Database and its Applications

Data Models and Mathematical Foundations

Pooja T S

Computer Applications

Database and its Applications

Data Models and Mathematical Foundations

Relational vs Non-relational Models

Pooja T S

Computer Applications

Database and its Applications

Relational vs Non-relational Models

Figure: Understanding two major database paradigms

Database and its Applications

Data Models

- A **data model** is a way of organizing and structuring data
- Determines how data is stored, accessed, and managed
- **Two broad categories**:
  - Relational (SQL)
  - Non-relational (NoSQL)
- A **Database model** defines the logical design and structure of a database

Database and its Applications

Relational Model – Overview

- Based on **tables (relations)** → rows (tuples) & columns (attributes)
- Uses schema with predefined rules.
- Uses **SQL** (Structured Query Language)
- Focus on **relationships between data entities**
- Example: Student–Course enrollment database .
- Database Systems: PostgreSQL, MySQL.

Database and its Applications

Key Features of Relational Model

- Structure: Organized into tables with rows and columns.
- Integrity: Enforces primary keys, foreign keys, and constraints.
- Consistency: **ACID compliance** (Atomicity, Consistency, Isolation, Durability)
- Querying: Rich SQL operations (joins, subqueries, aggregations).
- Best for **transactional systems**

Database and its Applications

When to Use Relational Models?

- Applications requiring strong consistency.
- Structured data with predefined schema.
- Complex queries and transactions.
- Examples: Banking, ERP systems.

Database and its Applications

Advantages of Relational Models

- Structured and well-defined schema.
- Supports complex queries and joins.
- Strong consistency guarantees (ACID).
- Seasoned ecosystem with many tools.

Database and its Applications

Non-relational Model – Overview

- Not table-based → **flexible structures** (key-value, document, graph, column)
- Often called **NoSQL databases**
- Querying via APIs or specific query languages.
- **Schema-less** or dynamic schema
- Example: MongoDB storing JSON documents

Database and its Applications

Key Features of Non-relational Model

- **Schema flexibility** : Supports documents, key-value pairs, graphs.
- Schema-less: Data structures can vary across records.
- **Horizontal scalability** → handles Big Data
- **High availability & performance**
- Best for **real-time, large-scale, unstructured data or semi-strucured data**

Database and its Applications

Types of Non-relational Models

- **Key-Value Stores** → Redis, DynamoDB
- **Document Stores** → MongoDB, CouchDB
- **Column-oriented** → Cassandra, HBase
- **Graph Databases** → Neo4j, ArangoDB

Database and its Applications

When to Use Non-relational Models?

- Applications needing flexibility and rapid iteration.
- Large-scale, high-velocity data.
- Semi-structured or unstructured data.
- Examples: Social media, IoT, recommendation engines.

Database and its Applications

Advantages of Non-relational Models

- Flexible and schema-less.
- High scalability for big data.
- Optimized for distributed environments.
- Faster development cycles.

Database and its Applications

Relational vs Non-relational Models

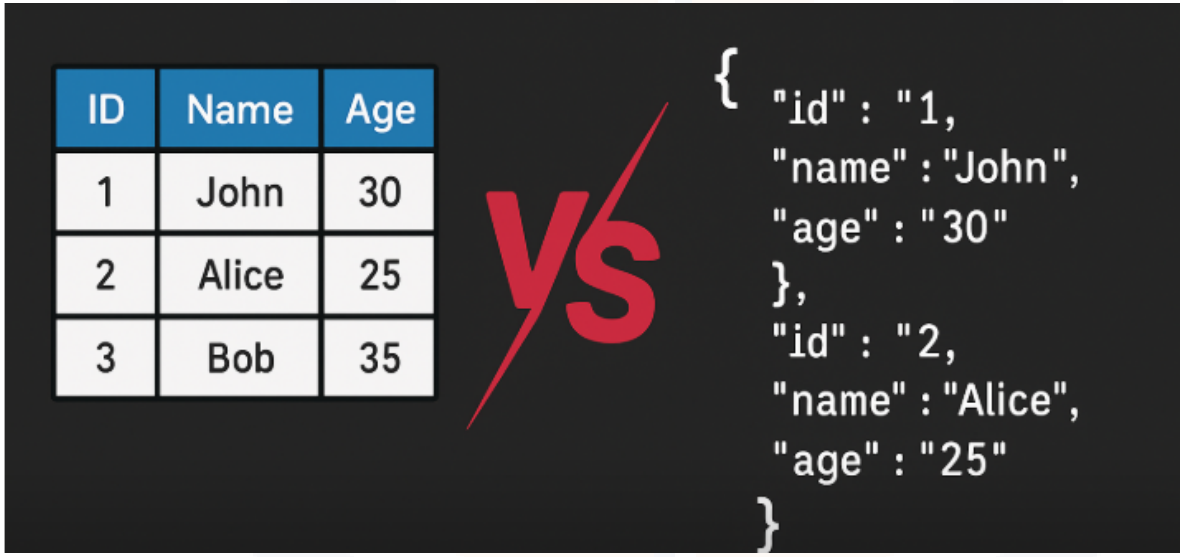

Figure: Tabular (Relational) vs JSON (Non-relational) Data Representation

Database and its Applications

Relational vs Non-relational Models

PES UNIVERSITY

Aspect	Relational (SQL)	Non-relational (NoSQL)
Data Structure	Tables with rows and columns	Key-Value, Document, Graph, etc.
Schema	Fixed, predefined schema	Flexible schema
Query Language	SQL	Varies (JSON queries, APIs)
Transactions	ACID (Atomicity, Consistency, Isolation, Durability)	BASE (Basically Available, Soft state, Eventual consistency)
Best Use Case	Banking, ERP systems	Social Media, IoT, Big Data

Database and its Applications

Activity: Database Match-up

- Instructions:
  - You will see a list of applications and database models.
  - Match each application with the correct database type (Relational or Non-relational).
  - Justify your choice in 1–2 sentences.
- Applications List:
  - Banking system
  - Instagram feed
  - E-commerce product catalog
  - ERP/CRM system
  - IoT sensor data stream
  - Social media chat app

Database and its Applications

Expected Matches

- Banking system → Relational (ACID, transactions)
- Instagram feed → Non-relational (flexibility, scalability)
- E-commerce catalog → Non-relational (document store, product metadata)
- ERP/CRM → Relational (structured, schema-based)
- IoT sensor data → Non-relational (high velocity, real-time)
- Chat app → Non-relational (scalable, unstructured messages)

Thank You

Pooja T S

Assistant Professor

Department of Computer Applications

poojats@pes.edu

080-26721983 Extn: 233