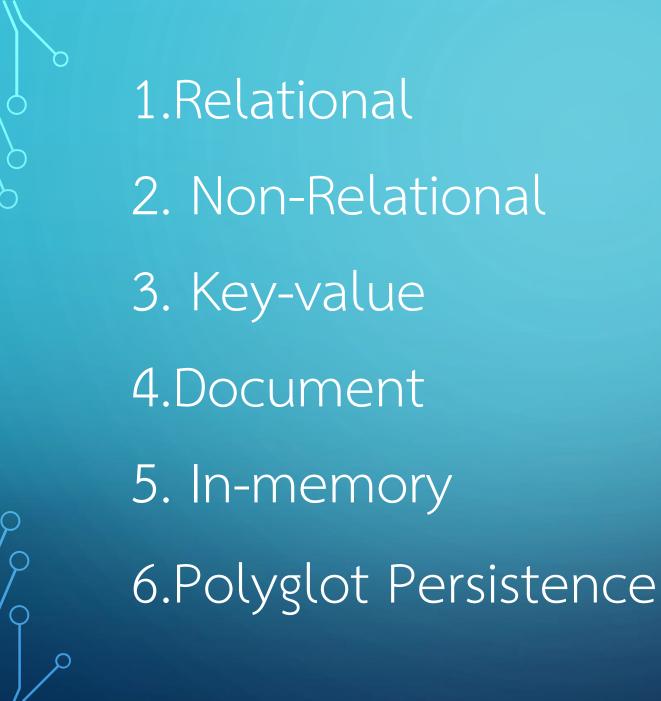
Data collection format

0

 \bigcirc



Introduction to Relational Databases •Definition of a relational database •Key characteristics of relational databases •Importance of relational databases in modern data management

Data Storage in Relational Databases Concept of tables and rows •Columns and data types •Primary keys and foreign keys •Relationships between tables (one-tomany, many-to-many)

Data Organization and Normalization •Normalization principles (1NF, 2NF, 3NF) •Benefits of normalization (data integrity, redundancy reduction) •Challenges and trade-offs in normalization

Query Language and Operations Introduction to SQL (Structured Query Language) •Basic SQL commands (SELECT, INSERT, UPDATE, DELETE) •Joins, filters, and aggregations

Advantages of Relational Database Design

Scalability and performance
Data consistency and integrity
Flexibility and adaptability
Widespread adoption and support

Books Table

BookID	Title	Author	Publisher	PublicationYear	Genre	Price
1	To Kill a Mockingbird	Harper Lee	J.B. Lippincott & Co.	1960	Fiction	9.99
2	The Great Gatsby	F. Scott Fitzgerald	Charles Scribner's Sons	1925	Fiction	12.99
3	1984	George Orwell	Secker & Warburg	1949	Fiction	8.99
4	Harry Potter and the Sorcerer's Stone	J.K. Rowling	Bloomsbury	1997	Fantasy	14.99
5	The Kite Runner	Khaled Hosseini	Riverhead Books	2003	Fiction	11.99

Customers Table

CustomerID	FirstName	LastName	Email	PhoneNumber	Address
1	John	Smith	john.smith@email.com	555-1234	123 Main St, Anytown USA
2	Jane	Doe	jane.doe@email.com	555-5678	456 Oak Rd, Somewhere City
3	Michael	Johnson	michael.johnson@email.com	555-9012	789 Elm St, Othertown
4	Emily	Williams	emily.williams@email.com	555-3456	321 Pine Ave, Newtown
5	David	Brown	david.brown@email.com	555-7890	654 Cedar Ln, Oldtown

Non-Relational

"Non-Relational" is a term used to describe database management systems that do not follow the traditional relational database model. In other words, non-relational databases are databases that do not use the standard table-based structure with rows and columns found in relational databases.

Introduction to Non-Relational Databases

 Definition of Non-Relational (NoSQL) databases

Key characteristics: flexible schema, horizontal scaling, high availability

Types of Non-Relational Databases

- Key-Value Stores (e.g., Redis, Amazon DynamoDB)
 Document-Oriented Databases (e.g., MongoDB, Apache CouchDB)
- Wide-Column Stores (e.g., Apache Cassandra, Apache HBase)
- Graph Databases (e.g., Neo4j, Amazon Neptune)

Key-Value Stores

Example: Amazon DynamoDB
Data model: key-value pairs
Use cases: caching, user session management, real-time applications

Document-Oriented Databases

Example: MongoDB
Data model: flexible, schema-less documents (e.g., JSON, BSON)
Use cases: content management systems, mobile apps, loT data

Wide-Column Stores

- Example: Apache Cassandra
- Data model: tables with dynamic columns
- Use cases: time-series data, user activity tracking, real-time analytics

Graph Databases

- Example: Neo4j
- Data model: nodes, relationships, and properties
 Use cases: social networks, recommendation engines, fraud detection

"In-memory database"

"In-memory database" refers to a type of database management system where the entire database is stored in the main memory (RAM) of the computer, as opposed to being stored on a persistent storage device like a hard disk or solid-state drive. Introduction to In-Memory Databases

Definition of in-memory databases Key characteristics: data stored in RAM, high performance, volatile data

Architecture of In-Memory Databases

Data storage in main memory (RAM) Advantages of in-memory architecture: low latency, high throughput Challenges: managing data persistence and recovery

Use Cases for In-Memory Databases

Real-time analytics and business intelligence Caching and session management High-speed transactions and trading systems Internet of Things (IoT) and sensor data processing

Popular In-Memory Database Solutions

- Redis: Open-source key-value store
- Apache Ignite: Distributed in-memory data fabric
- SAP HANA: In-memory database management system for analytics
- MemSQL: Distributed in-memory database with scalability



- Advantages: Extremely high performance, low latency, real-time data processing
- Tradeoffs: Dependence on sufficient RAM, risk of data loss on system failure
- Considerations for choosing in-memory databases: data volume, performance requirements, persistence needs

Throughout the presentation, you can include visual aids, such as diagrams or charts, to illustrate the key concepts and examples of in-memory databases. Additionally, you can highlight the specific use cases and industries where in-memory databases are most beneficial, as well as discuss the potential challenges and trade-offs associated with this database architecture.