TOPIC 1 – INTRODUCTION TO DATABASE

LEARNING OUTCOMES

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

1. To describe traditional file systems, its limitations and the concepts and needs of database
2. To identify evolution of database
3. To discuss the different types Database Architecture
4. To identify roles of people in managing database

INTRODUCTION

This chapters aim to introduce database concepts and its environment.

1.1 Introduction to Database

"Hey there! Let's talk about databases and how they play a role in organizing and communicating information effectively.

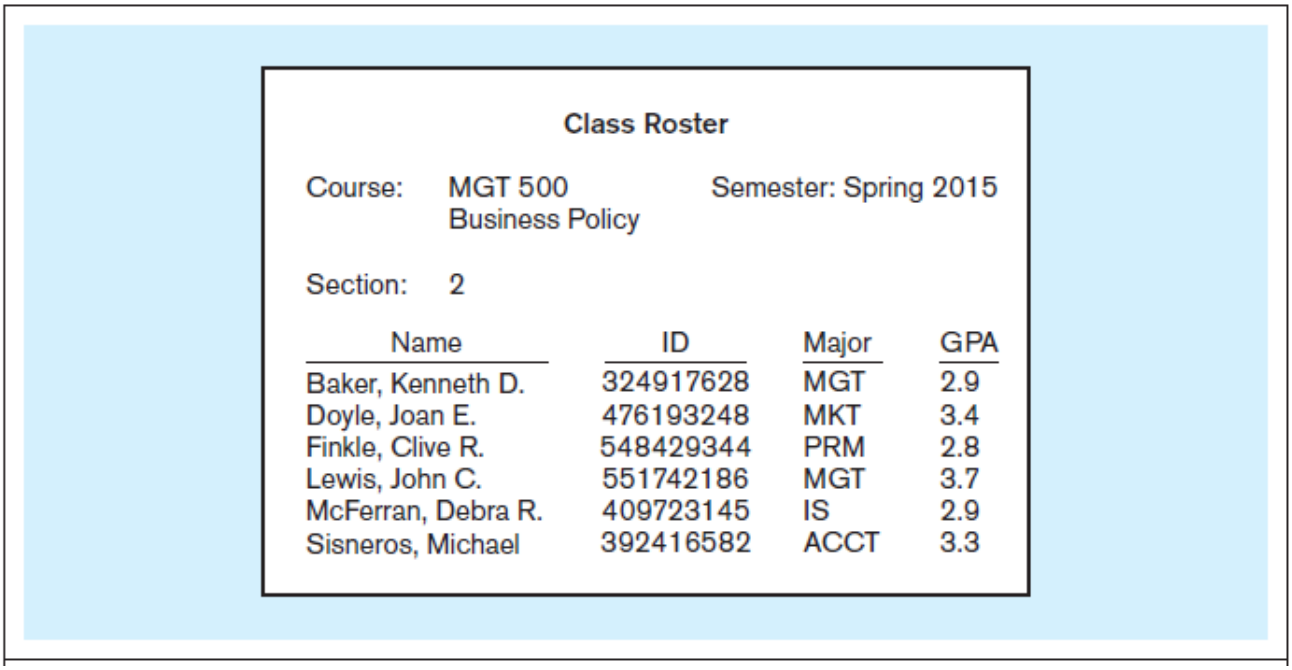
So, what is a database? Well, it's like an organized collection of data that is logically related. But here's the thing: data by itself isn't very useful. It needs to go through some processing, summarizing, and organizing to become valuable for decision makers and knowledge workers like yourself.

Once the data is processed, it transforms into information. This information is often presented to users in the form of reports or graphical displays. You know, those neat charts and graphs that help us make sense of the data at a glance.

Now, let's dive into metadata. Metadata refers to the underlying structure of the data in a database. When you design a database, you're actually specifying its metadata. And when you start populating the database, that's when you're putting the actual data into it.

Imagine this: Figure 1 displays a report that showcases various types of data entities. In this report, you'll find courses and the sections of those courses. Moreover, it also includes information about the students enrolled in each section. It's fascinating how just by organizing the raw data, we can transform it into something truly valuable and informative.

So, you see, databases are all about facilitating communication and providing meaningful insights. They enable decision makers and knowledge workers to access and understand data in a way that supports better decision-making and enhances our overall knowledge.



Class Roster

Course: MGT 500 Business Policy Semester: Spring 2015

Section: 2

Name	ID	Major	GPA
Baker, Kenneth D.	324917628	MGT	2.9
Doyle, Joan E.	476193248	MKT	3.4
Finkle, Clive R.	548429344	PRM	2.8
Lewis, John C.	551742186	MGT	3.7
McFerran, Debra R.	409723145	IS	2.9
Sisneros, Michael	392416582	ACCT	3.3

Figure 1 – Data Context (Source from Database Systems, Thomas Conolly,2020)

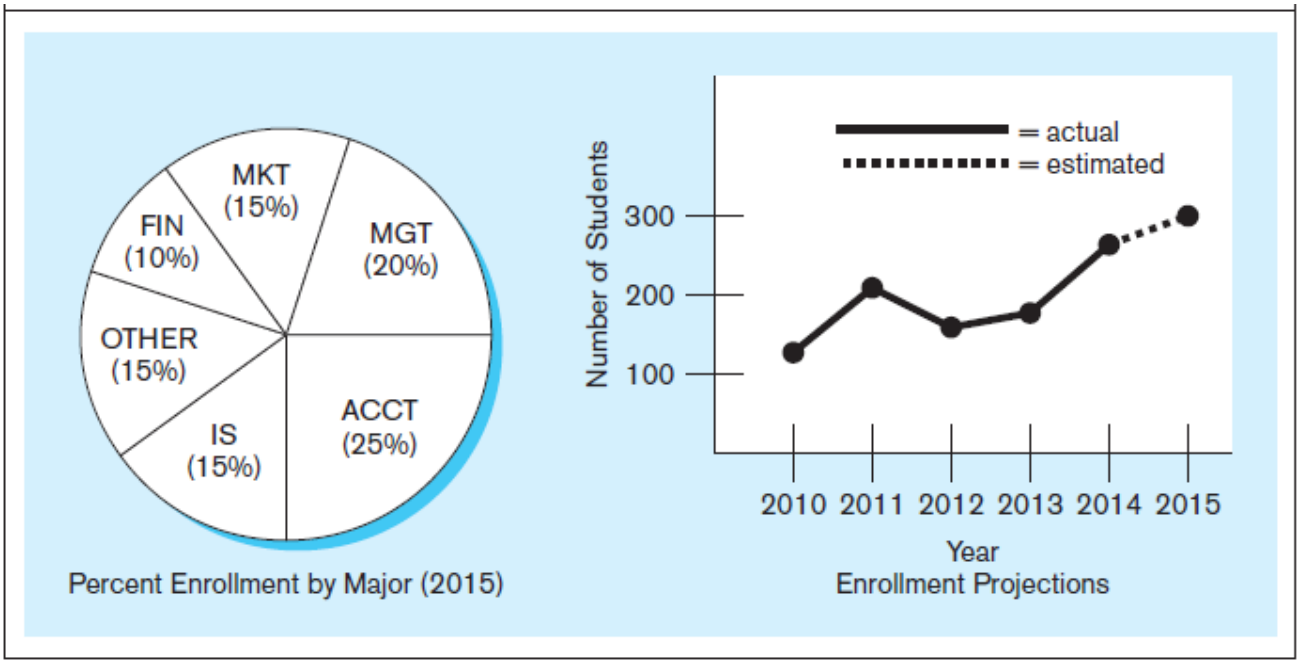


Figure 2 – Summarize Data Figure 1 – Data Context (Source from Database Systems, Thomas Conolly,2020)

In Figure 2, we have summaries of the data. Instead of looking at individual data units, the data has been processed and grouped into aggregates and categories. This helps us get a broader understanding of the data and identify patterns or trends.

When we talk about aggregated data, we're referring to the sums and averages that represent a collection of individual data points. It's like zooming out and seeing the bigger picture. By aggregating the data, we can gain insights and make comparisons that wouldn't be possible when dealing with individual units alone.

This process of summarizing and aggregating data is yet another way of turning raw data into information that is not only useful but also actionable. It helps decision makers and knowledge workers like yourself to derive meaningful conclusions and take informed actions based on the patterns and trends revealed by the aggregated data.

So, whether it's through organizing data into categories or aggregating it into summaries, the goal is always to transform raw data into valuable information that supports decision-making and drives action.

(Source from Database Systems, Thomas Conolly,2020)

TABLE 1-1 Example Metadata for Class Roster

Data Item	Metadata					
Name	**Type**	**Length**	**Min**	**Max**	**Description**	**Source**
Course	Alphanumeric	30			Course ID and name	Academic Unit
Section	Integer	1	1	9	Section number	Registrar
Semester	Alphanumeric	10			Semester and year	Registrar
Name	Alphanumeric	30			Student name	Student IS
ID	Integer	9			Student ID (SSN)	Student IS
Major	Alphanumeric	4			Student major	Student IS
GPA	Decimal	3	0.0	4.0	Student grade point average	Academic Unit

(Source from Database Systems, Thomas Conolly,2020)

Metadata is not data per se. Instead it is a description of how data is to be stored and organized into a database. Example of meta data is shown in Table 1.1.

1.1.1 Limitations of File Based

Prior to the advent of databases, data was stored in individual files, each being used by a separate program. This was the traditional file processing approach to data storage. As business applications became more complex, it became evident that traditional file processing systems had a number of shortcomings and limitations as listed below:

a) Program-Data Dependence
   - All programs maintain metadata for each file they use

b) Duplication of Data
   - Different systems/programs have separate copies of the same data

c) Limited Data Sharing
   - No centralized control of data

d) Lengthy Development Times
   - Programmers must design their own file formats

e) Excessive Program Maintenance
   - 80% of information systems budget

However, there is still a lot of data that is stored in traditional file systems. Legacy systems still abound with traditional files. Even Excel spreadsheets, which are relatively modern, would be considered to fall within the same category as file systems. Many companies store their important data in myriad spreadsheets, and as their businesses become more complex they run up against the limitations of these storage methods.

The solution for the above limitations of File Based Processing is the Database Management System (DBMS).

1.1.2 Database Management System

The database management system (DBMS) is a software system that is used to create, maintain, and provide controlled access to user databases.

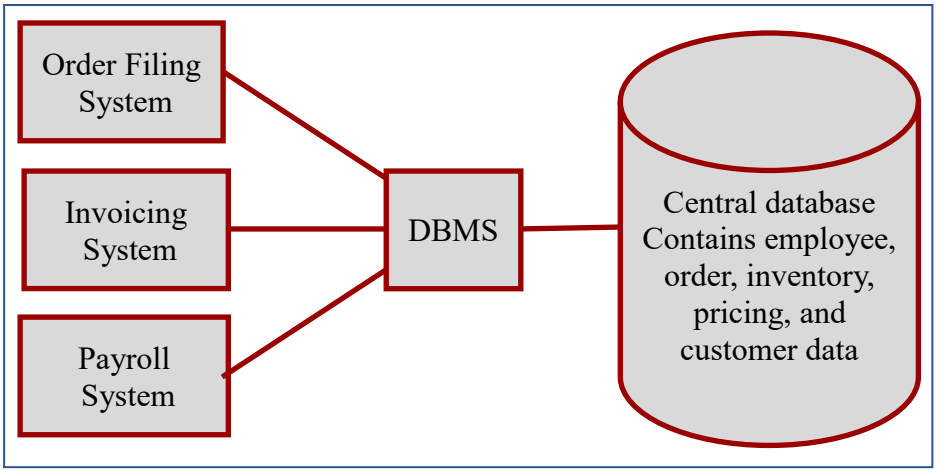


Figure 1.3 – Database management Systems
(Source: Fundamental of Database, Kroenki(2020)

The majority of modern database management systems (DBMSs) are designed as relational databases. These databases organize data into tables, where data relationships are established through common values in related tables. Throughout this course, we will delve into the intricacies of relational databases.

It's worth noting that the centralization of data within a database eliminates the necessity for separate systems and programs to maintain their individual copies of the data. This eradicates duplication and enhances data integrity.

By centralizing the data in a single database, organizations can ensure that all users and applications access the most up-to-date and consistent information. This eliminates discrepancies that can arise when multiple copies of data are stored and managed independently. Consequently, data duplication is reduced, promoting accuracy and reliability.

The adoption of a centralized database system brings numerous benefits, including streamlined data management, improved data integrity, and simplified maintenance processes. It provides a solid foundation for effective data sharing and collaboration across different applications and users.

As we progress in this course, we will explore the advantages and intricacies of relational databases in greater detail. Should you have any questions or require further clarification, please feel free to ask!

Table 1.2 - Example of Database Application Systems

Application	Example Users	Number of Users	Typical Size	Remarks
Sales Contact Manager	Salesperson	1	2,000 rows	Products such as GioldMine and Act! Are database centric
Patient appointment (doctor, dentist)	Medical office	15 to 50	100,000 rows	Vertical market software vendors incorporate databases into their software products
Customer relationship management (CRM)	Sales, marketing, or customer service departments	500	10 million rows	Major vendors such as Microsoft and Oracle PeopleSoft Enterprise build applications around the database
Enterprise resource planning (ERP)	An entire organization	500	10 million+ rows	SAP uses a database as a central repository for ERP data.
E-commerce site	Internet users	Possibly millions	1 billion+ rows	Drugstore.com has a database that grows at the rate of 20 million rows per day!
Digital dashboard	Senior managers	500	100,000 rows	Extractions, summaries, and consolidations of operational databases.
Data mining	Business analysts	25	100,000 to millions+	Data are extracted, reformatted, cleaned, and filtered for use by statistical mining tools.

1.1.3 ADVANTAGES OF DATABASE APPROACH

There are many advantages of databases comparing to traditional files. Below are among the advantages of database approach:

- Program-data independence
- Planned data redundancy
- Improved data consistency
- Improved data sharing
- Increased application development productivity
- Enforcement of standards
- Improved data quality
- Improved data accessibility and responsiveness
- Reduced program maintenance

1.1.4 LIMITATIONS OF DATABASE APPROACH

Even database approach has many advantages over file-based approach, it has limitation too. It is not an easy thing to convert from traditional file processing systems to databases. Although the end result is usually beneficial to the organization, the costs of converting are significant, especially for a large company. Other than that it also incur other cost such as cost in:

- Recruiting new specialized personnel
- Installation and management cost and complexity
- Conversion costs

- backup and recovery

SELF CHECK 1.1

1. Describe the disadvantages in File based approach
2. Explain the concept of DBMS

Note: Please complete the Self Check 1.1 in the LMS

1.2 TYPES OF DATABASE DESIGN

There are three (3) types of database design:

- From existing data
  - Analyze spreadsheets and other data tables
  - Extract data from other databases
  - Design using normalization principles

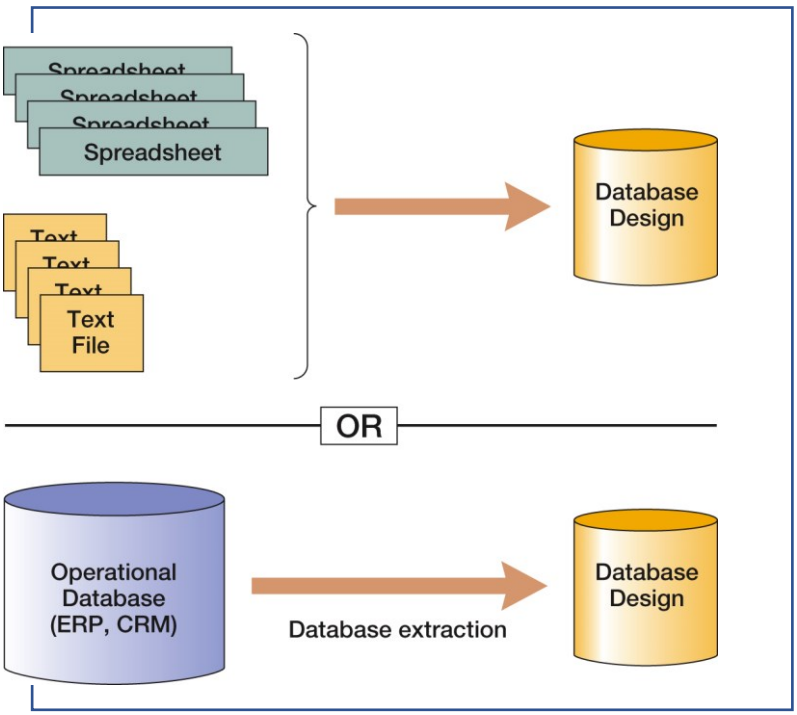


Figure 1.4 – Database design from existing data

- New systems development
  - Crete data model from application requirements
  - Transform data model into database design
- Database redesign
  - Migrate databases to newer databases
  - Integrate two or more databases

– Reverse-engineer and design new database using normalization principles and data model transformation

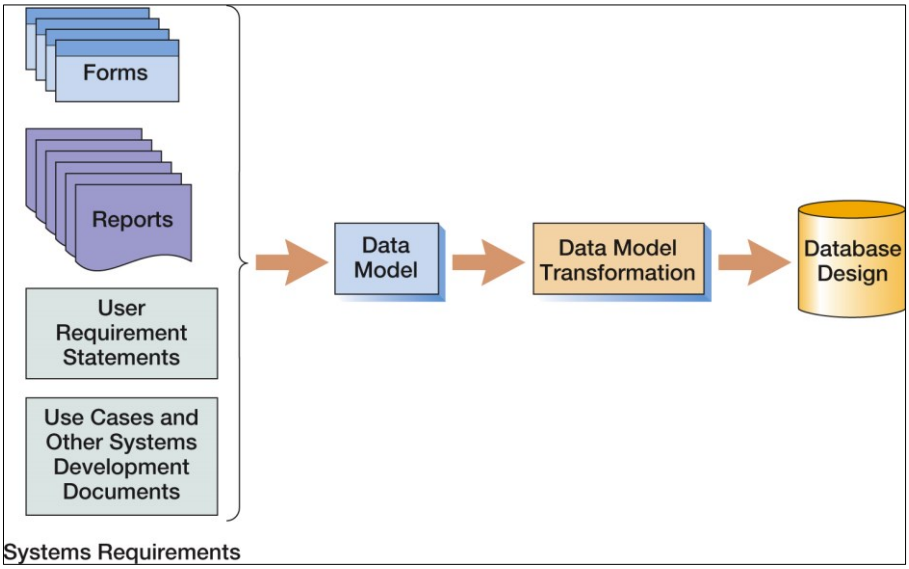


Figure 1.5 New Database Design
(Source: Fundamentals of Database, Kroenke,2020)

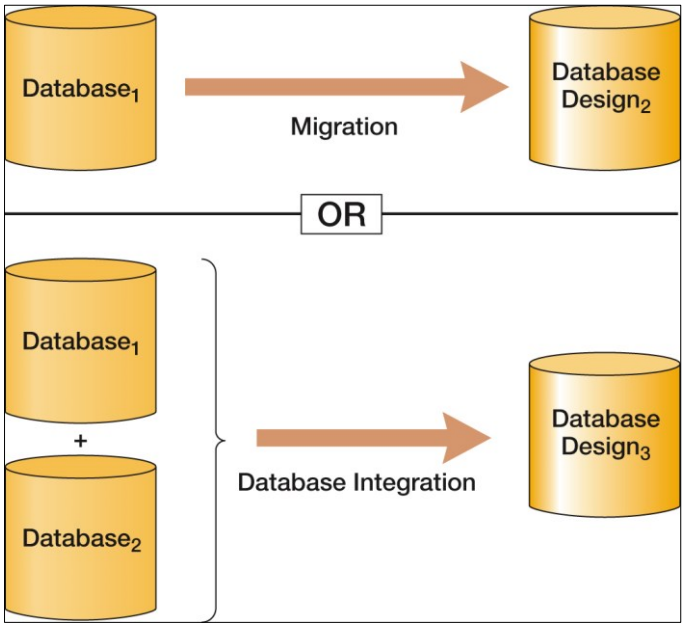


Figure 1.6 - Database Redesign
(Source: Fundamentals of Database, Kroenke,2020)

SELF CHECK 1.2

1. Describe the phases involved in designing database

Note: Please complete the Self Check 1.2 in the LMS

1.3 EVOLUTION OF DATABASE SYSTEMS

1.3.1 Database evolutions are driven by the following four main objectives:

- Need for program-data independence ➔ reduced maintenance
- Desire to manage more complex data types and structures
- Ease of data access for less technical personnel
- Need for more powerful decision support platforms

1.3.2 Database Architecture

The hierarchical and network database models was the first attempt to structure data according to relationships between entities. But they fell short because they were very inflexible. For example, many-to-many relationships are impossible in hierarchical databases, and although they are possible in network models they are difficult to modify in these structures.

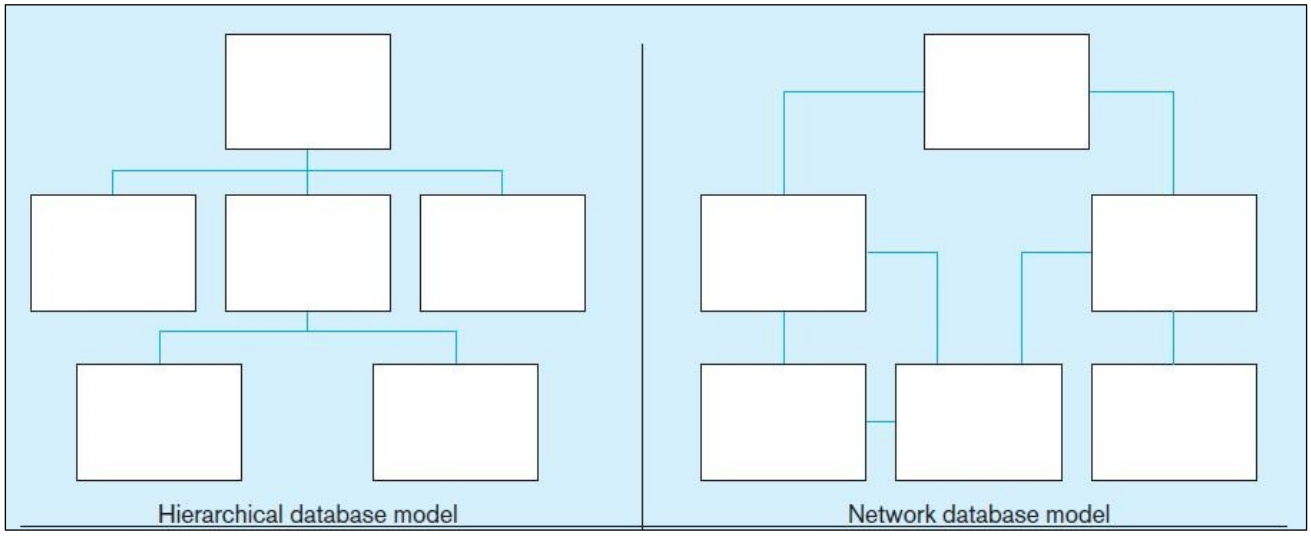


Figure 1.7 – Hierarchical and Network Database Model
(Source from Database Systems, Thomas Conolly,2020)

The relational model is the most ubiquitous, and represents relationships as primary-to-foreign key associations in different “relations”. Here, see some confusing terminology. There is the concept of “relationship”, which was shown as lines between boxes and there is another concept of “relation”, which is really like a database table.

Object-oriented databases are interesting in that they allow for a sort of inheritance between classes and subclasses. Also, unlike relations (tables) in relational databases, objects in object-oriented databases are capable of behaviors (program code) in the form of “methods”. If you take a course in Java or another object-oriented language, you will become familiar with these ideas.

Other than above models, multidimensional models are typically based on data warehouses, and are used for decision support purposes. The term Online Analytical Processing (OLAP) refers to the types of systems that use multidimensional data.

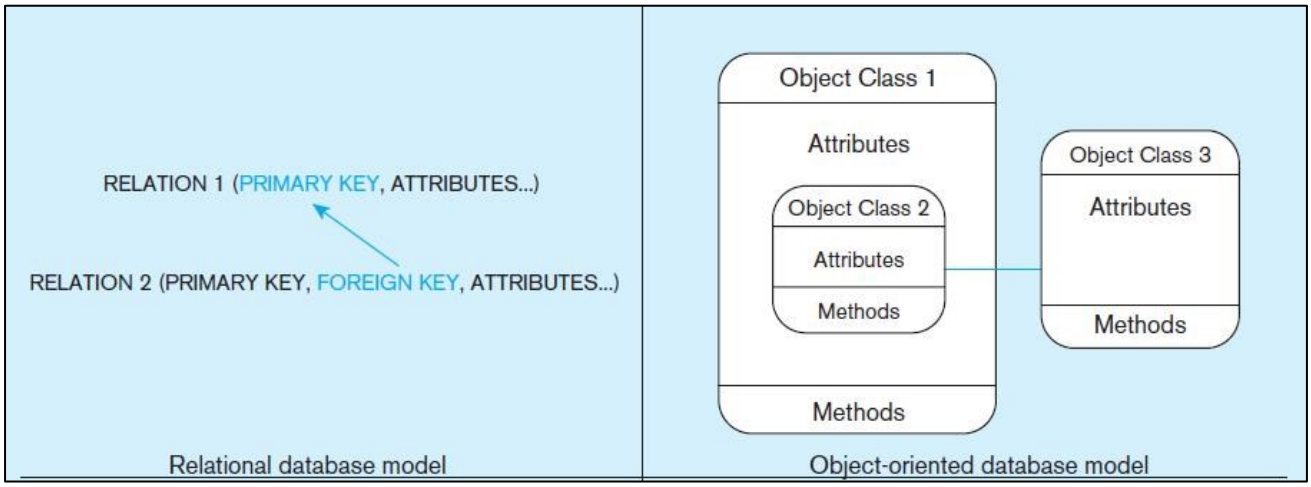


Figure 1.8 – Relational and Object-Oriented Database Model

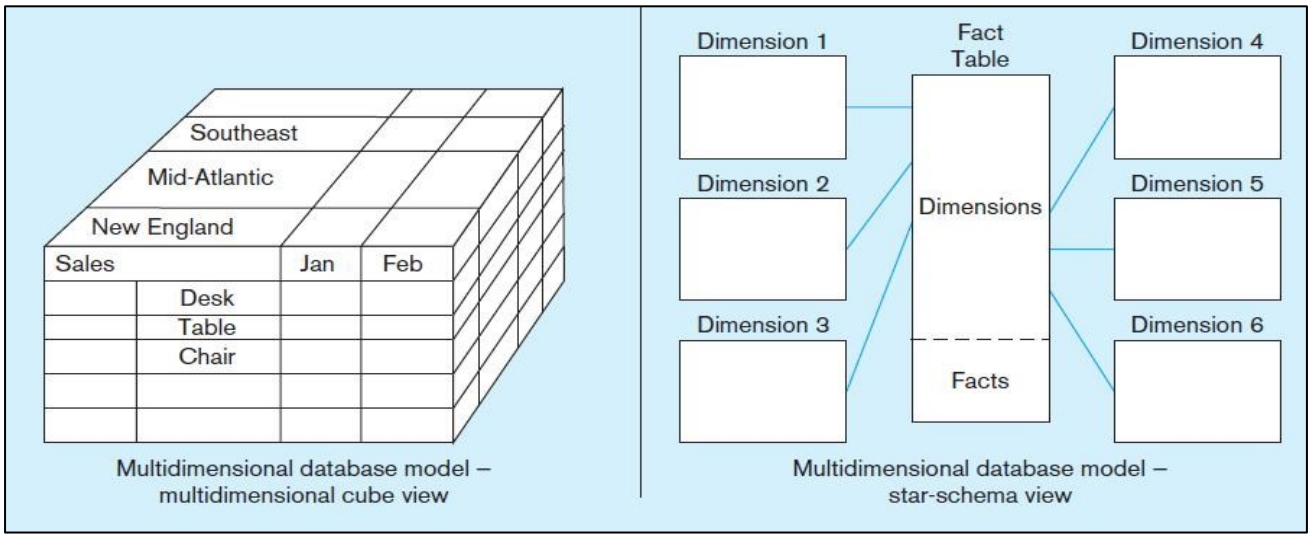


Figure 1.9 Multidimensional Database Model
(Source from Database Systems, Thomas Conolly,2020)

Multitier client/server and ERP systems are typical for the normal operational activities of most companies. Data warehouses tend to be large because they collect and maintain historical data over time. Most web applications follow the 3-tier approach. Databases are typically at an enterprise tier, application program code is at an application/Web tier, and user interfaces for different users are at the client tier.

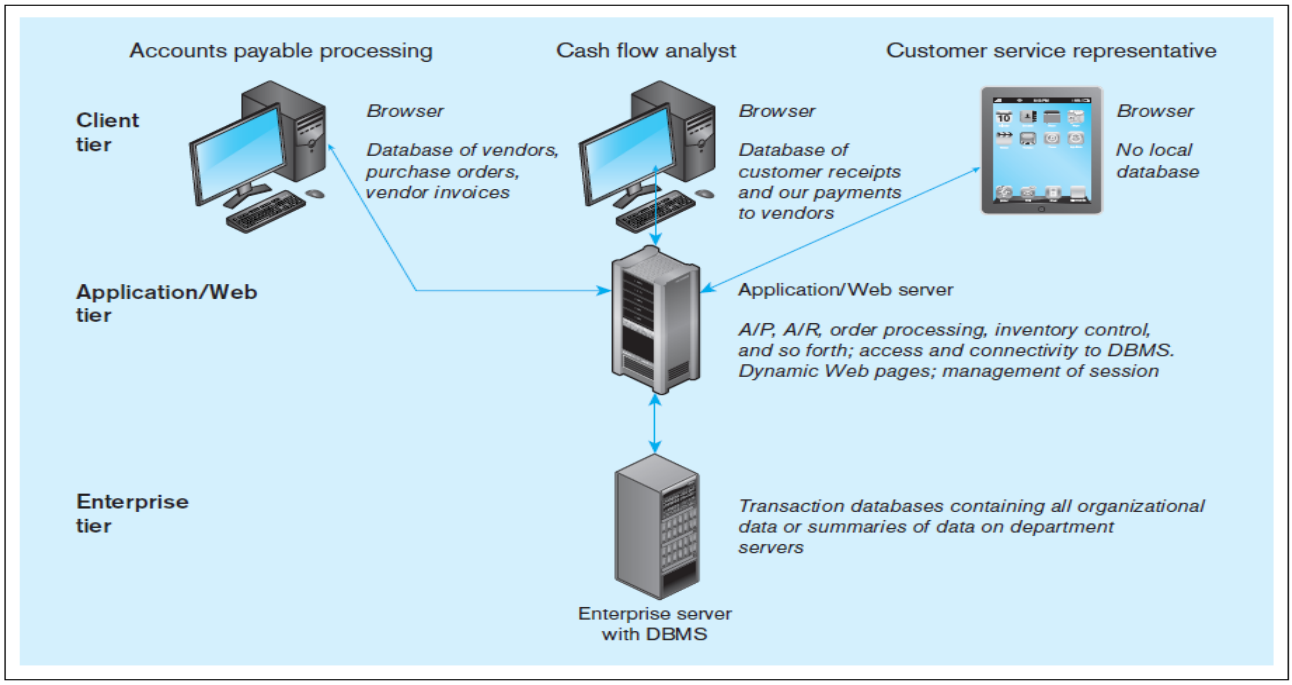


Figure 1.10 – Multi Tier Database
(Source from Database Systems, Thomas Conolly,2020)

Table 1.3 and 1.4 summarizes the history of database used before 1970's to current trend of database.

Era	Years	Important Products	Remarks
Web Databases	1995-Present	IIS, Apache, PHP, ASP.NET, and Java	Stateless characteristic of HTTP was a problem at first. Early applications were simple one-stage transactions. Later, more complex logic developed.
Open source DBMS products	1995-Present	MySQL, PostgresQL, and other products	Open source DBMS products provide much of the functionality and features of commercial DBMS products at reduced cost.
XML, and Web services	1998-Present	XML, SOAP, WSDL, UDDI, and other standards	XML provides tremendous benefits to Web-based database applications. Very important today. May replace relational databases during your career. See Chapter 11 and Appendix
Big Data and the NoSQL movement	2009-present	Hadoop, Cassandra, Hbase, CouchDB, Arango DB, Mongo DB, JSON and other products	Web applications such as Facebook and Twitter use Big Data technologies. The NoSQL movement is geared toward processing large data sets using NoSQL data models which replace relational databases with nonrelational data structures such as XML and JSON, and which may supplant relational databases during your career. See Chapter 12 and Appendices K and L.

Era	Years	Important Products	Remarks
Predatabase	Before 1970	File Managers	All data were stored in separate files. Data integration was very difficult. File storage space was expensive and limited.
Early Database	1970-1980	ADABAS, System2000, Total, IDMS, IMS	First products to provide related tables. CODASYL DBTG and hierarchical data models (DL/I) were prevalent.
Emergence of relational model	1978-1985	DB2, Oracle Database, Ingres	Early relational DBMS products had substantial inertia to overcome. In time, the advantages weighed out.
Microcomputer DBMS products	1982-1992+	dBase- R:base, Paradox, Microsoft Access	Amazing! A database on a micro. All micro DBMS products were eliminated by Microsoft Access in the early 1990s.
Object-oriented DBMS	1985-2000	Oracle ODBMS, Gemstone, O2, Versant	Never caught on. Required relational database to be converted. Too much work for perceived benefit.

(Source from Database Systems, Thomas Conolly,2020)

SELF CHECK 1.3

1. Describe the FIVE (5) types of DBMS

1.4 DATABASE SYSTEMS USERS

Users may be divided into Those who actually use and control the database content, and those who design, develop and maintain database applications (called "Actors on the Scene"), and
Those who design and develop the D B M S software and related tools, and the computer systems operators (called "Workers Behind the Scene").

1.4.1 Database Actors on the Scene

- **Database administrators:**
  Responsible for authorizing access to the database, for coordinating and monitoring its use, acquiring software and hardware resources, controlling its use and monitoring efficiency of operations.

- **Database Designers:**
  Responsible to define the content, the structure, the constraints, and functions or transactions against the database. They must communicate with the end-users and understand their needs.

- **End-users:** They use the data for queries, reports and some of them update the database content. End-users can be categorized into:
  - **Casual**: access database occasionally when needed
  - **Naïve** or Parametric: they make up a large section of the end-user population.
    - They use previously well-defined functions in the form of "canned transactions" against the database.
    - Users of Mobile Apps mostly fall in this category
    - Bank-tellers or reservation clerks are parametric users who do this activity for an entire shift of operations
    - Social Media Users post and read information from websites
- **Sophisticated:**
  - These include business analysts, scientists, engineers, others thoroughly familiar with the system capabilities.
  - Many use tools in the form of software packages that work closely with the stored database.
- **Stand-alone:**
  - Mostly maintain personal databases using ready-to-use packaged applications.
  - An example is the user of a tax program that creates its own internal database.

- Another example is a user that maintains a database of personal photos and videos.

- **System Analysts and Application Developers**
  - This category currently accounts for a very large proportion of the I T work force.
  - **System Analysts**: They understand the user requirements of naïve and sophisticated users and design applications including canned transactions to meet those requirements.
  - **Application Programmers:** Implement the specifications developed by analysts and test and debug them before deployment.
  - **Business Analysts**: There is an increasing need for such people who can analyze vast amounts of business data and real-time data ("Big Data") for better decision making related to planning, advertising, marketing etc.

1.4.2 Actors Behind the Scene

- **System Designers and Implementors:**
  Design and implement D B M S packages in the form of modules and interfaces and test and debug them. The D B M S must interface with applications, language compilers, operating system components, etc.

- **Tool Developers**:
  Design and implement software systems called tools for modeling and designing databases, performance monitoring, prototyping, test data generation, user interface creation, simulation etc. that facilitate building of applications and allow using database effectively.

- **Operators and Maintenance Personnel:**
  They manage the actual running and maintenance of the database system hardware and software environment.

SUMMARY

In this topic you have learnt that:

Finally, we are at the and of this topic. In this topic you have learnt a basic introduction of Database evolution which started from file based system. Then we also discuss different types of Database Architecture. Finally, we have touched on the existence of different role in managing and maintaining database systems.

That is all about Topic 1 - A brief introduction on Database. See you all in the next topic, Topic 2 - Data Modelling & Entity Relationship Diagram. Bye!

KEY TERMS

- File Based Systems
- Database
- Database Management System
- Database Architecture
- System Development Life Cycles
- Prototyping

REFERENCES

David M. Kroenke, David J.Auer and Robert CY.2019 Database Processing: Fundamentals, Design, and Implementation, 15th ed.Pearson

Thomas Conolly, Database Systems. 7th ed. Boston: Pearson/Addison Wesley RS