CLOUD APPLICATION DEVELOPMENT (GROUP 1)

PHASE 5 : ASSIGNMENT NOTEBOOK SUBMISSION

NAME :SOMESH D

EMAIL : someshsomi28@gmail.com

GitHub Repository URL : https://github.com/somesh2809/somesh.git

Title of the Project :-

PROJECT 7 : Data warehouse on IBM Cloud Foundary

# Data Warehousing

## **Abstract:**

Data warehousing is a collection of decision support technologies, aimed at enabling the knowledge worker to make better and faster decisions. A data warehouse is a subject-oriented, integrated, time varying, non-volatile collection of data that is used primarily in organizational decision making. Data warehouse supports on-line analytical processing, the functional and performance requirements of which are quite different from those of the on-line transaction processing applications traditionally supported by the operational databases. In this paper author suggest that for a strategic solution, the hub and spoke / centralised architecture is the more likely choice

[

import logging

import time

import mysql.connector

# Set up logger

logger = logging.getLogger(\_\_name\_\_)

logger.setLevel(logging.INFO)

formatter = logging.Formatter("%(asctime)s - %(name)s - %(levelname)s - %(message)s")

# Log to console

handler = logging.StreamHandler()

handler.setFormatter(formatter)

logger.addHandler(handler)

# Also log to a file

file\_handler = logging.FileHandler("cpy-errors.log")

file\_handler.setFormatter(formatter)

logger.addHandler(file\_handler)

def connect\_to\_mysql(config, attempts=3, delay=2):

attempt = 1

# Implement a reconnection routine

while attempt < attempts + 1:

try:

return mysql.connector.connect(\*\*config)

except (mysql.connector.Error, IOError) as err:

if (attempts is attempt):

# Attempts to reconnect failed; returning None

logger.info("Failed to connect, exiting without a connection: %s", err)

return None

logger.info(

"Connection failed: %s. Retrying (%d/%d)...",

err,

attempt,

attempts-1,

)

# progressive reconnect delay

time.sleep(delay \*\* attempt)

attempt += 1

return None

Amazon Redshift:-

Amazon Redshift is a cloud-based fully managed petabytes-scale data warehouse By the Amazon Company. It starts with just a few hundred gigabytes of data and scales to petabytes or more. This enables the use of data to accumulate new insights for businesses and customers. It is a relational database management system (RDBMS) therefore it is compatible with other RDBMS applications. Amazon Redshift offers quick querying capabilities over structured data by the use of SQL-based clients and business intelligence (BI) tools using standard ODBC and JDBC connections. Amazon Redshift is made around industry-standard SQL, with additional practicality to manage massive datasets and support superior analysis and reporting of these data. It helps to work quickly and easily along with data in open formats, and simply integrates with and connects to the AWS scheme. Also query and export data to and from the data lake. No alternative cloud data warehouse tool makes it straightforward to query data and writes data back to the data lake in open formats. It focuses on simple Use and Accessibility. MySQL and alternative SQL-based systems are one in all the foremost well-liked and simply usable interfaces for database management. Redshift’s easy query-based system makes platform adoption and acclimatization a light breeze. It is incredibly quick once it involves loading data and querying it for analytical and reporting functions.

SELECT subjects, s\_name, mark, dense\_rank()

OVER ( partition by subjects order by mark desc )

AS 'dense\_rank' FROM result;

Microsoft Azure:

Azure is a cloud computing platform that was launched by Microsoft in 2010. Microsoft Azure is a cloud computing service provider for building, testing, deploying, and managing applications and services through Microsoft-managed data centers. Azure is a public cloud computing platform that offers Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). The Azure cloud platform provides more than 200 products and cloud services such as Data Analytics, Virtual Computing, Storage, Virtual Network, Internet Traffic Manager, Web Sites, Media Services, Mobile Services, Integration, etc. Azure facilitates simple portability and genuinely compatible platform between on-premise and public Cloud. Azure provides a range of cross-connections including virtual private networks (VPNs), caches, content delivery networks (CDNs), and ExpressRoute connections to improve usability and performance. Microsoft Azure provides a secure base across physical infrastructure and operational security. Azure App offers a completely managed web hosting service that helps in building web applications, services, and Restful APIs. It offers a variety of plans to meet the requirements of any application, from small to globally scaled web applications. Running virtual machines or containers in the cloud is one of the most popular applications of Microsoft Azure.

rank():

This function assigns a rank to each row within a partition that has gaps. In this case, ranks are assigned in a non-consecutive manner i.e if there two records have the same value then they will be assigned the same rank, and the next rank value will be the previous rank plus the number of duplicates.

SELECT subjects, s\_name, mark, rank()

OVER ( partition by subjects order by mark desc )

AS 'rank' FROM result;

**Google BigQuery:**

BigQuery is a serverless data warehouse that allows scalable analysis over petabytes of data. It’s a Platform as a Service that supports querying with the help of ANSI SQL. It additionally has inbuilt machine learning capabilities. BigQuery was declared in 2010 and made available for use there in 2011. Google BigQuery is a cloud-based big data analytics web service to process very huge amount of read-only data sets. BigQuery is designed for analyzing data that are in billions of rows by simply employing SQL-lite syntax. BigQuery can run advanced analytical SQL-based queries beneath big sets of data. BigQuery is not developed to substitute relational databases and for easy CRUD operations and queries. It is oriented for running analytical queries. It is a hybrid system that enables the storage of information in columns; however, it takes into the NoSQL additional features, like the data type, and the nested feature. BigQuery is a better option than Redshift since we have to pay by the hour. BigQuery may also be the best solution for data scientists running ML or data mining operations since they deal with extremely large datasets. Google Cloud also offers a set of auto-scaling services that enables you to build a data lake that integrates with your existing applications, skills, and IT investments. In BigQuery, most of the time is spent on metadata/initiation, but the actual execution time is very small.

# Python script for ETL process

import sqlite3

# Connect to source database (transactional database)

source\_conn = sqlite3.connect('source\_database.db')

source\_cursor = source\_conn.cursor()

# Connect to destination database (data warehouse)

warehouse\_conn = sqlite3.connect('data\_warehouse.db')

warehouse\_cursor = warehouse\_conn.cursor()

# Extract data from source

source\_cursor.execute('SELECT \* FROM transactions')

data\_to\_load = source\_cursor.fetchall()

# Transform data if needed (for simplicity, let's assume no transformation for this example)

# Load data into data warehouse

warehouse\_cursor.executemany('INSERT INTO fact\_table (column1, column2, column3) VALUES (?, ?, ?)', data\_to\_load)

warehouse\_conn.commit()

# Close connections

source\_conn.close()

warehouse\_conn.close()

**Snowflake:**

Snowflake is a cloud computing-based data warehousing built on top of the Amazon Web Services or Microsoft Azure cloud infrastructure. The Snowflake design allows storage and computes to scale independently, thus customers can use and pay money for storage and computation individually. In Snowflake data processing is simplified: Users will do data blending, analysis, and transformations against varied forms of data structures with one language, SQL. Snowflake offers dynamic, scalable computing power with charges primarily based strictly on usage. With Snowflake, computation and storage are fully separate, and also the storage value is that the same as storing the data on Amazon S3. AWS tried to handle this issue by introducing Redshift Spectrum, which allows querying data that exists directly on Amazon S3; however, it’s not as seamless as Snowflake. With Snowflake, we can clone a table, a schema, or perhaps a database in no time and occupying no extra space. This is often because the cloned table creates pointers that point to the kept data, however, not the actual data. In alternative words, the cloned table solely has data that’s completely different from its original table.

Implementation:-

Once the Planning and Design stages are complete, the project to implement the current Data Warehouse iteration can proceed quickly. Necessary hardware, software and middleware components are purchased and installed, the development and test environment is established, and the configuration management processes are implemented. Programs are developed to extract, cleanse, transform and load the source data and to periodically refresh the existing data in the Warehouse, and the programs are individually unit tested against a test database with sample source data. Metrics are captured for the load process. The metadata repository is loaded with transformational and business user metadata. Canned production reports are developed and sample ad-hoc queries are run against the test database, and the validity of the output is measured. User access to the data in the Warehouse is established. Once the programs have been developed and unit tested and the components are in place, system functionality and user acceptance testing is conducted for the complete integrated Data Warehouse system. System support processes of database security, system backup and recovery, system disaster recovery, and data archiving are implemented and tested as the system is prepared for deployment. The final step is to conduct the Production Readiness Review prior to transitioning the Data Warehouse system into production. During this review, the system is evaluated for acceptance by the customer organization.

Architecture Review and Design:-

The Architecture is the logical and physical foundation on which the Data Warehouse will be built. The Architecture Review and Design stage, as the name implies, is both a requirements analysis and a gap analysis activity. It is important to assess what pieces of the architecture already exist in the organization (and in what form) and to assess what pieces are missing which are needed to build the complete Data Warehouse architecture.

During the Architecture Review and Design stage, the logical Data Warehouse architecture is developed. The logical architecture is a configuration map of the necessary data stores that make up the Warehouse; it includes a central Enterprise Data Store, an optional Operational Data Store, one or more (optional) individual business area Data Marts, and one or more Metadata stores. In the metadata store(s) are two different kinds of metadata that catalog reference information about the primary data.

Once the logical configuration is defined, the Data, Application, Technical and Support Architectures are designed to physically implement it. Requirements of these four architectures are carefully analyzed so that the Data Warehouse can be optimized to serve the users. Gap analysis is conducted to determine which components of each architecture already exist in the organization and can be reused, and which components must be developed (or purchased) and configured for the Data Warehouse.

The Data Architecture organizes the sources and stores of business information and defines the quality and management standards for data and metadata.

The Application Architecture is the software framework that guides the overall implementation of business functionality within the Warehouse environment; it controls the movement of data from source to user, including the functions of data extraction, data cleansing, data transformation, data loading, data refresh, and data access (reporting, querying).

The Technical Architecture provides the underlying computing infrastructure that enables the data and application architectures. It includes platform/server, network, communications and connectivity hardware/software/middleware, DBMS, client/server 2-tier vs.3-tier approach, and end-user workstation hardware/software. Technical architecture design must address the requirements of scalability, capacity and volume handling (including sizing and partitioning of tables), performance, availability, stability, chargeback, and security.

Tool Selection:-

The purpose of this stage is to identify the candidate tools for developing and implementing the Data Warehouse data and application architectures, and for performing technical and support architecture functions where appropriate. Select the candidate tools that best meet the business and technical requirements as defined by the Data Warehouse architecture, and recommend the selections to the customer organization. Procure the tools upon approval from the organization.

It is important to note that the process of selecting tools is often dependent on the existing technical infrastructure of the organization. Many organizations feel strongly for various reasons about using tools for the Data Warehouse applications that they already have in their "arsenal" and are reluctant to purchase new application packages. It is recommended that a thorough evaluation of existing tools and the feasibility of their reuse be done in the context of all tool evaluation activities. In some cases, existing tools can be form-fitted to the Data Warehouse; in other cases, the customer organization may need to be convinced that new tools would better serve their needs.

It may even be feasible that this series of activities is skipped altogether, if the organization is insistent that particular tools be used (no room for negotiation), or if tools have already been assessed and selected in anticipation of the Data Warehouse project.

Tools may be categorized according to the following data, technical, application, or support functions:

- Source Data Extraction and Transformation

- Data Cleansing

- Data Load

- Data Refresh

- Data Access

- Security Enforcement

- Version Control/Configuration Management

- Backup and Recovery

- Disaster Recovery

- Performance Monitoring

- Database Management

- Platform

- Data Modeling

- Metadata Management

BENEFITS OF DATA WAREHOUSE METHODS

With data warehousing, you can provide a common data model for different interest areas regardless of data's source. In this way, it becomes easier to report and analyze information.

Find out inconsistencies and resolved before loading of information in data warehousing, this makes the reporting and analyzing process simpler.

The best part of data warehousing is that the information is under the control of users, so that in case the system gets purged over time, information can be easily and safely stored for longer time period.

Because of being different from operational systems, a data warehouse helps in retrieving data without slowing down the operational system.

Data warehousing enhances the operational business applications values and customer relationship management systems.

Data warehousing give a way to proper functioning of support system applications like trend reports, exception reports and the actual performance analyzing reports.

Precisely, a data warehouse system proves to be helpful in providing collective information to all its users. It is mainly created to support different analysis, queries that need extensive searching on a larger scale.

Functions of Data Warehouse Tools and Utilities

The following are the functions of data warehouse tools and utilities −

Data Extraction − Involves gathering data from multiple heterogeneous sources.

Data Cleaning − Involves finding and correcting the errors in data.

Data Transformation − Involves converting the data from legacy format to warehouse format.

Data Loading − Involves sorting, summarizing, consolidating, checking integrity, and building indices and partitions.

Refreshing − Involves updating from data sources to warehouse.

Update-Driven Approach

This is an alternative to the traditional approach. Today's data warehouse systems follow update-driven approach rather than the traditional approach discussed earlier. In update-driven approach, the information from multiple heterogeneous sources are integrated in advance and are stored in a warehouse. This information is available for direct querying and analysis.**Micro Focus Vertica**:

Micro Focus Vertica: Micro Focus Vertica is developed to use in data warehouses and other big data workloads where speed, scalability, simplicity, and openness are crucial to the success of analytics. It is a self-monitored MPP database and offers scalability and flexibility that other tools don’t. It is used on commercial hardware, therefore we can scale the database as required. It is designed significantly in-database advanced analytics capabilities to improve query performance over traditional relational database systems and unverified open source offerings. For example, Vertica is a column-oriented relational database; therefore, it might not qualify as a NoSQL database. A NoSQL database is best outlined as being a non-relational, shared-nothing, horizontally scalable database while not ACID guarantees. Vertica differs from normal RDBMS within the approach that it stores data by grouping data at once on disk by column instead of by row, Vertica reads the columns documented by the query, rather than scanning the complete table as row-oriented databases should do. Vertica offers the foremost advanced unified analytical warehouse that allows the organization to stay up with the dimensions and complexness of huge amounts of data volumes. With Vertica, businesses can perform tasks like predictive maintenance, client remembrance, economic compliance and network optimization, and far more.

**Amazon DynamoDB:**

Amazon DynamoDB is a fully managed proprietary NoSQL data warehouse service that supports key-value and document data structures and is obtainable by Amazon.com as a part of the Amazon Web Services portfolio. DynamoDB has an identical data model and encompasses a completely different underlying implementation. A partition key value is used in DynamoDB as input to an enclosed hash function. The output from the hash function determines the partition within which the item is going to be kept. All items with identical partition key values are stored together, in sorted order by sort key value. It offers customers high availability, dependability, and progressive scalability, with no limits on dataset size or request output for a given table. DynamoDB is meant for OLTP use cases high-speed data access wherever you are operative on many records at a time. However, users even have a desire for OLAP access patterns massive, analytical queries over the complete dataset to search out common things, or a variety of orders by day, or different insights. DynamoDB is aligned with the values of Serverless applications: automatic scaling consistent with your application load, pay-per-what-you-use rating, simple to induce started with, and no servers to manage. This makes DynamoDB an awfully common selection for Serverless applications running in AWS.

**Process of data warehousing:-**

Data warehousing is a process of collecting, integrating, and organizing data from various sources into a centralized repository for analytical purposes. Data warehousing involves data cleaning, data integration, and data consolidation. Data warehousing supports analytical reporting, structured and/or ad hoc queries, and decision making. Data warehousing also helps in improving data quality, faster queries, better business analytics, and historical insight.

Some of the basic concepts of data warehousing are:

- Data Warehouse: A data warehouse is a centralized storage system that allows for the storing, analyzing, and interpreting of data in order to facilitate better decision-making. A data warehouse is constructed by integrating data from multiple heterogeneous sources that support analytical reporting, structured and/or ad hoc queries, and decision making¹²³.

- Data Source: A data source is any system or application that produces or contains data that can be used for data warehousing. Data sources can be transactional databases, operational systems, external sources, files, etc¹².

- Data Integration: Data integration is the process of combining data from different sources into a consistent and unified view. Data integration involves data extraction, transformation, and loading (ETL) processes that move data from source systems to the data warehouse¹²⁴.

- Data Quality: Data quality is the degree to which data meets the expectations and requirements of the data consumers. Data quality involves ensuring the accuracy, completeness, consistency, timeliness, validity, and uniqueness of the data in the data warehouse¹⁴.

- Data Model: A data model is a logical representation of the structure and meaning of the data in the data warehouse. A data model defines the entities, attributes, relationships, constraints, and business rules of the data. A data model can be conceptual, logical, or physical¹⁴.

- Data Mart: A data mart is a subset of a data warehouse that is tailored to meet the specific needs of a particular business unit or department. A data mart can be independent or dependent on the data warehouse. A data mart can have its own data model, ETL processes, and analytical tools¹²⁴.

from \_\_future\_\_ import print\_function

from datetime import date, datetime, timedelta

import mysql.connector

cnx = mysql.connector.connect(user='scott', database='employees')

cursor = cnx.cursor()

tomorrow = datetime.now().date() + timedelta(days=1)

add\_employee = ("INSERT INTO employees "

"(first\_name, last\_name, hire\_date, gender, birth\_date) "

"VALUES (%s, %s, %s, %s, %s)")

add\_salary = ("INSERT INTO salaries "

"(emp\_no, salary, from\_date, to\_date) "

"VALUES (%(emp\_no)s, %(salary)s, %(from\_date)s, %(to\_date)s)")

data\_employee = ('Geert', 'Vanderkelen', tomorrow, 'M', date(1977, 6, 14))

# Insert new employee

cursor.execute(add\_employee, data\_employee)

emp\_no = cursor.lastrowid

# Insert salary information

data\_salary = {

'emp\_no': emp\_no,

'salary': 50000,

'from\_date': tomorrow,

'to\_date': date(9999, 1, 1),

}

cursor.execute(add\_salary, data\_salary)

# Make sure data is committed to the database

cnx.commit()

cursor.close()

cnx.close()

- Online Analytical Processing (OLAP): OLAP is a technique for analyzing multidimensional data using various operations such as slice and dice, drill down, roll up, pivot, etc. OLAP enables users to interactively explore and manipulate large amounts of data from different perspectives and levels of detail¹²⁴.

- Data Mining: Data mining is the process of discovering hidden patterns and trends from large volumes of data using various techniques such as classification, clustering, association analysis, regression analysis, etc. Data mining helps in extracting useful information and knowledge from the data warehouse for decision making¹²⁴.

(4) Introduction to Data Warehousing: Definition, Concept, and Techniques. https://www.digitalvidya.com/blog/introduction-to-data-warehousing/.

**Background**   
A Database Management System (DBMS) stores data in the form of tables, uses ER model and the goal is [ACID properties](https://www.geeksforgeeks.org/acid-properties-in-dbms/). For example, a DBMS of college has tables for students, faculty, etc.

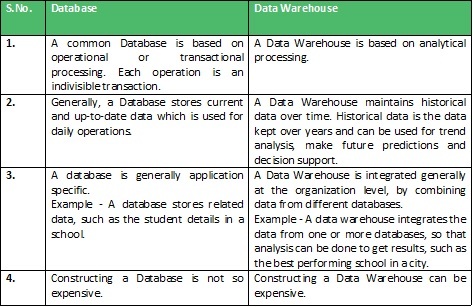
A **Data Warehouse**is separate from DBMS, it stores a huge amount of data, which is typically collected from multiple heterogeneous sources like files, DBMS, etc. The goal is to produce statistical results that may help in decision makings. For example, a college might want to see quick different results, like how the placement of CS students has improved over the last 10 years, in terms of salaries, counts, etc.

**Need for Data Warehouse**  
An ordinary Database can store MBs to GBs of data and that too for a specific purpose. For storing data of TB size, the storage shifted to Data Warehouse. Besides this, a transactional database doesn’t offer itself to analytics. To effectively perform analytics, an organization keeps a central Data Warehouse to closely study its business by organizing, understanding, and using its historic data for taking strategic decisions and analyzing trends.

**Benefits of Data Warehouse:**

1. **Better business analytics:**Data warehouse plays an important role in every business to store and analysis of all the past data and records of the company. which can further increase the understanding or analysis of data to the company.
2. **Faster Queries:**Data warehouse is designed to handle large queries that’s why it runs queries faster than the database.
3. **Improved data Quality:**In the data warehouse the data you gathered from different sources is being stored and analyzed it does not interfere with or add data by itself so your quality of data is maintained and if you get any issue regarding data quality then the data warehouse team will solve this.
4. **Historical Insight:**The warehouse stores all your historical data which contains details about the business so that one can analyze it at any time and extract insights from it

**Data Warehouse vs DBMS** 

[](https://www.geeksforgeeks.org/difference-between-database-system-and-data-warehouse/)

**Example Applications of Data Warehousing**   
Data Warehousing can be applied anywhere where we have a huge amount of data and we want to see statistical results that help in decision making.

* **Social Media Websites:** The social networking websites like Facebook, Twitter, Linkedin, etc. are based on analyzing large data sets. These sites gather data related to members, groups, locations, etc., and store it in a single central repository. Being a large amount of data, Data Warehouse is needed for implementing the same.
* **Banking:**Most of the banks these days use warehouses to see the spending patterns of account/cardholders. They use this to provide them with special offers, deals, etc.
* **Government:** Government uses a data warehouse to store and analyze tax payments which are used to detect tax thefts.

|  |
| --- |
| # importing required libraries  **import** mysql.connector    dataBase **=** mysql.connector.connect(    host **=**"localhost",    user **=**"user",    passwd **=**"password",    database **=** "gfg"  )    # preparing a cursor object  cursorObject **=** dataBase.cursor()    query **=** "SELECT \* FROM STUDENT LIMIT 2 OFFSET 1"  cursorObject.execute(query)    myresult **=** cursorObject.fetchall()    **for** x **in** myresult:      print(x)    # disconnecting from server  dataBase.close() |

**Output:**

**('Nikhil', 'CSE', 98, 'A', 18)**

**('Nisha', 'CSE', 99, 'A' , 18)**

### PostgreSQL:

### It is an extremely stable database management system, backed by over twenty years of community development that has contributed to its high levels of resilience, integrity, and correctness. PostgreSQL is employed because the primary data store or data warehouse for several web, mobile, geospatial, and analytics applications. SQL Server is a database management system that is especially used for e-commerce and providing different data warehousing solutions. PostgreSQL is a sophisticated version of SQL that provides support to various functions of SQL like foreign keys, subqueries, triggers, and other user-defined varieties and functions. Postgres is a feature-rich database that can handle advanced complicated queries and big databases. MySQL is a less complicated database that is comparatively simple to line up and manage, fast, reliable, and well-understood. PostgreSQL performs well in OLTP/OLAP systems once read/write speeds are needed and intensive data analysis is required. PostgreSQL additionally works well with Business Intelligence applications however is best suited to data warehousing and data analysis applications that require quick read/write operations speed.

### Amazon S3:

### Amazon S3 is object storage engineered to store and retrieves any quantity of data from any place. It is an easy storage service that provides business-leading sturdiness, accessibility, performance, security, and nearly unlimited scalability at very low prices. AWS S3 is a key-value store, one of the foremost classes of NoSQL databases used for accumulating voluminous, mutating, unstructured, or semi-structured data. Features like metadata support, prefixes, and object tags enable users to arrange data consistent with their desires. The S3 object storage cloud service offers subscriber access to similar systems that Amazon uses to run its own websites. Amazon S3 is object storage capable of storing massive objects, up to 5TB in size. S3 allows customers to access, store and download practically any file or object that’s up 5 TB in size with the biggest single upload capped at 5 gigabytes (GB). S3 is often used for storing pictures, videos, logs, and alternative varieties of files. There’s no limit on the number of objects that may be stored in an S3 bucket. Every object in S3 includes a URL that might be used to download the object. S3 provides unlimited storage at a comparatively low cost than DynamoDB; however, scan operations are abundant slower than DynamoDB, although it can perform HTTP queries for the same. Amazon S3 sets the quality once it involves business cloud storage whereas simple use is not a part of that standard but, top-quality security, extreme flexibility, and total integration are.

### Teradata:

### Teradata is one of the admired Relational Database Management systems. It is appropriate for building big data warehousing applications. Teradata accomplishes this with the help of parallelism. Teradata database system is built on Massively Parallel Processing (MPP) architecture. The Teradata system primarily splits the work among its processes and runs them in parallel to reduce workload and also makes sure that the task is accomplished quickly and successfully. Teradata provides real-time, intelligent answers by processing 100% of the appropriate data, despite the volume of the query. Teradata fulfills all the requirements in terms of Integration or ETL with the capabilities of consuming, analyzing and managing the data. Data in an exceeding data warehouse is organized to support analysis instead of processing real-time transactions as in online transaction processing systems (OLTP). Although it is geared towards OLAP. It’s one of the most powerful data integration and analytics database solutions within the market. Teradata is employed or has been utilized in past by most business enterprises. It processes enormous amounts of data very easily. It’s simple to navigate and a sensible graphical user interface helps Business users use it with basic training and query knowledge, however, big data processing is a challenge because of its existing architectures. Features :

**Centralized Data Repository:** Data warehousing provides a centralized repository for all enterprise data from various sources, such as transactional databases, operational systems, and external sources. This enables organizations to have a comprehensive view of their data, which can help in making informed business decisions.

**Data Integration:**Data warehousing integrates data from different sources into a single, unified view, which can help in eliminating data silos and reducing data inconsistencies.

**Historical Data Storage:** Data warehousing stores historical data, which enables organizations to analyze data trends over time. This can help in identifying patterns and anomalies in the data, which can be used to improve business performance.

**Query and Analysis:**Data warehousing provides powerful query and analysis capabilities that enable users to explore and analyze data in different ways. This can help in identifying patterns and trends, and can also help in making informed business decisions.

**Data Transformation:**Data warehousing includes a process of data transformation, which involves cleaning, filtering, and formatting data from various sources to make it consistent and usable. This can help in improving data quality and reducing data inconsistencies.

**Data Mining:** Data warehousing provides data mining capabilities, which enable organizations to discover hidden patterns and relationships in their data. This can help in identifying new opportunities, predicting future trends, and mitigating risks.

**Data Security:**Data warehousing provides robust data security features, such as access controls, data encryption, and data backups, which ensure that the data is secure and protected from unauthorized access.

### Advantages:

**Improved data quality:**Data warehousing can help improve data quality by consolidating data from various sources into a single, consistent view.

**Faster access to information:** Data warehousing enables quick access to information, allowing businesses to make better, more informed decisions faster.

**Better decision-making:** With a data warehouse, businesses can analyze data and gain insights into trends and patterns that can inform better decision-making.

**Reduced data redundancy:** By consolidating data from various sources, data warehousing can reduce data redundancy and inconsistencies.

**Scalability:**Data warehousing is highly scalable and can handle large amounts of data from different sources.

### Disadvantages:

**Cost:**Building a data warehouse can be expensive, requiring significant investments in hardware, software, and personnel.

**Complexity:**Data warehousing can be complex, and businesses may need to hire specialized personnel to manage the system.

**Time-consuming:**Building a data warehouse can take a significant amount of time, requiring businesses to be patient and committed to the process.

**Data integration challenges:**Data from different sources can be challenging to integrate, requiring significant effort to ensure consistency and accuracy.

**Data security:**Data warehousing can pose data security risks, and businesses must take measures to protect sensitive data from unauthorized access or breaches.

There can be many more applications in different sectors like E-Commerce, telecommunications, Transportation Services, Marketing and Distribution, Healthcare, and Retail.