Name :- Aditya S. Khandagale
Div :- AIEC-2
RollNo :- 03
Sub :- Data Engineering

Assignment - I

Q1

a] A healthcare organization wants to digitze patient records. Explain the Structured, Semi Structured data type they would encounter and how the data lifecycle supports this transformation

Ans when a healthcare organization digitizes patient record, it will face three main types of data

i] Structured Data :-

a] organized and Stored in relational databases

b] Examples : Patiend Id, Name, Age, contanct, Blood Test Results

c] Easy to query using SQL (rows and columns)

ii] Semi-structured Data :-

a] Data that has Some Structure but not staictly tabular

b] Example :- Electronic prescription, lab reports in xml / JSON format a wearable device logs

c] flexible and can evolve without fixed schema

iii] Unstructured Data :-

a] No predefined structure, often large in size

b] Example : MRI Scans, X-ray images, doctor's notes, audio / video consulation recording

c] Need Big Data / NoSQL Systems for storage and analysis

Data lifecycle Support

- Collection :- Patient data gathered from hospital System, labs and IOT devices
- Storage :- Structured data → relation DB, Semi structured → JSON/XML, unstructured → image repositories
- Processing :- Transform into usable formats
- Analysis :- Data mining for disease patterns, predictive analytics
- Sharing :- Secure sharing between doctor and insurance agencies
- Archival :- old records stored or deleted as per compliance

b] A logistics company wants to design a system to analyze vehicle tal telementry data. Describe the phases of the data lifecycle applicable to their system

Ans Telemetry data includes GPS, speed fuel consumption egine temperature etc. Data lifecykle phases applicable

i] Data collection :- Sensors in vehicles capture ~~tom~~ telemetry continuously
ii] Data storage :- Data stored in real-time database or cloud (time series DB like influxDB)
iii] Data Processing :- Streaming (ai) engines (Kafka Spark) process live data
iv] Data Analysis :- Analystics dashboards track fuel usage, routes, and driver behavior
v] Data Sharing :- Reports shared with managers for route optimization
vi] Data Archival :- Historical telemetry stored for compliance or deleted after retention period

Q)

a] A university plans to depoly a research database. Compare and analyze Key-value and document based No SQL database for this use case

Ans • Key-value Store
i] Structure :- Data Stored as Key-value pairs (like a dictionary)
ii] Pros :- Extremely fast lookups, good for caching research IDs.
iii] Cons :- No Support for complex queries, poor handling of relationships
iv] Example : Redis, ~~Dyan~~ DynamoDB

- Document - Based Store

i] Structure :- JSON/BSON documents with flexible Schema

ii] Pros : Can store complex research data (metadata, experiment results, references) in one document

iii] Good for queries, indexing and semi-structured data

iv] Example :- Mongo DB, Couch DB

Analysis :-
for a research databases where data varies (paper, metadata, authors, experiment datasets) document - based No SQL is better because of flexible Schema and query support. Key value is faster but too simple for research use cases

b An e-commerce platform receives high volume clickstream data. Explain How Big data charac-teristic (volume, variety, velocity) influence the data engineering process

Ans
- volume :- Massive clickstream logs generated daily by millions of users. Require distributed Storage (HDFS, NoSQL)
- variety : clickstream data include page visits device type, location, Session logs (structured + unstructured). Needs schema-flexible system like MongoDB

- Velocity : Data generated at high speed in real time. Needs stream processing frameworks (Kafka, SparkStreaming)

Impact on Data Engineering
- Data pipelines must be Scalable
- ETL processes need to handle multi format data
- Real-time analytics required for personalization and fraud detection

Big Data characteristic drive design of data ingestion, storage and processing pipelines

Q3
a A Startup is evaluating NoSQL option for a recommendation engine. Distinguish between Document and Graph database in term of performance and structure

Ans i] Document Database
- Store data as JSON document
- Best for user profiles, purchase history
- Performance : High for reading/writting document
- Limitation : Not efficient for complex relationships

ii] Graph Database
- Store entities as node and relationships as edges
- Best for recommendation systems

Q4 a] A food delivery app aggregates reviews and ratings. Explain how MongoDB's JSON document structure help store heterogeneous data

Ans MongoDB uses a flexible JSON-like format allowing heterogenous data in a single collection

Example

```
{ "restaurant" : "Pizza Hut",
  "review" : "Great taste!",
  "rating" : 4.5,
  "user" : { "name" : "Johny", "location" : "Mum" },
  "Images" : ["img1.jpg", "img2.jpg"]
}
```

- Some reviews may have images, others may not
- No fixed schema - easy to handle varied user data
- Supports nested fields, arrays and flexible attributes

b] A medical records system must be queried efficiently. Analyze how mongoDB index and aggregation can support analytical dashboards

Ans

- Indexes :-

Improve query performance (eg. searching patients by ID, disease or date)

- Example

```
db.records.createIndex ({patientId : 1}).
```

- Aggregation :-
- used to group and summarize data for dashboards
- Example :- Average patient age, disease-wise case counts
- Query

```
db.records.aggregate([
  {$group: {_id: "$disease", total:
{$sum: 1}}}
])
```

with indexes, queries become fast and with aggregation, insights can be visualized in real time dashboards for medical decision-making

Q5 a] An IoT data hub captures reading every second discuss the advantages of mongoDB in handling high-ingestion use cases

Ans when an IoT data hub captures sensor reading every second the database must handle high ingestion rate, scalability and flexible schema MongoDB provides several advantages

i] High write Throughput

- MongoDB is optimized for rapid insert operation allowing millions of sensor readings per second
- uses memory-mapped storage engine for fast ingestion

ii] Horizontal Scalability

- IoT hubs generate huge amounts of data continuously

- MongoDB Supports sharding, distributing data across multiple servers for balanced load

iii] Flexible Schema

- IoT data may very vary (Temperature, pressure humidity)
- MongoDB's JSON/BSON documents allow inserting new sensor fields without altering Schema

b] A developer uses MongoDB compass to model data Analyze its usefulness in query visualization and performance tunning

Ans MongoDB Compass is a GUI tool that helps developer's interact with MongoDB visually instead of command-line. Its usefulness includes

i] Data Modeling and Exploration

- Developer can view JSON documents in tree/table view
- Helpful for understanding schema in semi-structured IOT or e-commerce datasets

ii] Query Visualization

- Provide a visual query builder
- user can filter, Sort and project fields without writing full mongo queries
- Queries show real time result with execution stats

iii] Performance Tuning

- Show query execution ~~slow~~ plans with detail

like index usage and scan performance
- Helps identify slow queries and optimize them using indexes

iv] Index Management
- Allows creation and monitoring of indexes from GUI
- Developer can test how indexes affect query response time

v] Aggregation pipeline Builder
- Provides drag and drop interface to build multi-stage pipelines for analytics
- useful for dashboards and reports