

# Database Connections and Data Extraction with Python

## Connecting to Relational Databases with Python



**Pinal Dave**

Data Expert

@pinaldave | [blog.sqlauthority.com](http://blog.sqlauthority.com)

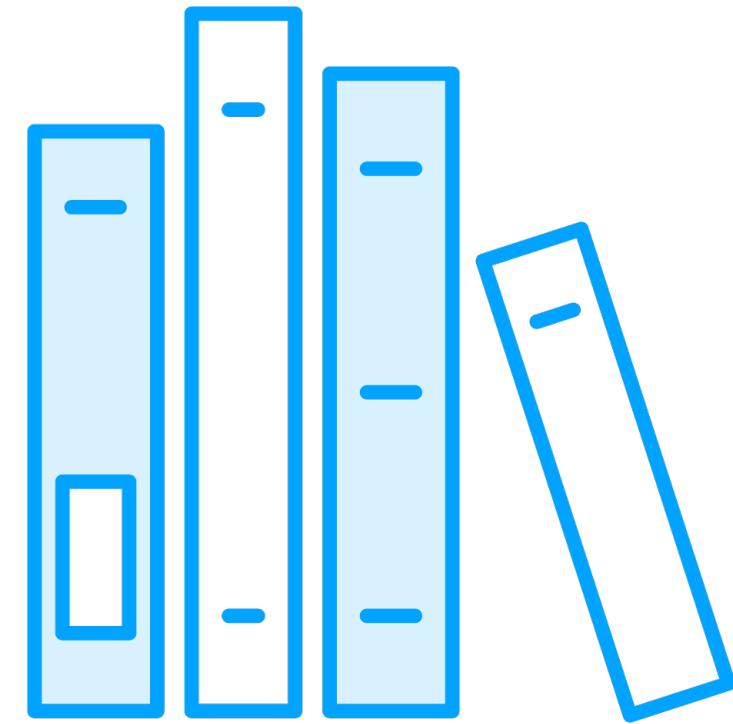
# What This Course Covers

This course will teach you how to establish, manage, and optimize database connections using Python. You'll explore relational and NoSQL databases, execute queries efficiently, and extract data for analysis.

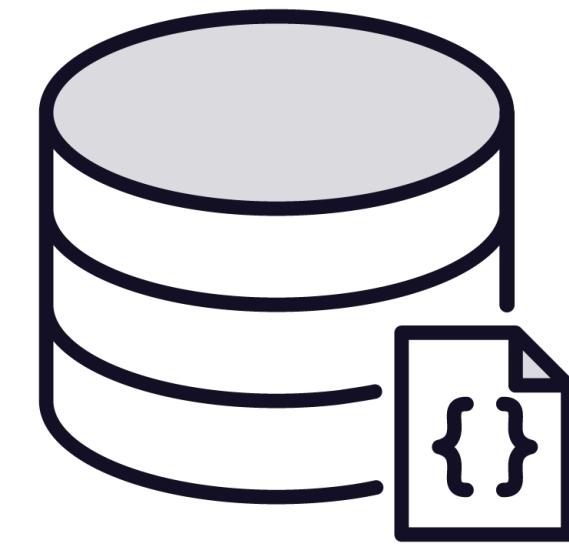
To keep things practical, we'll focus on MySQL for relational databases and MongoDB for NoSQL, while covering essential details for others.



# Python's Role in Database Connectivity



Offers multiple  
database libraries



Works with both  
relational (SQL) and  
NoSQL databases



Great for automation  
and data analysis



# Popular Python Database Libraries - Relational

Library	Database supported	Key features
<code>psycopg2</code>	<b>PostgreSQL</b>	<b>Native PostgreSQL adapter, fast execution</b>
<code>SQLAlchemy</code>	<b>Multiple SQL databases</b>	<b>ORM-based, supports MySQL, PostgreSQL, SQLite, etc.</b>
<code>pyodbc</code>	<b>Microsoft SQL Server &amp; more</b>	<b>Supports ODBC-compatible databases (SQL Server, Oracle, etc.)</b>
<code>mysql-connector-python</code>	<b>MySQL</b>	<b>Official MySQL driver, direct connection</b>



# Popular Python Database Libraries – NoSQL

Library	Database supported	Key features
pymongo	MongoDB	Direct MongoDB interaction, supports CRUD operations
boto3	AWS DynamoDB	AWS SDK for Python, supports scalable NoSQL operations
cassandra-driver	Apache Cassandra	Works with Cassandra, optimized for distributed queries
redis-py	Redis	Supports key-value store, caching, and message brokering



# How Python Connects to Databases

1<sup>ST</sup>

**Install necessary libraries** (pip install library)

2<sup>ND</sup>

**Establish a database connection**

3<sup>RD</sup>

**Execute queries** (SELECT, INSERT, UPDATE, DELETE)

4<sup>TH</sup>

**Handle errors and optimize performance**

