

Understanding Databases

This paper is submitted in partial fulfillment
of the requirements for
Technical English

孟芙蘭
Shu-Te University
October 2025

About the Author

Franzyn Leane Manzanades 孟芙蘭 (s24101304@stu.edu.tw) is an international student from Argao, Cebu, Philippines—a town rich in heritage and strong community values. Growing up in this environment, Franzyn developed a deep sense of hard work, humility, and respect for others. Excelling academically from an early age, she graduated as class valedictorian in both elementary and junior high school, actively participating in extracurriculars and academic competitions that broadened her horizons.



Franzyn pursued her senior high school studies in the Science, Technology, Engineering, and Mathematics (STEM) strand, graduating with high honors. She briefly studied Mechanical Engineering and Military Science at a recognized state university before choosing to further her education in Taiwan. Currently, she is undertaking a Bachelor's degree in Information Management at Shu-Te University, where she combines technical knowledge with practical experience.

Her motivation to learn is deeply rooted in the desire to improve herself and support her family. Living and working in Taiwan has enriched her both personally and professionally—teaching her discipline, responsibility, and cultural understanding. Franzyn hopes to bring back the lessons and skills she has gained to her home country, contributing to her community and inspiring others to pursue their dreams.

Table of Contents

- I. Preface
- II. Introduction to Databases
- III. Types of Databases
- IV. Components of a Database
- V. What Is a Database Management System (DBMS)
- VI. SQL: The Language of Databases
- VII. Database Design Best Practices
- VIII. Database Security
- IX. Trends in Modern Databases
- X. Careers in Databases
- XI. Conclusion
- XII. References

Preface

In today's digital world, data is often called the “new oil” — a valuable resource that drives innovation, decision-making, and everyday operations across industries. Whether you're a student, a professional, or simply curious about how information is organized and managed, understanding databases is essential.

This article aims to provide a clear and comprehensive overview of databases — from their basic components and types to advanced topics like database management systems, SQL, and modern trends. Along the way, you'll find practical insights into best practices, security, and career opportunities in the field.

As data continues to grow in both volume and importance, having a solid grasp of databases is a skill that opens doors to many possibilities. This guide is designed to be accessible to beginners while still offering valuable knowledge for those looking to deepen their understanding.

I hope this article serves as a helpful starting point for your journey into the world of databases and inspires you to explore the exciting possibilities within this vital field.

I. Introduction to Databases

In today's digital world, data is everywhere. From logging into your favorite social media account to making an online purchase, almost every digital action involves a database. But what exactly is a database?

A database is a collection of information that is organised and kept in a computer so that it can be examined using a computer programme to get information from it and in theory, consists of two major components: data and information. So the ultimate objective is to manage the data so that it can be transformed into the appropriate information and data retrieval, deletion, and editing can be done conveniently, effectively, efficiently, and accurately (B. Rawat, S. Purnama, & Mulyati, 2021).

A database is an organized collection of data that can be easily accessed, managed, and updated. It allows individuals and organizations to store information in a structured format, making it easier to retrieve and manipulate as needed. Whether you're booking a flight, streaming music, or checking your grades online—databases are working behind the scenes.

II. Types of Databases

In this piece, we'll explore the main types of databases, how they differ from one another, and where each is most effectively used. Over time, different types of databases have been developed to meet various data needs and usage scenarios. In this section, we'll explore the main types of databases, how they differ in structure and functionality, and where each type is most effectively used. From traditional relational databases to modern NoSQL systems and specialized options like graph or time-series databases, understanding these types helps in choosing the right database for a specific task. The most common types are as follows:

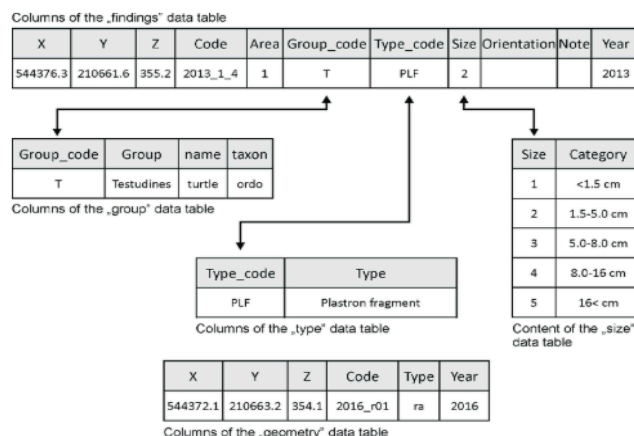


Figure 3. Example of a Relational Database Model
(Source: ResearchGate)

1. Relational Database Management System (RDBMS)

It organizes data into tables (relations) composed of rows and columns. Uses primary keys to uniquely identify rows and foreign keys to establish relationships between

tables. Queries are written in SQL (Structured Query Language), which allows for efficient data manipulation and retrieval.

Examples: MySQL oracle, Microsoft SQL Server and Postgre SQL.

2. NoSQL DBMS

They are designed to handle large-scale data and provide high performance for scenarios where relational models might be restrictive. They store data in various non-relational formats, such as key-value pairs, documents, graphs or columns. These flexible data models enable rapid scaling and are well-suited for unstructured or semi-structured data.

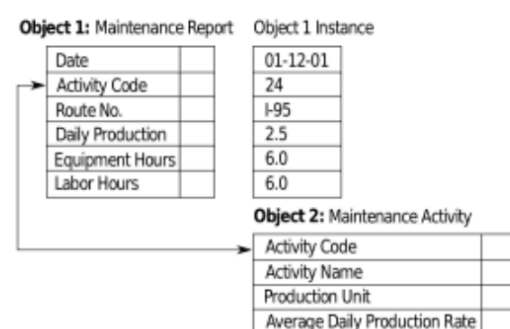
Examples: MongoDB, Cassandra, DynamoDB and Redis.

3. Object-Oriented DBMS (OODBMS)

It integrates object-oriented programming concepts into the database environment, allowing data to be stored as objects. Supports complex data types and relationships, making it ideal for applications requiring advanced data modeling and real-world simulations.

Examples: ObjectDB, db4o.

Object-Oriented Model



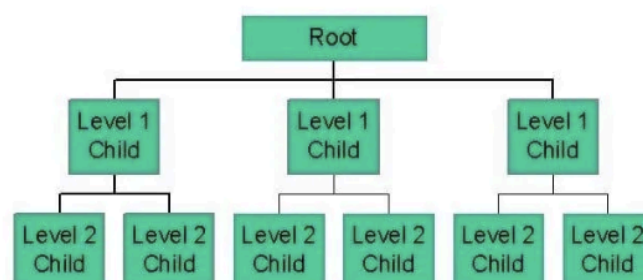
*An example of an Object-Oriented Database Model
(Source: Data Integration Glossary)*

4. Hierarchical Database

Organizes data in a tree-like structure, where each record (node) has a single parent and has multiple children. This model is similar to a file system with folders and subfolders. It is efficient for storing data with a clear hierarchy, such as organizational charts or file directories. Navigation is fast and predictable due to the fixed structure. It lacks flexibility and is difficult to restructure or handle complex many-to-many relationships.

Example: IBM Information Management System (IMS).

Hierarchical database model

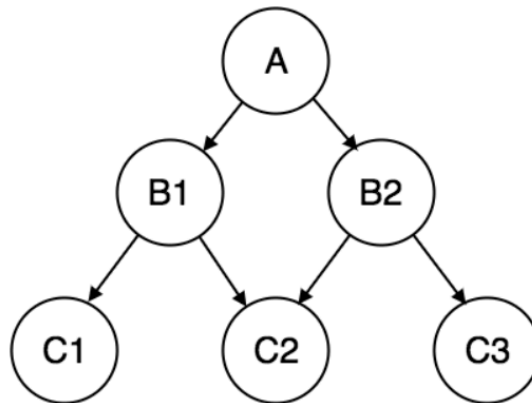


Hierarchical Database Model (Source: DBMS Internals)

5. Network Database

It uses a graph-like model to allow more complex relationships between entities.

Unlike the hierarchical model, it permits each child to have multiple parents, enabling many-to-many relationships. Data is represented using records and sets, where sets define the relationships. It is more flexible than the hierarchical model and better suited for applications with complex data linkages.



A Network Database Model (Source: The Intact One)

Example: Integrated Data Store (IDS), TurboIMAGE

6. Cloud-Based Database

They are hosted on cloud computing platforms like AWS, Azure or Google Cloud. They offer on-demand scalability, high availability, automatic backups and remote accessibility. These databases can be relational (SQL) or non-relational (NoSQL) and are maintained by cloud service providers, reducing administrative overhead. They support modern application requirements, including distributed access and real-time analytics.

Example: Amazon RDS (for SQL), MongoDB Atlas (for NoSQL), Google BigQuery

III. Components of a Database

To understand how a database works, it's important to know its key components. These parts work together to organize, store, and retrieve data efficiently. Whether you're using a simple database or a complex system, the basic building blocks—like tables, fields, and keys—remain the same. In this section, we'll look at the essential components that make up a database and how they help manage and structure data effectively.

- **Tables:** Basic units where data is stored.
- **Fields (columns):** Describe the type of data.
- **Records (rows):** Individual entries.
- **Primary Key:** Unique identifier for a record.

- **Foreign Key:** Connects data across tables.
 - **Index:** Improves data retrieval speed.
 - **View:** Virtual table created from a query.
 - **Schema:** Blueprint of the database structure.
-

IV. What is a DBMS?

A **Database Management System (DBMS)** is the software that makes it possible to interact with a database. It serves as the bridge between users, applications, and the data itself—making it easier to store, access, and manage information efficiently. In this section, we'll look at the key roles a DBMS plays, such as ensuring data security, maintaining integrity, and supporting backup and recovery. We'll also touch on some of the most popular DBMS options used today.

Roles of a DBMS:

- Data storage and retrieval
- Security and access control
- Backup and recovery
- Data integrity enforcement

Popular DBMS: MySQL, PostgreSQL, Oracle, MongoDB.

V. SQL: The Language of Databases

SQL (Structured Query Language) is the standard language used to interact with most relational databases. It allows users to perform a wide range of tasks—from retrieving and updating data to managing database structures. SQL is essential for anyone working with data, as it provides a clear and powerful way to communicate with the database. In this section, we'll cover some of the most common SQL commands and how they are used to handle data and relationships between tables.

Basic SQL commands:

- ❖ **CREATE:** to create a database and its objects like (table, index, views, store procedure, function and triggers)
 - ❖ **ALTER:** Alters the structure of the existing database
 - ❖ **DROP:** Delete all records from a database table
 - ❖ **JOINS:** Used to combine data from multiple tables based on relationships
 - ❖ **SELECT:** Retrieves data from a database
 - ❖ **INSERT:** Adds new data into a table
 - ❖ **UPDATE:** Modifies existing data within a table
 - ❖ **DELETE:** Removes data
-

VI. Database Design Best Practices

Designing a database properly from the start is crucial for ensuring efficiency, accuracy, and scalability. Good database design helps prevent data redundancy, improves performance, and makes it easier to manage and grow the system over time. In this section, we'll look at key best practices—such as normalization, the use of keys, and proper indexing—that lead to a well-structured and reliable database.

- **Normalization:** Organize data to reduce redundancy.
 - **Use of primary/foreign keys:** Ensure data consistency.
 - **Data types and constraints:** Define rules for valid data.
 - **Scalability planning:** Prepare for future growth.
 - **Regular indexing:** For faster queries.
-

VII. Database Security

As organizations increase their reliance on, possibly distributed, information systems for daily business, they become more vulnerable to security breaches even as they gain productivity and efficiency advantages. Though a number of techniques, such as encryption and electronic signatures, are currently available to protect data when transmitted across sites, a truly comprehensive approach for data protection

must also include mechanisms for enforcing access control policies based on data contents, subject qualifications and characteristics, and other relevant contextual information, such as time. It is well understood today that the semantics of data must be taken into account in order to specify effective access control policies. Also, techniques for data integrity and availability specifically tailored to database systems must be adopted. In this respect, over the years, the database security community has developed a number of different techniques and approaches to assure data confidentiality, integrity, and availability (E. Bertino and R. Sandhu, 2005). **Database security** is essential for protecting sensitive information and ensuring that data remains accurate, private, and accessible only to authorized users. With growing concerns about data breaches and cyber threats, it's important to follow best practices to secure your database. In this section, we'll cover key security measures such as user access control, data encryption, regular backups, and protection against common attacks like SQL injection.

- Setting user roles and permissions
 - Encrypting sensitive data
 - Regular backups and recovery plans
 - Preventing SQL injection through parameterized queries
-

VIII. Trends in Modern Databases

In today's fast-paced digital world, data is one of the most valuable assets for businesses. Whether it's used to understand customer behavior, improve operational efficiency, or drive decision-making, the ability to store, manage, and analyze data effectively is crucial. As technology advances, the demands placed on databases have grown, with businesses needing systems that can handle more data, more complex queries, and faster processing. The rise of cloud computing, big data, and artificial intelligence (AI) has made these needs even more pressing, pushing companies to rethink how they approach database management (N. Miryala, 2022).

The world of databases is constantly evolving to meet the demands of modern applications and growing data volumes. New technologies and approaches are shaping how data is stored, managed, and accessed. In this section, we'll explore current trends in modern databases, such as the rise of cloud-based solutions, the use of AI for smarter data management, and the growing importance of distributed systems and big data integration.

- **Cloud databases** offer flexibility and remote access.
 - **Database-as-a-Service (DBaaS)** like Firebase simplifies management.
 - **Big Data integration** combines large datasets for insights.
 - **AI in DBMS** helps with automation, optimization, and predictive analytics.
 - **Distributed databases** improve performance and fault tolerance.
-

IX. Careers in Databases

If you enjoy working with data, here are some career paths:

- **Database Administrator (DBA)**
- **Data Analyst**
- **Data Engineer**
- **Backend Developer**
- **Business Intelligence (BI) Specialist**

As data becomes more central to decision-making and technology, careers in database-related fields are in high demand. Whether you're interested in maintaining databases, analyzing data, or building data-driven applications, there are many paths to explore. In this section, we'll look at some key roles—like Database Administrator, Data Analyst, and Data Engineer—and how each contributes to managing and making sense of large volumes of data.

X. Theoretical Framework

This framework integrates these theories to explain how databases handle data integrity, query efficiency, and scalability. It positions databases as a bridge between raw data and actionable insights, emphasizing their role in fields like business intelligence, AI, and cloud computing. This article aims to discuss and understand databases thus, this is related to the Relational Model Theory by Dr. Edgar F. Codd (1970). Databases evolved as a response to the need for organized data storage in the digital era. The theoretical foundations began with early file systems in the 1960s and matured with the relational model in the 1970s.

**Relational Model Theory
by Dr. Edgar F. Codd**



**revolutionized databases by modeling
data as relations (tables) based on
mathematical set theory and
relational algebra**



Understanding Databases

XI. Conclusion

Databases are a foundational part of the digital age. From websites and apps to entire industries, the ability to store and manage data efficiently is essential. Understanding how databases work is not only valuable for IT professionals but for anyone living in an increasingly data-driven world. Whether you're building software, analyzing trends, or just curious—learning about databases opens up a world of possibilities. In an era where data powers every aspect of our digital lives, databases stand as the unsung heroes behind seamless online experiences, informed decision-making, and technological innovation. In a whole, this article has explored the fundamental concepts of databases—from their basic components and diverse types (such as relational and NoSQL systems) to advanced topics like SQL querying, database management systems, design best practices, security measures, and emerging trends in cloud computing and AI integration.

References

- B. Rawat, S. Purnama, & Mulyati (2021). MySQL Database Management System (DBMS) On FTP Site LAPAN Bandung. International Journal of Cyber and IT Service Management (IJCITSM), 1(2), 173-179. Retrieved from <https://iast-journal.org/ijcitsm/index.php/IJCITSM/article/view/47>
- E. Bertino and R. Sandhu (2005). Database security - concepts, approaches, and challenges in IEEE Transactions on Dependable and Secure Computing, vol. 2, no. 1, pp. 2-19, Jan.-March 2005, doi: 10.1109/TDSC.2005.9. keywords: {Databases;Data security;Access control;Information security;Protection;Grid computing;Information systems;Productivity;Cryptography;Qualifications;Index Terms- Data confidentiality;data privacy;relational and object databases;XML.} Retrieved from <https://ieeexplore.ieee.org/abstract/document/1416861>
- N. Miryala (2022). Emerging Trends and Challenges in Modern Database Technologies: A Comprehensive Analysis. Retrieved from https://www.researchgate.net/profile/Naresh-Kumar-Miryala/publication/386275409_Emerging_Trends_and_Challenges_in_Modern_Database_Technologies_A_Comprehensive_Analysis/links/674b41dd359dcb4d9d46352f/Emerging-Trends-and-Challenges-in-Modern-Database-Technologies-A-Comprehensive-Analysis.pdf