

Database Management Systems

Advantages of RDBMS

Vikas Thammanna Gowda

01/24/2025

1 Comparing Lists and Relational Databases

1.1 Lists: A Simple Data Management Tool

A list is a simple, linear collection of items, often used for straightforward tasks like maintaining a to-do list or a basic contact directory. While suitable for managing limited and uncomplicated data, lists encounter significant challenges when handling complex datasets:

1.1.1 Challenges of Using Lists

- **Data Redundancy:** Lists can lead to duplicate entries, making data management inefficient and increasing the risk of inconsistencies.
- **Lack of Structure:** Without a defined schema, lists struggle to represent intricate relationships between data points, hindering effective data retrieval and analysis.
- **Scalability Issues:** As data volume grows, lists become cumbersome to manage, leading to decreased performance and increased difficulty in data handling.

1.2 Relational Databases: Structured Data Management

In contrast, a relational database organizes data into structured tables with predefined schemas, enabling the establishment of relationships between different data entities. This structure offers several advantages:

1.2.1 Advantages of Relational Databases

- **Data Integrity and Consistency:** Relational databases enforce data integrity through constraints and relationships, ensuring that the data remains accurate and reliable.
- **Complex Querying Capabilities:** Utilizing Structured Query Language (SQL), relational databases allow for sophisticated queries, facilitating efficient data retrieval and manipulation.
- **Scalability and Performance:** Designed to handle large volumes of data, relational databases can scale to meet increasing demands, maintaining performance and reliability.
- **Support for ACID Transactions:** Relational databases adhere to ACID (Atomicity, Consistency, Isolation, Durability) properties, ensuring reliable and secure transactions.

2 Example 1

Consider a scenario where you're managing information about students and the courses they enroll in. Using a simple list, you might represent this data as follows:

Student Name	Course Name	Instructor	Enrollment Date
Alice Smith	Mathematics	Dr. Johnson	2024-09-01
Bob Johnson	Physics	Dr. Lee	2024-09-02
Alice Smith	Chemistry	Dr. Brown	2024-09-03
Carol White	Mathematics	Dr. Johnson	2024-09-04

2.1 Challenges with Using Lists

- **Data Redundancy:** Information such as the student's name and instructor's name is repeated for each course enrollment. For instance, "Alice Smith" appears multiple times, leading to unnecessary duplication.
- **Data Inconsistency:** If a student's name changes (e.g., due to marriage), updating every occurrence in the list is error-prone and may lead to inconsistencies if some entries are overlooked.
- **Difficulty in Data Retrieval:** Extracting specific information, such as all courses taught by a particular instructor or all students enrolled in a specific course, requires scanning the entire list, which is inefficient and cumbersome.
- **Scalability Issues:** As the number of students and courses increases, the list becomes increasingly difficult to manage, leading to potential performance degradation and higher chances of errors.

2.2 Relational Database Example

A relational database addresses these challenges by organizing data into structured tables with defined relationships. Here's how the same information can be represented:

Students Table

StudentID	Student Name
1	Alice Smith
2	Bob Johnson
3	Carol White

Courses Table

CourseID	Course Name	Instructor
101	Mathematics	Dr. Johnson
102	Physics	Dr. Lee
103	Chemistry	Dr. Brown

Enrollments Table

EnrollmentID	StudentID	CourseID	Enrollment Date
1	1	101	2024-09-01
2	2	102	2024-09-02
3	1	103	2024-09-03
4	3	101	2024-09-04

2.2.1 Advantages of Relational Databases

- **Data Integrity and Consistency:** By assigning unique identifiers (e.g., `StudentID`, `CourseID`), each entity is stored once, reducing redundancy. Updates to a student's name are made in a single location (the `Students` table), ensuring consistency across the database.
- **Efficient Data Retrieval:** Structured Query Language (SQL) allows for complex queries. For example, to find all courses a particular student is enrolled in, you can perform a `JOIN` operation between the `Students` and `Enrollments` tables, streamlining data retrieval.
- **Scalability and Maintenance:** Adding new students, courses, or enrollments involves inserting new records into the respective tables without altering existing data structures. This modular approach simplifies maintenance and supports scalability.
- **Data Relationships:** Relational databases explicitly define relationships between different data entities through foreign keys (e.g., `StudentID` in the `Enrollments` table referencing the `Students` table), ensuring referential integrity and enabling complex data associations.

VIKAS

3 Example 2

In a simple list, you might record each employee's details along with their project assignments in a single entry:

Employee Name	Department	Project Name	Role	Start Date
John Doe	IT	Website Revamp	Developer	2024-01-15
Jane Smith	Marketing	Ad Campaign	Coordinator	2024-02-01
John Doe	IT	App Development	Lead Developer	2024-03-10
Emily Davis	HR	Recruitment Drive	Recruiter	2024-04-05

3.0.1 Challenges with Using a Simple List

- **Data Redundancy:** Employee information, such as "John Doe," is repeated for each project assignment, leading to duplicate data entries.
- **Data Inconsistency:** If an employee's department changes, updating each occurrence manually increases the risk of errors and inconsistencies.
- **Limited Query Capabilities:** Retrieving specific information, such as all projects assigned to a particular department, requires scanning the entire list, which is inefficient.
- **Scalability Issues:** As the number of employees and projects grows, the list becomes unwieldy, making data management and retrieval more complex.

3.1 Using a Relational Database

A relational database organizes this information into structured tables with defined relationships, enhancing data integrity and retrieval efficiency.

3.1.1 Employees Table

EmployeeID	Employee Name	Department
1	John Doe	IT
2	Jane Smith	Marketing
3	Emily Davis	HR

3.1.2 Projects Table

ProjectID	Project Name
101	Website Revamp
102	Ad Campaign
103	App Development
104	Recruitment Drive

3.1.3 Assignments Table

AssignmentID	EmployeeID	ProjectID	Role	Start Date
1	1	101	Developer	2024-01-15
2	2	102	Coordinator	2024-02-01
3	1	103	Lead Developer	2024-03-10
4	3	104	Recruiter	2024-04-05

3.1.4 Advantages of Using a Relational Database

- **Data Integrity and Consistency:** Employee and project information is stored once in their respective tables, reducing redundancy. Updates to an employee's department are made in a single location, ensuring consistency across the database.

- **Efficient Data Retrieval:** Structured Query Language (SQL) enables complex queries. For example, to find all projects assigned to the IT department, you can join the **Employees** and **Assignments** tables based on **EmployeeID**.
- **Scalability and Maintenance:** Adding new employees, projects, or assignments involves inserting new records into the respective tables without altering existing data structures, facilitating scalability and ease of maintenance.
- **Data Relationships:** Relational databases explicitly define relationships between different data entities through foreign keys (e.g., **EmployeeID** in the **Assignments** table referencing the **Employees** table), ensuring referential integrity and enabling complex data associations.

VIKAS

4 Problems

1. Garden Glory is a partnership that provides gardening and yard maintenance services to individuals and organizations. Garden Glory is owned by two partners. They employ two office administrators and a number of full- and part-time gardeners. Garden Glory will provide one-time garden services, but it specializes in ongoing service and maintenance. Many of its customers have multiple buildings, apartments, and rental houses that require gardening and lawn maintenance services.

1. Create a sample list of owners, properties and services. Your list should include, at a minimum, owner name, phone, and billing address, as well as property name, type, and address.
2. Describe modification problems that are likely to occur if Garden Glory attempts to maintain the list in a spreadsheet.
3. Split the list into tables such that each has only one theme. Create appropriate ID columns. Use a linking column to represent the relationship between a property and an owner.
4. Demonstrate that the modification problems you identified in 2 have been eliminated.
5. Keys?

VIKAS

2. The Queen Anne Curiosity Shop sells both antiques and current-production household items that complement or are useful with the antiques. For example, the store sells antique dining room tables and new tablecloths. The antiques are purchased from both individuals and wholesalers, and the new items are purchased from distributors. The store's customers include individuals, owners of bed-and-breakfast operations, and local interior designers who work with both individuals and small businesses. The antiques are unique, although some multiple items, such as dining room chairs, may be available as a set (sets are never broken). The new items are not unique, and an item may be reordered if it is out of stock. New items are also available in various sizes and colors (for example, a particular style of tablecloth may be available in several sizes and in a variety of colors).

1. Create a sample list of purchased inventory items and vendors and a second list of customers and sales. The first list should include inventory data, such as a description, manufacturer and model (if available), item cost, and vendor identification and contact data you think should be recorded. The second list should include customer data you think would be important to The Queen Anne Curiosity Shop, along with typical sales data.
2. Describe problems that are likely to occur when inserting, updating, and deleting data in these spreadsheets.
3. Attempt to combine the two lists you created in 1 into a single list. What problems occur as you try to do this?
4. Split the spreadsheets you created in 1 into tables such that each has only one theme. Create appropriate ID columns.
5. Explain how the tables in your answer to 4 will eliminate the problems you identified in 2.
6. What is the relationship between the tables you created from the first spreadsheet and the tables you created from the second spreadsheet? If your set of tables does not already contain this relationship, how will you add it into your set of tables?
7. Keys?