

# CH: 1

## Introduction to Data Warehousing

### 1. What is Data Warehouse?

#### Why Data Warehousing?

- Data explosion (blast) in data base management systems (DBMS).
  - Inefficient retrieval of required information
- Needs of Decision Support Systems (DSS) to facilitate decision making.
  - Extracting, cleaning, transforming, and filtering data from DBMS and provide efficient access to required information
- Data warehouse comes to free.

#### Who needs data warehouse?

- Decision makers who rely on mass amount of data
- Those who use customized, complex processes to obtain information from various data sources
- Those who want to use simple technology to access data
- Those who require systematic approach for decision.

#### Two major functions of data warehousing

- Extracting necessary information for decision making from heterogeneous data sources and stored in the data warehouse.
- Providing queries and decision analyses to users.

#### EXMAPLE:

#### Why Data Warehousing?

#### Typical DW Queries ,,

- What was the total revenue for Scotland in the third quarter of 2004?
- What was the total revenue for property sales for each type of property in Great Britain in 2003?
- What are the three most popular areas in each city for the renting of property in 2004 and how does this compare with the figures for the previous two years?
- What is the monthly revenue for property sales at each branch office, compared with rolling 12-monthly prior figures?
- What would be the effect on property sales in the different regions of Britain if legal costs went up by 3.5% and Government taxes went down by 1.5% for properties over £100,000?

- Which type of property sells for prices above the average selling price for properties in the main cities of Great Britain and how does this correlate to demographic data?
- What is the relationship between the total annual revenue generated by each branch office and the total number of sales staff assigned to each branch office?

**A data warehouse is a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision-making process (Bill Inmon, 1993).**

## **CHARACTERISTICS of DATA WAREHOUSE:**

A common way of introducing data warehousing is to refer to the characteristics of a data warehouse.

- Subject Oriented
- Integrated
- Non-volatile
- Time Variant

**Subject Oriented:** Data warehouses are designed to help you analyze data. For example, to learn more about your company's sales data, you can build a warehouse that concentrates on sales. Using this warehouse, you can answer questions like "Who was our best customer for this item last year?" This ability to define a data warehouse by subject matter, sales in this case, makes the data warehouse subject oriented.

**Integrated:** Integration is closely related to subject orientation. Data warehouses must put data from disparate sources into a consistent format. They must resolve such problems as naming conflicts and inconsistencies among units of measure. When they achieve this, they are said to be integrated.

**Non-volatile:** Non-volatile means that, once entered into the warehouse, data should not change. This is logical because the purpose of a warehouse is to enable you to analyze what has occurred.

**Time Variant:** In order to discover trends in business, analysts need large amounts of data. This is very much in contrast to online transaction processing (OLTP) systems, where performance requirements demand that historical data be moved to an archive. A data warehouse's focus on change over time is what is meant by the term time variant. Typically, data flows from one or more online transaction processing (OLTP) databases into a data warehouse on a monthly, weekly, or daily basis. The data is normally processed in a staging file before being added to the data warehouse. Data warehouses commonly range in size from tens of gigabytes to a few terabytes. Usually, the vast majority of the data is stored in a few very large fact tables.

## Data Warehouse Applications

Data Warehouse helps the business executives in organize, analyse and use their data for decision making. Data Warehouse serves as a soul part of a plan-execute-assess "closed-loop" feedback system for enterprise management. Data Warehouse is widely used in the following fields:

- Financial services
- Banking Services
- Consumer goods
- Retail sectors.
- Controlled manufacturing

## Data Warehouse Types

Information processing, Analytical processing and Data Mining are the three types of data warehouse applications that are discussed below:

- **Information processing** - Data Warehouse allow us to process the information stored in it. The information can be processed by means of querying, basic statistical analysis, reporting using crosstabs, tables, charts, or graphs.
- **Analytical Processing** - Data Warehouse supports analytical processing of the information stored in it. The data can be analysed by means of basic OLAP operations, including slice-and-dice, drill down, drill up, and pivoting.
- **Data Mining** - Data Mining supports knowledge discovery by finding the hidden patterns and associations, constructing analytical models, performing classification and prediction. These mining results can be presented using the visualization tools.

## 2. Data Warehousing Today

For many businesses, the important question that often comes with understanding big data is "Where are we to put all this information?" It's not exactly a simple task, for there are volumes of material that get recorded each and every day. Moreover, it's not like companies can use most database practices today, since it's not always necessary to utilize databases like the ones found in most businesses that handle customers or transactions. Often, it's mere statistical recordkeeping that experts can analyze later. That's why data warehousing is an important factor in making the most of predictive analytics through the use of custom BI solutions.

### Not just a dumping ground

While a data warehouse functions similarly to its physical namesake, it doesn't necessarily translate into a place where a business simply stores excess information for big data purposes. Database expert James Serra noted that many companies simply utilize a common database for this function, rendering it into a simple dumping ground for the reams of material they record

on a regular basis. That's not practical for many reasons, namely that the amount of data required to make actionable and informed decisions tends to be massive.

*"A data warehouse functions more like a curated library than temporary storage space."*

Instead, how data warehouses function is that a company extracts data from specific sources on a regular basis. Then, using automation or a team of preparers, the data gets cleaned and properly formatted for placement within the database.

It is this method of organization and delivery that makes data warehousing extremely important to business looking to embrace big data. It allows a lot of integration and flexibility within the confines of business intelligence. For example, a company can create ad-hoc reports and analysis without needing to interfere with the source systems, especially if they happen to be transactional like MySQL and others. Data warehousing can provide full-fledged reports with a higher degree of accuracy because of the ability to drill down details that an analyst can't find from reading individual bits of information. Finally, the potential for data mining for historical trends to better enable predictive analytics is possible. All of these show that data warehousing is an essential part of big data.

Cloud-based technology has revolutionized the business world, allowing companies to easily retrieve and store valuable data about their customers, products and employees. This data is used to inform important business decisions.

Many global corporations have turned to data warehousing to organize data that streams in from corporate branches and operations centers around the world. It's essential for IT students to understand how data warehousing helps businesses remain competitive in a quickly evolving global marketplace.

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

**Ensure consistency.** Data warehouses are programmed to apply a uniform format to all collected data, which makes it easier for corporate decision-makers to analyse 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.

**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.

**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.

### 3. Future Trends in Data Warehouse

Data warehouses are here for the long term. Much has been invested in building them and many people and business functions depend on them. But sustainability demands that we rethink the data warehouse. Data warehouse architecture can no longer stand alone. We must think purpose, placement, and positioning of the data warehouse in broader data management architecture.

Architecture, of course, is only the beginning. The data warehouse is alive but it faces many challenges. It doesn't scale well, it has performance bottlenecks, it can be difficult to change, and it doesn't work well for big data. In a future of data warehouse modernization we'll need to consider cloud data warehousing, data warehousing with Hadoop, data warehouse automation as well as architectural modernization.

This understanding has given rise to the concept of business intelligence (BI), the use of data mining, big data, and data analytics to analyze raw data and create faster, more effective business solutions. However, while the concept of BI is not necessarily new, traditional BI tactics are no longer enough to keep up and ensure success in the future. Today, traditional BI must be combined with agile BI (the use of agile software development to accelerate traditional BI for faster results and more adaptability) and big data to deliver the fastest and most useful insights so that businesses may convert, serve, and retain more customers.

Essentially, for a business to survive, BI must continuously evolve and adapt to improve agility and keep up with data trends in this new customer-driven age of enterprise. This new model for BI is also driving the future of data warehousing, as we will see moving forward.

#### **Older BI Deployments Cannot Keep Pace for Success**

As valuable as older BI applications and deployments have been over the years, they simply cannot keep pace with customer demands today. In fact, decision-makers in IT and business have reported a number of challenges when they have only deployed traditional BI. These include:

Inability to accurately quantify their BI investments' ROI. Newer BI deployments implement methodologies for measuring ROI and determining the value of BI efforts.

A breakdown in communication and alignment between IT and business teams.

Inability to properly manage operational risk, resolve latency challenges, and/or handle scalability. While BI is intended to improve all of these, traditional BI is falling behind.

Difficulty with platform migration and/or integration.

Poor data quality. Even if data mining is fast and expansive, if the quality of the data is not up to par, it will not be useful in creating actionable intelligence for important business decisions.

#### **Keeping Up with Customer Demand through New BI Deployments**

So how can combining traditional BI, agile BI, and big data help businesses grow and succeed in today's market? Consider that big data gives businesses a more complete view of the

customer by tapping into multiple data sources. At the same time, agile BI addresses the need for faster and more adaptable intelligence. Combine the two, along with already existing traditional BI, and efforts that were once separate can work together to create a stronger system of insight and analytics.

Through this new BI strategy, businesses can consistently harness insights and create actionable data in less time. Using the same technology, processes, and people, it allows businesses to manage growth and complexity, react faster to customer needs, and improve collaboration and top-line benefits – all at the same time.

## **The Drive for a New Kind of Data Warehousing**

A new kind of data warehousing is essential to this new BI deployment, as much of the inefficiency in older BI deployments lies in the time and energy wasted in data movement and duplication. A few factors are driving the development and future of data warehousing, including:

**Agility** – To succeed today, businesses must use collaboration more than ever. Instead of having separate departments, teams, and implementations for things like data mining and analysis, IT, BI, business, etc., the new model involves cross-functional teams that engage in adaptive planning for continuous evolution and improvement. This kind of model cannot function with old forms of data warehousing, with just a single server (or set of servers) where data is stored and retrieved.

**The Cloud** – More and more, people and businesses are storing data on the cloud. Cloud-based computing offers the ability to access more data from different sources without the need for massive amounts of data movement and duplication. Thus, the cloud is a major factor in the future of data warehousing.

**The Next Generation of Data** – We are already seeing significant changes in data storage, data mining, and all things related to big data, thanks to the Internet of Things. The next generation of data will (and already does) include even more evolution, including real-time data and streaming data.

## **How New Data Warehousing Solves Problems for Businesses**

So how do new data warehouses change the face of BI and big data? These new data warehousing solutions offer businesses a more powerful and simpler means to achieve streaming, real-time data by connecting live data with previously stored historical data.

Before, business intelligence was an entirely different section of a company than the business section, and data analytics took place in an isolated bubble. Analysis was also restricted to only looking at and analyzing historical data – data from the past. Today, if businesses only look at historical data, they will be behind the curve before they even begin. Some of the solutions to this, which new data warehousing techniques and software provide, include:

**Data lakes** – Instead of storing data in hierarchical files and folders, as traditional data warehouses do, data lakes have a flat architecture that allows raw data to be stored in its natural form until it is needed.

**Data fragmented across organizations** – New data warehousing allows for faster data collection and analysis across organizations and departments. This is in keeping with the agility model and promotes more collaboration and faster results.

**IoT streaming data** – Again, the Internet of Things, is a major game changer, as customers, businesses, departments etc. share and store data across multiple devices.

## Examples of the Future of Data Warehousing

And what exactly will the future of data warehousing look like? Companies like SAP are working on that right now. With the launch of the BW/4HANA data warehousing solution running on premise and Amazon Web Services (AWS) and others like it, we can see how businesses can combine historical and streaming data for better implementation and deployment of new BI strategies. This system and others like it work with Spark and Hadoop, as well as other programming frameworks to bring data and systems of insight into the 21st century and beyond.

## 4. Data Warehouse Architecture

### Basic Steps to develop data warehouse architecture.

In general, building any data warehouse consists of the following steps:

1. Extracting the transactional data from the data sources into a staging area
2. Transforming the transactional data
3. Loading the transformed data into a dimensional database
4. Building pre-calculated summary values to speed up report generation
5. Building (or purchasing) a front-end reporting tool

#### Extracting Transactional Data

A large part of building a DW is **pulling data from various data sources and placing it in a central storage area**. In fact, this can be the most difficult step to accomplish due to the reasons mentioned earlier: Most people who worked on the systems in place have moved on to other jobs. Even if they haven't left the company, you still have a lot of work to do: You need to figure out which database system to use for your staging area and how to pull data from various sources into that area.

#### Transforming Transactional Data

An equally important and challenging step after extracting is **transforming and relating the data** extracted from multiple sources. As I said earlier, your source systems were most likely built by many different IT professionals. Let's face it. Each person sees the world through their own eyes, so each solution is at least a bit different from the others. The data model of your mainframe system might be very different from the model of the client-server system.

## Creating a Dimensional Model

The third step in building a data warehouse is **coming up with a dimensional model**. Most modern transactional systems are built using the relational model. The relational database is highly normalized; when designing such a system, you try to get rid of repeating columns and make all columns dependent on the primary key of each table. The relational systems perform well in the On-Line Transaction Processing (OLTP) environment.

The dimensional model consists of the fact and dimension tables. The fact tables consist of foreign keys to each dimension table, as well as measures. The *measures* are a factual representation of how well (or how poorly) your business is doing (for instance, the number of parts produced per hour or the number of cars rented per day). *Dimensions*, on the other hand, are what your business users expect in the reports—the details about the measures. For example, the time dimension tells the user that 2000 parts were produced between 7 a.m. and 7 p.m. on the specific day; the plant dimension specifies that these parts were produced by the Northern plant.

## Loading the Data

After you've built a dimensional model, it's time to **populate it with the data** in the staging database. This step only sounds trivial. It might involve combining several columns together or splitting one field into several columns. You might have to perform several lookups before calculating certain values for your dimensional model.

## Generating Pre-calculated Summary Values

The next step is **generating the pre-calculated summary values** which are commonly referred to as *aggregations*. This step has been tremendously simplified by SQL Server Analysis Services (or OLAP Services, as it is referred to in SQL Server 7.0). After you have populated your dimensional database, SQL Server Analysis Services does all the aggregate generation work for you.

## Building (or Purchasing) a Front-End Reporting Tool


After you've built the dimensional database and the aggregations you can decide how sophisticated your **reporting tools** need to be. If you just need the drill-down capabilities, and your users have Microsoft Office 2000 on their desktops, the Pivot Table Service of Microsoft Excel 2000 will do the job. If the reporting needs are more than what Excel can offer, you'll have to investigate the alternative of building or purchasing a reporting tool. The cost of building a custom reporting (and OLAP) tool will usually outweigh the purchase price of a third-party tool. That is not to say that OLAP tools are cheap (not in the least!).



# Architectural Component of Data warehouse:

- Multi-Dimensional Database
- ETL
- OLAP
- Meta Data

## Multi Dimension Database

 **Data Mart are normally a multi-dimensional database using industry standard STAR Schema approach. This will include:**

Dimensions tables

Fact Tables

Hierarchies (For Drill down and drill across)

Role Models (Multiple references of Dimension to the Fact table)

Summary / Aggregate tables

Snow Flaking and Normalization as required

 **Optimized database design for better performance**

STAR Transformation


Bit Map Indexes

Database Partitioning

## ETL

- Create Source and Target Meta Data
- Identify the Source & the Target data mart data structures
- Create Extraction Programs using ETL tool (e.g. Ascential Data Stage / Informatica)
- Define data cleansing, transformation and aggregation rules
- Perform data capture and enrichment processing / Evaluate incremental (delta) refresh option of the data mart to reduce the production cycle

## OLAP

 **OLAP (Online Analytical Processing) tool provides the front end analytical capabilities**

 **Functionality provided**


Slice & Dice

Drill up, Drill down & Drill across

Pivoting

Trend analysis across Time

Easy interface to tools such as Excel / MS Word

 **Implemented in a three tier web based architecture**

## Meta Data

The fact tables contain quantitative data that might be queried or acquired to measure and the dimension tables are smaller and hold descriptive data that relates to measure and reflect upon the data hierarchy in the database.

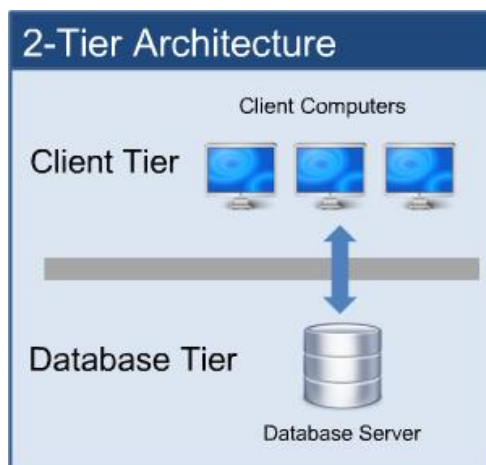
## Data Warehouse system Architecture (Two tier and three tier Architecture)

Difference:

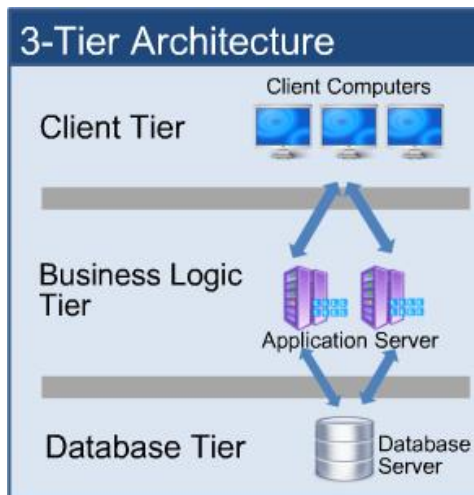
Two-tier architecture is a client/server architecture, where a request to do some task is sent to the server and the server responds by performing the task.

Whereas a three-tier or a multi-tier architecture has client, server and database. Where the client request is sent to the server and the server in turn sends the request to the database. The database sends back the information/data required to the server which then sends it to the client.

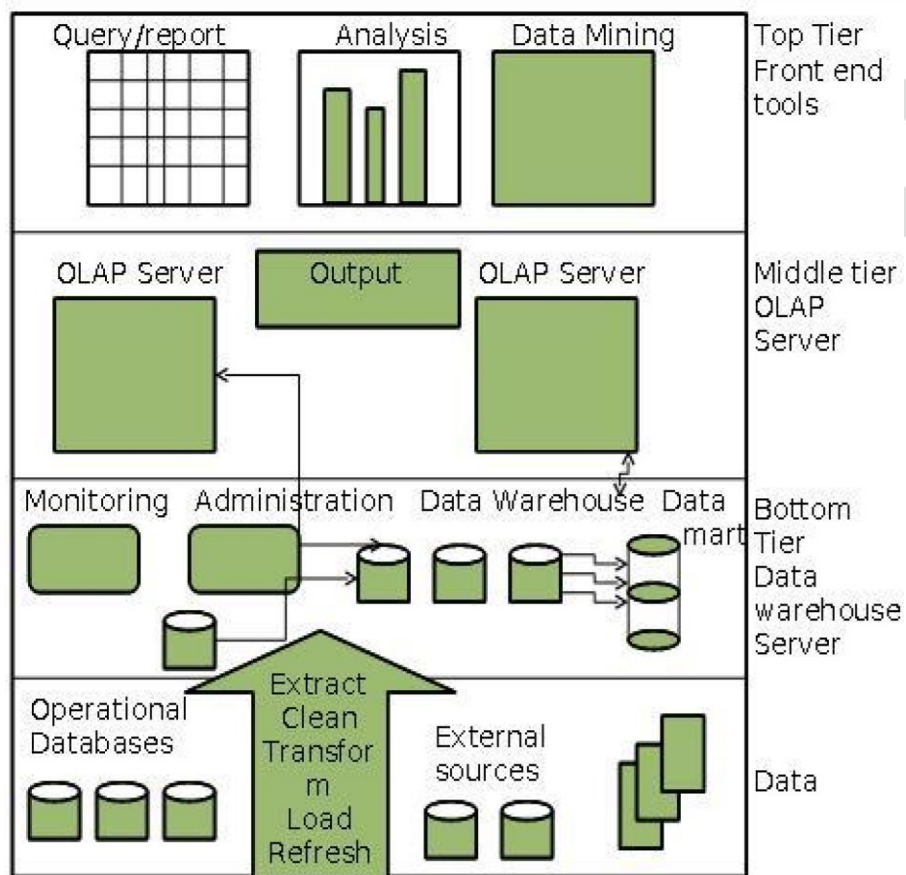
### Two tier Architecture



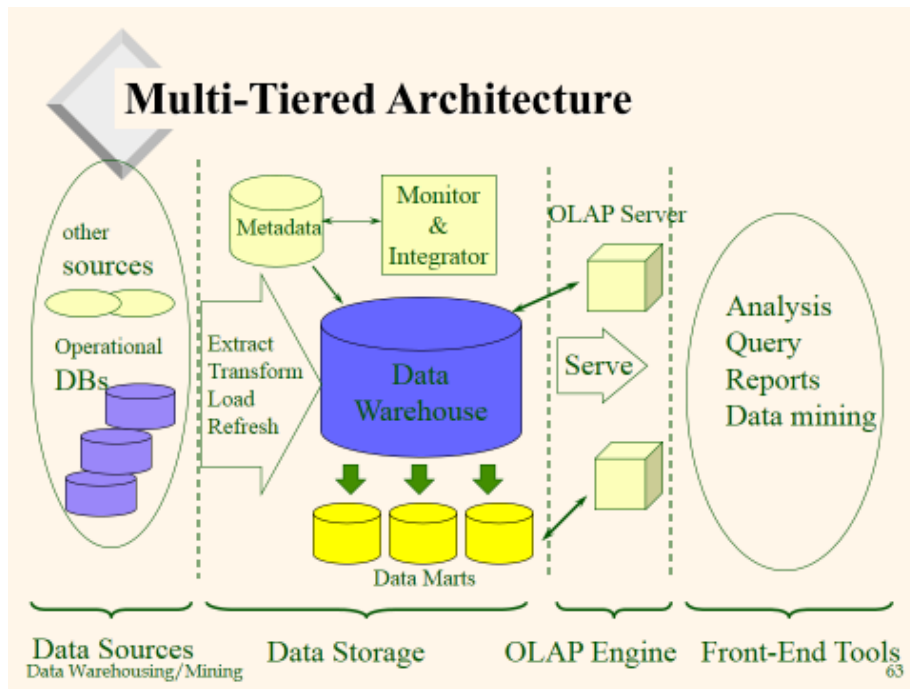
### Three tier Architecture



### 3 tier Data warehouse Architecture



OR



	1-Tier	2-Tier	Multi-Tier
<b>Benefits</b>	<ul style="list-style-type: none"> <li>• Very simple</li> <li>• Inexpensive</li> <li>• No server needed</li> </ul>	<ul style="list-style-type: none"> <li>• Good security</li> <li>• More scalable</li> <li>• Faster execution</li> </ul>	<ul style="list-style-type: none"> <li>• Exceptional security</li> <li>• Fastest execution</li> <li>• “Thin” client</li> <li>• Very scalable</li> </ul>
<b>Issues</b>	<ul style="list-style-type: none"> <li>• Poor security</li> <li>• Multi user issues</li> </ul>	<ul style="list-style-type: none"> <li>• More costly</li> <li>• More complex</li> <li>• “Thick” client</li> </ul>	<ul style="list-style-type: none"> <li>• Very costly</li> <li>• Very complex</li> </ul>
<b>Users</b>	<ul style="list-style-type: none"> <li>• Usually 1 (or a few)</li> </ul>	<ul style="list-style-type: none"> <li>• 2-100</li> </ul>	<ul style="list-style-type: none"> <li>• 50-2000 (+)</li> </ul>

**Data warehouses often adopt a three – tier architecture,**

1. The bottom tier is a warehouse database server that is almost always a relational database system. “How are the data extracted from this tier in order to create the data warehouse?” Data from operational databases and external sources (such as customer profile information Provided by external consultants) are extracted using application program interfaces known as gateways. A gateway is supported by the underlying DBMS and allows client programs to generate SQL code to be executed at a server. Examples of gateways include ODBC (Open Database Connection) and OLE – DB (Open Linking and Embedding for Databases), by Microsoft, and JDBC (Java Database Connection).

2. The middle tier is an OLAP server that is typically implemented using either (1) a relational OLAP (ROLAP) model, that is, an extended relational DBMS that maps operations on multidimensional data to standard relational operations, or (2) a multidimensional OLAP

(MOLAP) model, that is, a special – purpose server that directly implements multidimensional data and operations. OLAP servers are discussed in Section 2.3.3

3. The top tier is a client, which contains query and reporting tools, analysis tools, and / or data mining tools (e.g., trend analysis, prediction, and so on). From the architecture point of view, there are three data warehouse models: the enterprise warehouse, the data mart, and the virtual warehouse.

**Enterprise warehouse:** An enterprise warehouse collects all of the information about subjects spanning the entire organization. It provides corporate – wide data integration, usually from one or more operational systems or external information providers, and is cross – functional in scope. It typically contains detailed data as well as summarized data, and can range in size from a few gigabytes to hundreds of gigabytes, terabytes, or beyond. An enterprise data warehouse may be implemented on traditional mainframes, UNIX superservers, or parallel architecture platforms. It requires extensive business modeling and may take years to design and build.

**Data mart:** A data mart contains a subset of corporate – wide data that is of value to a specific group of users. The scope is confined to specific selected subjects. For example, a marketing data mart may confine its subjects to customer, item, and sales. The data contained in data marts tend to be summarized. Depending on the sources of data, data marts can be categorized as independent or dependent. Independent data marts are sourced from data captured from one or more operational systems or external information providers, or from data generated locally within a particular department or geographic area. Dependent data marts are sourced directly from enterprise data warehouses.

**Virtual warehouse:** A virtual warehouse is a set of views over operational databases. For efficient query processing, only some of the possible summary views may be materialized. A virtual warehouse is easy to build but requires excess capacity on operational database servers.

## 5. Data Flow Architecture

A data warehouse system has two main architectures: the data flow architecture and the system architecture. The *data flow architecture* is about how the data stores are arranged within a data warehouse and how the data flows from the source systems to the users through these data stores. The *system architecture* is about the physical configuration of the servers, network, software, storage, and clients. In this chapter, I will discuss the data flow architecture first and then the system architecture.

In data warehousing, the data flow architecture is a configuration of data stores within a data warehouse system, along with the arrangement of how the data flows from the source systems through these data stores to the applications used by the end users. This includes how the data flows are controlled, logged, and monitored, as well as the mechanism to ensure the quality of the data in the data stores. I discussed the data flow architecture briefly in Chapter 1, but in this chapter I will discuss it in more detail, along with four data flow architectures: single DDS,

NDS + DDS, ODS + DDS, and federated data warehouse. The data flow architecture is different from data architecture. Data architecture is about how the data is arranged in each data store and how a data store is designed to reflect the business processes. The activity to produce data architecture is known as data modeling.

Data stores are important components of data flow architecture.

I'll begin the discussion about the data flow architecture by explaining what a data store is.

A data store is one or more databases or files containing data warehouse data, arranged in a particular format and involved in data warehouse processes.

**Based on the user accessibility, you can classify data warehouse data stores into three types:**

- A user-facing data store is a data store that is available to end users and is queried by the end users and end-user applications.
- An internal data store is a data store that is used internally by data warehouse components for the purpose of integrating, cleansing, logging, and preparing data, and it is not open for query by the end users and end-user applications.
- A hybrid data store is used for both internal data warehouse mechanisms and for query by the end users and end-user applications.

A master data store is a user-facing or hybrid data store containing a complete set of data in a data warehouse, including all versions and all historical data.

**Based on the data format, you can classify data warehouse data stores into four types:**

- A stage is an internal data store used for transforming and preparing the data obtained from the source systems, before the data is loaded to other data stores in a data warehouse.
- A normalized data store (NDS) is an internal master data store in the form of one or more normalized relational databases for the purpose of integrating data from various source systems captured in a stage, before the data is loaded to a user-facing data store.
- An operational data store (ODS) is a hybrid data store in the form of one or more normalized relational databases, containing the transaction data and the most recent version of master data, for the purpose of supporting operational applications.
- A dimensional data store (DDS) is a user-facing data store, in the form of one or more relational databases, where the data is arranged in dimensional format for the purpose of supporting analytical queries.