Database security

- Database security is the technique that protects and secures the database against intentional or accidental threats.
- Security concerns will be relevant not only to the data resides in an organization's database: the breaking of security may harm other parts of the system, which may ultimately affect the database structure.
- Consequently, database security includes hardware parts, software parts, human resources, and data
- To efficiently do the uses of security needs appropriate controls, which are distinct in a specific mission and purpose for the system.

- There are three layers of database security: the database level, the access level, and the perimeter level.
- Security at the database level occurs within the database itself, where the data live.
- Access layer security focuses on controlling who is allowed to access certain data or systems containing it.
- Database security at the perimeter level determines who can and cannot get into databases. Each level requires unique security solutions.

Security Level	Database Security Solutions
Database Level	•Masking •Tokenization •Encryption
Access Level	•Access Control Lists •Permissions
Perimeter Level	•Firewalls •Virtual Private Networks

We consider database security about the following situations:

- Theft and fraudulent.
- Loss of confidentiality or secrecy.
- Loss of data privacy.
- Loss of data integrity.
- Loss of availability of data.

Threats in a Database

- **Availability loss** − Availability loss refers to non-availability of database objects by legitimate users.
- **Integrity loss** − Integrity loss occurs when unacceptable operations are performed upon the database either accidentally or maliciously. This may happen while creating, inserting, updating or deleting data. It results in corrupted data leading to incorrect decisions.
- **Confidentiality loss** − Confidentiality loss occurs due to unauthorized or unintentional disclosure of confidential information. It may result in illegal actions, security threats and loss in public confidence.

- **Security models**

- A security model establishes the external criteria for the examination of security issues in general, and provides the context for database considerations, including implementation and operation. Specific DBMSs have their own security models which are highly important in systems design and operation.
- Any faults in the security model will translate either into insecure operation or clumsy systems. Access control The purpose of access control must always be clear.
- Access control is expensive in terms of analysis, design and operational costs. It is applied to known situations, to known standards, to achieve known purposes.
- Do not apply controls without all the above knowledge. Control always has to be appropriate to the situation. The main issues are introduced below.

Authentication

- the DBMS is concerned, is an authorization-identifier. Authentication does not give any privileges for particular tasks. It only establishes that the DBMS trusts that the user is who he/she claimed to be and that the user trusts that the DBMS is also the intended system. Authentication is a prerequisite for authorization.

Authorization

- Authorization relates to the permissions granted to an authorized user to carry out particular transactions, and hence to change the state of the database (write item transactions) and/or receive data from the database (read-item transactions). The result of authorization, which needs to be on a transactional basis, is a vector: Authorization (item, auth-id, operation).
- At a logical level, the system structure needs an authorization server, which needs to co-operate with an auditing server. There is an issue of server-to-server security and a problem with amplification as the authorization is transmitted from system to system. Amplification here means that the security issues become larger as a larger number of DBMS servers are involved in the transaction

Access Control Models:

Access control is the combination of policies and technologies that decide which authenticated users may access which resources. Security requirements, infrastructure, and other considerations lead companies to choose among the four most common access control models:

- Mandatory Access Control (MAC)
- Discretionary Access Control (DAC)
- Role-Based Access Control (RBAC)

Mandatory access control(MAC):

- Mandatory access control uses a centrally managed model to provide the highest level of security. A non-discretionary system, MAC reserves control over access policies to a centralized security administration.
- MAC works by applying security labels to resources and individuals. These security labels consist of two elements:
- **Classification and clearance** — MAC relies on a classification system (restricted, secret, top-secret, etc.) that describes a resource's sensitivity. Users' security clearances determine what kinds of resources they may access.
- **Compartment —** A resource's compartment describes the group of people (department, project team, etc.) allowed access. A user's compartment defines the group or groups they participate in.

Advantages of MAC

- **Enforceability** — MAC administrators set organization-wide policies that users cannot override, making enforcement easier.
- **Compartmentalization** — Security labels limit the exposure of each resource to a subset of the user base.

Disadvantages of MAC

- **Collaboration —** MAC achieves security by constraining communication. Highly collaborative organizations may need a less restrictive approach.
- **Management burden —** A dedicated organizational structure must manage the creation and maintenance of security labels.

Discretionary Access Control (DAC)?

- Discretionary access control decentralizes security decisions to resource owners. The owner could be a document’s creator or a department’s system administrator. DAC systems use access control lists (ACLs) to determine who can access that resource. These tables pair individual and group identifiers with their access privileges.
- The sharing option in most operating systems is a form of DAC. For each document you own, you can set read/write privileges and password requirements within a table of individuals and user groups. System administrators can use similar techniques to secure access to network resources.

Advantages of DAC

- **Conceptual simplicity** — ACLs pair a user with their access privileges. As long as the user is in the table and has the appropriate privileges, they may access the resource.
- **Responsiveness to business needs** — Since policy change requests do not need to go through a security administration, decision-making is more nimble and aligned with business needs.

Disadvantages of DAC

- **Over/underprivileged users** — A user can be a member of multiple, nested workgroups. Conflicting permissions may over- or under privilege the user.
- **Limited control** — Security administrators cannot easily see how resources are shared within the organization. And although viewing a resource's ACL is straightforward, seeing one user's privileges requires searching every ACL.
- **Compromised security** — By giving users discretion over access policies, the resulting inconsistencies and missing oversight could undermine the organization's security posture.

Role-based Access Control (RBAC):

- Role-based access control grants access privileges based on the work that individual users do. A popular way of implementing “least privilege” policies, RBAC limits access to just the resources users need to do their jobs.
- Implementing RBAC requires defining the different roles within the organization and determining whether and to what degree those roles should have access to each resource.
- Accounts payable administrators and their supervisor, for example, can access the company’s payment system. The administrators’ role limits them to creating payments without approval authority. Supervisors, on the other hand, can approve payments but may not create them.

Advantages of RBAC

- **Flexibility** — Administrators can optimize an RBAC system by assigning users to multiple roles, creating hierarchies to account for levels of responsibility, constraining privileges to reflect business rules, and defining relationships between roles.
- **Ease of maintenance** — With well-defined roles, the day-to-day management is the routine on-boarding, off-boarding, and cross-boarding of users' roles.
- **Centralized, non-discretionary policies** — Security professionals can set consistent RBAC policies across the organization.
- **Lower risk exposure** — Under RBAC, users only have access to the resources their roles justify, greatly limiting potential threat vectors.

Disadvantages of RBAC

- **Complex deployment** — The web of responsibilities and relationships in larger enterprises makes defining roles so challenging that it spawned its own subfield: role engineering.
- **Balancing security with simplicity** — More roles and more granular roles provide greater security, but administering a system where users have dozens of overlapping roles becomes more difficult.
- **Layered roles and permissions** — Assigning too many roles to users also increases the risk of over-privileging users.

Intrusion Detection:

- Intrusion detection is an important countermeasure for most applications, especially client-server applications like web applications and web services. Many newer technologies are beginning to include integrated services such as a single device that incorporates a firewall, IDS, and limited IPS functionality. Logging is an important aspect of intrusion detection, but is best viewed as a way to record intrusion-related activity, not to determine what is an intrusion in the first place. The vast majority of applications do not detect attacks, but instead try their best to fulfill the attackers' requests.
- Intrusion detection systems are used to detect anomalies with the aim of catching hackers before they do real damage.

- Database management systems (DBMS), which are the ultimate layer in preventing malicious data access or corruption, implement several security mechanisms to protect data. However these mechanisms cannot always stop malicious users from accessing the data by exploiting system vulnerabilities.
- In fact, when a malicious user accesses the database there is no effective way to detect and stop the attack in due time. This practical experience report presents a tool that implements concurrent intrusion detection in DBMS.
- This tool analyses the transactions the users execute and compares them with the profile of the authorized transactions that were previously learned in order to detect potential deviations. The tool was evaluated using the transactions from a standard database benchmark (TPC-W) and a real database application. Results show that the proposed intrusion detection tool can effectively detect SQL-based attacks with no false positives and no overhead to the server

SQL Injection

- SQL injection is a technique used to exploit user data through web page inputs by injecting SQL commands as statements. Basically, these statements can be used to manipulate the application’s web server by malicious users.
- SQL injection is a code injection technique that might destroy your database.
- SQL injection is one of the most common web hacking techniques.
- SQL injection is the placement of malicious code in SQL statements, via web page input.

Example of SQL Injection

Suppose we have an application based on student records. Any student can view only his or her own records by entering a unique and private student ID. Suppose we have a field like below:
Student id:
And the student enters the following in the input field:
12222345 or 1=1.
So this basically **translates to :**
SELECT * from STUDENT where STUDENT-ID == 12222345 or 1 = 1

Now this **1=1** will return all records for which this holds true. So basically, all the student data is compromised. Now the malicious user can also delete the student records in a similar fashion.

Consider the following SQL query.

```
SELECT * from USER where USERNAME = “” and PASSWORD=””
```

Now the malicious can use the ‘=’ operator in a clever manner to retrieve private and secure user information. So instead of the above-mentioned query the following query when executed, retrieves protected data, not intended to be shown to users.

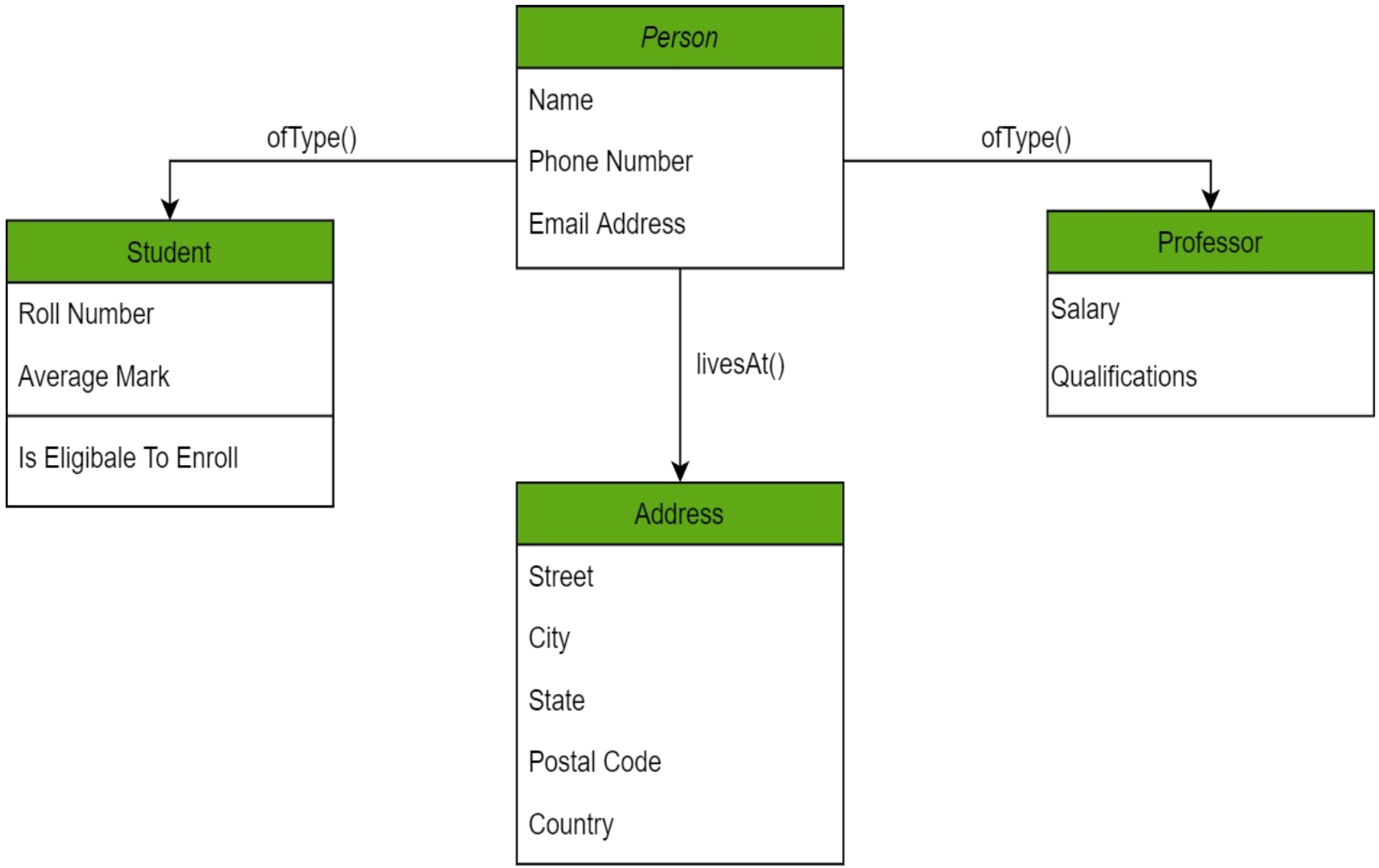
```
Select * from User where (Username = “” or 1=1) AND (Password=”” or 1=1).
```

Since **1=1** always holds true, user data is compromised.

- **Impact of SQL Injection**
  The hacker can retrieve all the user-data present in the database such as user details, credit card information, social security numbers and can also gain access to protected areas like the administrator portal. It is also possible to delete the user data from the tables.
  Nowadays, all online shopping applications, bank transactions use back-end database servers. So in-case the hacker is able to exploit SQL injection, the entire server is compromised.
- **Preventing SQL Injection**
- User Authentication: Validating input from the user by pre-defining length, type of input, of the input field and authenticating the user.
- Restricting access privileges of users and defining as to how much amount of data any outsider can access from the database. Basically, user should not be granted permission to access everything in the database.
- Do not use system administrator accounts.

Object-Oriented Databases:

- Those familiar with the Object-Oriented Programming Paradigm would be able to relate to this model of databases easily. Information stored in a database is capable of being represented as an object which response as an instance of the database model. Therefore, the object can be referenced and called without any difficulty. As a result, the workload on the database is substantially reduced.


Person
Name
Phone Number
Email Address
ofType()
ofType()
Student
Roll Number
Average Mark
Is Eligibale To Enroll
livesAt()
Professor
Salary
Qualifications
Address
Street
City
State
Postal Code
Country

Object oriented database

- An Object relational model is a combination of a Object oriented database model and a Relational database model. So, it supports objects, classes, inheritance etc. just like Object Oriented models and has support for data types, tabular structures etc. like Relational data model.
- One of the major goals of Object relational data model is to close the gap between relational databases and the object oriented practises frequently used in many programming languages such as C++, C#, Java etc.

Advantages of Object Relational model

- **Inheritance**The Object Relational data model allows its users to inherit objects, tables etc. so that they can extend their functionality. Inherited objects contains new attributes as well as the attributes that were inherited.
- **Complex Data Types** Complex data types can be formed using existing data types. This is useful in Object relational data model as complex data types allow better manipulation of the data.
- **Extensibility**The functionality of the system can be extended in Object relational data model. This can be achieved using complex data types as well as advanced concepts of object oriented model such as inheritance.

Disadvantages of Object Relational model

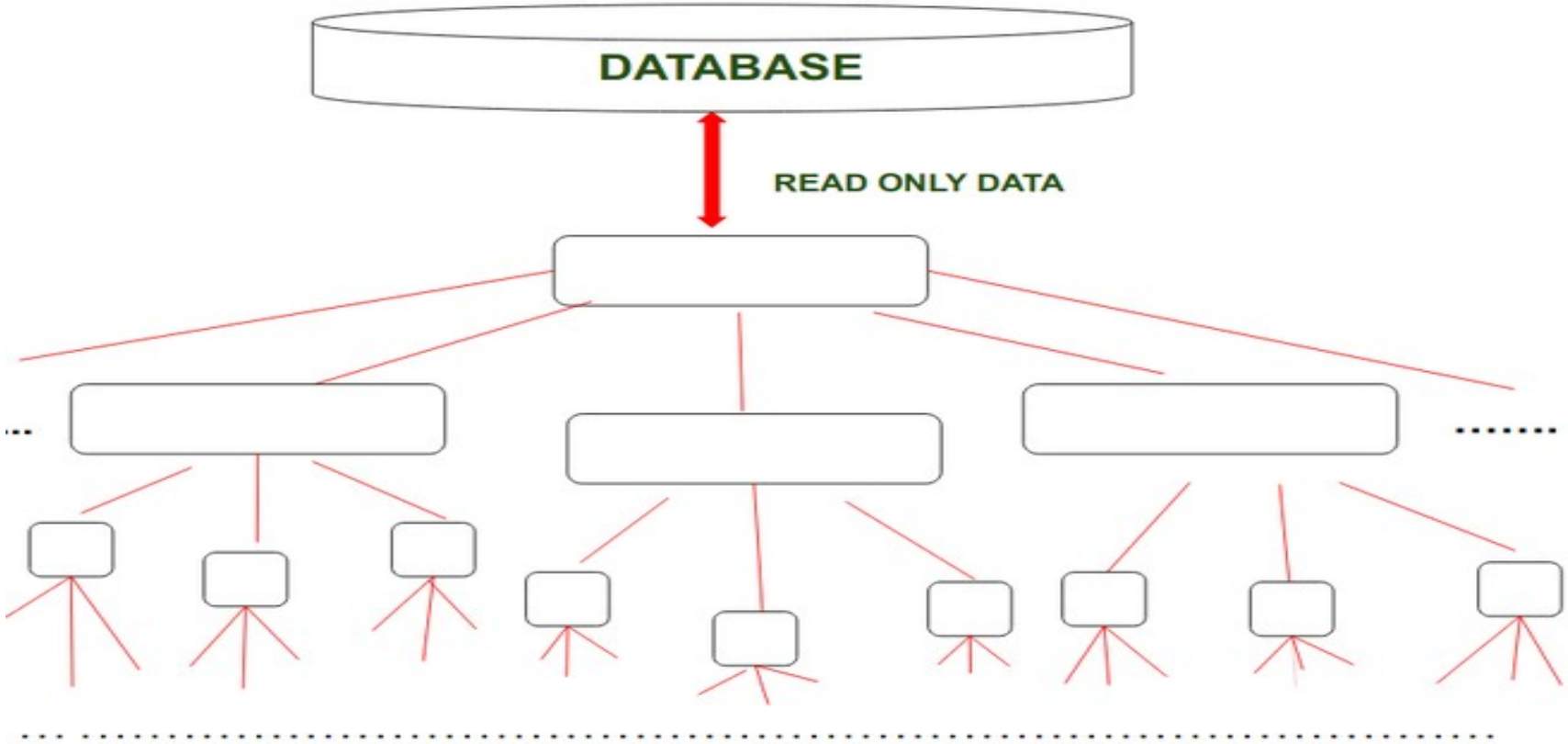
- The object relational data model can get quite complicated and difficult to handle at times as it is a combination of the Object oriented data model and Relational data model and utilizes the functionalities of both of them.

Logical Database

A Logical Database is a special type of ABAP (Advance Business Application and Programming) that is used to retrieve data from various tables and the data is interrelated to each other. Also, a logical database provides a read-only view of Data.

- **Structure Of Logical Database:**

A Logical database uses only a hierarchical structure of tables i.e. Data is organized in a Tree-like Structure and the data is stored as records that are connected to each other through edges (Links). Logical Database contains Open SQL statements which are used to read data from the database. The logical database reads the program, stores them in the program if required, and passes them line by line to the application program.


DATABASE
READ ONLY DATA

Features of Logical Database:

In this section, let us look at some features of a logical database:

- We can select only that type of Data that we need.
- Data Authentication is done in order to maintain security.
- Logical Database uses hierarchical Structure due to this data integrity is maintained.

Goal Of Logical Database:

The goal of Logical Database is to create well-structured tables that reflect the need of the user. The tables of the Logical database store data in a non-redundant manner and foreign keys will be used in tables so that relationships among tables and entities will be supported.

Web Database

- A web database is a system for storing information that can then be accessed via a website. For example, an online community may have a database that stores the username, **password**, and other details of all its members. The most commonly used database system for the internet is MySQL due to its integration with PHP — one of the most widely used server side programming languages.
- At its most simple level, a web database is a set of one or more tables that contain data. Each table has different fields for storing information of various types. These tables can then be linked together in order to manipulate data in useful or interesting ways. In many cases, a table will use a **primary key**, which must be unique for each entry and allows for unambiguous selection of data.

Distributed Database

A distributed database is basically a database that is not limited to one system, it is spread over different sites, i.e, on multiple computers or over a network of computers. A distributed database system is located on various sites that don’t share physical components. This may be required when a particular database needs to be accessed by various users globally. It needs to be managed such that for the users it looks like one single database.

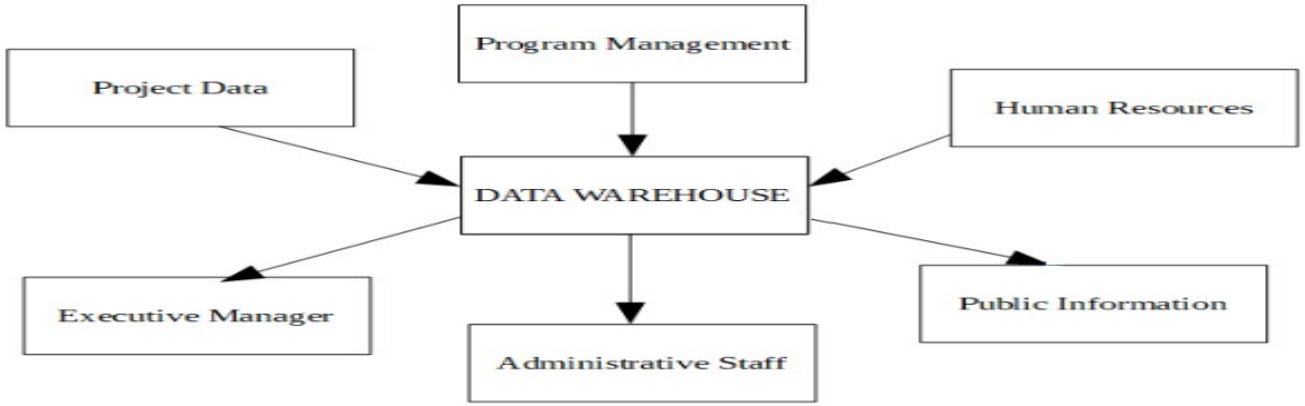
- **Types:**
- **1. Homogeneous Database:**
  In a homogeneous database, all different sites store database identically. The operating system, database management system, and the data structures used – all are the same at all sites. Hence, they’re easy to manage.
- **2. Heterogeneous Database:**
  In a heterogeneous distributed database, different sites can use different schema and software that can lead to problems in query processing and transactions. Also, a particular site might be completely unaware of the other sites. Different computers may use a different operating system, different database application. They may even use different data models for the database. Hence, translations are required for different sites to communicate.
-

Data Warehousing

- Data warehousing is a collection of tools and techniques using which more knowledge can be driven out from a large amount of data. This helps with the decision-making process and improving information resources.
- Data warehouse is basically a database of unique data structures that allows relatively quick and easy performance of complex queries over a large amount of data. It is created from multiple heterogeneous sources.

Characteristics of Data Warehousing

- Integrated
- Time variant
- Non-volatile
- The purpose of Data warehouse is to support the decision making process. It makes information easily accessible as we can generate reports from the data warehouse. It usually contains historical data derived from transactional data but can also include data from other sources. Data warehouse is always kept separated from transactional data.



Data mining

Data mining refers to extracting knowledge from large amounts of data. The data sources can include databases, data warehouse, web etc.

Knowledge discovery is an iterative sequence:

- Data cleaning – Remove inconsistent data.
- Data integration – Combining multiple data sources into one.
- Data selection – Select only relevant data to be analyzed.
- Data transformation – Data is transformed into appropriate form for mining.

Data mining – methods to extract data patterns.

- Pattern evaluation – identify interesting patterns in the data.
- Knowledge representation- visualization and knowledge representation techniques are used.

What kind of data that can be mined?

- Database Data
- Data Warehouse
- Transactional Data

Scope of Data mining

- Automated Prediction of trends and behaviours: Data mining automates the process of finding the predictive information in large databases. For example : Consider a marketing company. In this company, data mining uses the past promotional mailing to identify the targets to maximize the return.
- Automated discovery of previously unknown patterns: Data mining sweeps through the database and identifies previously hidden patterns. For example: In a retail store data mining will go through the entire database and find the pattern for the items which are usually brought together.