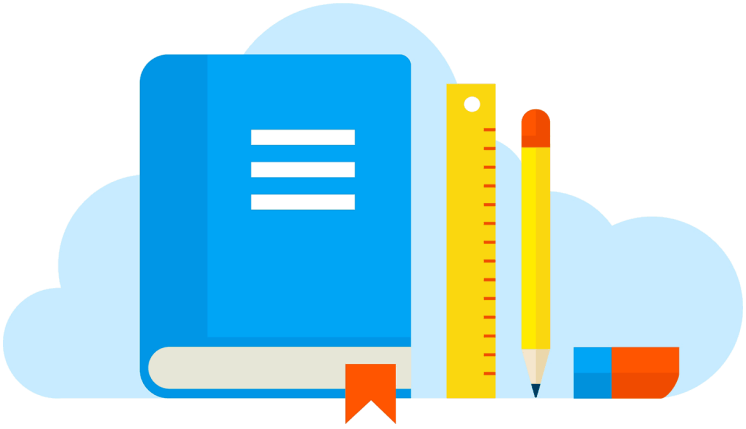

Class Notes

Date: 06/01/2025

History of Database Management Systems (DBMS)

The evolution of **Database Management Systems (DBMS)** spans several decades, driven by the increasing need to efficiently store, manage, and retrieve vast amounts of data. Here's a detailed timeline of the history of DBMS:

History of Database Management Systems (DBMS)

The evolution of **Database Management Systems (DBMS)** spans several decades, driven by the increasing need to efficiently store, manage, and retrieve vast amounts of data. Here's a detailed timeline of the history of DBMS:

2. The Birth of DBMS (1960s)

In the 1960s, the need for better data management became evident. Data storage was growing rapidly, and organizations required more efficient ways to handle it.

- **Hierarchical Model (1960s)**:

  - In 1960, **IBM's IMS (Information Management System)** became one of the first DBMSs. IMS was based on a **hierarchical data model**, where data was represented in a tree-like structure with parent-child relationships. It was used primarily by large organizations like banks and government agencies.
  - While IMS solved many issues related to managing large datasets, it was inflexible, as each relationship between data elements had to be pre-defined.

- **Network Model (Late 1960s)**:
  - Around the late 1960s, another model emerged, the **network model**, which was introduced by the **Conference on Data Systems Languages (CODASYL)**. It improved on the hierarchical model by allowing more complex relationships between data elements. It used a graph-like structure where records could have multiple parent-child relationships.
  - The **IDMS (Integrated Data Management System)** is a notable example of a network DBMS.

Objectives

- **Centralized Data Storage**: Consolidate customer, product, sales, and inventory data into one unified system.

- **Data Integrity**: Ensure that data across multiple systems (online store, physical stores, and warehouses) is consistent.

- **Real-Time Updates**: Facilitate real-time updates of inventory and sales data across all platforms.

- **Improved Customer Experience**: Personalize marketing and sales offers based on customer preferences and purchase history.

- **Scalability**: The DBMS should be able to scale as the business expands into new markets and products.

Data Model

1. **Entities**:
    - **Customers**: Customer details such as name, address, contact information, and purchase history.
    - **Products**: Product details including name, price, category, stock quantity, and supplier information.
    - **Sales**: Information about each sale, including product, quantity, sale price, date, and customer.
    - **Inventory**: Track stock levels across different warehouses and stores.
    - **Suppliers**: Information about the suppliers who provide the products.

2. **Relationships**:
    - One customer can make many sales.
    - One product can appear in many sales.
    - A product is supplied by one or more suppliers.
    - Inventory is linked to specific products, but it may differ by location.

Normalization: The DBMS was designed to ensure data normalization up to the third normal form (3NF) to reduce redundancy and improve data integrity. This means:

- The tables were designed in a way that each attribute in a table is atomic and free from partial and transitive dependencies.

Implementation Process

1. **Planning and Design**:
    - The first phase involved understanding the business requirements, designing the database schema, and deciding on the necessary hardware and software infrastructure.

2. **Database Setup**:
    - MySQL was installed and configured, and the necessary tables and relationships were created based on the business needs.
3. **Integration with Existing Systems**:
    - RetailMart's e-commerce platform, point-of-sale (POS) systems in stores, and inventory management software were integrated with the new DBMS. APIs were created for real-time data exchange between these systems.

4. **Data Migration**:
    - Data from legacy systems (such as old inventory systems and manual customer records) were migrated to the new DBMS. Data validation was performed to ensure data integrity during the migration process.

5. **Testing and Optimization**:
    - After initial deployment, thorough testing was conducted to check the performance of the DBMS under heavy traffic. Indexing and query optimization were performed to ensure fast response times for large data sets.

Results and Benefits

- **Improved Data Accuracy**: The centralization of data in the DBMS helped eliminate discrepancies between systems and provided a single source of truth.

- **Operational Efficiency**: Real-time updates across the retail chain improved inventory management, reduced out-of-stock situations, and streamlined sales processes.

- **Personalized Customer Experience**: With detailed customer profiles, RetailMart could offer personalized promotions and discounts, leading to increased customer satisfaction and retention.

- **Better Decision Making**: Management could now access detailed reports and analytics that helped them optimize operations, forecast demand, and reduce costs.

- **Scalability**: As RetailMart expanded into new regions and added more product categories, the DBMS was able to handle the increased data load without performance degradation.

- **Objectives**

- **Centralized Data Storage**: Consolidate customer, product, sales, and inventory data into one unified system.

- **Data Integrity**: Ensure that data across multiple systems (online store, physical stores, and warehouses) is consistent.

- **Real-Time Updates**: Facilitate real-time updates of inventory and sales data across all platforms.

- **Improved Customer Experience**: Personalize marketing and sales offers based on customer preferences and purchase history.

- **Scalability**: The DBMS should be able to scale as the business expands into new markets and products.