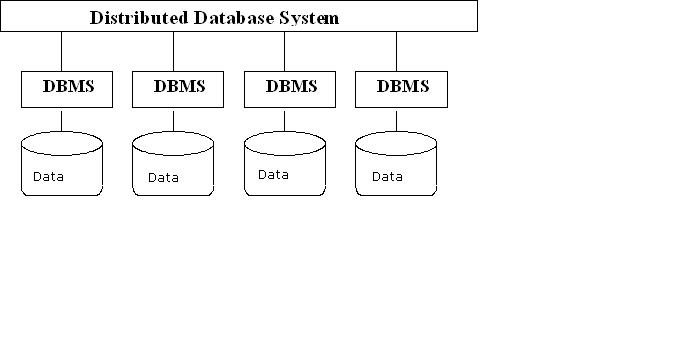
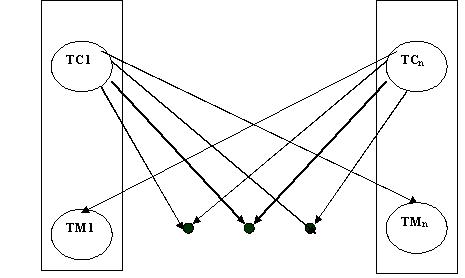
A distributed database is a database that consists of two or more files located in different sites either on the same network or on entirely different networks. Portions of the database are stored in multiple physical locations and processing is distributed among multiple database nodes.



**Overall System Architecture**  
  
  
****

The main difference between centralized and distributed database systems is that, in the former, the data reside in one single location, whereas in the latter, the data reside in several locations. This distribution of data is the case of many difficulties in transaction processing and query processing.

**Advantages of a DDBS**

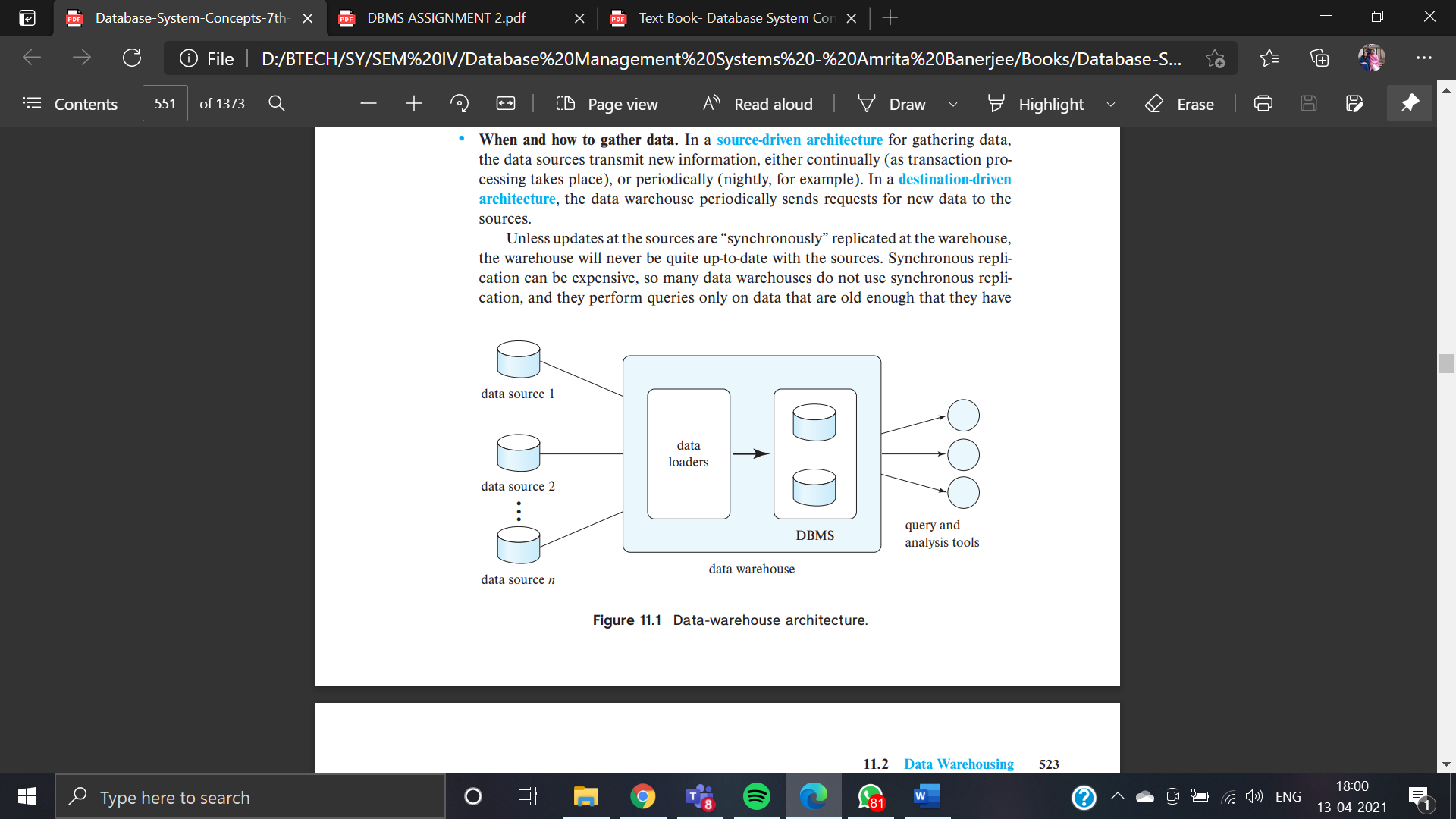
* Modularity
* Fault Tolerance
* High Performance
* Data Sharing
* Low Cost Component**s**

**Issues**

* Data Distribution
* Exploiting Parallelism
* Concurrency and Recovery
* Heterogeneity



A data warehouse is a repository (or archive) of information gathered from multiple sources, stored under a unified schema, at a single site. Once gathered, the data are stored for a long time, permitting access to historical data. Thus, data warehouses provide the user a single consolidated interface to data, making decision-support queries easier to write. Moreover, by accessing information for decision support from a data warehouse, the decision maker ensures that online transaction-processing systems are not affected by the decision-support workload.





Data Mining is a process of finding potentially useful patterns from huge data sets. It is a multi-disciplinary skill that uses machine learning, statistics, and AI to extract information to evaluate future events probability. The insights derived from Data Mining are used for marketing, fraud detection, scientific discovery, etc.

Data Mining is all about discovering hidden, unsuspected, and previously unknown yet valid relationships amongst the data. Data mining is also called Knowledge Discovery in Data (KDD), Knowledge extraction, data/pattern analysis, information harvesting, etc.



Database Encryption involves an algorithm which converts the data into incomprehensible cipher text. This cipher text is unusable till it is decrypted back.

Two main advantages of data encryption as follows;

* It makes the data unreadable. It can only be read or updated again by first decrypting it with a corresponding decryption key. As a result, the data becomes strongly secure and unauthorized access can be easily prevented.
* Hackers cannot manipulate this data as data integrity is also guaranteed. Even if a hackers accesses the encrypted data for manipulation or misuse. They cannot do it as the encrypted data does not make sense.



An application developer needs to identify and prevent attacks like cross-site scripting and SQL injection attacks to provide better security to a system.

• When an attacker is able to insert and execute malicious code to a web for input, it is called an SQL injection attack.

• The attacker then not only gains access to resources or data but also manipulate and alter the data

• Any SQL query string which is constructed by concatenating the user supplied parameters; is prone to the SQL injection attack.

**The working of an SQL injection attack can be detailed as follows;**

o The attacker basically terminates a text string and attaches a new command to it.

o Also, the attacker will terminate the freshly attached string with a comment mark.

o This is done to facilitate further insertion of command or attaching an additional string to it before it is executed.

• The implication of such a thing can be understood by an example. Consider an application which executes a query to check a user’s password as below;

String id= request.getParameter(“userid”);

String pwd= request.getParameter(“password”);

executeQuery(“SELECT password FROM users WHERE userid= ‘“+id+“’”;

• Now, suppose the user entered the value for the user id as ‘admin’,

SELECT password FROM users WHERE userid==’admin’;

• Obviously, this can be disastrous for the security of an application revealing unauthorized information.

**An SQL injection attack can be failed by taking the following precautions;**

o Never concatenate a user input values directly into a query string.

o Use prepared statement in the various libraries to ensure that special characters like quotes and question marks are escaped as fitting to the target database.

o Initially validate or constraint the input in the application for type, length, range and format.

o Use parameters with Dynamic SQL and particular stored procedures.