

Module_7

Data Warehouse

CS 3410 Introduction to Database



Data Warehouse

- Most organizations have a plethora of data but cannot access it. This is a reason why there is a knowledge gap between so many organizations.
- The first reason is because organizations have a fragmented way in which they created their information systems and databases. Because these organizations have time constraints and resource constraints, they have been developing information systems one at a time. This creates inconsistent and uncoordinated databases.

Data warehousing

- Databases are usually based on different purchased applications, software platforms, and hardware that have concluded in from a variety of organizational acquisitions, mergers, and reorganizations.
- From these circumstances, it can be very difficult, most likely impossible, for managers to find and use accurate information, that has to be synthesized across different systems.

Data warehousing

- The second reason that there is an information gap is because most systems are created to help operational processing, in which there is little to no thought given about information and analytical tools need to make decisions.
- Operation processing, also known as transactional processing, takes, stores, and changes data to help daily operations of an organization. It usually focuses database design to optimize access on a small set of data that relates to a transaction.

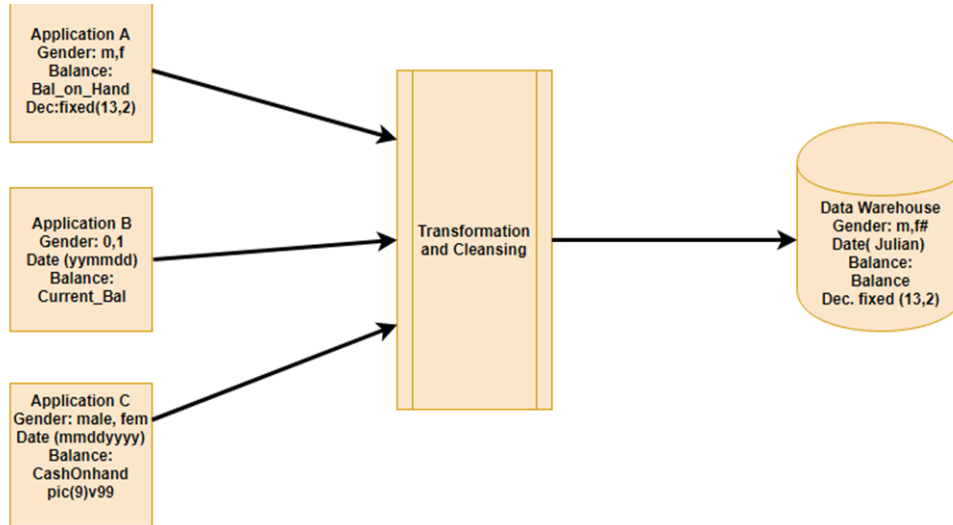
Data warehousing

- Information processing is an analysis of data or other forms of data to help decision making.
- Large swatches of data are needed to derive information, for example: sales of total products, of several years, from all sales regions.
- Systems that are mostly developed internally or bought from outside vendors were designed to help operational processing, of which very little thought is given to information processing.

Data warehousing

- The data warehouse is a collection of data that is nonupdateable, time-variant, integrated, and subject-oriented that is used in the support of decision-making management processes and business intelligence.
- The meaning of each of these terms is important for defining what a data warehouse is.

Data warehousing



Data warehousing

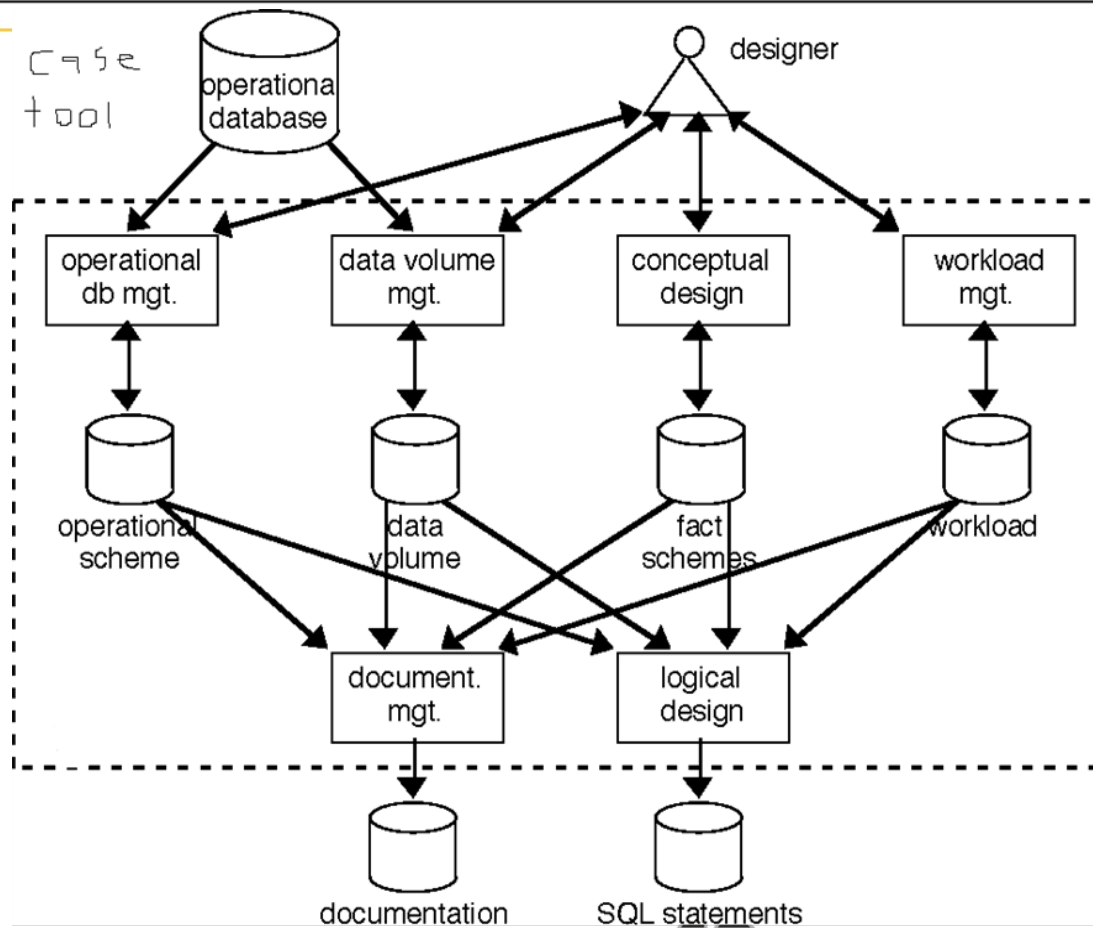
- Now I am going to talk about the architecture of a data warehouse.
- A simple data warehouse design is where all the different forms of data are stored in the central repository of the data warehouse. The repository gets information put into it by the sources of data then is used for data analysis and mining.

Data warehousing

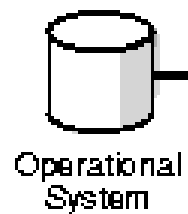
- Next we have simple architecture design with a staging area included in it. All of the non-required data must be prepared before going to the warehouse because only the necessary data is wanted. The staging part of the warehouse is put in before it actually goes into the data warehouse to simplify and clean the data .

Data warehouse

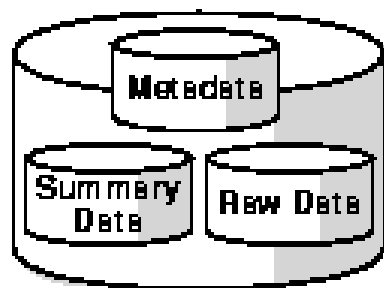
case
tool



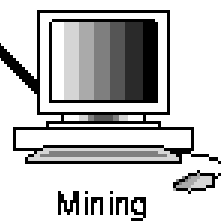
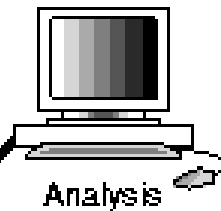
Data Sources



Warehouse



Users



Data warehousing

- One thing that I haven't mentioned yet is the tools and utilities of a data warehouse. These can be very useful when navigating the warehouse.
- Data Extraction is used to get data from several different outside sources and make it easier to view.
- Data Cleaning can also be a very useful tool considering it finds the errors inside of your data and corrects them for you.

Data warehousing

- This can also be used to get rid of duplicate data you do not need anymore.
- . Data Transformation is used for converting the format type of the data sets. If the data is in legacy format it can easily be transformed to warehouse for you to easier analyze data.
- Data Loading is used mostly for sorting and checking liability of data. The refresh tool is just for simply updating the sources into warehouse format.

Client/Server intro and background

Client/server topology has been around since the beginning of the internet content model

- One-to-connections are not feasible for web hosted applications where multiple users are connecting to the host for the same content
- This is relevant to the design and purpose of databases do the way the database must store and fetch files for the front end application
 - A database which serves a client/server application would be designed differently than one which operates with applications that utilize a single server for a single client (peer to peer)

Client/Server intro and background

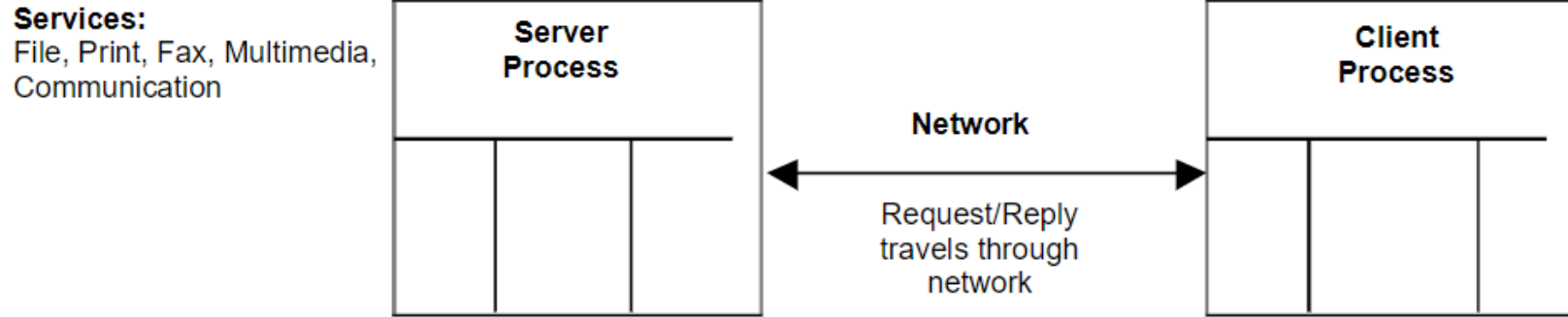


Figure 1. A graphic demonstrating the architecture of client/server systems. From “*An Introduction to Client/Server Computing*” by Yadav SC, Singh SK, 2009.

Multi Tier Architecture - Overview

- Separation and duplication of a server/database system by decentralizing data and compute resources
- Higher reliability, throughput
- No reason to not use this method of system layout
 - Considered the standard for modern computing and storage systems

Multi Tier Architecture - 7 parts

- Persistence: The database which serving application
- Accessor: The SQL server (compute)
- Logic: The applications use data on the database
- Presentation: Web browser languages such as HTML or XML.
- Requester/consumer: The web browser itself (Chrome, IE, Firefox, etc)
- Elsewhere: AWS, Azure, or GCP, or other foreign sources

Multi Tier Architecture - 7 parts

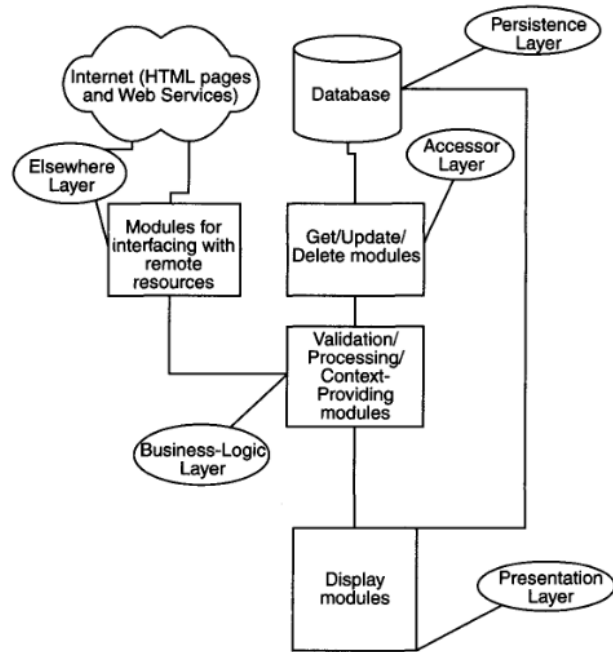


Figure 2. A graphic demonstrating the architecture of multi tier architecture. From “*Multi-Tier Application Programming with PHP : Practical Guide for Architects and Programmers*” by Wall, D, Morgan Kaufmann, 2004.

3 Main Components of Client/Server Systems

- Presentation
 - High-level interactive portion of the database. Where the end-user can input requests and receive output on their host device.
 - GUI elements which the user interacts with
 - HTML
- Processing
 - The intermediate coding between the front end and the database itself
 - The “backend”
 - C, C++, Java, Python
- Storage
 - The physical database
 - The DBMS (database management system)

Two-tier vs Three-tier architecture

- In two-tier architectures, applications run on a client and communicate with the server directly
- Data layer or structure is then stored on a server
- Database as server model

Two-tier vs Three-tier architecture

- Three-tier architectures include another layer, the application layer and client layer are broken out
- Developers are able to concurrently work on each tier of the application as they are designing it
- More modern than two-tier architecture

Section Concise Summary

- The client-server model is the standard for modern web-facing servers. The core functionality of data transfer is efficient and well defined.
 - Customization is available to fit most/all application needs
 - More flexible than two-tier architecture.
- Abstraction layers serves to make the application UI as simple to use as possible, making this structure desirable to end users and developers alike.

Reference List

- Akram, Muhammad. “Qualitative & Quantitative analysis of tiered Architecture of Web-Applications.” 2007. PDF.
- Yadav SC, Singh SK. An Introduction to Client/Server Computing. New Delhi: New Age International; 2009.
- Wall, D, Morgan Kaufmann. Multi-Tier Application Programming with PHP : Practical Guide for Architects and Programmers; 2004.