Why Databases?

Understanding the Foundation of Modern Data Management

Learning Objectives

By the end of this session, you will be able to:

- **Identify** how data is present in everyday life
- **Analyze** problems with non-database data storage methods
- **Explain** how database systems solve these problems
- **Describe** the core functions of a DBMS

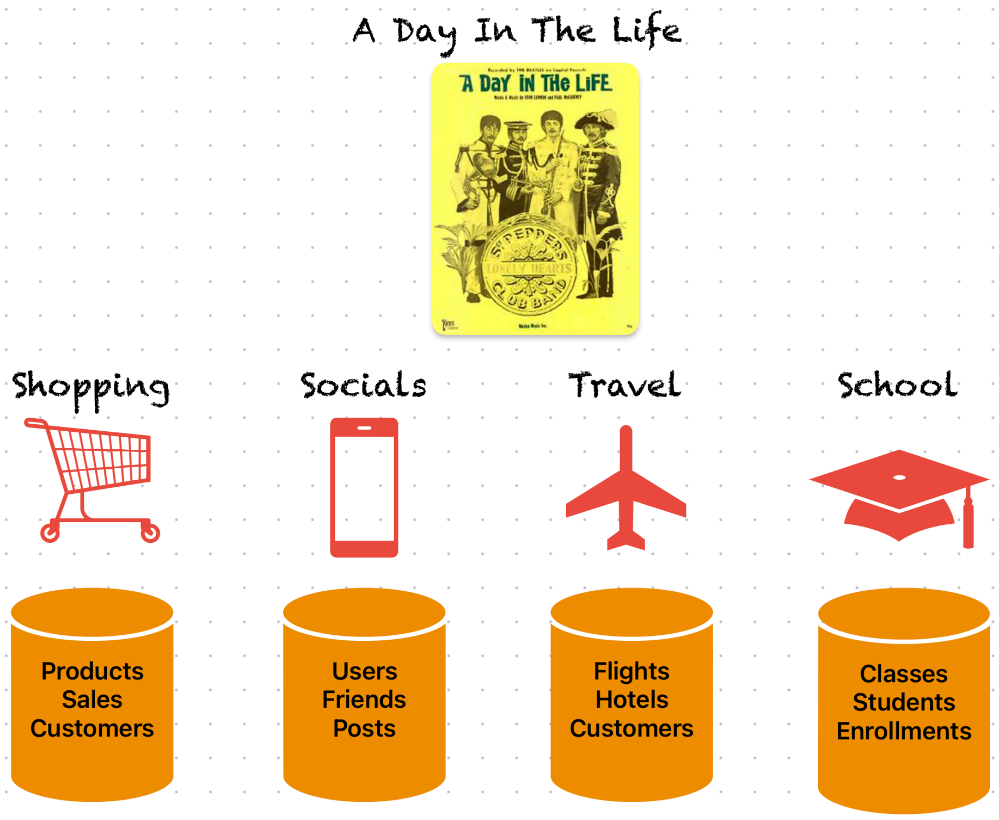
Part 1: Data Is Ubiquitous

You Can't Escape Data!

Your Data Journey

From Birth to Death (and Everything In Between)

- **Birth**: Birth certificate 📄
- **Education**: School records, grades 🎓
- **Shopping**: Purchase history, preferences 🛍️
- **Social**: Social media posts, connections 👥
- **Travel**: Bookings, loyalty programs ✈️
- **Health**: Medical records, prescriptions 🏥
- **Work**: Employment records, payroll 💼
- **Death**: Death certificate ⚰️



Products
Sales
Customers

Users
Friends
Posts

Flights
Hotels
Customers

Classes
Students
Enrollments

🎯 Active Learning: Personal Data Touchpoints

Think-Pair-Share (3 minutes)

1. **Think** (1 min): List 10 data touchpoints from TODAY
2. **Pair** (1 min): Share with your neighbor
3. **Share** (1 min): What surprised you?

Examples to get started:

- Alarm clock app
- Transit card swipe
- Email login
- Credit card purchase

Who Manages All This Data?

Organizations (school, government, business, ...)!

1. Store vast amounts of data 💾
2. Secure sensitive information 🔒
3. Maintain data consistency ✅
4. Retrieve useful insights 🔍

How? → DATABASES!

Part 2: Life Without Databases

The Dark Ages of Data Management

Storage 1: Paper Files 📁

Organization Methods:

- Filing cabinets
- Folders & labels
- Manual indexing

Major Issues:

- **Slow** retrieval and reporting
- **Labor-intensive** aggregation
- **Physical space** requirements
- **No backup** for disasters

🎯 Thought Exercise: Paper Nightmare

Scenario:

You have **1,000 index cards** with student records

Your Task:

Calculate the average GPA of all California CS majors

Consider:

- How long would this take? ⏱️
- What if you made an error? 😰
- What if someone asks for a different report tomorrow? 🔄

Storage 2: Computer Files 💻

The Digital Evolution (But Still Problems!)

Example: Customer Orders File

Order ID	Customer Name	Email	Product	Price	Quantity
1001	John Smith	john@email.com	Laptop	$999.99	1
1001	John Smith	john@email.com	Mouse	$29.99	2
1002	Jane Doe	jane@email.com	Keyboard	$79.99	1

Three Major Problems

1. Structural Dependency 🏗️

- Programs depend on file structure
- Adding a field = rewrite all programs
- No ad-hoc queries possible

2. Data Dependency 🔗

- Programs depend on data types
- Changing integer to float = modify all programs
- Physical format tied to logical format

3. Data Redundancy 🔄

- Same data repeated everywhere
- Storage waste + integrity nightmares
- More on this next

The Redundancy Problem

Why Is Redundancy Bad?

1. **Storage Cost** 💰
   - Duplicate data = wasted space
2. **Data Integrity** ⚠️
   - Multiple copies = inconsistency risk
3. **Security Risk** 🔓
   - More copies = more vulnerability points

Data Anomalies: The Triple Threat

1. Update Anomalies ✏️

Problem: Change John's email → Update multiple rows

Risk: Miss some → Inconsistent data

2. Insert Anomalies ➕

Problem: Can't add customer without order

Risk: Incomplete data representation

3. Delete Anomalies ❌

Problem: Delete all orders → Lose customer info

Risk: Unintended data loss

🎯 Active Learning: Anomaly Hunt

Box Office Data Exercise (5 minutes)

Movie Title	Director	Actor	Theater	Show Date	Tickets	Revenue
Top Gun	J. Kosinski	Tom Cruise	AMC	2024-07-15	285	$4,417
Top Gun	J. Kosinski	J. Connelly	AMC	2024-07-15	285	$4,417
Avatar 2	J. Cameron	S. Worthington	AMC	2024-07-16	312	$4,836

Questions:

1. What redundancies do you see?
2. What happens if we change the theater name?
3. Can we store a movie without a showing?

Storage 3: Spreadsheets 📊

The Double-Edged Sword

✅ Pros:

- User-friendly interface
- Quick calculations
- Accessible to non-programmers

❌ Cons:

- **NOT a database!**
- No concurrency control
- Limited data integrity
- Poor security model
- Still has redundancy issues

Part 3: Enter Database Systems

The Solution to Our Problems!

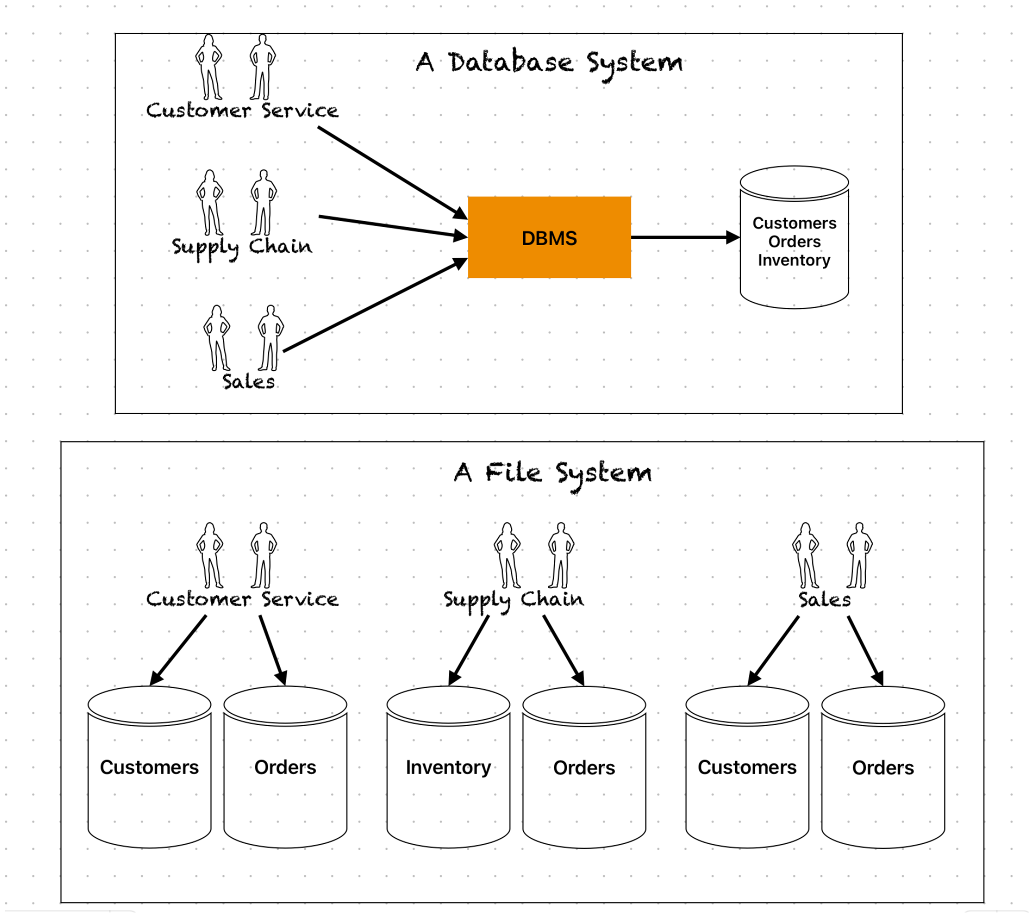
What Is a Database System?

Core Concept:

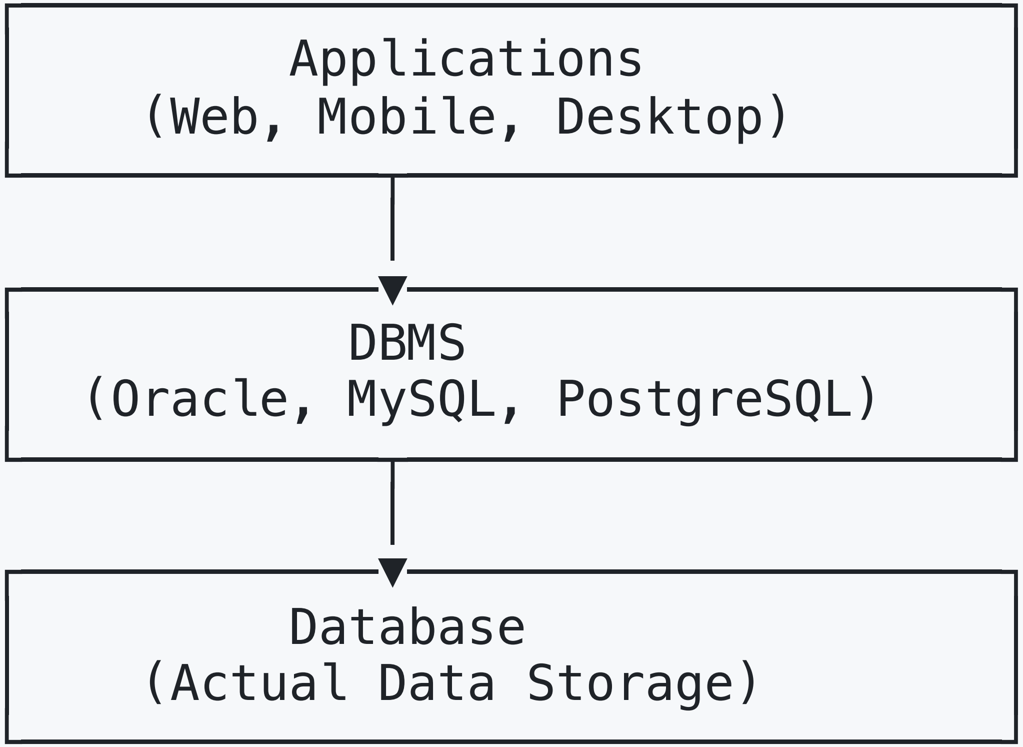
Logically related data in a **single logical repository**

Key Components:

- **Database**: The data itself
- **DBMS**: Database Management System (the software)
- **Users**: Applications and people



DBMS Architecture



Applications
(Web, Mobile, Desktop)

DBMS
(Oracle, MySQL, PostgreSQL)

Database
(Actual Data Storage)

The 8 Core Functions of a DBMS

1. Data Dictionary Management 📚

- Maintains **metadata** (data about data)
- Provides data abstraction
- Eliminates structural/data dependence

2. Data Storage Management 💾

- Manages physical storage structures
- Optimizes data access performance
- Handles indexing and caching

Core Functions (continued)

3. Data Transformation 🔄

- Converts user input to required format
- Example: US date (MM/DD) → UK date (DD/MM)

4. Security Management 🔐

- **Authentication**: Who are you?
- **Authorization**: What can you do?
- Row-level and column-level security

Core Functions (continued)

5. Concurrency Control 🔀

- Multiple users, same time, no conflicts
- Transaction management
- Locking mechanisms

6. Backup and Recovery 💾

- Regular backups
- Point-in-time recovery
- Disaster recovery planning

Core Functions (final)

7. Data Integrity Management ✅

- Enforces business rules
- Prevents invalid data
- Maintains relationships

8. Access Languages & APIs 🗣️

- **SQL**: Structured Query Language
- APIs for programming languages
- Query optimization

SQL: The Universal Language

Declarative, Not Procedural

Traditional Programming:

```
1. Open file
2. Read each record
3. Check if city = "NYC"
4. If yes, add to count
5. Return count
```

SQL:

```sql
SELECT COUNT(*)
FROM customers
WHERE city = 'NYC'
```

You say WHAT you want

Not HOW to get it

🎯 Active Learning: Function Matching

Match the Scenario to the DBMS Function (3 minutes)

Scenarios:

A. "Two tellers withdraw from same account simultaneously"

B. "Need to change phone format without breaking apps"

C. "Power outage during finals week"

D. "Junior employee can't see executive salaries"

Functions:

1. Security Management
2. Concurrency Control
3. Backup & Recovery
4. Data Dictionary Management

Spreadsheet vs Database

The Final Verdict

Feature	Spreadsheet	Database
Multi-user	❌ Limited	✅ Full support
Data Integrity	❌ Manual	✅ Automatic
Security	❌ Basic	✅ Granular
Scalability	❌ Small data	✅ Big data
Relationships	❌ Manual	✅ Built-in
Recovery	❌ Manual saves	✅ Automatic

Conclusion: Spreadsheets are tools, databases are systems!

Key Takeaways 🎯

1. **Data is everywhere in modern life**

2. **Non-database solutions have critical limitations:**

    - Dependencies, redundancy, anomalies

3. **Database systems solve these problems through:**

    - Centralized management
    - 8 core DBMS functions
    - SQL for universal access

4. **Choose the right tool: Spreadsheet ≠ Database**