

Contents

Introduction – What is data warehousing?	2
Overview	2
Data Ware house architecture:	3
Advantages of DW	4
Disadvantages	4
Why is data warehousing important?	4
Data Warehouse Design Methodologies	5
Inmon’s top-down approach	5
Kimball’s dimensional design approach	6
Requirements gathering –	8
Business Objectives	8
Current Problems	8
Desired Future state	8
Success Criteria (What does “winning” look like?)	9
Identify Users	9
Additional questions:	9
Different ways how to build Data Warehouse (DW)	10
1. Creating a DW using a SQL Merge Statement	10
2. Create a DW using Change Data Capture (CDC)	10
Change Data Capture Vs. SQL Merge Statement	19
Some key features of data warehousing:	20

Introduction – What is data warehousing?

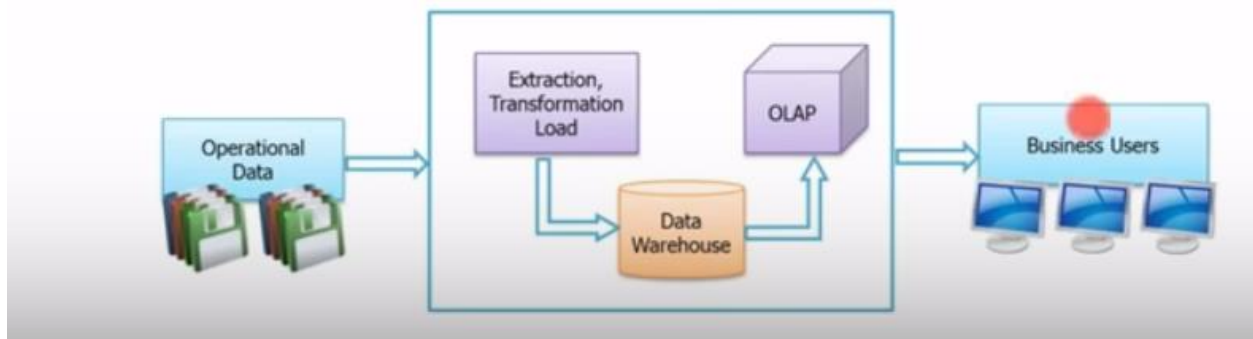
Overview – Data warehouse is an information system that contains historical and commutative data from single or multiple sources. It simplifies reporting and analysis process of the organization.

It is a single version of truth for any company for decision making and forecasting. It is a technique for collecting and managing data from varied sources to provide meaningful business insights. It is a blend of technologies and components which allows the strategic use of data.

In the world of computing, data warehouse is defined as a system that is used for data analysis and reporting. Also known as enterprise data warehouse, this system combines methodologies, user management system, data manipulation system and technologies for generating insights about the company. Considered as repositories of data from multiple sources, data warehouse stores both current and historical data. They are then used to create analytical reports that can either be annual or quarterly in nature.

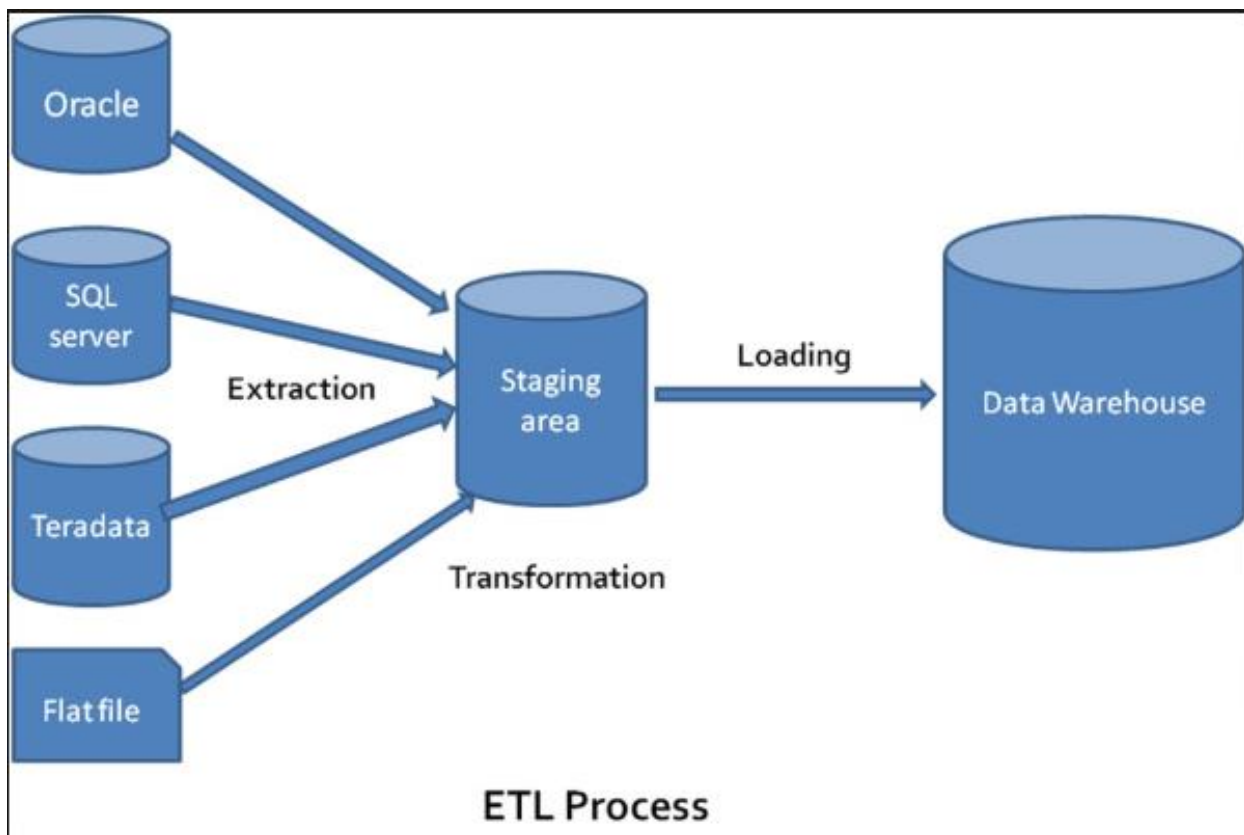
It is electronic storage of a large amount of information by a business which is designed for query and analysis instead of transaction processing. It is a process of transforming data into information and making it available to users in a timely manner to make a difference.

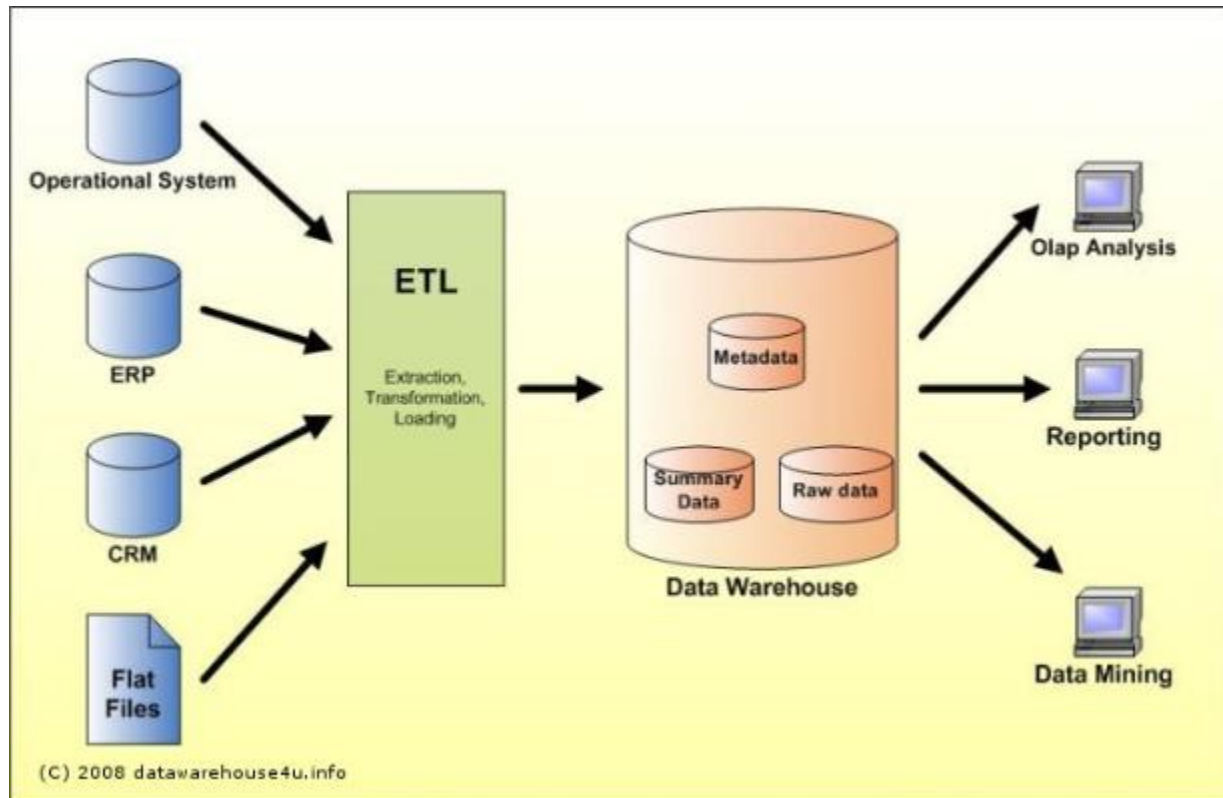
Basically, a data warehouse is a central location where consolidated data from multiple locations (databases) are stored. A Data warehouse is maintained separately from the organization's database and the end users access it whenever any information needed.



Note that Data warehouses are not loaded every time new data is added to the database.

Data Ware house architecture:





Advantages of DW

- Strategic questions can be answered by studying trends
- Data warehousing is faster and more accurate
- Speedy Data Retrieving
- Error Identification & Correction
- Easy Integration

Disadvantages

- Time Consuming Preparation
- Difficulty in Compatibility
- Maintenance costs
- Limited Use Due to Confidential Information

Why is data warehousing important?

Data warehousing is an increasingly important business intelligence tool, allowing organizations to:

1. **Ensure consistency.** Data warehouses are programmed to apply a uniform format to all collected data, which makes it easier for corporate decision-makers to analyze and share data insights with their colleagues around the globe. Standardizing data from different sources also reduces the risk of error in interpretation and improves overall accuracy.
2. A data warehouse functions more like a **curated library** than temporary storage space
3. **Make better business decisions.** Successful business leaders develop data-driven strategies and rarely make decisions without consulting the facts. Data warehousing improves the speed and efficiency of accessing different data sets and makes it easier for corporate decision-makers to derive insights that will guide the business and marketing strategies that set them apart from their competitors.
4. **Improve their bottom line.** Data warehouse platforms allow business leaders to quickly access their organization's historical activities and evaluate initiatives that have been successful — or unsuccessful — in the past. This allows executives to see where they can adjust their strategy to decrease costs, maximize efficiency and increase sales to improve their bottom line.

Data Warehouse Design Methodologies

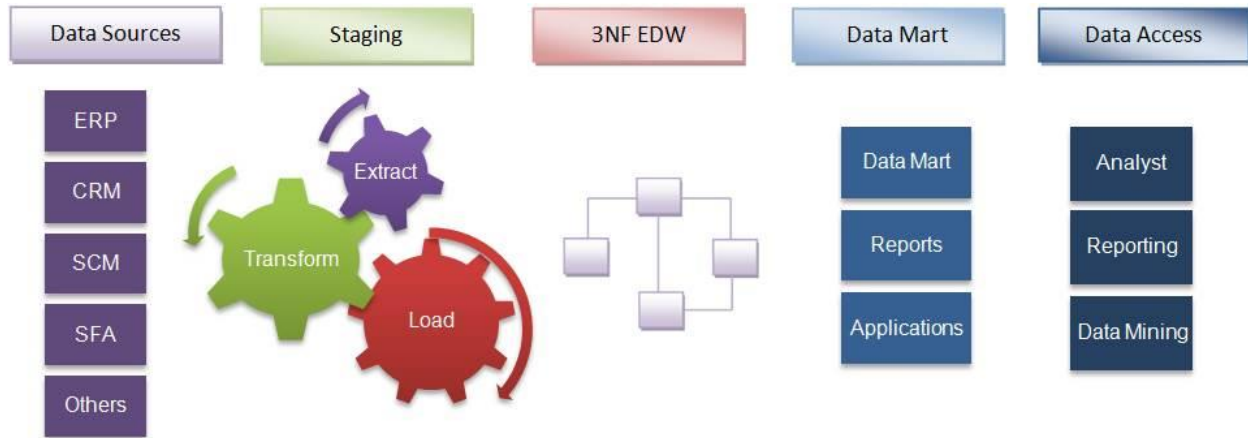
When it comes to designing a data warehouse for your business, the two most commonly discussed methods are the approaches introduced by **Bill Inmon** and **Ralph Kimball**

Inmon's top-down approach

In this approach (the top-down design), a normalized data model is designed first, then the dimensional data marts, which contain data required for specific business processes or specific departments, are created from the data warehouse.

Inmon defines a data warehouse as a centralized repository for the entire enterprise. A data warehouse stores the “atomic” data at the lowest level of detail. Dimensional data marts are created only after the complete data warehouse has been created. Thus, the data warehouse is at the center of the

corporate information factory (CIF), which provides a logical framework for delivering business intelligence.



This approach of DW design is

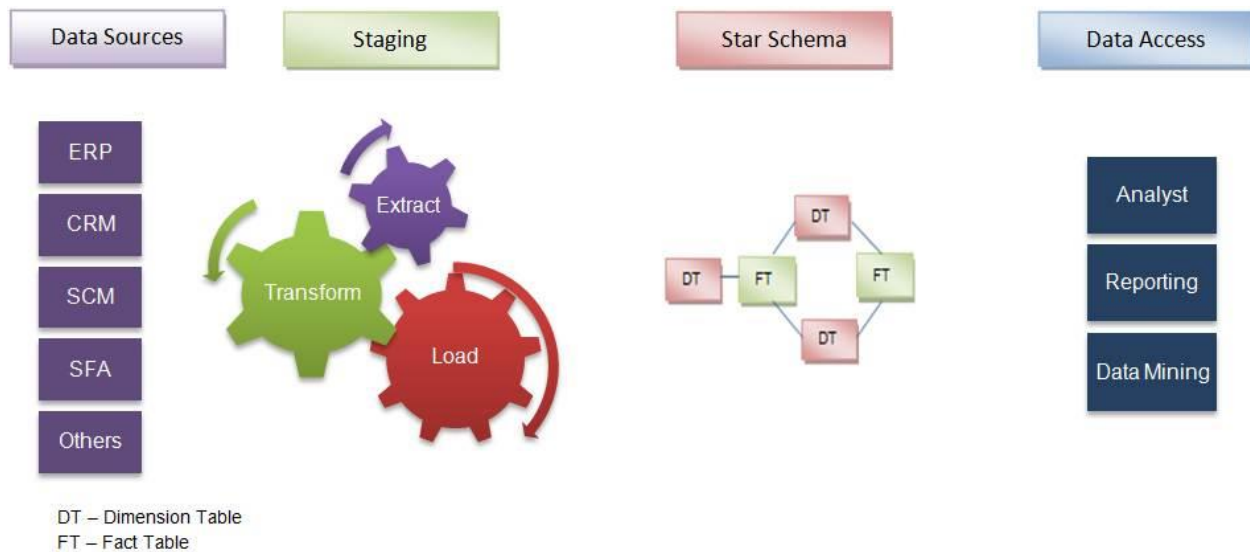
1. **Subject-oriented:** The data in the data warehouse is organized so that all the data elements relating to the same real-world event or object are linked together.
2. **Time-variant:** The changes to the data in the database are tracked and recorded so that reports can be produced showing changes over time.
3. **Non-volatile:** Data in the data warehouse is never overwritten or deleted. Once committed, the data is static, read-only and retained for future reporting.
4. **Integrated:** The database contains data from most or all of an organization's operational applications, and that this data is made consistent.

Kimball's dimensional design approach

In this approach (the bottom-up design), the data marts facilitating reports and analysis are created first; these are then combined to create a broad data

warehouse. Kimball defines data warehouse as “a copy of transaction data specifically structured for query and analysis”.

Kimball’s data warehousing architecture is also known as data warehouse bus (BUS). Dimensional modelling focuses on ease of end-user accessibility and provides a high level of performance to the data warehouse.



Difference between Inmon and Kimball

	Inmon	Kimball
Building Data warehouse	Time Consuming	Takes lesser time
Maintenance	Easy	Difficult, often redundant and subject to revisions
Cost	High initial cost; Subsequent project development costs will be much lower	Low initial cost; Each subsequent phase will cost almost the same
Time	Longer start-up time	Shorter time for initial set-up
Skill Requirement	Specialist team	Generalist team
Data Integration requirements	Enterprise-wide	Individual business areas

Requirements gathering –

What to ask: -

Business Objectives

1. What is your business process?
2. What are your goals in developing this system?
3. What decisions are you trying to make?
4. What are the most important, strategic questions that need to be answered?
5. Who are the key stakeholders and users? Do their goals differ? If so, how?
6. How do the system goals map to business goals?
7. What is the most important business goal of the system?
8. How will the system change the way you are doing things now?
9. How will the system help you be more efficient?
10. What are the system deliverables?
11. What will the new system accomplish that is not currently accomplished manually or with other systems?

Current Problems

1. What are the current problems you are facing today without the system?
2. What problems should this system solve?
3. What do you have to do manually that you would like to automate?
4. What performance problems need to change?
5. What functional limitations would you like to change?
6. Which reports do you currently use? What data on the report is important? How do you use the information?
7. Where are there specific bottlenecks to getting at information?
8. How do you analyze the information you currently receive? What type of data is used? How do you currently get the data? How often do you get new data?
9. What type of ad hoc analysis do you typically perform? Who requests ad hoc information? What do you do with the information?
10. What will the new system do?

Desired Future state

1. What business requirements will this system address?
2. What information do you need from this system that you don't have now?
3. Where is any of this data currently captured in another corporate system?

4. How would you like to see this information?
5. What functionality do you need from the system?
6. What data and/or functionality is shared by other (many) business areas?
7. If the reports were dynamic, what would they do differently?
8. How much historical information is required?
9. Do you want to track any data changes like, change in marital status, change in address or phone no.

Success Criteria (What does “winning” look like?)

1. What is most important for success of the application?
2. What do we need to accomplish to make this project successful?
3. What do we need to change to make this project successful?
4. What buy-in do we need?
5. What critical elements such as budget, resource allocation, or support are we lacking?

Identify Users

1. Who will be using the system?
2. What are the titles and roles of the people who will use the system?
3. What are their levels of expertise?

Additional questions:

- What is the format of data we are getting? - audio, video, images, different file formats?
Size of the data
- What kind of data are they tracking
- What kind of report you want
- What are the current reports they are using
- What changes do they want to see?
- Security- who will have access to the data.
- Who can run the reports, access the parameters?
- What attributes you want to capture in the history?
- Business season
- Business Calendar year
- Inter-relationship between the performance/data of different departments.
- Do you want to break it down to data marts?
- Will it be department oriented

- How long do you want to data to retained? What would be the archival mechanism
- What is the frequency of the reports- daily, weekly, monthly,
- How often will the data warehouse be refreshed?
- How big is the DW going to be-How much storage space is needed
- Server-Database Credentials
- Disaster recovery mechanism in case of unforeseen data loss.

Different ways how to build Data Warehouse (DW)

1. Creating a DW using a SQL Merge Statement

In previous versions of SQL Server, we had to write separate statements to INSERT, UPDATE, or DELETE data based on certain conditions, but now, using MERGE statement we can include the logic of such data modifications in one statement that even checks when the data is matched then just update it and when un-matched, then insert it.

- MERGE is a new feature in 2008 that provides an efficient way to perform multiple DML operations.
- You can perform insert, update, or delete operations in a single statement
- It allows you to join a data source with a target object and then perform multiple actions against the target objects based on the result of that join
- Link for Better understanding:
<http://www.made2mentor.com/2013/05/writing-t-sql-merge-statements-the-right-way/>

2. Create a DW using Change Data Capture (CDC)

- A process to capture insert, update, and delete activity from a data source.
- Most accurate technique to detect a **Business Event**.
 - **Business Event** – Individual actions performed by people or an organization during the execution of business process.

Why Does Capturing Business Events Matter?

- Most accurate technique to build consistent business metrics.
- Important business metrics are many times lost due to poor Operational Application Database Design.
- Many Operational Applications have a lack of analytical foresight.

What Techniques Can Be Used to Incorporate CDC For A Data Warehouse?

- Database Triggers.
 - o Detection of insert, update, and delete events.
 - o Supported by all major databases.
- Data Modification Stamping (also known as Change Tracking). Involves the addition of tracking columns to a table.
 - o Time Stamping.
 - o Status Indicator.
 - o Version Number.
- Database transaction log scanning, decode database transaction logs to detect database insert, update, and delete events.
 - o Examples:
 - Microsoft SQL Server Change Data Capture.
 - IBM infosphere Change Data Capture.
 - Oracle Goldengate.

Example of CDC Detection Point

- Sales Person becomes customer by purchasing a product.
 - o Sales Person changes the Person Type from “**Prospect**” to “**Customer**” in the Sales Application.
 - o This changes the Person Code Type from “**P**” to “**C**” on the “**Person**” table.
 - o This update action overwrites the information about a person prior “**Prospect**” status.

What Valuable Information Was Lost?

- You no longer know the time it took to convert this person into a customer.
- The time this conversion takes may be key to increasing sales, especially if this could be correlated to other business events.

So, What to Do?

- Build an infrastructure and a process to capture business events.
 - o Choose a method to detect business events on your database. (**Database Triggers Change columns, or transaction log scanning**)
 - o Build a table to store your business events.
 - o Use the business events as the seeds to populate your BI fact table.

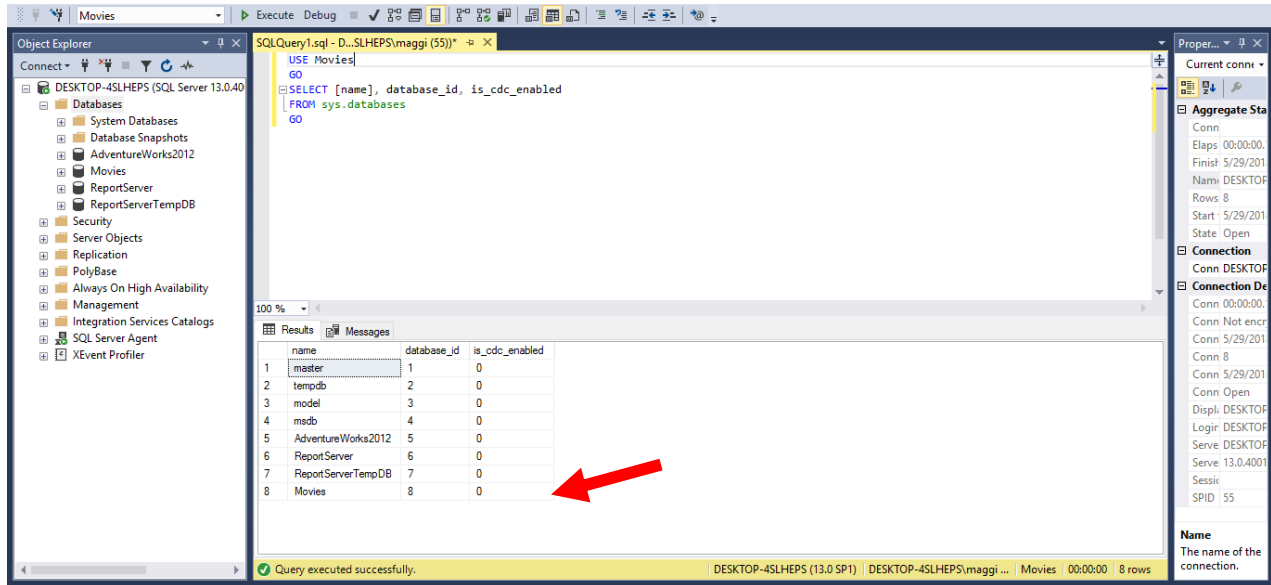
Event ID	Business Event Type ID	Event Date Stamp	Table	Column	Key Value	Data Value
1	1	2012-07-05 12:15 PM	Person	Person_Type_Codes	102	P
2	1	2012-07-07 09:23AM	Person	Person_Type_Codes	102	C

Rewards Achieved

- A central location to drive the population of all fact and Slowly Changing Dimension tables.
- Elimination for the need to flush and reload fact and dimension tables.
- Potential to process data in real time.
- Ability to capture business events for status columns.
- Capability to retain history of business events over time.

Enabling Change Data Capture In Database

- This query will return the entire database name along with a column that shows whether CDC is enabled.

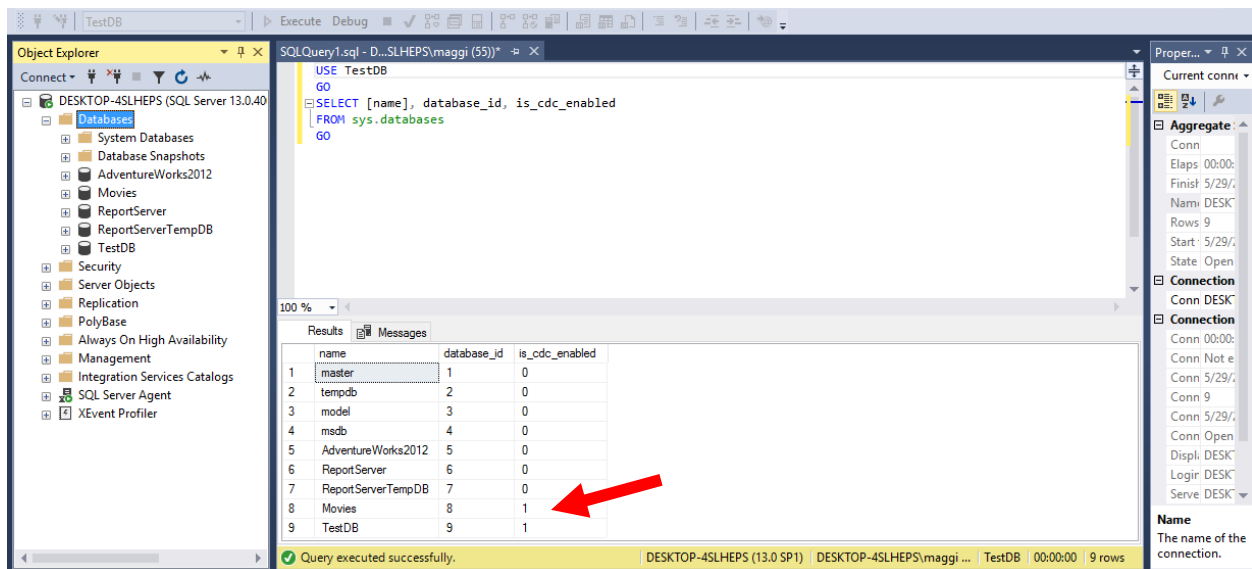


SQLQuery1.sql - D:\SLHEPS\maggi (55)*

```
USE [Movies]
GO
SELECT [name], database_id, is_cdc_enabled
FROM sys.databases
GO
```

	name	database_id	is_cdc_enabled
1	master	1	0
2	tempdb	2	0
3	model	3	0
4	msdb	4	0
5	AdventureWorks2012	5	0
6	ReportServer	6	0
7	ReportServerTempDB	7	0
8	Movies	8	0

Query executed successfully. DESKTOP-4SLHEPS (13.0 SP1) | DESKTOP-4SLHEPS\maggi ... | Movies | 00:00:00 | 8 rows



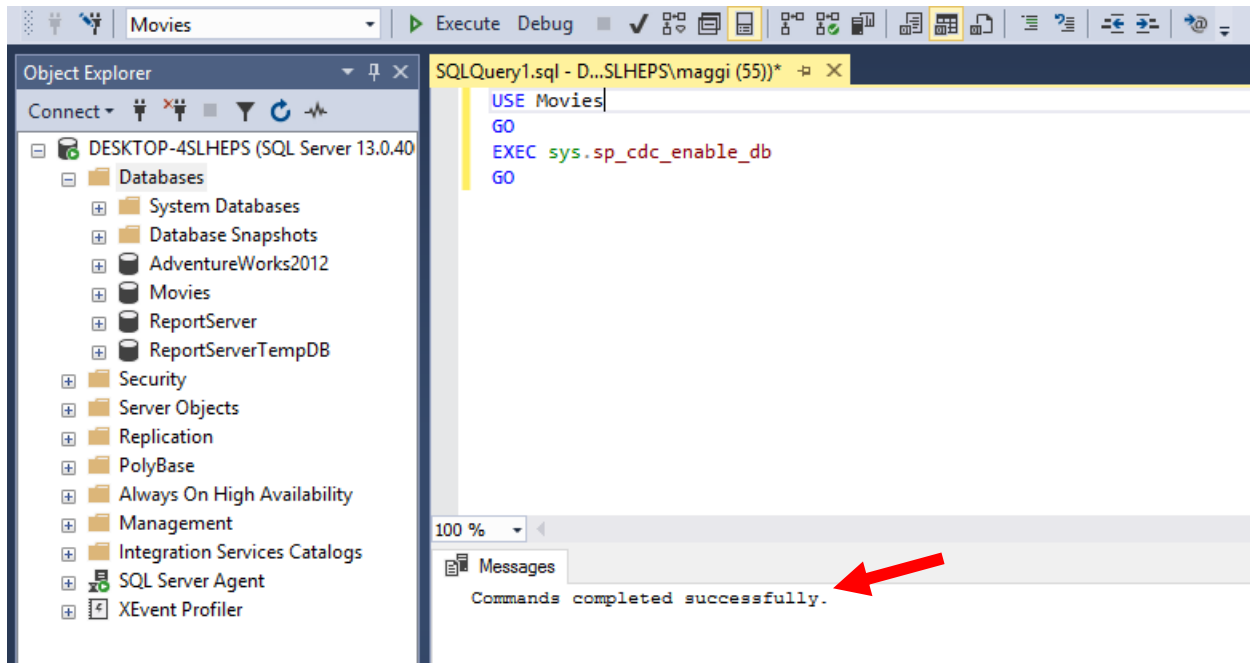
SQLQuery1.sql - D:\SLHEPS\maggi (55)*

```
USE [TestDB]
GO
SELECT [name], database_id, is_cdc_enabled
FROM sys.databases
GO
```

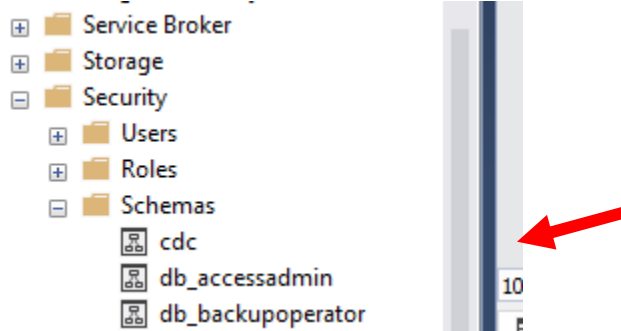
	name	database_id	is_cdc_enabled
1	master	1	0
2	tempdb	2	0
3	model	3	0
4	msdb	4	0
5	AdventureWorks2012	5	0
6	ReportServer	6	0
7	ReportServerTempDB	7	0
8	Movies	8	1
9	TestDB	9	1

Query executed successfully. DESKTOP-4SLHEPS (13.0 SP1) | DESKTOP-4SLHEPS\maggi ... | TestDB | 00:00:00 | 9 rows

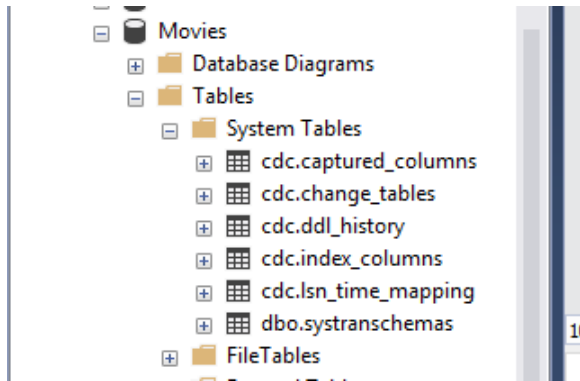
- You can run this stored procedure in the context of each database to enable CDC at database level. (The following script will enable CDC in **Movies** database.)



- Additionally, in the database **Movies**, you will see that a schema with the name 'cdc' has now been created.



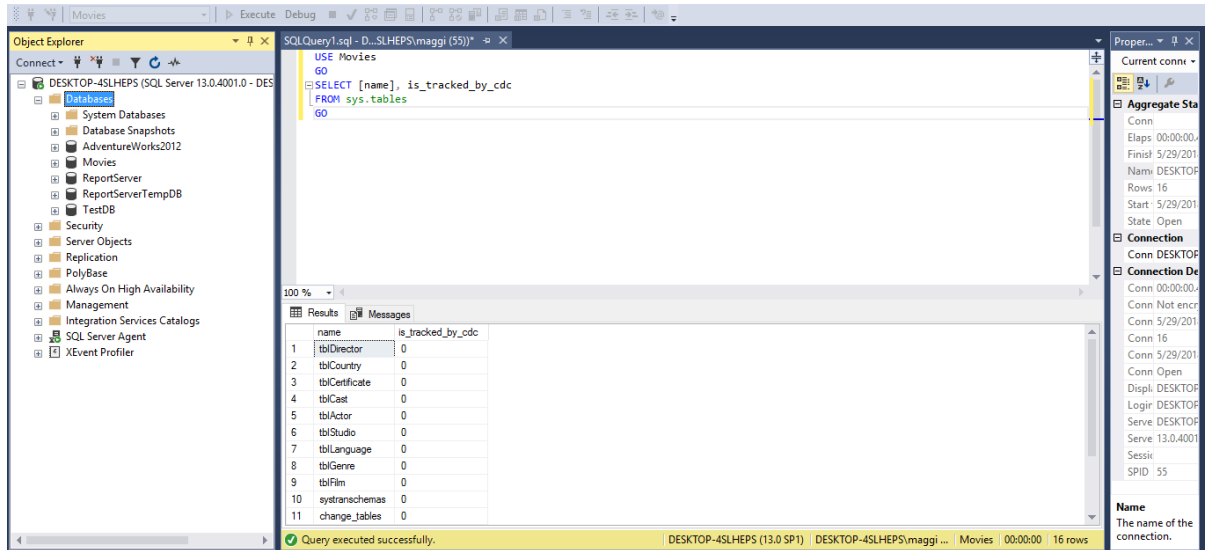
- Some System Tables will have been created within the **Movies** database as part of the CDC schema.



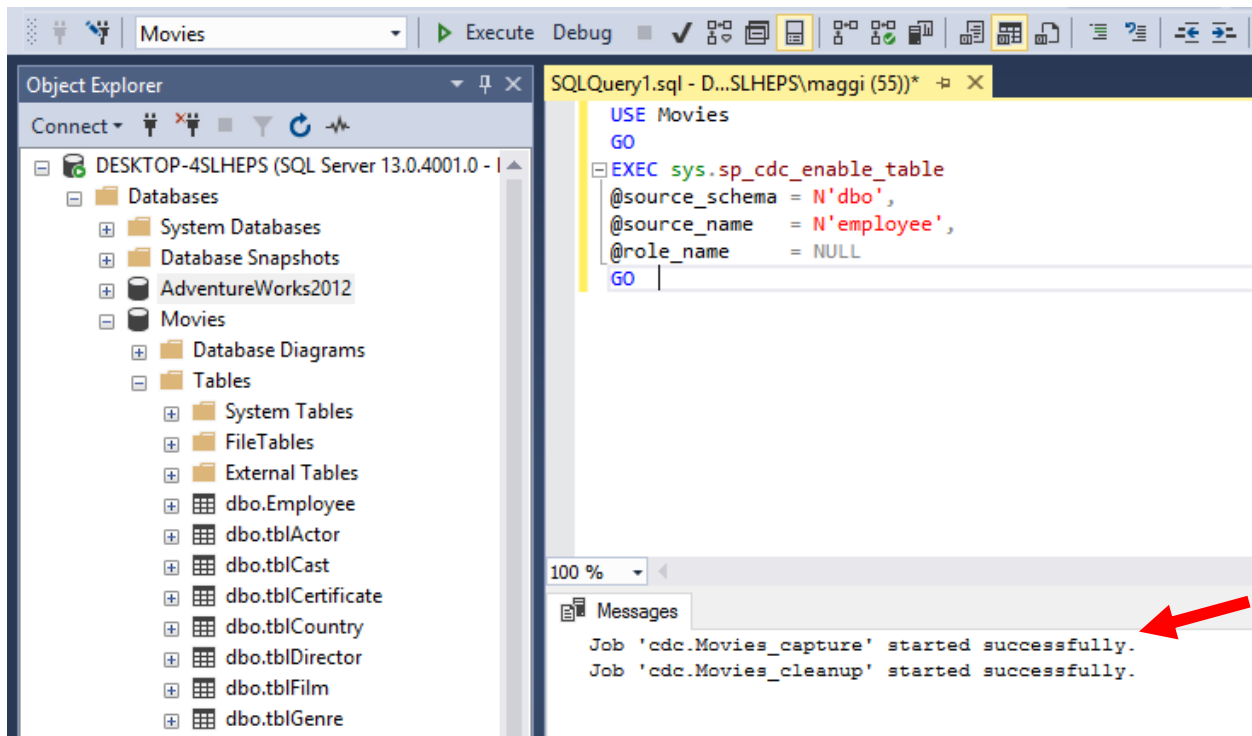
- The table which have been created are listed here.
 - o **cdc.captured_columns** – This table returns result for list of captured column.
 - o **cdc. change_tables** – This table returns list of all the tables which are enabled for capture.
 - o **cdc.ddl_history** – This table contains history of all the DDL changes since capture data enabled.
 - o **cdc.index_columns** – This table contains indexes associated with change table.
 - o **cdc.lsn_time_mapping** – This table maps LSN number (for which we will learn later) and time.

Enabling Change Data Capture on one or more Database Tables

- The CDC feature can be applied at the table-level to any database for which CDC is enabled. It has to be enabled for any table which needs to be tracked. First run following query to show which tables of database have already been enabled for CDC.

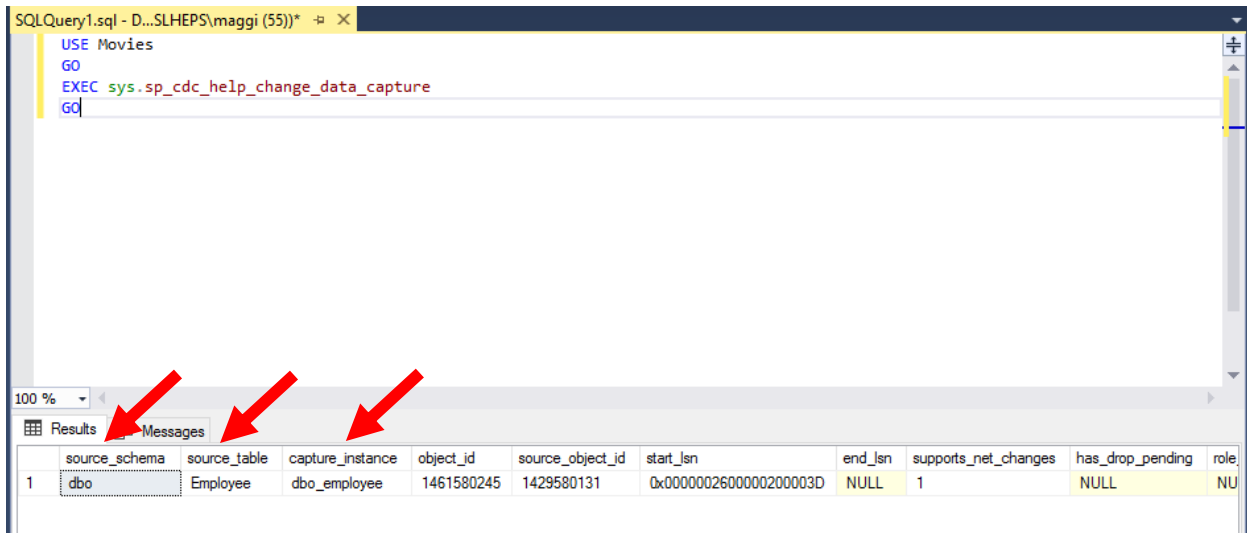


- Following script will enable CDC on **dbo.Employee** table.



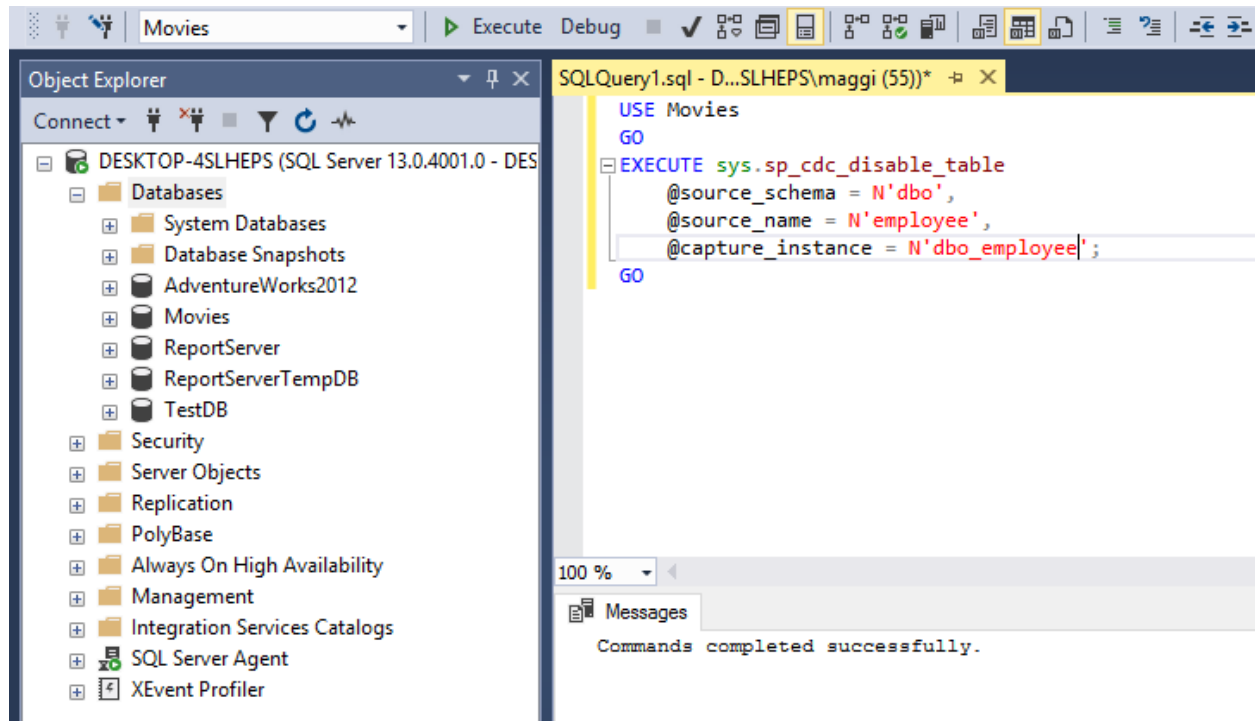
Disabling Change Data Capture feature on Database Tables

- For dropping any tracking of any table, we need three values the Source Schema, the Source Table name, and the Capture Instance. It is very easy to get schema and table name. In our case, the schema is **dbo** and table name is **Employee**, however we do not know the name of the Capture Instance. We can retrieve it very easily by running following T-SQL Query.
- This will return a result which contains all the three required information for disabling CDC on table.

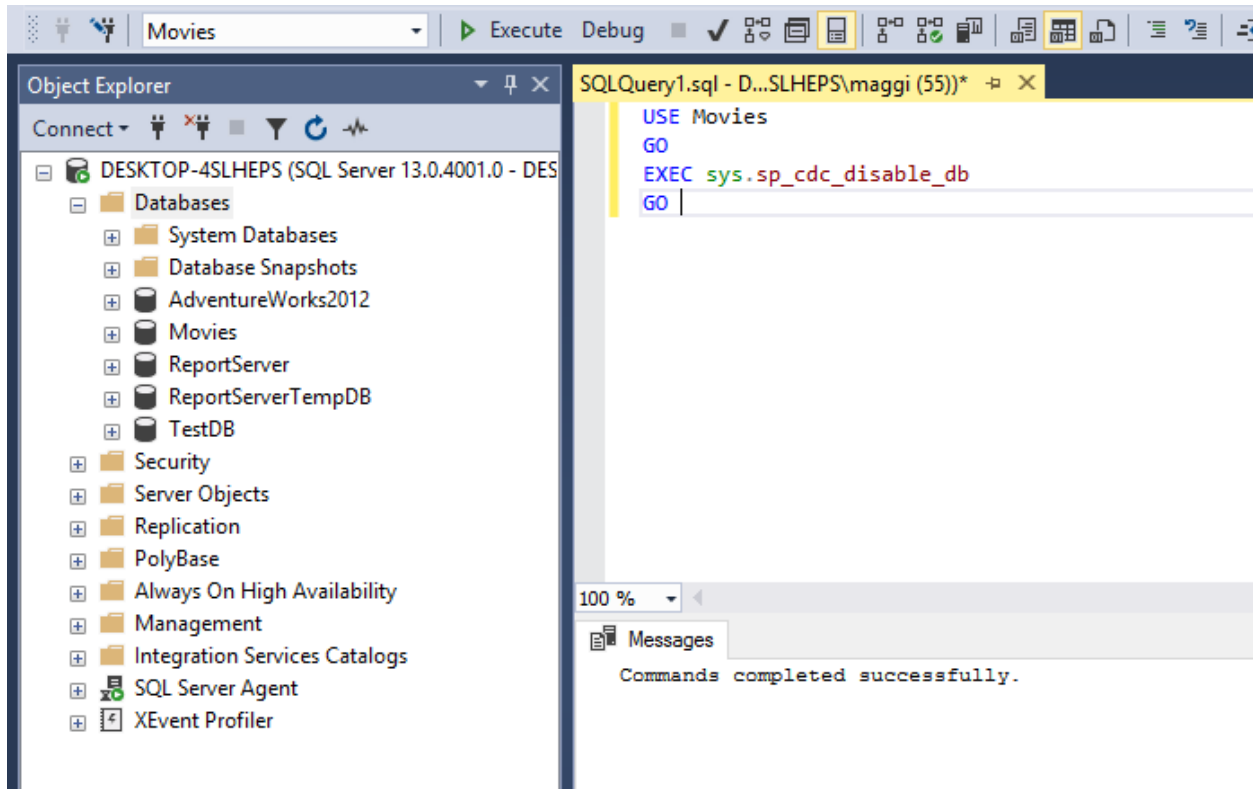


	source_schema	source_table	capture_instance	object_id	source_object_id	start_lsn	end_lsn	supports_net_changes	has_drop_pending	role
1	dbo	Employee	dbo_employee	1461580245	1429580131	0x000000026000002000003D	NULL	1	NULL	NU

- The Stored Procedure **sys.sp_cdc_help_change_data_capture** provides lots of other useful information as well. Once we have name of the capture instance, we can disable tracking of the table by running this T-SQL query.



Disabling Change Data Capture feature on a Database



On tutorial on how to use Insert, Update , & Delete Operation in Change Data Capture, Visit :- <https://www.red-gate.com/simple-talk/sql/learn-sql-server/introduction-to-change-data-capture-cdc-in-sql-server-2008/>

Change Data Capture Vs. SQL Merge Statement

- MERGE is simplified version to do all DML operations by means of single statement (INSERT/UPDATE/DELETE) based on key comparison. It can be between multiple objects and can even include derived tables as the source.
- CDC on the other hand is just a mechanism to track changes within single object itself like updates, inserts etc. happening on it and gives an idea on how various column values changed

Some key features of data warehousing:**It provides companies with comprehensive decision-making support**

As the core components of any company involves making plans and developing methodologies and techniques to achieve organizational goals, data warehouse can support great support to help them to do this. This is because data that is conceptualized and compiled in a proper manner, can go a long way in helping companies to strategies and create long term plans.

Data warehouse helps in subject orientation

A important feature of data warehouse is that it is oriented towards the subject. As data is gathered from numerous sources, data warehouse helps companies to use specific data that applies to their own field. This helps a company to gain insight into how data can be used in a manner, that all the sectors of the company are benefited in a proper manner. By helping a company handle specific areas like management or IT, data warehouse can help them grow in a strategic and comprehensive manner.

Data warehouse helps to integrate data

After data is compiled from various sources, data warehouse allows for data integration. This means that data is dynamic and applicable to various departments. Integration of data is therefore one of the most key features of data warehouse.

It allows for flexibility in time

As data is stored in a strategic manner, data has a specific time duration. This makes it easier for companies to access data for a particular time period. It is always better to have data structured in a time specific manner, because it can help companies to find loopholes in management and over all functioning on one hand and make effective comparison on the other hand.

Data warehouse keeps data safe and secure

Before the development of data warehouse, secondary storage was considered as the best way to save data. However, data warehouse supports integration, cohesiveness and multi-application of data, making them a more suitable choice. This is because data warehouse helps to preserve data for future use as well. As data in a warehouse is secure, data warehouse is one of the effective methods to store data for future use.

Data warehouse allows companies to store large volumes of data

Today the data available to companies is almost limitless. And data warehouse is more than capable of meeting this challenge as the size of the warehouse can be increased depending on the amount of data. Different organizations have different amounts of data that they would want to save for future use, so data warehouse is one of the perfect ways to meet that requirement in an effective manner.

Data warehouse is accurate and grounded

Data in a data warehouse is completely accurate and grounded, as it contains all techniques and theories. As a lot of companies, depend on data insights to take future decisions, this is an extremely important feature. If data is incorrect, it can affect the progress and growth of the company, as a number of technologies is involved in protecting data in warehouse, companies can be assured that the data they have is effective, discrete and multi-dimensional.

Data warehouse is the future of all companies, be it big or small

Since data warehouse was officially introduced in the year 2002, it has steadily grown in popularity and has become an integral part of many companies and brands. As many companies use data warehouse to preserve and gain insights about data, there are many advancements in this field by engineers that are making data warehouse more progressive and advanced. One of the most effective techniques to save substantial amounts of dynamic data, data

warehouse is something that all companies must consider for reaching the next stage of growth and development.