MongoDB

mongoDB®

Session 1 - Introduction

XANSA IT SOLUTIONS simplifying IT

Mongo or

The confusion.

XANSA IT SOLUTIONS simplifying IT

Humongous

huge, enormous.

Origin of the term

Humongous

reflects its ability to manage and store extremely large amounts of data

Origin of the term

NoSQL Databases

Student Profile

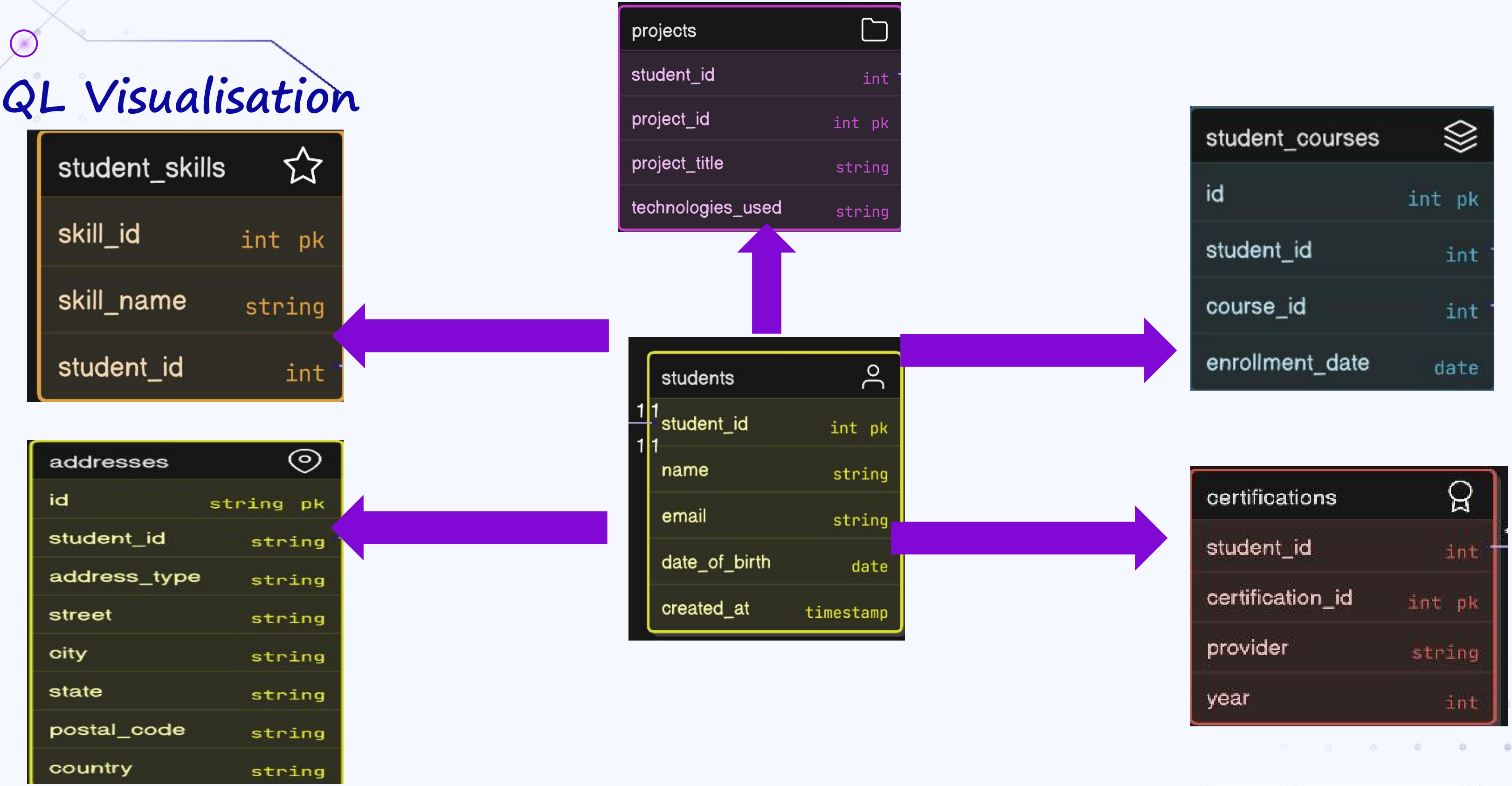
- Can a student have multiple phone numbers?
- Can a student have multiple addresses?
- Can a student possess multiple skills?
- Can a student enroll in multiple courses?

Student Profile

• Can a student have multiple phone numbers?	Yes
• Can a student have multiple addresses?	Yes
• Can a student possess multiple skills?	Yes
• Can a student enroll in multiple courses?	Yes

XANSA IT SOLUTIONS simplifying IT

SQL Visualisation



student_skills	
skill_id	int pk
skill_name	string
student_id	int

addresses	
id	string pk
student_id	string
address_type	string
street	string
city	string
state	string
postal_code	string
country	string

projects	
student_id	int
project_id	int pk
project_title	string
technologies_used	string

students	
student_id	int pk
name	string
email	string
date_of_birth	date
created_at	timestamp

student_courses	
id	int pk
student_id	int
course_id	int
enrollment_date	date

certifications	
student_id	int
certification_id	int pk
provider	string
year	int

Joins

Joins

Joins

& More..

Joins

MongoDB Visualisation

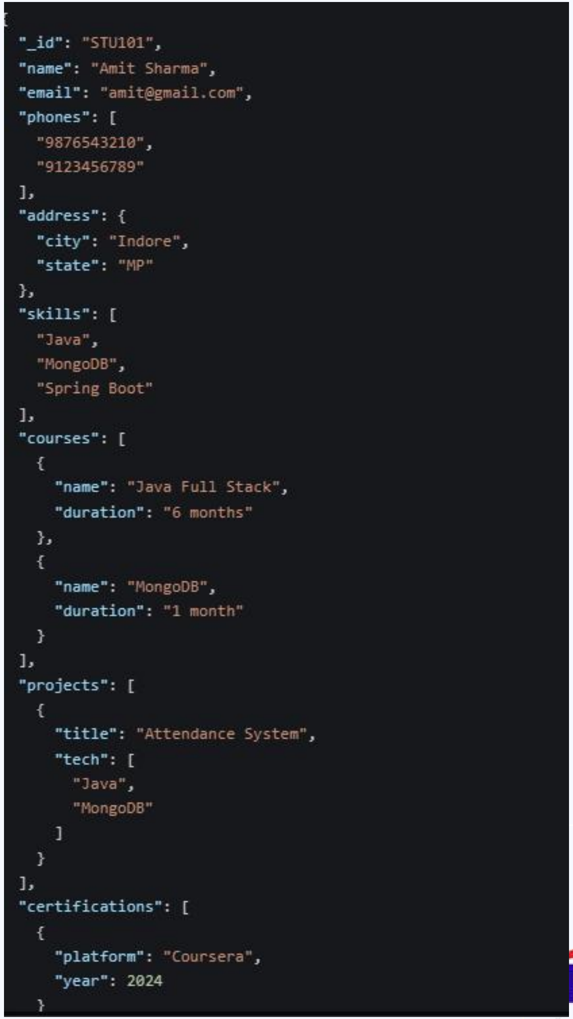
name

email

phones [ ]

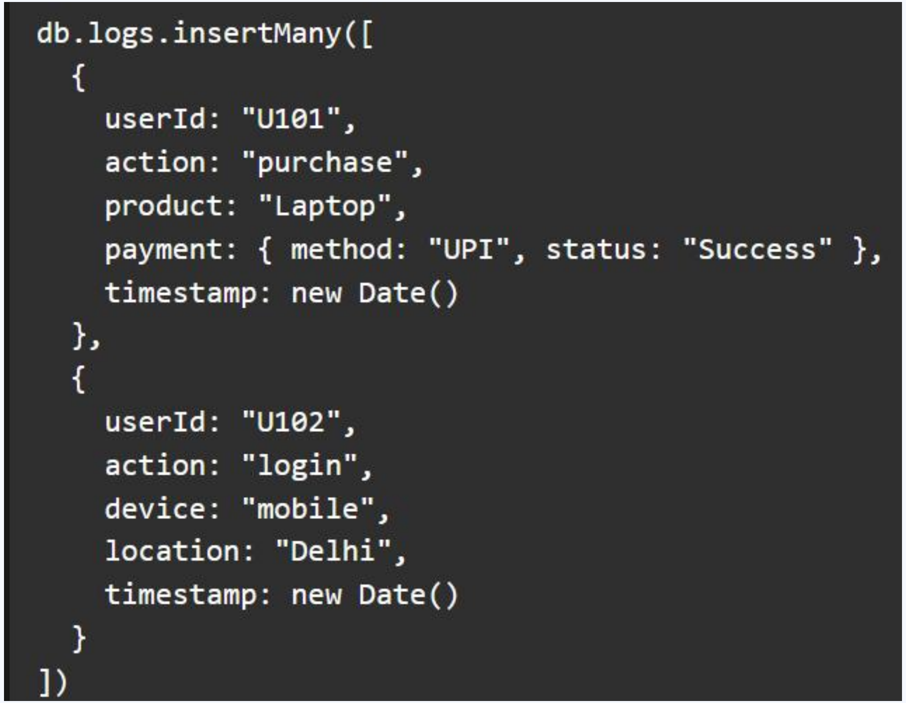
skills [ ]

projects [ ]



```
{
  "_id": "STU101",
  "name": "Amit Sharma",
  "email": "amit@gmail.com",
  "phones": [
    "9876543210",
    "9123456789"
  ],
  "address": {
    "city": "Indore",
    "state": "MP"
  },
  "skills": [
    "Java",
    "MongoDB",
    "Spring Boot"
  ],
  "courses": [
    {
      "name": "Java Full Stack",
      "duration": "6 months"
    },
    {
      "name": "MongoDB",
      "duration": "1 month"
    }
  ],
  "projects": [
    {
      "title": "Attendance System",
      "tech": [
        "Java",
        "MongoDB"
      ]
    }
  ],
  "certifications": [
    {
      "platform": "Coursera",
      "year": 2024
    }
```

MongoDB Visualisation



```
db.logs.insertMany([
  {
    userId: "U101",
    action: "purchase",
    product: "Laptop",
    payment: { method: "UPI", status: "Success" },
    timestamp: new Date()
  },
  {
    userId: "U102",
    action: "login",
    device: "mobile",
    location: "Delhi",
    timestamp: new Date()
  }
])
```

Same table structure?

“MongoDB stores data the way developers think.”

Fixed data → SQL
Growing / changing / nested data → MongoDB

What is NoSQL?

- NoSQL stands for Not Only SQL
- A type of database that does not use traditional tables
- Designed to handle:
  - Large volume of data
  - High speed read/write
  - Unstructured or semi-structured data
- Works well with modern web & mobile applications

Limitations of Traditional RDBMS

- Fixed schema (hard to change structure)
- Vertical scaling is expensive
- Poor performance with large, unstructured data
- Complex joins reduce speed
- Not ideal for real-time & big data systems

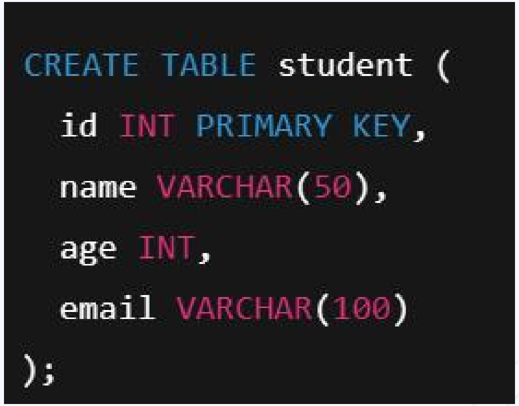
Schema

- A schema defines the structure, format, and constraints of data stored in a database.

Schema = Design + Rules of data storage

Schema in Traditional Databases (RDBMS)

- In relational databases like MySQL, Oracle, SQL Server, a schema defines:
  - Table names
  - Column names
  - Data types (INT, VARCHAR, DATE, etc.)
  - Constraints (PRIMARY KEY, NOT NULL, UNIQUE, FOREIGN KEY)



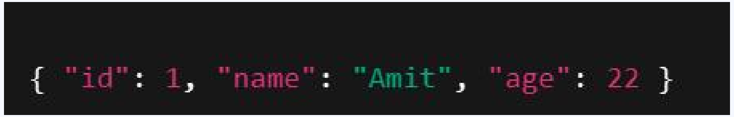
```
CREATE TABLE student (
  id INT PRIMARY KEY,
  name VARCHAR(50),
  age INT,
  email VARCHAR(100)
);
```

Every row must follow this structure

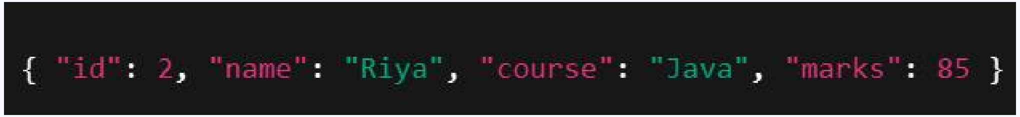
You cannot insert data with extra or missing columns

Schema in MongoDB (NoSQL)

- MongoDB is schema-less, meaning:
  - No fixed structure is required
  - Documents in the same collection can have different fields
  - Structure can change anytime



```
{ "id": 1, "name": "Amit", "age": 22 }
```



```
{ "id": 2, "name": "Riya", "course": "Java", "marks": 85 }
```

Both documents are valid
No predefined column structure

Key Characteristics of NoSQL

- Schema-less or flexible schema
- Horizontal scaling (scale out)
- High availability
- Distributed architecture
- Faster read/write operations

Types of NoSQL Databases

- Document-Based
  - Stores data as documents - Example: MongoDB
- Key-Value
  - Simple key–value pairs - Example: Redis
- Column-Based
  - Data stored in columns - Example: Cassandra
- Graph-Based
  - Data stored as nodes & edges - Example: Neo4j

When to Use NoSQL?

- Rapid application development
- Large scale applications
- Real-time analytics
- Big Data & IoT systems
- Cloud-based applications

Why MongoDB is Used

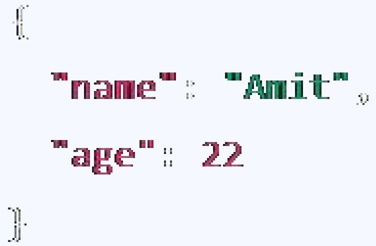
XANSA
IT SOLUTIONS
simplifying IT

What is MongoDB?

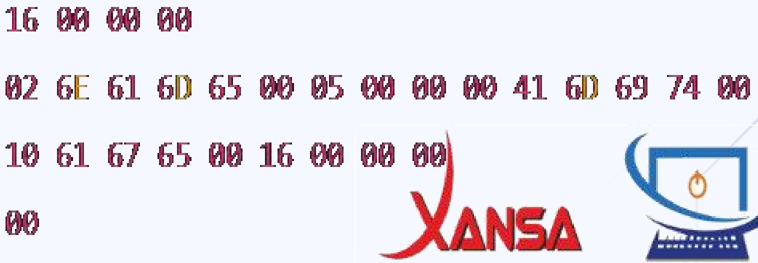
- Open-source NoSQL document database
- Stores data in BSON (Binary JSON)
- Schema-less and flexible
- High performance & scalability

What is BSON?

- BSON stands for Binary JSON.
  - BSON is a binary-encoded format to store data efficiently.
- JSON is:
  - Human-readable
  - Text-based (slower to process)
- BSON is:
  - Binary format
  - Faster to read/write
  - Supports more data types



```
{
  "name": "Amit",
  "age": 22
}
```



```
16 00 00 00
02 6E 61 6D 65 00 05 00 00 00 41 6D 69 74 00
10 61 67 65 00 16 00 00 00
00
```

XANSA IT SOLUTIONS simplifying IT

JSON vs BSON

Feature	JSON	BSON
Full form	JavaScript Object Notation	Binary JSON
Format	Text	Binary
Human readable	Yes	No
Processing speed	Slower	Faster
Data types	Limited	More (Date, Binary, Int64)
Storage efficiency	Less efficient	More efficient

Why MongoDB is Popular

- Easy to learn and use
- Flexible data model
- Faster development
- Scales easily
- Strong community support

Key Features of MongoDB

- Document-oriented storage
- Automatic sharding
    - Automatic sharding is a feature where large data is automatically split and distributed across multiple servers to improve performance, scalability, and availability.
- Replication for high availability
- Rich query language
- Powerful aggregation framework

MongoDB Vs. RDBMS

Structural Comparison

RDBMS	MongoDB
Database	Database
Table	Collection
Row	Document
Column	Field
Primary Key	_id

Schema Comparison

- RDBMS
  - Fixed schema
  - Structure must be defined before inserting data
- MongoDB
  - Schema-less
  - Different documents can have different fields

Performance & Scalability

- RDBMS
  - Vertical scaling
  - Limited scalability
-

  MongoDB
  - Horizontal scaling
  - Handles large data efficiently

Query & Joins

- RDBMS
  - Complex joins
  - Slower for large datasets
- MongoDB
  - Embedded documents reduce joins
  - Faster read operations

Use Case Comparison

- Use RDBMS when:
  - Data is highly structured
  - Strong ACID compliance is required

- Use MongoDB when:
  - Data structure changes frequently
  - Performance & scalability are critical

Thank You

XANSA

IT SOLUTIONS

simplifying IT