MongoDB Assignment

Theoretical Questions

What are the key differences between SQL and NoSQL databases

Feature	SQL (Relational)	NoSQL (Non-relational, e.g., MongoDB)
Data model	Tables (rows, columns)	Documents, key-value, graphs, wide-column
Schema	Fixed schema	Flexible schema
Query language	SQL	Database-specific (MongoDB uses MQL)
Relationships	Joins supported	Embedded documents, manual references
Scaling	Vertical scaling	Horizontal scaling (sharding)
ACID compliance	Strong	Varies; MongoDB supports ACID for multi-doc transactions

What makes MongoDB a good choice for modern applications

 Flexible schema — no need to predefine columns.
 JSON-like documents (BSON) easy for developers to use.
 Horizontal scaling using sharding.
 High availability with replication.
 Rich query capabilities and aggregation.
 Good for big data & real-time analytics.

Explain the concept of collections in MongoDB

 A **collection** is a group of MongoDB documents, equivalent to a table in SQL.
 Documents in a collection don’t have to share the same fields.
 Stored in BSON format.
 Created automatically when inserting the first document.

How does MongoDB ensure high availability using replication

 Replication means storing copies of the same data on multiple servers.
 MongoDB uses **replica sets**:

- **Primary node** — handles reads & writes.
- **Secondary nodes** — maintain copies, handle failover if primary goes down.

 Ensures fault tolerance and redundancy.

What are the main benefits of MongoDB Atlas

 Fully managed cloud service.
 Automated backups & monitoring.
 Built-in security & access control.
 Global distribution.
 Easy scaling without downtime.

What is the role of indexes in MongoDB, and how do they improve performance

Role of indexes in MongoDB

- Indexes make queries faster by reducing the number of documents MongoDB must scan.
- Types:
  - Single field
  - Compound
  - Multikey
  - Text
  - Geospatial
- Without indexes, MongoDB does a **collection scan** (slow).

Describe the stages of the MongoDB aggregation pipeline

Stages of the aggregation pipeline

1. **$match** — Filter documents.
2. **$group** — Group by a field and perform aggregations.
3. **$project** — Select/transform fields.
4. **$sort** — Sort results.
5. **$limit** — Limit results.
6. **$skip** — Skip a number of documents.
7. **$lookup** — Join with another collection.
8. **$unwind** — Flatten arrays.

What is sharding in MongoDB? How does it differ from replication.

Sharding vs replication

- **Sharding** — Splits data across multiple servers (**horizontal scaling**).
- **Replication** — Copies the same data to multiple servers (**high availability**).
- **Difference**: Sharding partitions data; replication duplicates it.

What is PyMongo, and why is it used

 Official Python driver for MongoDB.
 Allows Python programs to connect to and work with MongoDB databases.

Schema validation

- MongoDB can enforce rules on document structure.
- Uses JSON Schema to define required fields, data types, and allowed values.

What are the ACID properties in the context of MongoDB transactions

ACID properties in MongoDB transactions

- **Atomicity** — All or nothing execution.
- **Consistency** — Data moves from one valid state to another.
- **Isolation** — Transactions don't interfere with each other.
- **Durability** — Data is permanently stored after commit.

What is the purpose of MongoDB's explain() function

Purpose of explain()

- Shows how MongoDB executes a query.
- Helps in **query optimization** by revealing index usage, execution time, and stages.

How does MongoDB handle schema validation

 MongoDB can enforce rules on document structure.
 Uses JSON Schema to define required fields, data types, and allowed values.

What is the difference between a primary and a secondary node in a replica set

Primary vs secondary node

- **Primary** — Accepts writes and reads.
- **Secondary** — Reads (if enabled), syncs from primary, takes over if primary fails.

What security mechanisms does MongoDB provide for data protection

MongoDB security mechanisms

- Authentication (SCRAM, LDAP, Kerberos).
- Authorization (role-based access control).
- Encryption (TLS/SSL in transit, encryption at rest).
- IP whitelisting.

Explain the concept of embedded documents and when they should be used

Embedded documents

- A document inside another document.
- Used when related data is always accessed together (e.g., order with its items).

What is the purpose of MongoDB's $lookup stage in aggregation

Purpose of $lookup

- Performs a **join** between collections.
- Syntax:

{ $looku **Advantages for horizontal scaling**

- Sharding distributes data across servers.
- Allows near-linear scalability.
- Handles massive datasets without overloading a single machine.

```
p: {
  from: "otherCollection",
  localField: "fieldInThisCollection",
  foreignField: "fieldInOtherCollection",
  as: "outputArray"
}}
```

What are some common use cases for MongoDB

Common use cases for MongoDB

- Real-time analytics.
- Content management systems.
- IoT data storage.
- Product catalogs.
- Mobile apps with offline sync.

What are the advantages of using MongoDB for horizontal scaling

MongoDB vs SQL transactions

- SQL transactions are always ACID and span multiple tables easily.
- MongoDB supports multi-document ACID transactions but performance may be lower than single-document ops.

How do MongoDB transactions differ from SQL transactions

Capped vs regular collections

Feature	Capped Collection	Regular Collection
Size	Fixed	Grows dynamically
Overwrite	Old docs overwritten when full	No overwrite
Use case	Logging, caching	General storage

What are the main differences between capped collections and regular collections

Feature	Capped Collection	Regular Collection
Size	Fixed	Grows dynamically
Overwrite	Old docs overwritten when full	No overwrite
Use case	Logging, caching	General storage

What is the purpose of the $match stage in MongoDB’s aggregation pipeline

- Filters documents early in the aggregation pipeline to reduce processing.
- Equivalent to SQL WHERE.

How can you secure access to a MongoDB database

- Enable authentication.
- Use role-based access control.
- Restrict network access (firewall, IP whitelist).
- Use TLS/SSL encryption.

What is MongoDB’s WiredTiger storage engine, and why is it important?

- Default storage engine in MongoDB.
- Provides:
  - Compression for data.
  - Document-level concurrency.
  - Better performance and scalability.
  - Journaling for durability.

1.Write a Python script to load the Superstore dataset from a CSV file into MongoDB

```
client = MongoClient("mongodb://localhost:27017/")  # Change URI if using
MongoDB Atlas
db = client["superstore_db"]  # Database name
orders_collection = db["Orders"]
```

2. Retrieve and print all documents from the Orders collection

```
df = pd.read_csv("Superstore.csv")  # Path to your dataset
# Convert DataFrame to dictionary
data_dict = df.to_dict("records")
# Insert into collection
orders_collection.insert_many(data_dict)
print("Data inserted successfully!")
print("\nAll Orders:")
for doc in orders_collection.find():
    print(doc)
```

3.Count and display the total number of documents in the Orders collection

```
print("\nAll Orders:")
for doc in orders_collection.find():
    print(doc)
```

Write a query to fetch all orders from the "West" region

```
print("\nOrders from 'West' region:")
for doc in orders_collection.find({"Region": "West"}):
    print(doc)
```

Write a query to find orders where Sales is greater than 500

```
print("\nOrders with Sales > 500:")
for doc in orders_collection.find({"Sales": {"$gt": 500}}):
    print(doc)
```

Fetch the top 3 orders with the highest Profit

```
print("\nTop 3 Orders by Profit:")
for doc in orders_collection.find().sort("Profit", -1).limit(3):
    print(doc)
```

Update all orders with Ship Mode as "First Class" to "Premium Class.

```
update_result = orders_collection.update_many(
    {"Ship Mode": "First Class"},
    {"$set": {"Ship Mode": "Premium Class"}}
)
print(f"\nUpdated {update_result.modified_count} documents from 'First Class' to
'Premium Class'.")
```

8< Delete all orders where Sales is less than 50

```
delete_result = orders_collection.delete_many({"Sales": {"$lt": 50}})
print(f"\nDeleted {delete_result.deleted_count} orders with Sales < 50.")
```

Use aggregation to group orders by Region and calculate total sales per region

```
print("\nTotal Sales per Region:")
pipeline = [
    {"$group": {"_id": "$Region", "total_sales": {"$sum": "$Sales"}}}
]
for doc in orders_collection.aggregate(pipeline):
    print(doc)
```

Fetch all distinct values for Ship Mode from the collection< 66<

```
ship_modes = orders_collection.distinct("Ship Mode")
print("\nDistinct Ship Modes:", ship_modes)
```

Count the number of orders for each category.

```
print("\nOrders count per Category:")
pipeline = [
    {"$group": {"_id": "$Category", "order_count": {"$sum": 1}}}
]
for doc in orders_collection.aggregate(pipeline):
    print(doc)
```