CS660 – DATABASE SYSTEMS

UNIT 1 – DATABASE SYSTEMS

Dr. John Conklin

COLORADO TECHNICAL UNIVERSITY

FOUNDED 1965

Administrative Items

- Ensure you read **ALL** announcements and send the acknowledgment once you downloaded all required files (SQL code files and sample project file).
- Ensure you follow **ALL** instructions posted for discussion boards, individual projects, and anything I note in the announcements.
- Initial discussion posts are due on Thursday evening. Any responses are not due until Saturday.
- The first individual project is due **Sunday**.
- All writings should be in APA 7 format, with any sources used in the writing properly cited in the text and listed in the reference section in proper APA format.

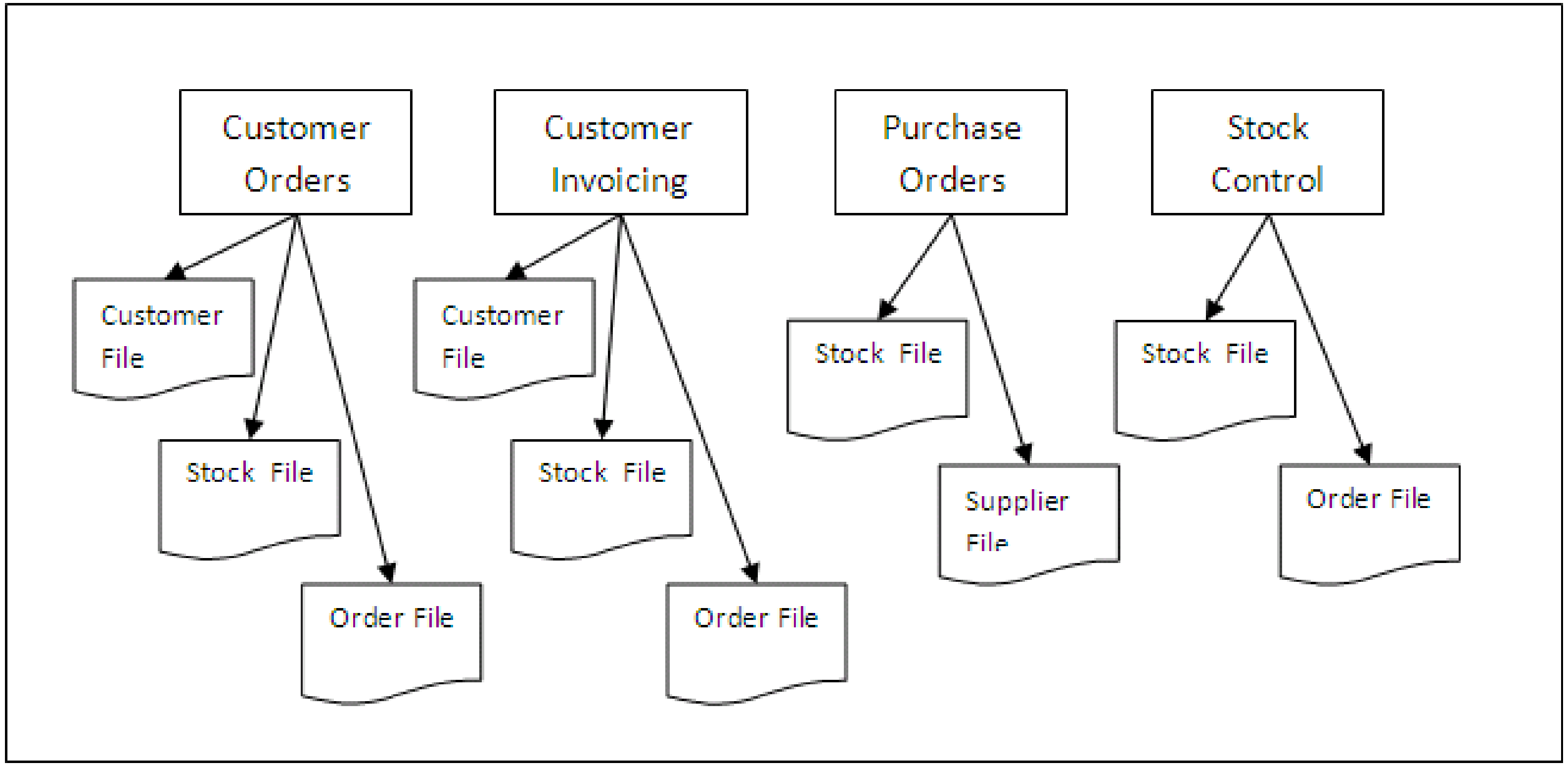
Agenda

- File-Based Approach
- DBMS Components
- Database Structure History
- Database Challenges
- Individual Project

FILE BASED APPROACH

File-based approach

- The phrase "***file-based approach***" describes a situation in which data is kept in one or more distinct computer files organized and controlled by various application applications.
- Computer applications refer to any software or, occasionally, a group of connected programs.

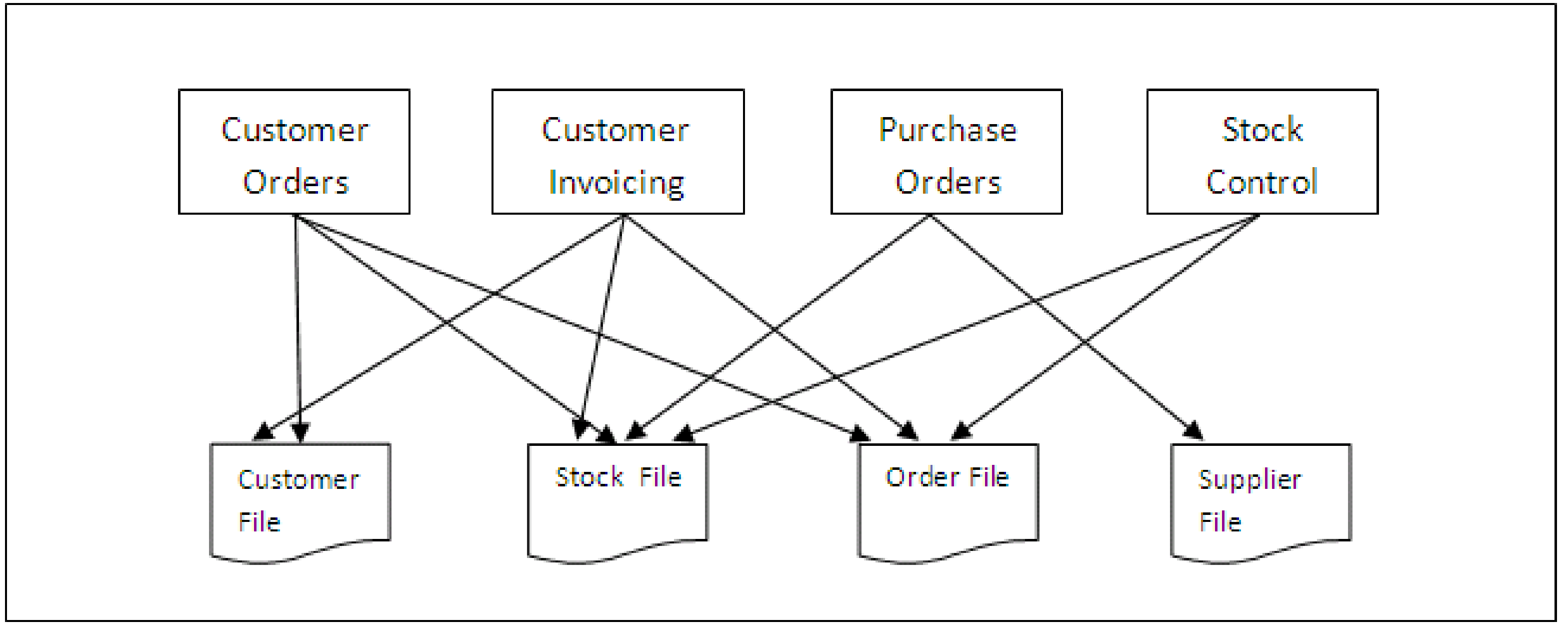


File-based approach

The graphic on the left demonstrates how several apps will each have a copy of the files they require to do the tasks they oversee.

File-based approach

Sharing files between programs is one way to get around the issue of each application having its own collection of files. The graphic below demonstrates how this can help with the issue of data duplication and inconsistency between several applications.



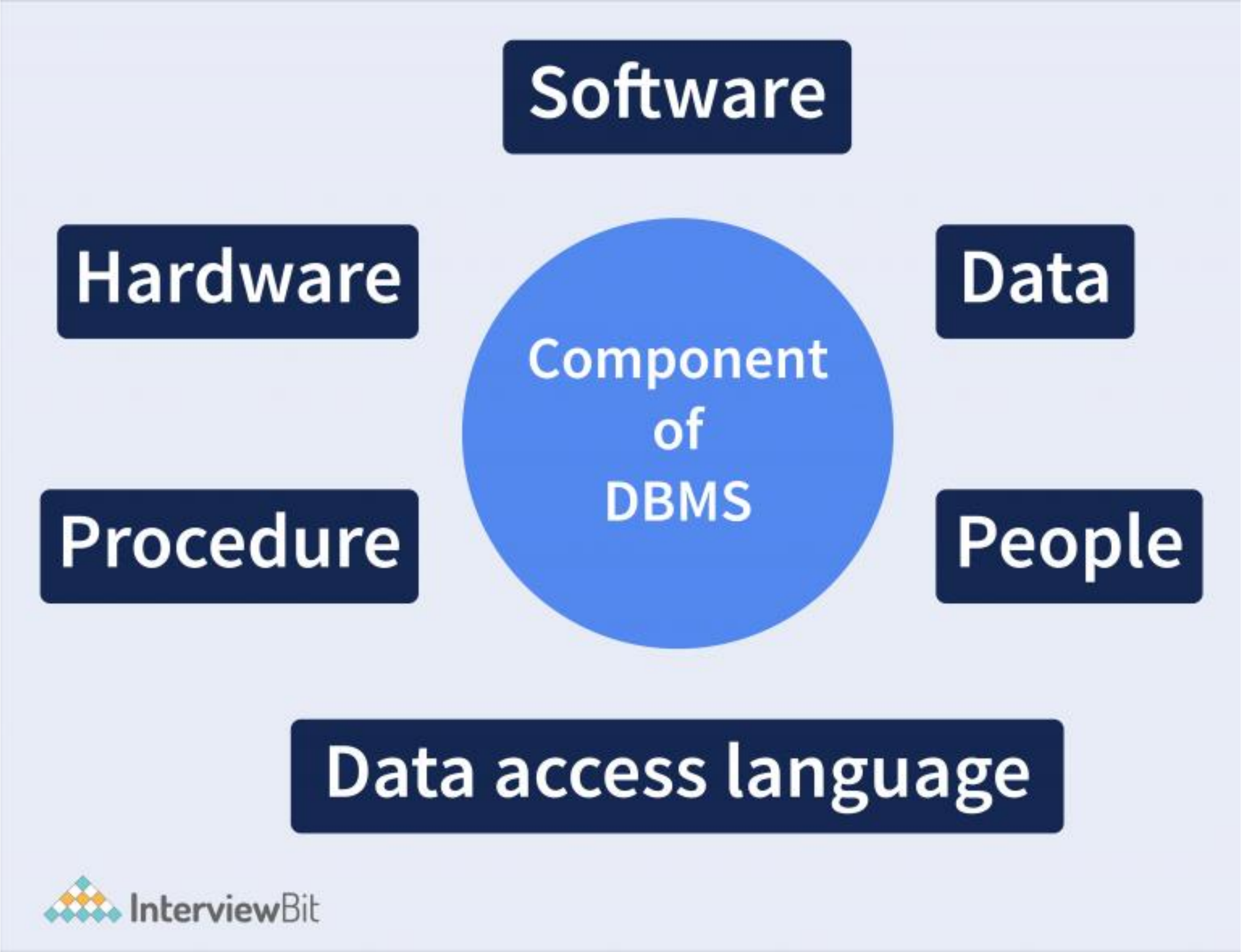
File-based approach

- The introduction of shared files solves the problem of duplication and inconsistent data across different versions of the same file held by different departments, but other problems may emerge, including:
  - **File incompatibility.**
  - **Difficult to control access.**
  - **Physical data dependence.**
  - **Difficult to implement concurrency.**

DBMS COMPONENTS

DBMS Components

- The database system is made up of several parts.
  - Essential functions.
  - Interconnected elements.
  - Governs usage.



https://www.interviewbit.com/blog/wp-content/uploads/2022/02/Components-768x590.png

DBMS Components

The database's environment's components are listed below. The DBMS may be divided into five primary categories, each of which is described in the image on the left:

DBMS Components - Hardware

- Input-output devices.
- Any information system's hardware, which includes the devices like computers, scanners, and printers required to acquire, convert, and output data to users, is its most obvious component.
- Processing is usually done on the user's computer.
- Workstations.

DBMS Components - Software

- The software makes up **most** of a database management system.
- Software is a grouping of programs or instructions that give instructions to a computer.
- Apps needed to access the data.
- Can interpret the Database Access Language.
- As the software that governs everything, this is the essential element.
- Examples of DBMS applications.

DBMS Components - Data

- The collection of information kept in the database is referred to as data.
- Structured, non-structural, and logical data are all data kept in the database.
- **Unstructured** data is a collection of many different forms of data that are saved in their native formats, as opposed to structured data, which is very particular and is kept in a preset manner.
- To store and preserve the data within the database is the major motivation for the invention of the database management system.
- The database management system's most crucial element is it. The standard database includes both the actual (operational) data and the metadata (information about the data).

DBMS Components - Procedures

- Procedures are **broad guidelines for utilizing a database management system**.
- Access control, data validation, and reducing network traffic between clients and DBMS servers are all possible using procedures.
- Database Functions and Database Procedures are related.

DBMS Components - People

- Individuals.
- Database administrators, software developers, and end users are among the participants.
- Active in creating and developing the components of DBMSs.
- All contemporary online and mobile applications now store user data.

DBMS Components – Database Access Language

- Users can create commands in the straightforward Database Access Language to perform the needed actions on the database's stored data.
- Data stored in a database may be accessed, updated, and deleted using instructions written in a database access language.
- Before submitting commands to the database for execution, users can write them using the Database Access Language.
- Database languages include My Access, Oracle, and SQL (structured query language).

DBMS Components – Database Access Language

- The following commands serve as the base for all DDL commands:
  - ALTER<object>
  - COMMENT
  - CREATE<object>
  - DESCRIBE<object>
  - DROP<object>
  - SHOW<object>
  - USE<object>

DBMS Components – Database Access Language

- The following commands serve as the base for all DML commands:
  - INSERT
  - UPDATE
  - DELETE
  - LOCK
  - CALL
  - EXPLAIN PLAN

DBMS Components

- End users may add, read, update, and remove data in a database thanks to a DBMS.
- You may utilize a database virtually daily, regardless of whether you are unaware.

DATABASE STRUCTURE HISTORY

Database Structure History

- A Relational Model of Data for Large Shared Banks was the title of an academic article written by IBM computer scientist Edgar F. Codd in June 1970.
  - It described a novel approach to modeling data.
  - It described how to create a system of interconnected tables that allow you to store each piece of data just once.
  - A database with this structure could solve any question, provided the solution was contained somewhere in the database.
  - At a period when storage was expensive, disk space would be utilized effectively. With this study, databases entered a new era.

Database Structure History

- Relational databases became increasingly popular in the 1980s and 1990s, providing thorough indexes that made any query effective.
- Table joins—read operations that combine distinct records into a single one—and
- Transactions—a name for a collection of reads and particularly writes dispersed across the database—was crucial.
- Software developers learned to ask for what they needed and let the database determine how to supply it using SQL, or Structured Query Language, which has since emerged as the language of data. To prevent surprises, expected assurances were included in the database.

Database Structure History

The emergence of NoSQL databases

- Relational databases were designed with the idea that they would only use one machine to execute them.
- Single server.
- The only option is to go from a single database server to a cluster of database nodes cooperating when the workload becomes so intense that no single computer can handle it.

Database Structure History

- This task is challenging for typical SQL databases architected to run on a single server. It demands significant work commitments and frequent choices that forfeit many capabilities that first drew developers to these databases.
- By the late 2000s, SQL databases were still quite common.
- A few examples are Cassandra, HDFS, and Google BigTable.
- However, they have reduced functionality, usually due to fewer joins, fewer transactions, or fewer indexes.

Database Structure History

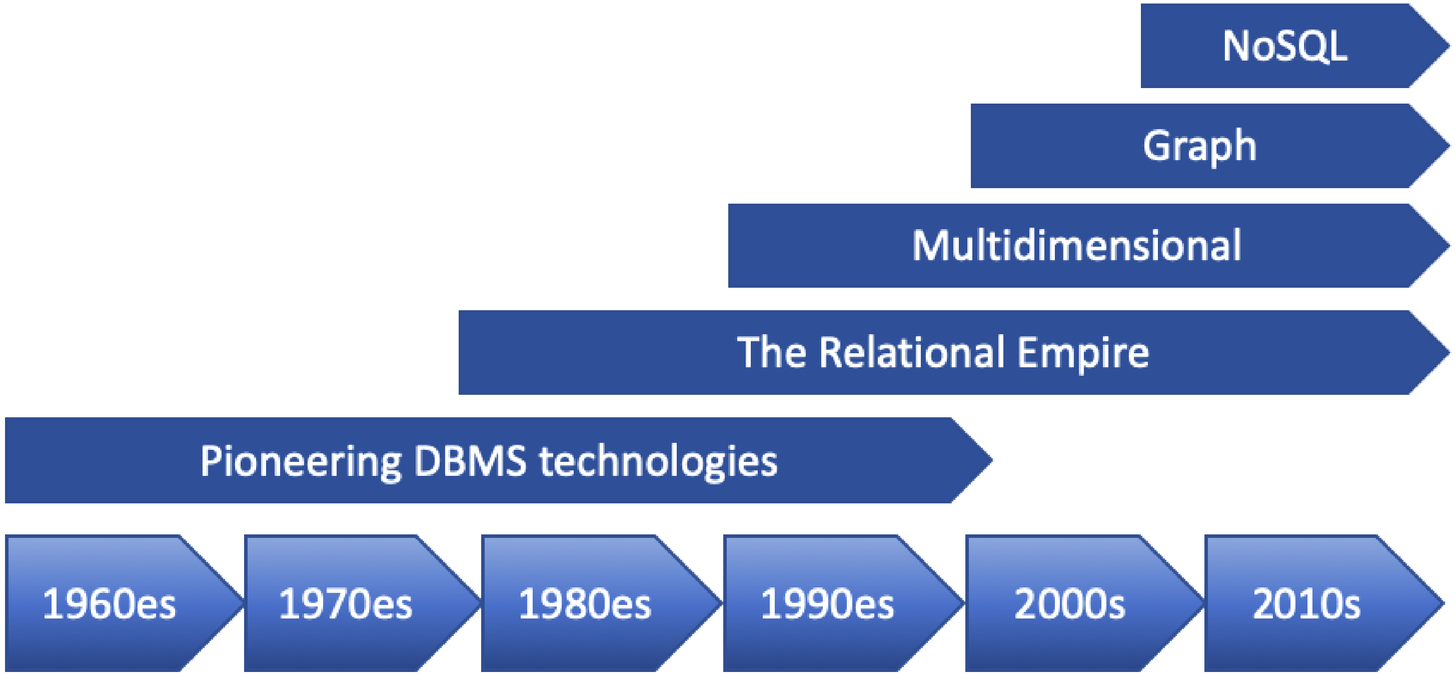
The next step in database evolution is **distributed SQL**.

- By adding capabilities to ease the pain of sharding, traditional SQL databases have attempted to overcome their scaling problem (and maintain their market dominance).
- Simultaneously, NoSQL systems have been developing a portion of the capability that SQL systems lack.

Database Structure History

- ***Google Research*** introduced Google Cloud Spanner, a database designed to distribute data globally and allow consistent transactions, in what has come to be known as the Spanner paper, released in **2012**.
- **Distributed SQL** is the name given to this novel type of database.
- A database may only be classified as distributed SQL if it satisfies the following five criteria:
  - scalability,
  - consistency,
  - resilience,
  - SQL, and
  - geo-replication.

Database Structure History



NoSQL

Graph

Multidimensional

The Relational Empire

Pioneering DBMS technologies

1960es

1970es

1980es

1990es

2000s

2010s

Five "waves" made up the evolution of the database:

- The first wave, which spanned from 1960 to 1999, was made up of network, hierarchical, inverted list, and (in the 1990s) object-oriented DBMSs.
- Around 1990, the relational wave debuted all of the SQL products (as well as a few non-SQL items), and around 2008, it started to lose consumers.
- Around 1990, the decision support wave, which is still going strong today, brought Online Analytical Processing (OLAP) and specialized DBMSs.
- The Semantic Web stack from the Worldwide Web Consortium launched the graph wave in 1999, and property graphs started to surface around 2008.
- Big data is just one aspect of the NoSQL revolution, which started in 2008.

DATABASE CHALLENGES

Database Challenges

9 Typical Database Management Issues:

- Managing scalability as data volume increases
- Maintaining database performance
- Multiple data storage
- Data safety
- Limitations on mitigation
- Data management and distribution
- Misconfigured or incomplete security
- Data Integration
- Speed

We will discuss each of these issues on the next few slide.

Database Challenges

Managing scalability as data volume increases

- 63% every month.
- Although specific tools and applications supply databases with larger datasets than others, it's also possible that data may be constantly updated and accessed.
- Think about how you'll use the data being held.
- Available storage.

Database Challenges

Maintaining database performance

- Your staff will find slow database performance annoying, but it will also cause apps to stop and negatively affect end users. The optimal experience for both your staff and clients is crucial.
- One of the finest methods for scaling databases that don't require frequent updates is caching to a distant site.
- Focus on enhancing query performance.
- Use workers who are more knowledgeable about databases and have more relevant expertise.

Database Challenges

Multiple Data Storage

- One of the biggest problems most firms face is having a large amount of data stored.
- Multiple data storage systems present a substantial barrier.
- As a result, ensure sure your data has a single source of truth.

Database Challenges

Data safety

- In addition to that, you also risk your company going down and losing its reputation.
- Make sure to back up your data constantly.

Database Challenges

Limitations on mitigation

- It consists of data servers. Innovative businesses that prioritize transaction volume are familiar with their computer systems, hardware setup, data structure, and catalog components.
- They are aware that any of them might cause data loss,

Database Challenges

Data management and distribution

- Data management has benefits and disadvantages.
- In addition, businesses need to understand how much power should be allocated to localities.

Database Challenges

Misconfigured or incomplete security

- There is no question in my mind. Particularly in cloud contexts, databases are significantly at risk from improperly configured security.
- Your data will be vulnerable to external attackers if your cloud security is insufficient or nonexistent. However, it's rather simple to neglect the proper settings or security fixes while handling many databases.

Database Challenges

Data Integration

- Managing databases used to be rather easy. New complications nevertheless appear when you grow databases further. You are now at a loss as to how to change your DBM.
- If you offer omnichannel services, you must combine data from many sources. You can accomplish this using software that was made with this goal in mind.

Database Challenges

Speed

- Everyone despises using sluggish computers. It's past time to optimize your systems if trying to access data consistently causes you to become anxious. Make sure your SQL queries don't have too many joins and that you index properly.
- If this isn't the problem, it's also past time to increase your bandwidth; else, a virus could have infected you. This is why you need to create frequent database health checks.

INDIVIDUAL PROJECT

Individual Project

Description

- The retail store has experienced significant growth in the recent months. The customers have expressed an interest in being able to purchase the retail store’s products electronically. Currently, the retail store has a simple Web site, but it is not clear that the system can support dynamic interaction on the Web site or online purchases. The main database system that is used in the physical store has not been working correctly, however, and this is an issue. The retail store wants to be able to eventually integrate an online database system with the current order entry system that is used to support sales made in the local store. The retail store also needs the database system to enforce the rules that have been established for their business practices. What can be done to support the goals and objectives for meeting the needs of their customers?

Individual Project

- It is assumed that Microsoft SQL Server Express Edition (with management tools) was chosen as the database platform. It can be downloaded for free at this Web site.
- You have been asked to develop a solution in the form of a Database System Development Plan for your retail store using the following format:
- Database System Development and Implementation Plan
  - Use Microsoft Word
  - Title page
    - Course number and name
    - Project name
    - Your name
    - Date
  - Table of contents (TOC)
    - Use auto-generated TOC
    - Maximum of 3 levels deep
    - Be sure to update the fields of the TOC so it is up-to-date before submitting your project.
  - Section headings (create each heading on a new page with TBD as content except for sections listed under new content below)

Individual Project

The project deliverables are as follows:

- A description of the general business environment for the case study organization (give me the name of the store, its location, how many employees work there, and the market you serve.)
- A database system goals and objectives statement
- A narrative that discusses how the proposed database system addresses the presenting business problem(s)
- A narrative that demonstrates how the proposed system aligns with the mission statement and strategic goals of the organization
- The student's analysis as to how this part of the project fulfills the mission and/or one or more goals of the case study organization
- All sources should be cited in-text, and references should be made using **APA 7 format**.
- Name the document "**yourname_CS660_IP1**"
- Submit the document for grading.

Individual Project

Reference:

- Microsoft (2023). Industry-leading Performance and Security with SQL Server 2019. www.microsoft.com/en-us/sql-server/sql-server-2019

Contact Information

Email:	Jconklin@coloradotech.edu
Phone:	602.796.5972
Website:	http://drjconklin.com
Office Hours:	Wednesdays: 6:00 PM – 7:00 PM (CST)
	Saturdays: 11:00 AM – 12:00 PM (CST)
Live Chats:	Wednesdays: 7:00 PM – 8:00 PM (CST)

References

Frisendal, T. (2023). *History of Data Modeling. http://graphdatamodeling.com/GraphDataModeling/History.html*

interviewbit.com. (2023). Components of DBMS (Database Management System). https://www.interviewbit.com/blog/components-of-dbms/

Kelly, D. (2022). A brief history of databases: From relational, to NoSQL, to distributed SQL. https://www.cockroachlabs.com/blog/history-of-databases-distributed-sql/

Modise, M. (n.d.). Chapter 1. Introduction to the Module. In. https://www.cs.uct.ac.za/mit_notes/database/htmls/chp01.html#

tudip.com. (2022). *9 Common Database Management Challenges and How to Fix them. https://tudip.com/blog-post/9-common-database-management-challenges-and-how-to-fix-them/*