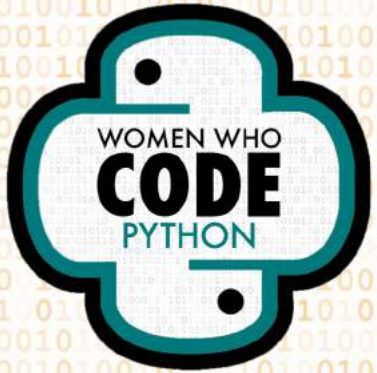
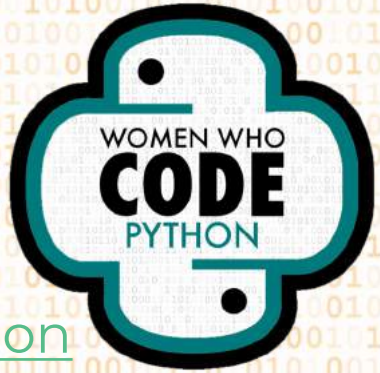
Women Who Code Python Track

Databases with Python Series



Session#4 - MongoDB with Python

Welcome Everyone!!



- The slides available here on GitHub: https://github.com/WomenWhoCode/WWCodePython
- Our social media and events here: https://linktr.ee/wwcodepython
- Please make sure your chat is set to "**All panelists and attendees**".
- Few housekeeping rules:
  - Everyone will be muted throughout the webinar!
  - Please share your thoughts on the chat and/or ask questions in the Q&A.
  - Our team is here. Please reach out to us with any technical questions!

WELCOME

WOMEN WHO CODE

Hi! We are ...

Hala

WWCode Python Volunteer

Ramya

WWCode Python Lead

Karen

WWCode Python Lead

OUR MISSION

Inspiring women to excel in technology careers.

WOMEN WHO CODE

OUR VISION

A world where women are representative as technical executives, founders, VCs, board members and software engineers.

WOMEN WHO CODE

OUR TARGET

Engineers with two or more years of experience looking for support and resources to strengthen their influence and levelup in their careers.

WOMEN WHO CODE

CODE OF CONDUCT

WWCode is an inclusive community, dedicated to providing an empowering experience for everyone who participates in or supports our community, regardless of gender, gender identity and expression, sexual orientation, ability, physical appearance, body size, race, ethnicity, age, religion, socioeconomic status, caste, creed, political affiliation, or preferred programming language(s).

Our events are intended to inspire women to excel in technology careers, and anyone who is there for this purpose is welcome. We do not tolerate harassment of members in any form. Our **Code of Conduct** applies to all WWCode events and online communities.

Read the full version and access our incident report form at **womenwhocode.com/codeofconduct**

WOMEN WHO CODE

250,000+

Members

In 95 cities and 122 countries
with 70 networks,
10K+ events,
$1025 daily Conference tickets,
$2M Scholarships and
Access to jobs + resources
Infinite connections

WOMEN WHO CODE

OUR MOVEMENT

As the world changes, we can be a connecting force that creates a sense of belonging while the world is being asked to isolate.

WOMEN WHO CODE

AGENDA – Session#4

- Overview of previous sessions
- Difference b/w SQL vs NoSQL | Examples
- Its real-time applications and Use cases
- Terminologies of NoSQL and SQL
- Data Types in MongoDB
- Overview of Mongo Shell and Compass
- CRUD implementation with MongoDB & Python - programmatically

Overview – previous sessions

SQL + NoSQL with **Python** Series

#1 - SQL and Introduction	#2 - SQLite	#3 - Firebase
Database	It's Syntaxes with Python	It's Syntaxes with Python
Relational Database	GUI software & installation	Google Colab & Implementation
Data Model	Sample Database	NoSQL with Firebase
RDBMS & Designing	SQL Data Types	Use-cases of Firebase
Examples of RDBMS	SQL essential commands	SQL essential commands

Difference between NoSQL & SQL

Demystifying differences between two:

NoSQL	SQL
Non-**S**tructured **Q**uery **L**anguage	**S**tructured **Q**uery **L**anguage
Non-Relational Database	Relational Database
Suitable for unstructured data	Suits structured data
Key-value pairs & Documents based	Table based structure
Dynamic Schema	Pre-defined schema
Horizontal Scaling – adding more machines to an existing pool	Vertical Scaling – adding more resources (CPU, RAM, Memory) to an existing machine

Examples of SQL and NoSQL RDBMS

SQL

PostgreSQL

NoSQL

Amazon DynamoDB

Real-time applications & Use cases

- MongoBD applied in real-time applications – Ex: eBay, MetLife, Shutterfly, Adhar etc
- Gaming applications – Ex: EA, FIFA online game
- Content Management Systems, Full-stack web applications, Products data management systems
- Mobile apps
- Real-time Analytics
- Its implementation in IoT applications
- It can be used in combo with SQL database

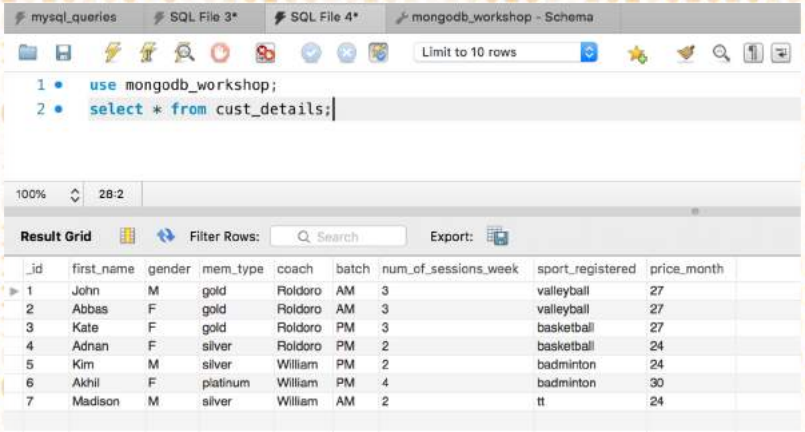
Terminologies - NoSQL (aka Non-SQL)

Demystifying NoSQL & its building blocks:

NoSQL (MongoDB)		SQL
Database	⟷	Database
Collection	⟷	Table
Document	⟷	Row / Record
Field / Key	⟷	Column

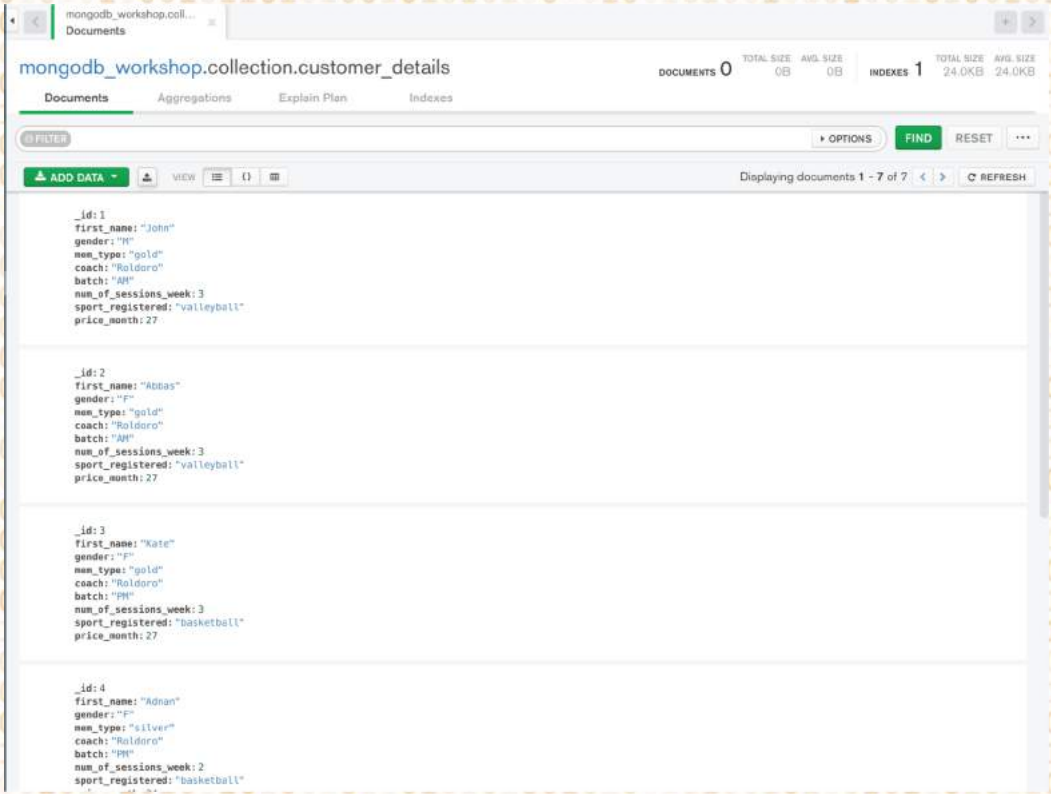
Structure of Data Storage

SQL



_id	first_name	gender	mem_type	coach	batch	num_of_sessions_week	sport_registered	price_month
1	John	M	gold	Roldoro	AM	3	valleyball	27
2	Abbas	F	gold	Roldoro	AM	3	valleyball	27
3	Kate	F	gold	Roldoro	PM	3	basketball	27
4	Adnan	F	silver	Roldoro	PM	2	basketball	24
5	Kim	M	silver	William	PM	2	badminton	24
6	Akhil	F	platinum	William	PM	4	badminton	30
7	Madison	M	silver	William	AM	2	tt	24

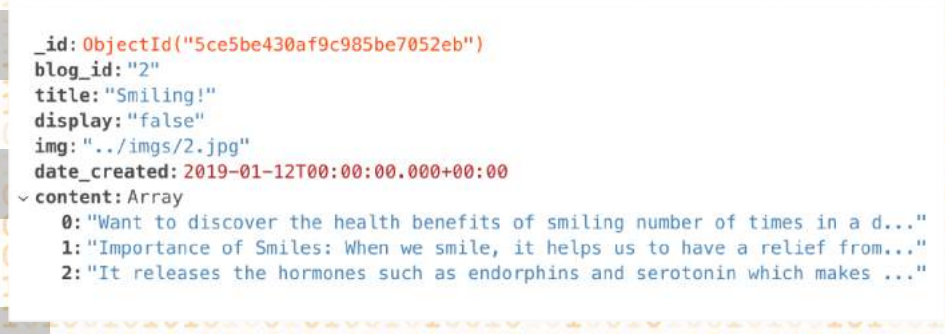
NoSQL (MongoDB)



Data Types in MongoDB

Most common data types supported in MongoDB:

Data Type	Brief	Example
String	Stores sequence of characters – must be of UTF-8 valid	
Integer	Numerical values – 32 bit or 64 bit	
Boolean	True or False values	
Double	Stores float/decimal values	23.32
Arrays	For multiple values in arrays or list	
Timestamp	To record when document added or modified	
Null	Stores null values	null
Date	Stores current date	date_added: new Date()
Object ID	To store Document's ID	



To communicate with MongoDB

1. Mongo Shell
2. MongoDB Compass
3. MongoDB Atlas
4. With Python (Pymongo / Mongo Engine)
5. Jupyter Notebook
6. PyCharm / VS Code / Atom

Tools & Requirements

Installation resources:

- MongoDB Server – Community server available @ HERE
- Mongo Shell – @ Shell
- MongoDB Compass Community - @ Compass
- Python installation – @ HERE

Tools for now:

1. MongoDB with Python via Jupyter Notebook:

- 1. Jupyter Notebook installation via:

  i) PIP

  OR

  ii) Anaconda

```
1 192:~ ramyan$ pip3 install notebook
2 .....
3 192:~ ramyan$ jupyter notebook
```

- 2. Python installation prior to step-1 above

- 3. MongoDB Server & Compass community version

Coding pre-requisites:

|| CRUD operations with MongoDB & Python via Jupyter Notebook ||

After installation & Start the below tools!

- MongoDB server
- MongoDB Compass
- Jupyter Notebook

Note: Keep the database server on!

1. Communicate via Python & Jupyter Notebook

1a) Import Module & Connect:

```python
import pymongo
# from pymongo import MongoClient

# To connect to MongoDB server:
client = pymongo.MongoClient('localhost', 27017)

# To create Database:
db = client.mongodb_workshop

# To create MongoDB's Collection object:
collection = db.collection["customer_details"]
```

1b) Create/Insert Documents (aka records/rows):

```python
# To insert one document:

# One document/record:
cust_1 = {"_id": 1, "first_name": "John", "gender": "M", "mem_type": "gold", "coach": "Roldoro",
"batch": "AM", "num_of_sessions_week": 3, "sport_registered": "valleyball", "price_month": 27}

# To execute the insert command with data:
collection.insert_one(cust_1)
```

→ To insert single document

```python
# Storing more than one documents in a list:
cust_docs = [
    {"_id": 2, "first_name": "Abbas", "gender": "F", "mem_type": "gold", "coach": "Roldoro",
"batch": "AM", "num_of_sessions_week": 3, "sport_registered": "valleyball", "price_month": 27},
    {"_id": 3, "first_name": "Kate", "gender": "F", "mem_type": "gold", "coach": "Roldoro", "batch":
"PM", "num_of_sessions_week": 3, "sport_registered": "basketball", "price_month": 27},
    {"_id": 4, "first_name": "Adnan", "gender": "F", "mem_type": "silver", "coach": "Roldoro",
"batch": "PM", "num_of_sessions_week": 2, "sport_registered": "basketball", "price_month": 24},
    {"_id": 5, "first_name": "Kim", "gender": "M", "mem_type": "silver", "coach": "William",
"batch": "PM", "num_of_sessions_week": 2, "sport_registered": "badminton", "price_month": 24},
    {"_id": 6, "first_name": "Akhil", "gender": "F", "mem_type": "platinum", "coach": "William",
"batch": "PM", "num_of_sessions_week": 4, "sport_registered": "badminton", "price_month": 30},
    {"_id": 7, "first_name": "Madison", "gender": "M", "mem_type": "silver", "coach": "William",
"batch": "AM", "num_of_sessions_week": 2, "sport_registered": "tt", "price_month": 24}
]

# To insert more than one document:
collection.insert_many(cust_docs)
```

→ To insert more than one document

1c) Read/Fetch Documents (aka records/rows):

```
1 collection.find()
2 # OR
3 collection.find({})
```

→ To find all documents

```
1 # To fetch the specific document by passing in index/id:
2 collection.find_one(2)
```

→ To find single document

1d) Update Documents (aka records/rows):

```python
# Storing required document in variable:
doc_update = {"first_name": "John"}
val_update = {"$set": {"first_name": "Johnson"}}

# To update one document:
collection.update_one(doc_update, val_update)
```

→ To update single document

```python
# Stroing values in variables:
docs_update = {"first_name": {"$regex": "^A"}}
vals_update = {"$set": {"coach": "Deo"}}

# To update more than one document:
collection.update_many(docs_update, vals_update)
```

→ To update more than one document

1e) Delete Documents (aka records/rows):

```python
# To delete document with auto assigned ID:
collection.delete_one({"_id": ObjectId('605dbe03bc5d1e410a476345')})

#OR
collection.delete_one({"_id": 14})
```

➔ To delete single document

```python
# Store the data in a variable:
del_docs = { "first_name": {"$regex": "^c"} }

# To delete more than one document:
collection.delete_many(del_docs)
```

➔ To delete more than one document

2. MongoDB Shell

```
192:~ ramyan$ mongo
MongoDB shell version v4.0.2
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 4.0.2
Server has startup warnings:
2021-03-06T11:59:13.956+0000 I CONTROL  [initandlisten]
.
.
.
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

>
```

Alternative to Option-1

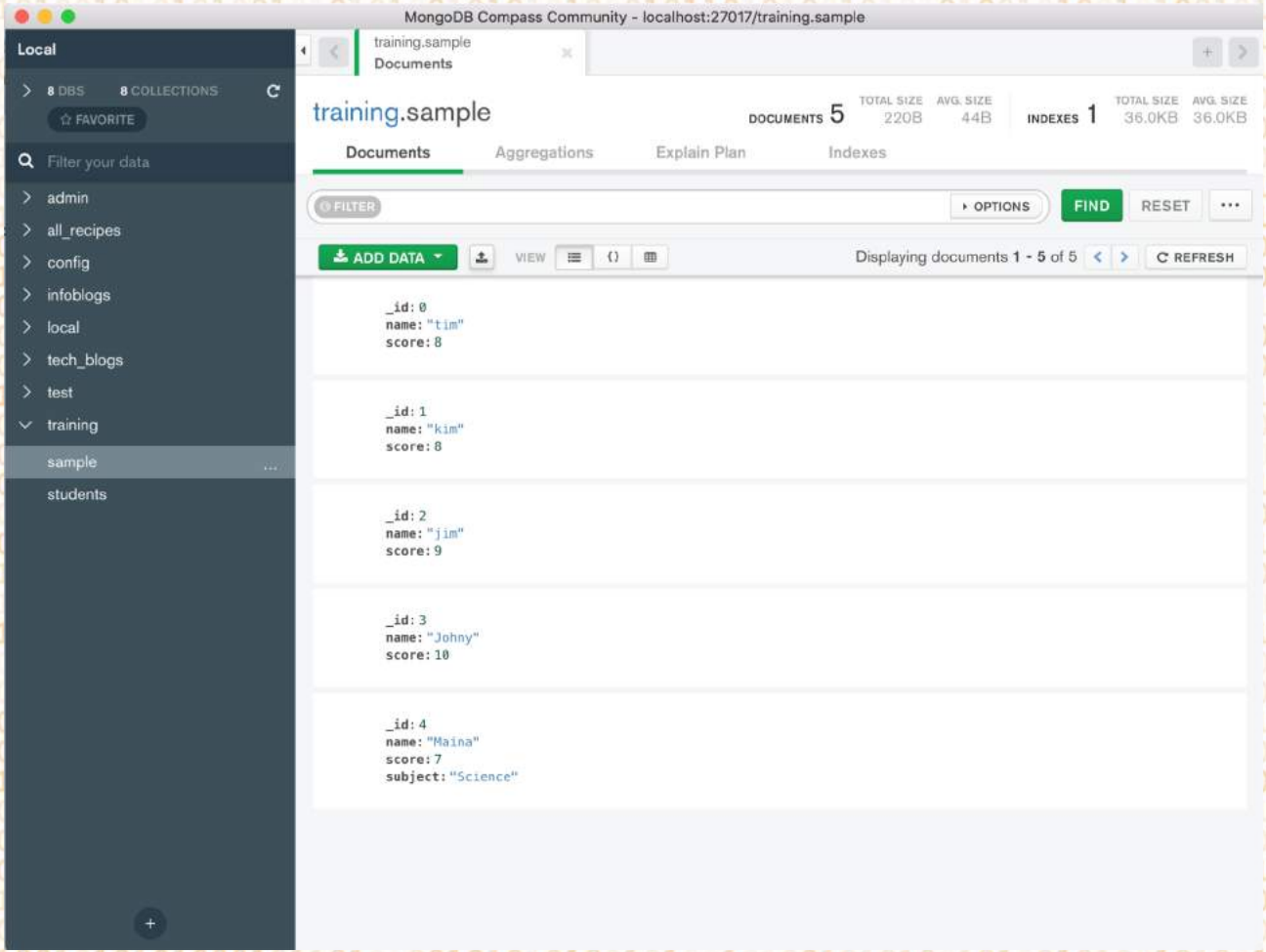
WHY

- ➔ Interact with MongoDB Database
- ➔ Querying Database

WHAT

- ➔ It's a standalone product
- ➔ It's an open-source CLI application

3. With MongoDB Compass



View of MongoDB Database, Collection & Documents in Compass (GUI) software

Let’s dive into Code Walkthrough

Link for Today’s Session:

➔ **https://github.com/rnedesigns/S4_WWCodePython_MongoDB**

MongoDB Server

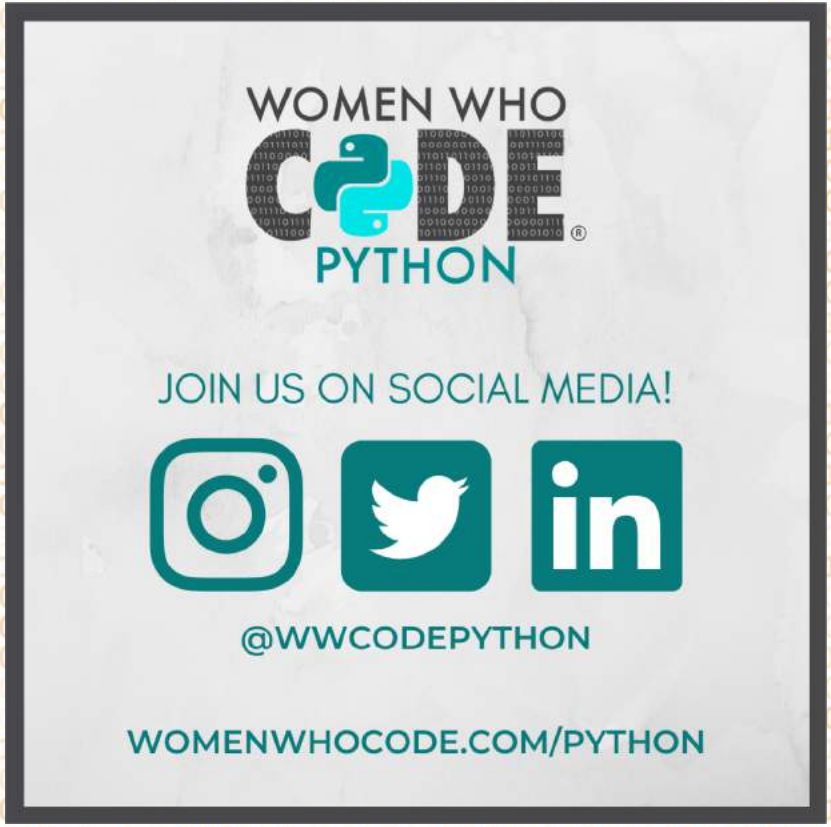
```
192:~ ramyan$ sudo mongod
Password:
2021-03-06T11:59:12.630+0000 I CONTROL  [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2021-03-06T11:59:12.681+0000 I CONTROL  [initandlisten] MongoDB starting : pid=36406 port=27017 dbpath=/data/db 64-bit host=192.168.1.103
2021-03-06T11:59:12.681+0000 I CONTROL  [initandlisten] db version v4.0.2
.
.
.
2021-03-06T11:59:18.028+0000 I NETWORK  [conn1] received client metadata from 127.0.0.1:63692 conn1: { application: { name: "MongoDB Shell" }, driver: { name: "MongoDB Internal Client", version: "4.0.2" }, os: { type: "Darwin", name: "Mac OS X", architecture: "x86_64", version: "17.7.0" } }
```

Summary

Topics we covered today:

- Differences b/w NoSQL & SQL
- Applications/Use cases of MongoDB
- Its building blocks
- Data Types
- MongoDB tools & installation
- Its CRUD implementation with Python

To Join Women Who Code Python



Upcoming Events

SAT 03 APR	🔥 Introduction to Deep Learning for Edge Devices Session 4: Hardware on the Edge🔥 *Featured* 8:00 PM – 9:30 PM (EDT) | Zoom	Register
THU 08 APR	✨Intro to Data Structures with Python: Ace the Technical Interview (Session #4: Stacks & Queues) ✨ *Featured* 8:00 PM – 9:30 PM (EDT) | Zoom	Register
SAT 17 APR	🔥 Introduction to Deep Learning for Edge Devices Session 5: Pruning🔥 *Featured* 8:00 PM – 9:30 PM (EDT) | Zoom	Register

WOMEN WHO CODE PYTHON

SQL & SQLite | NoSQL & Firebase + MongoDB with Python