PLAN OF WORK

TABLE OF CONTENTS

[COLLECTING DATA 2](#_Toc170766140)

[ESTABLISHING RELATIONS 3](#_Toc170766141)

[CREATING A DATABASE 5](#_Toc170766142)

[HOW TO BUILD A DBMS 7](#_Toc170766143)

[WEBOGRAPHY 8](#_Toc170766144)

COLLECTING DATA

* **HOW, WHERE, WHEN?**

There are several ways to collect the data:

1. Obtain spatial data, cartographic layers, and map series from government agencies, commercial data providers, and online repositories. This can be done through data downloads, API integrations, or direct data sharing agreements.

Digitize and catalogue historical map collections and other legacy data sources.

Conduct field surveys and use remote sensing techniques.

1. Establish a customer relationship management (CRM) system to record client information, order details, and delivery history.

Gather feedback from clients to understand their needs and improve the cartographic products and services offered.

Analyse order data to identify trends, optimize inventory, and plan for future demand.

ESTABLISHING RELATIONS

Here's an example of the different entities, their attributes, and their relations in a database for a cartographic department:

1. **Spatial Data**

* Attributes: feature \_id, feature\_type, geometry, image
* Relations: Spatial data can be used to create various cartographic layers.

1. **Maps**

* Attributes: map\_id, map\_name, map\_type, scale, publication\_date
* Relations: Maps are composed of one or more cartographic layers. Maps can be part of a map series.

1. **Map Series (Atlas)**

* Attributes: series\_id, series\_name, series\_type, coverage\_area
* Relations: A map series contains multiple related maps.

1. **Data Sources**

* Attributes: source\_id, source\_name, source\_type, data\_quality, contact\_info
* Relations: Spatial data and cartographic layers are acquired from various data sources.

1. **Personnel**

* Attributes: employee\_id, first\_name, last\_name, job\_title, contact\_info, expertise
* Relations: Employees are assigned to and participate in cartographic workflows.

1. **Equipment and Software**

* Attributes: item\_id, item\_name, item\_type, model, version, specifications, purchase\_date
* Relations: Equipment and software are used in the cartographic workflows.

1. **Clients and Orders**

* Attributes: client\_id, client\_name, contact\_info, order\_id, order\_date, product\_id, quantity, delivery\_method, payment\_info
* Relations: Clients place orders for cartographic products, which are fulfilled by the department.

These entities and their relationships would allow the cartographic department to manage spatial data, create and maintain maps, track workflows and personnel, monitor equipment and software, and manage client orders and deliveries.

CREATING A DATABASE

For the creation of our database, we shall use the client-server-DBMS architecture

These are its components:

* + **Client(s):** Represented by a computer or device running a software application.
  + **Client Application:** This software interacts with the user and sends requests/data to the server.
  + **Server:** A powerful computer running database management software.
  + **Database Management System (DBMS):** Software that manages the database on the server.
  + **Database:** The central storage location for structured data, often represented as tables.

**How it works**

* + User interacts with the client application.
  + Client application formulates a request (e.g., read/write data).
  + Client application sends the request to the server.
  + Server receives the request and communicates with the DBMS.
  + DBMS processes the request on the database.
  + Server sends the response (data or confirmation) back to the client application.
  + Client application displays the response to the user.

**Why do we use this architecture?**

1. The server acts as a central hub for data storage and management. This allows for:

* Easier implementation of security measures like user access controls and data encryption.
* Consistent data management practices to ensure data integrity.
* Easier backups and disaster recovery procedures.

1. The server can be a powerful machine specifically designed to handle database operations efficiently. This allows for:

* Supporting a large number of clients accessing the database concurrently.
* Handling complex data queries and manipulations smoothly.
* Upgrading the server's hardware or software to improve performance as needed.

1. Clients can be any device with minimal processing power or storage needs as they rely on the server for the heavy lifting. This allows for:

* A wider range of devices to access the database (e.g.: desktops, laptops, mobile phones).
* Easier updates and maintenance since updates are applied on the server, affecting all clients.
* The client application can be tailored for specific user needs without impacting the core database functionality.

**Physical architecture**

To determine the appropriate physical architecture for managing the cartographic department's data, we would need to consider several key factors:

1. Data Volume and Growth:
   * Estimate the current and future data volume requirements for the cartographic data, such as the size of raster images, vector data, and metadata.
   * Understand the expected growth rate of the data to ensure the physical architecture can scale accordingly.
2. Data Characteristics:
   * Analyse the types of data (e.g., raster images, vector data, metadata) and their specific storage and processing requirements.
   * Consider the data formats, access patterns, and the need for specialized indexing or querying capabilities.
3. Performance Requirements:
   * Identify the key performance metrics, such as response time for data retrieval, throughput for data processing, and concurrency requirements.
4. Availability and Reliability:
   * Determine the required levels of data availability, redundancy, and fault tolerance to ensure continuous service and data protection.
   * Consider the need for disaster recovery mechanisms and geographical distribution of the data.
5. Security and Access Control:
   * Evaluate the data sensitivity and the need for robust access control, encryption, and auditing mechanisms.
   * Identify the user roles and their respective data access requirements.

After this, you can now determine the hardware you can use to store your data depending on all the above factors.

Based on the requirements and considerations discussed earlier, such as data volume, performance, availability, and security, the following hardware components could be used to store and manage the cartographic department's data:

**1. Storage:**

* + For the large volumes of raster image data, vector data, and associated metadata, a combination of storage options would be recommended:
  + High-capacity hard disk drives (HDDs) for the main data storage: HDDs provide cost-effective, high-capacity storage for the bulk of the cartographic data.
  + Solid-state drives (SSDs) for performance-sensitive data and metadata: SSDs offer faster data access and retrieval times, which is beneficial for frequently accessed metadata and vector data.
  + The storage solution could be implemented as a distributed file system, such as HDFS or Amazon S3, to provide scalability, fault tolerance, and high availability.
  + Depending on the data growth rate and performance requirements, the storage capacity and mix of HDD and SSD could be scaled accordingly.

**2. Compute:**

* + For the data processing and analytics tasks, a cluster of powerful server machines would be appropriate:
  + High-performance CPUs with multiple cores to enable parallel processing of raster image analysis, spatial queries, and data transformations.
  + Substantial amounts of RAM to support in-memory caching and processing of large datasets.
  + Accelerators like GPUs or specialized hardware (e.g., FPGAs) could be considered for tasks that benefit from hardware acceleration, such as image processing or machine learning workloads.
  + The compute cluster could be implemented using a distributed data processing framework, like Apache Spark or Apache Flink, to leverage the scalability and fault tolerance of the underlying hardware.

**3. Network:**

* + High-speed network connectivity between the storage and compute components is crucial for efficient data transfer and processing.
  + Deployment of a high-bandwidth, low-latency network infrastructure, such as 10 GbE or 40 GbE Ethernet, would be recommended.
  + Depending on the geographical distribution of the data and users, the network topology could include edge computing devices, content delivery networks (CDNs), or remote data centres to optimize data access and minimize latency.

**4. High Availability and Disaster Recovery:**

* + To ensure the desired levels of data availability and resilience, the hardware architecture should include redundant and fault-tolerant components:
  + Redundant storage systems with RAID configurations or replication across multiple data centres.
  + Failover mechanisms for the compute clusters, such as load balancing and automatic failover.
  + Comprehensive backup and disaster recovery plans, including off-site data replication and the ability to quickly restore the system in the event of a failure or disaster.

**5. Security and Access Control:**

* + The hardware architecture should integrate with security mechanisms, such as:
  + Secure access controls and authentication for user and application access to the cartographic data.
  + Encryption of data at rest (on the storage devices) and in transit over the network.
  + Hardware-based security features, like trusted platform modules (TPMs) or security-hardened servers, to enhance the overall security posture.

The specific hardware configuration and sizing would depend on the detailed requirements and growth projections of the cartographic department's data and workloads. A thorough capacity planning and system design process would be necessary to determine the optimal hardware components and architecture.

Below is the estimated number of servers and other hardware devices required for a cartographic data management system:

**1. Storage:**

* + 1-2 high-capacity storage servers with a combination of HDDs and SSDs
  + The HDDs could provide the primary bulk storage, in the range of 20-50 TB
  + The SSDs could be used for caching and storing performance-sensitive data, around 2-5 TB
* Optionally, a distributed file system or object storage platform (e.g., HDFS, S3) could be implemented across the storage servers to provide scalability and high availability

**2. Compute:**

* + 1-2 application/compute servers, each with:
  + High-performance CPUs (e.g., 12-24 cores)
  + 64-128 GB of RAM
* Optionally, 1-2 GPU accelerators for tasks like image processing or machine learning

**3. Network:**

* + 1 high-speed network switch (e.g., 10 GbE or 40 GbE)
  + Depending on the geographical distribution, 1-2 edge computing devices or a content delivery network (CDN) could be considered to optimize data access and reduce latency

**4. High Availability and Disaster Recovery:**

* + site data replicas
* Optionally, a cloud-based disaster recovery solution could be implemented to provide offsite data protection and quick recovery

**5. Security and Access Control:**

* + Integrated security features within the storage and compute servers, such as TPM, hardware encryption, and access control mechanisms
  + 1 dedicated server or virtual machine for centralized identity management, authentication, and authorization

In summary, the estimated hardware setup would include:

* + 2-3 primary storage and compute servers
  + 1 backup/disaster recovery server
  + 1 high-speed network switch
  + Optionally, 1-2 edge computing devices or a CDN
  + Integrated security features and a dedicated server for access control

This setup provides a balance of performance, capacity, and redundancy to meet the cartographic data management requirements of a small organization. As the data volumes and workloads grow, the hardware configuration can be scaled accordingly.

HOW TO BUILD A DBMS

Here's an overview of the steps involved in designing a DBMS:

* Determine the purpose and requirements of the database: what kind of data will be stored, the relationships between the data, who will use the database, and what kind of queries will be performed on the data.
* Choose a database management system (DBMS) that fits your requirements. Examples of DBMS include MySQL, PostgreSQL, Microsoft SQL Server, and Oracle. Some are better than others for specific purposes (whether you’ll be using on a local computer or plan on having thousands or more users, etc.), and we’ll discuss this in a little more detail later.
* Design the database schema: create a visual representation of the database structure, including tables, columns, data types, and relationships between tables.
* Implement the database schema: create tables and columns in the chosen DBMS, and specify constraints such as primary keys and foreign keys.
* Populate the database with data: add data to the tables and verify that it meets the constraints.
* Test the database: perform queries on the data to ensure that it can be retrieved as required.
* Maintain the database: regularly back up the data, monitor performance, and make updates as needed to ensure that the database continues to meet the needs of its users.

WEBOGRAPHY

* 1. Collecting Data:[https://www.usgs.gov/data-management](https://www.usgs.gov/data-management%20)  22:33 01-07-2024
  2. Creating a database:<https://ebooks.inflibnet.ac.in/csp4/chapter/client-server-architecture> 22:38 01-07-2024
  3. How to build a DBMS:[https://interestingengineering.com/lists/build-a-database-part-1](https://interestingengineering.com/lists/build-a-database-part-1%2022:40)  22:40 01-07-2024