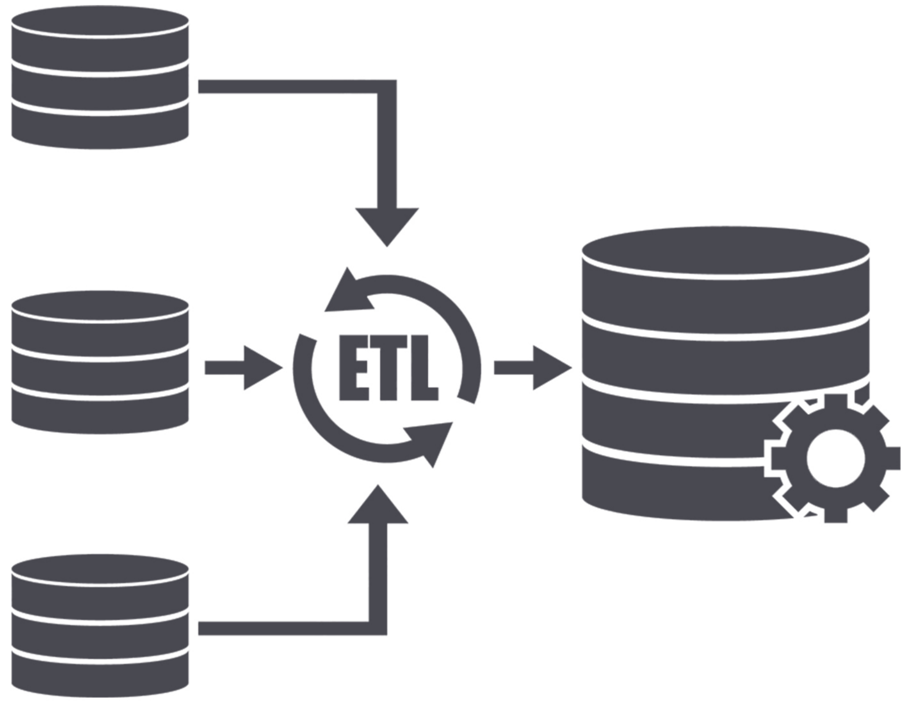


DATA WAREHOUSE

DATA WAREHOUSE

A data warehouse is a centralized repository that stores structured data (database tables, Excel sheets) and semi-structured data (XML files, webpages) for the purposes of reporting and analysis. The data flows in from a variety of sources, such as point-of-sale systems, business applications, and relational databases, and it is usually cleaned and standardized before it hits the warehouse. Because a data warehouse can store large amounts of information, it provides users with easy access to a wealth of historical data, which can be used for data mining, data visualization, and other forms of business intelligence reporting.

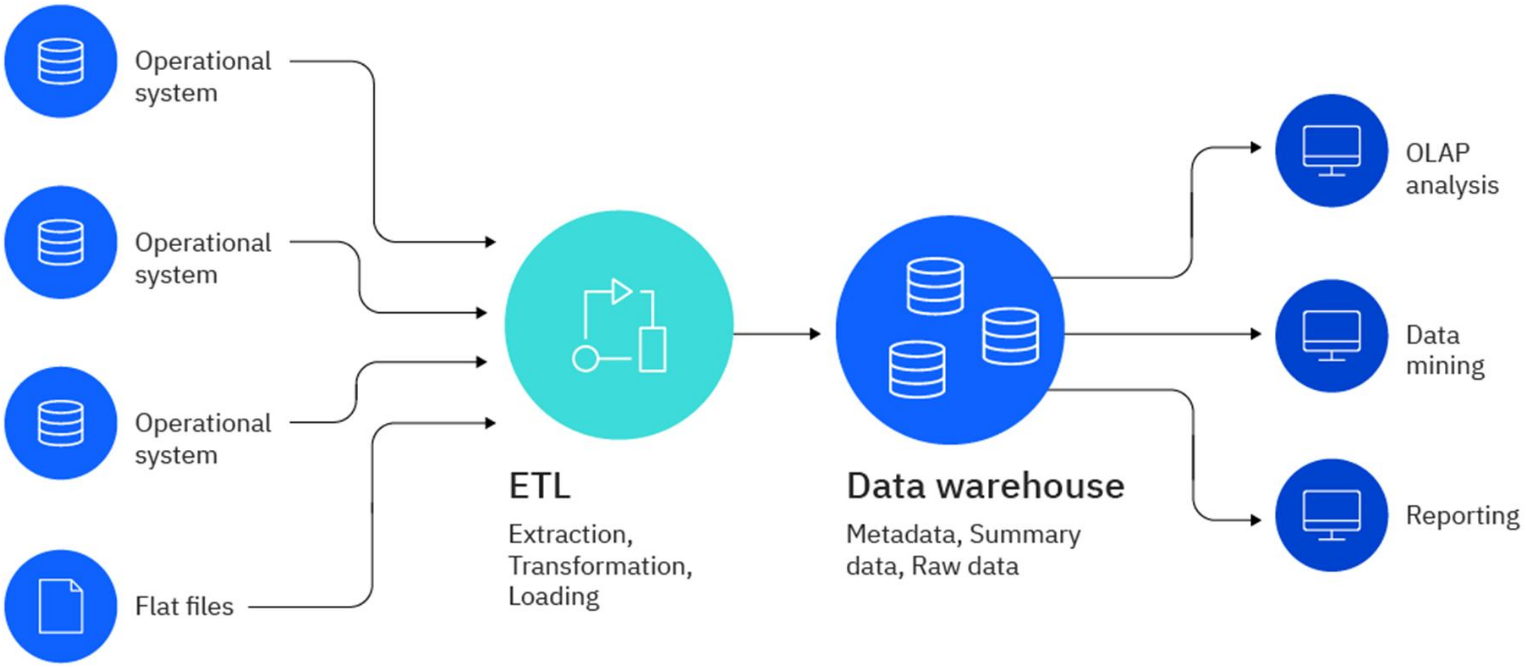
DATA WAREHOUSE ARCHITECTURE

Generally speaking, data warehouses have a three-tier architecture, which consists of a:

1. Bottom Tier

2. Middle Tier

3. Top Tier



1. BOTTOM TIER:

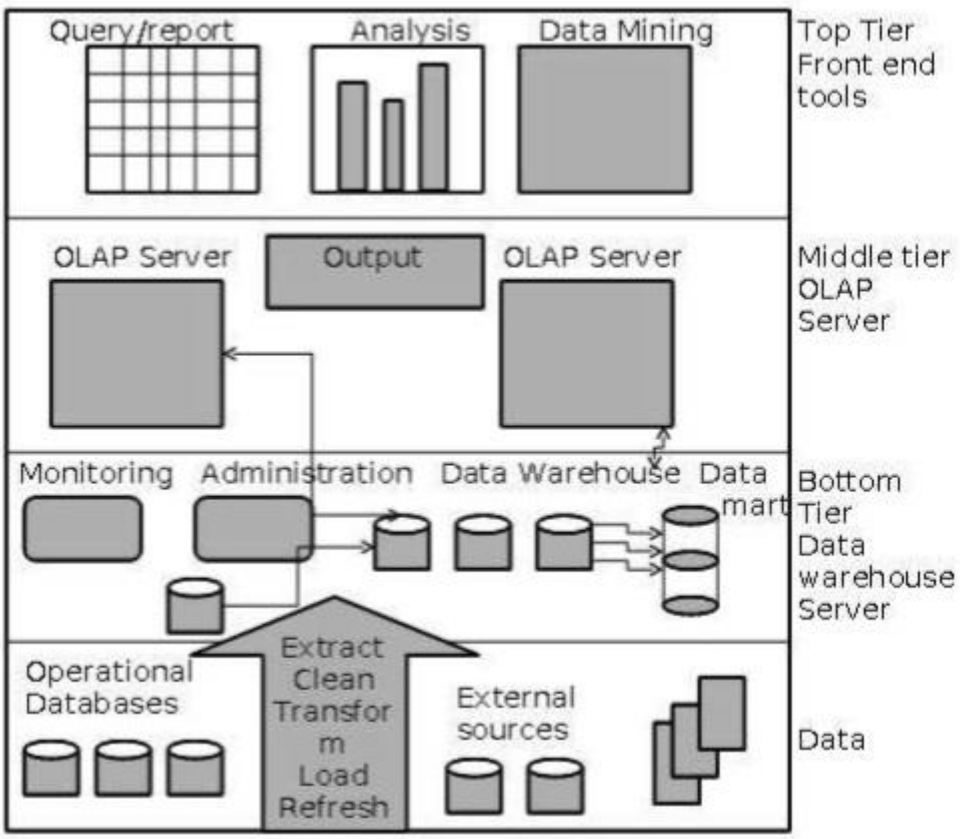
Data is ingested from multiple sources, then cleansed and transformed for other applications to use in a process called **extract, transform, & load (ETL).** For most organizations that use ETL, the process relies on automation, and is efficient, well-defined, continuous and batch-driven. The bottom tier is also where data is stored and optimized, which leads to faster query times and better performance overall.

2. MIDDLE TIER:

This is where you'll find the analytics engine, also known as the **online analytical processing (OLAP)** server. OLAP servers access large volumes of data from the data warehouse at a high speed, which leads to lightning-fast results. Three types of OLAP models can be used in this tier, which are known as ROLAP, MOLAP and HOLAP. The type of OLAP model used is dependent on the type of database system that exists.

3. TOP TIER:

The top tier is represented by some kind of front-end user interface or reporting tool, which enables end users to conduct ad-hoc data analysis on their business data



A SHORT HISTORY OF DATA WAREHOUSE ARCHITECTURE

Most data warehouses are built around a relational database system, either on-site or in the cloud, where data is stored and processed. They also have a metadata management system and an API connectivity layer to pull data from different sources and provide access to analytics and visualization tools.

A typical data warehouse has four main parts:

1. A central database
2. ETL tools (for extracting, transforming, and loading data)
3. Metadata
4. Access tools

All these components are designed for speed to get quick results and analyze data immediately.

Data warehouses have been around since the 1980s to optimize data analytics. As companies grew and generated more data, they needed systems

that could manage and analyze this data. Database administrators would pull data from operational systems, transform it, and load it into the data warehouse.

As data warehouse architecture evolved, more people in a company started using it to access structured data easily. This is where metadata became important. Reporting and dashboarding became common uses, and SQL (Structured Query Language) became the standard way to interact with the data.

COMPONENTS OF DATA WAREHOUSE ARCHITECTURE

1. ETL:

When database analysts want to move data from a data source into their data warehouse, this is the process they use. In short, ETL converts data into a usable format so that once it’s in the data warehouse, it can be analyzed / queried / etc.

2. METADATA:

Metadata is data about data. Basically, it describes all of the data that’s stored in a system to make it searchable. Some examples of metadata include authors, dates or locations of an article, create date of a file, the size of a file, etc. Think of it like the titles of a column in a spreadsheet. Metadata allows you to organize your data to make it usable, so you can analyze it to create dashboards and reports.

3. SQL QUERY PROCESSING:

SQL is the de facto standard language for querying your data. This is the language that analysts use to pull out insights from their data stored in the data warehouse. Typically data warehouses have proprietary SQL query processing technologies tightly coupled with the compute. This allows for very high performance when it comes to your analytics. One thing to note, however, is that the cost of a data warehouse can start getting expensive the more data and SQL compute resources you have.

4. DATA LAYER:

The data layer is the access layer that allows users to actually get to the data. This is typically where you’d find a data mart. This layer partitions segments of your data out depending on who you want to give access to, so you can get very granular across your organization. For instance, you may not want to give your sales team access to your HR team’s data, and vice versa.

5. GOVERNANCE & SECURITY:

This is related to the data layer in that you need to be able to provide fine-grained access and security policies across all your organization’s data. Typically data warehouses have very good data governance and security capabilities built in, so you don’t need to do a lot of custom data engineering work to include this. It’s important to plan for governance and security as you add more data to your warehouse and as your company grows.

6. DATA WAREHOUSE ACCESS TOOLS:

While access tools are external to your data warehouse, they can be seen as its business-user friendly front end. This is where you’d find your reporting and visualization tools, used by data analysts and business users to interact with the data, extract insights and create visualizations that the rest of the business can consume. Examples of these tools include Tableau, Looker, Power BI and Qlik.

UNDERSTANDING OLAP AND OLTP IN DATA WAREHOUSES

OLAP (Online Analytical Processing) is a type of software designed for performing high-speed, multidimensional analysis on large volumes of data stored in centralized data repositories such as data warehouses. In contrast, **OLTP (Online Transactional Processing)** supports real-time execution of a large number of database transactions by many users, typically over the internet. The primary distinction between OLAP and OLTP is in the name, that OLAP is analytical, while OLTP is transactional.

OLAP Tools:

- Designed for multidimensional analysis of data from data warehouses, which include both historical and transactional data.
- Common uses include data mining, business intelligence applications, complex analytical calculations, and predictive scenarios.
- Support business reporting functions like financial analysis, budgeting, and forecast planning.

OLTP Systems:

- Designed to support transaction-oriented applications by processing recent transactions as quickly and accurately as possible.
- Common uses include ATMs, e-commerce software, credit card payment processing, online bookings, reservation systems, and record-keeping tools.

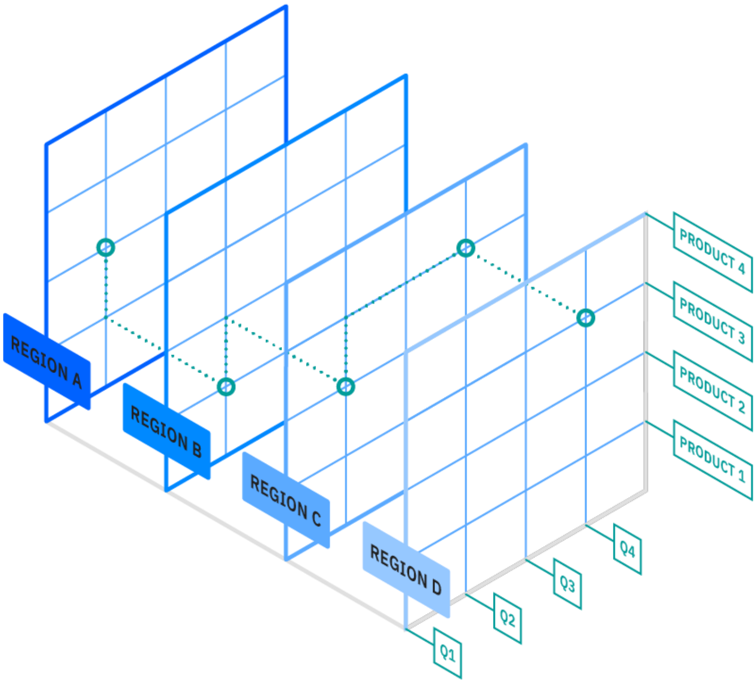
What is OLAP?

- Online analytical processing (OLAP) is a system for performing multi-dimensional analysis at high speeds on large volumes of data. Typically, this data is from a data warehouse, data mart or some other centralized data store. OLAP is ideal for data mining, business intelligence and complex analytical calculations, as well as business reporting functions like financial analysis, budgeting and sales forecasting.

- The core of most OLAP databases is the **OLAP cube**, which allows you to quickly query, report on and analyze multidimensional data. **What's a data dimension?** It's simply one element of a particular dataset. For example, sales figures might have several dimensions related to region, time of year, product models and more.

- The OLAP cube extends the row-by-column format of a traditional relational database schema and adds layers for other data dimensions. For example, while the top layer of the cube might organize sales by region, data analysts can also "drill-down" into layers for sales by state/province, city and/or specific stores. This historical, aggregated data for OLAP is usually stored in a **star schema** or **snowflake schema**.

- The following graphic shows the OLAP cube for sales data in multiple dimensions — by region, by quarter and by product:



What is OLTP?

Online transactional processing (OLTP) enables the real-time execution of large numbers of database transactions by large numbers of people, typically over the Internet. OLTP systems are behind many of our everyday transactions, from ATMs to in-store purchases to hotel reservations. OLTP can also drive non-financial transactions, including password changes and text messages.

OLTP systems use a relational database that can do the following:

- Process a large number of relatively simple transactions - usually insertions, updates & deletions to data.
- Enable multi-user access to the same data, while ensuring data integrity.
- Support very rapid processing, with response times measured in milliseconds.
- Provide indexed data sets for rapid searching, retrieval and querying.
- Be available 24/7/365, with constant incremental backups.

Many organizations use OLTP systems to provide data for OLAP. In other words, a combination of both OLTP and OLAP are essential in our data-driven world.

OLAP vs. OLTP

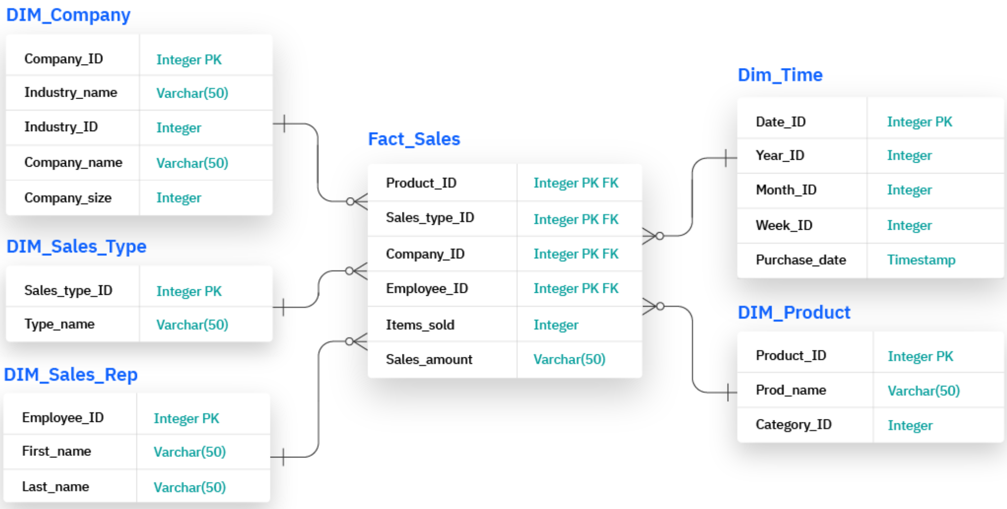
Aspect	OLAP (Online Analytical Processing)	OLTP (Online Transactional Processing)
Purpose	Analytical processing for complex queries and data analysis	Transactional processing for routine operations
Focus	Data analysis, business intelligence, and reporting	Real-time transaction processing
Data Source	Multi-dimensional schema (data warehouse, data marts)	Traditional DBMS (relational databases)
Data Type	Historical and aggregated data	Current, real-time data
Typical Users	Data scientists, business analysts, knowledge workers	Frontline workers (e.g., cashiers, bank tellers) and customer self-service
Processing Type	Complex analytical queries	Simple transaction queries (insertions, updates, deletions)
Response Time	Slower response times (minutes to hours)	Extremely fast response times (milliseconds)
Workload	Read-intensive, involving large data sets	Write-intensive, involving small data sets
Data Volume	Large volume of data for analysis	Large volume of transactions per second
Backup Frequency	Less frequent backups needed	Frequent or concurrent backups required
Availability	Not mission-critical, can tolerate downtime	Mission-critical, requires high availability
Data Integrity	Less critical, as data is historical and aggregated	Highly critical, ensures transactional consistency
Schema Design	Star schema or snowflake schema	Normalized schema (3NF)
Query Complexity	Supports complex queries involving large numbers of records	Supports simple queries involving one or few records
Use Cases	Financial analysis, budgeting, sales forecasting, data mining	ATMs, online banking, e-commerce transactions, reservation

SCHEMAS IN DATA WAREHOUSES

Schemas are ways in which data is organized within a database or data warehouse. There are two main types of schema structures, the star schema and the snowflake schema, which will impact the design of your data model.

Star schema:

This schema consists of one fact table which can be joined to a number of denormalized dimension tables. It is considered the simplest and most common type of schema, and its users benefit from its faster speeds while querying.



DIM_Company

Company_ID	Integer PK
Industry_name	Varchar(50)
Industry_ID	Integer
Company_name	Varchar(50)
Company_size	Integer

DIM_Sales_Type

Sales_type_ID	Integer PK
Type_name	Varchar(50)

DIM_Sales_Rep

Employee_ID	Integer PK
First_name	Varchar(50)
Last_name	Varchar(50)

Fact_Sales

Product_ID	Integer PK FK
Sales_type_ID	Integer PK FK
Company_ID	Integer PK FK
Employee_ID	Integer PK FK
Items_sold	Integer
Sales_amount	Varchar(50)

Dim_Time

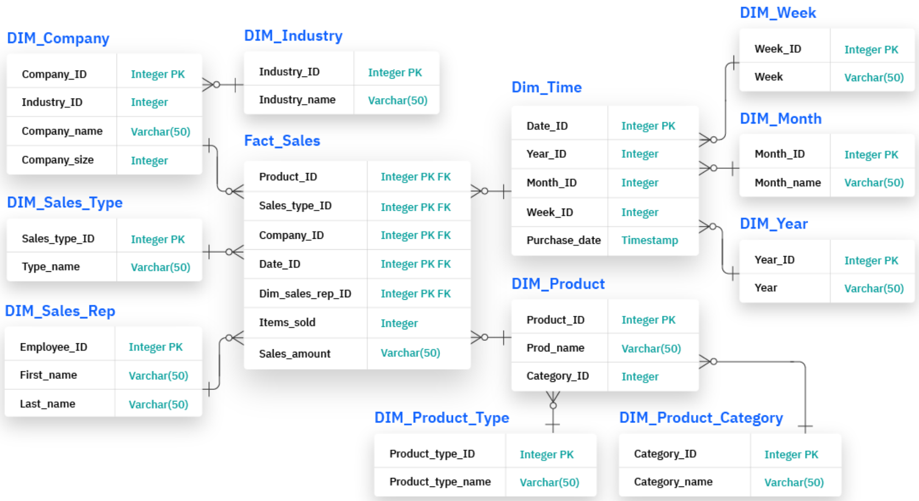
Date_ID	Integer PK
Year_ID	Integer
Month_ID	Integer
Week_ID	Integer
Purchase_date	Timestamp

DIM_Product

Product_ID	Integer PK
Prod_name	Varchar(50)
Category_ID	Integer

Snowflake schema:

While not as widely adopted, the snowflake schema is another organization structure in data warehouses. In this case, the fact table is connected to a number of normalized dimension tables, and these dimension tables have child tables. Users of a snowflake schema benefit from its low levels of data redundancy, but it comes at a cost to query performance.



DIM_Company

Company_ID	Integer PK
Industry_ID	Integer
Company_name	Varchar(50)
Company_size	Integer

DIM_Industry

Industry_ID	Integer PK
Industry_name	Varchar(50)

DIM_Sales_Type

Sales_type_ID	Integer PK
Type_name	Varchar(50)

DIM_Sales_Rep

Employee_ID	Integer PK
First_name	Varchar(50)
Last_name	Varchar(50)

Fact_Sales

Product_ID	Integer PK FK
Sales_type_ID	Integer PK FK
Company_ID	Integer PK FK
Date_ID	Integer PK FK
Dim_sales_rep_ID	Integer PK FK
Items_sold	Integer
Sales_amount	Varchar(50)

Dim_Time

Date_ID	Integer PK
Year_ID	Integer
Month_ID	Integer
Week_ID	Integer
Purchase_date	Timestamp

DIM_Week

Week_ID	Integer PK
Week	Varchar(50)

DIM_Month

Month_ID	Integer PK
Month_name	Varchar(50)

DIM_Year

Year_ID	Integer PK
Year	Varchar(50)

DIM_Product

Product_ID	Integer PK
Prod_name	Varchar(50)
Category_ID	Integer

DIM_Product_Type

Product_type_ID	Integer PK
Product_type_name	Varchar(50)

DIM_Product_Category

Category_ID	Integer PK
Category_name	Varchar(50)

DATA WAREHOUSE VS. DATABASE, DATA LAKE, & DATA MART

Data warehouse, database, data lake, and data mart are all terms that tend to be used interchangeably. While the terms are similar, important differences exist:

Data warehouse vs. data lake

Using a data pipeline, a data warehouse gathers raw data from multiple sources into a central repository, structured using predefined schemas designed for data analytics. A data lake is a data warehouse without the predefined schemas. As a result, it enables more types of analytics than a data warehouse. Data lakes are commonly built on big data platforms such as Apache Hadoop.

Data warehouse vs. data mart

A data mart is a subset of a data warehouse that contains data specific to a particular business line or department. Because they contain a smaller subset of data, data marts enable a department or business line to discover more-focused insights more quickly than possible when working with the broader data warehouse data set.

Data warehouse vs. database

A database is built primarily for fast queries and transaction processing, not analytics. A database typically serves as the focused data store for a specific application, whereas a data warehouse stores data from any number (or even all) of the applications in your organization.

A database focuses on updating real-time data while a data warehouse has a broader scope, capturing current and historical data for predictive analytics, machine learning, and other advanced types of analysis.

TYPES OF DATA WAREHOUSES

Cloud data warehouse:

A cloud data warehouse is a data warehouse specifically built to run in the cloud, and it is offered to customers as a managed service. Cloud-based data warehouses have grown more popular over the last five to seven years as more companies use cloud computing services and seek to reduce their on-premises data center footprint.

With a cloud data warehouse, the physical data warehouse infrastructure is managed by the cloud company, meaning that the customer doesn’t have to make an upfront investment in hardware or software and doesn’t have to manage or maintain the data warehouse solution.

Data warehouse software (on-premises/license):

A business can purchase a data warehouse license and then deploy a data warehouse on their own on-premises infrastructure. Although this is typically more expensive than a cloud data warehouse service, it might be a better choice for government entities, financial institutions, or other organizations that want more control over their data or need to comply with strict security or data privacy standards or regulations.

Data warehouse appliance:

A data warehouse appliance is a pre-integrated bundle of hardware and software—CPUs, storage, operating system, and data warehouse software—that a business can connect to its network and start using as-is. A data warehouse appliance sits somewhere between cloud and on-premises implementations in terms of upfront cost, speed of deployment, ease of scalability, and data management control.

BENEFITS OF A DATA WAREHOUSE

A data warehouse provides a foundation for the following:

Better data quality:

A data warehouse centralizes data from a variety of data sources, such as transactional systems, operational databases, and flat files. It then cleanses the operational data, eliminates duplicates, and standardizes it to create a single source of the truth.

Faster, business insights:

Data from disparate sources limit the ability of decision makers to set business strategies with confidence. Data warehouses enable data integration, allowing business users to leverage all of a company’s data into each business decision. Data warehouse data makes it possible to report on themes, trends, aggregations, and other relationships among data collected from an engineering lifecycle management (ELM) app.

Smarter decision-making:

A data warehouse supports large-scale BI functions such as data mining (finding unseen patterns and relationships in data), artificial intelligence, and machine learning—tools data professionals and business leaders can use to get hard evidence for making smarter decisions in virtually every area of the organization, from business processes to financial management and inventory management.

Gaining and growing competitive advantage:

All of the above combine to help an organization finding more opportunities in data, more quickly than is possible from disparate data stores.

If you would like to learn more & stay updated, please follow me on LinkedIn