**Database Structures Study Guide**

**Objective 5.2: Compare and contrast various database structures**

**1. Understanding Data Structure**

Data can range from highly structured to unstructured. The way we store and process this data determines the type of database we use.

* **Structured Data:** Organized into well-defined tables (e.g., spreadsheets, relational databases).
* **Semi-Structured Data:** Contains some structure but not in a rigid format (e.g., key-value pairs, JSON).
* **Unstructured Data:** Lacks a predefined structure (e.g., text documents, multimedia files).
* **Data Spectrum:** Data doesn’t always fit neatly into one category; it often exists on a spectrum between structured and unstructured.

**2. Structured Data: Relational Databases**

Relational databases store data in tables consisting of rows and columns.

* **Table Elements:**
  + **Field (Column):** Represents a single characteristic of data (e.g., CustomerName).
  + **Record (Row):** Represents a single instance of data (e.g., one customer).
* **Database Keys:**
  + **Primary Key:** Uniquely identifies each record within a table.
  + **Foreign Key:** Links one table to another by referencing the primary key of another table.

**Examples:**

* **Customer Table:**
  + Fields: CustomerID (Primary Key), FirstName, LastName, Address.
* **Order Table:**
  + Fields: OrderNumber (Primary Key), CustomerID (Foreign Key), OrderDate.
* **Relationship:**
  + The **CustomerID** in the Order Table acts as a **Foreign Key** linking to the **Customer Table**.
* **Business Rule (Constraint):** Prevents duplication of primary key values.

**3. Non-Relational Databases**

Non-relational databases (NoSQL) do not use structured tables. They are useful when data does not fit into a traditional tabular model.

* **Key-Value Stores:** Data is stored as a collection of key-value pairs.
  + **Example:**
    - **Key:** EmployeeID
    - **Value:** {FirstName: "Mike", LastName: "Chapple", Skill1: "Data Management"}
* **Document Stores:** Similar to key-value but optimized for storing large documents (e.g., JSON, XML).

**Advantages:**

* Efficient data retrieval.
* Flexibility in data structure.
* Suitable for applications with diverse data types.

**4. Database Schema**

A schema defines the structure of a database, including tables, fields, and relationships.

* **Example:** The **AdventureWorks** database schema includes multiple interconnected tables representing business data.
* **Schema Diagram:** Visual representation of how tables relate to one another.

**5. Comparing Relational and Non-Relational Databases**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Relational DB** | **Non-Relational DB** |
| **Data Structure** | Tables with rows and columns | Key-value pairs, documents |
| **Data Integrity** | Enforced through primary and foreign keys | Minimal constraints |
| **Query Language** | SQL | Varies (e.g., MongoDB uses BSON) |
| **Use Case** | Structured, predictable data | Flexible, varied data formats |

**6. Practice Questions and Answers**

**Question 1:**  
Which of the following statements most likely describes the relationship between the two tables (Books and Authors)?

* **Answer:** **C. AuthorID is a foreign key from Books to Authors.**
* **Explanation:** The **AuthorID** appears in both tables, establishing a natural relationship between a book and its author.

**Question 2:**  
You are examining the database used by a concert promoter and find a table named "Concerts." Where would you expect to find information about a single event?

* **Answer:** **C. In one row of the Concerts table.**
* **Explanation:** Each row in a relational database table contains information about a single instance of the table subject.

**7. Exam Tips**

* Understand the **differences between structured, semi-structured, and unstructured data**.
* Memorize the **key characteristics of relational databases** (tables, fields, records, keys).
* Know the **differences between primary keys and foreign keys**.
* Be familiar with **non-relational database types**, especially key-value and document stores.
* Practice identifying **primary and foreign keys** in given database scenarios.

**8. Key Takeaways**

* **Relational Databases:** Use structured data and maintain relationships through keys.
* **Non-Relational Databases:** Handle diverse, flexible data types, often using key-value pairs.
* **Database Schema:** A blueprint that shows table structure and relationships.
* **Exam Focus:** Be able to distinguish between database types, understand key concepts, and apply them in real-world scenarios.

Here are some additional practice questions with detailed explanations, focusing on key concepts

**Practice Question 3: Relational Database Concepts**

You are managing a database that tracks students and their enrolled courses. You have two tables: **Students** and **Courses**. Each student can enroll in multiple courses. Which of the following keys best represents the relationship between these two tables?

**A. StudentID as the primary key in both tables**  
**B. CourseID as the primary key in both tables**  
**C. StudentID as the primary key in the Students table and a foreign key in the Courses table**  
**D. A third table with StudentID and CourseID as a composite key**

**Answer: D. A third table with StudentID and CourseID as a composite key**

**Explanation:**

* In a **many-to-many relationship** (students can enroll in multiple courses, and each course can have multiple students), a third table, often called a **junction table**, is needed.
* The **junction table** will have two columns: **StudentID** and **CourseID**, and the combination of these two fields forms the **composite primary key**.
* This structure effectively links students to their enrolled courses.

**Practice Question 4: Identifying Primary and Foreign Keys**

Consider a **Library Database** with the following tables:

* **Books Table:** BookID (Primary Key), Title, AuthorID
* **Authors Table:** AuthorID (Primary Key), FirstName, LastName
* **Loans Table:** LoanID (Primary Key), BookID (Foreign Key), MemberID, LoanDate

Which of the following statements is correct?

**A. BookID in the Books table is a foreign key.**  
**B. AuthorID in the Books table is a foreign key.**  
**C. LoanID in the Loans table is a foreign key.**  
**D. MemberID in the Loans table is a primary key.**

**Answer: B. AuthorID in the Books table is a foreign key.**

**Explanation:**

* **BookID** in the **Books table** is the primary key, as it uniquely identifies each book.
* **AuthorID** in the **Books table** is a **foreign key** because it references the **AuthorID** in the **Authors table**, establishing a relationship between books and authors.
* **LoanID** is the **primary key** in the **Loans table**, while **BookID** and **MemberID** act as **foreign keys** linking to their respective tables.

**Practice Question 5: Non-Relational Database Types**

Which of the following is an advantage of using a **key-value store** over a relational database?

**A. Supports complex joins between tables**  
**B. Ideal for structured data with complex relationships**  
**C. Provides fast retrieval for simple data lookups**  
**D. Ensures strong data integrity through foreign keys**

**Answer: C. Provides fast retrieval for simple data lookups**

**Explanation:**

* Key-value stores are optimized for **quick retrieval**, making them suitable for simple lookups and storing data without requiring complex relationships.
* Unlike relational databases, they do **not support complex joins** or maintain **strong data integrity** through foreign keys.
* They are particularly useful when data structures are **simple and highly variable**.

**Practice Question 6: Understanding Semi-Structured Data**

Which of the following best describes **semi-structured data**?

**A. Data stored entirely in a table format**  
**B. Data with a rigid schema enforced at the database level**  
**C. Data containing elements of structured and unstructured data**  
**D. Unorganized data collected from sensors and devices**

**Answer: C. Data containing elements of structured and unstructured data**

**Explanation:**

* **Semi-structured data** does not fit completely into a structured table format but has some **organizational properties** (like tags or metadata) that make it more manageable than unstructured data.
* **Examples:** JSON documents, XML files, key-value pairs.

**Practice Question 7: Database Relationships**

In a **Customer-Order** relationship, which of the following best describes the **foreign key**?

**A. A field in the Orders table that uniquely identifies each order**  
**B. A field in the Customers table that stores customer names**  
**C. A field in the Orders table that references the CustomerID from the Customers table**  
**D. A field that combines OrderDate and CustomerName**

**Answer: C. A field in the Orders table that references the CustomerID from the Customers table**

**Explanation:**

* In a **one-to-many relationship** (one customer can place many orders), the **CustomerID** in the **Orders table** acts as a **foreign key** to link the order to a specific customer.

**Practice Question 8: Document Store Characteristics**

Which of the following best describes a **document store**?

**A. Stores data in a strictly tabular format**  
**B. Uses a relational schema to organize data**  
**C. Uses documents, often in JSON or XML format, as its data model**  
**D. Primarily used for complex transaction processing**

**Answer: C. Uses documents, often in JSON or XML format, as its data model**

**Explanation:**

* **Document stores** are a type of non-relational database designed for storing, retrieving, and managing document-oriented information.
* They are highly flexible, allowing for varied data structures within each document.

**Additional Study Tips:**

* **Understand Key Concepts:** Focus on differentiating structured, semi-structured, and unstructured data.
* **Practice Schema Analysis:** Analyze diagrams to identify keys and relationships.
* **Use Real-Life Examples:** Think about everyday applications of databases, such as customer management or inventory tracking.
* **Understand Database Models:** Be clear about when to use relational versus non-relational models.