and extracting insights from structured data.

Q2. Should I learn SQL or Python for data science?

A. Both SQL and Python are valuable for data science, but they serve different purposes. SQL is essential for handling and querying structured data in databases, while Python is widely used for data analysis, visualization, machine learning, and other tasks in data science. Learning both SQL and Python would be beneficial for a well-rounded data science skill set.

Q3. Is SQL worth learning for data science?

A. Yes, learning SQL is worth it for data science, especially if you’re working with databases or structured data. It provides a powerful tool for data manipulation, aggregation, filtering, and joining, which are fundamental tasks in data analysis and reporting.

Q4. How to start SQL for data science?

A. To start learning SQL for data science, you can begin with online tutorials, courses, or books that cover the basics of SQL syntax, querying databases, and data manipulation operations such as SELECT, JOIN, WHERE, GROUP BY, and ORDER BY. Additionally, practicing on real datasets and projects can help reinforce your learning and improve your SQL skills.

Article Url - https://www.analyticsvidhya.com/blog/2021/06/sql-for-data-science-a-beginners-guide/

Neelu Tiwari