Business Intelligence

Important Concepts

Unit 1

Introduction to data, information and Knowledge

- **Terminologies**:
  - It has been a long-held belief by many philosophers that ***knowledge*** is power and that knowledge stems from understanding of ***information***; information, in turn, is the assigning of meaning to ***data.***
  - To develop the understanding of information technology, we start by defining these three related concepts. The topics are hierarchical in that:

Introduction to data, information and Knowledge

- **Terminologies**:
  1. **Data**:

     The concept of data as it is used in the syllabus is commonly referred to as ‘raw’ data – a collection of text, numbers and symbols with no meaning. Data therefore must be processed, or provided with a context, before it can have meaning.

  - **Example**
    - 3, 6, 9, 12
    - Cat, dog, gerbil, rabbit
    - 161.2, 175.3, 166.4, 160.5

  These are meaningless sets of data. They could be the first four answers in the 3x table, a list of household pets and the heights of 15-year-old students but without a context we don’t know.

Introduction to data, information and Knowledge

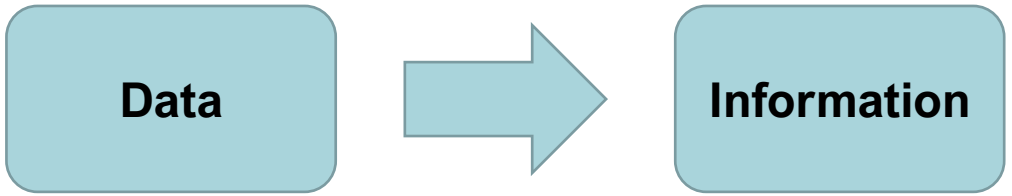
- **Terminologies**:

2. **Information**:

It is to learn the concept of what '*information*' is as used in information technology.

Information is the result of processing data, usually by computer. This results in facts, which enables the processed data to be used in context and have meaning. Information is data that has meaning.

- When does data become information?



Introduction to data, information and Knowledge

- **Terminologies**:

  2. **Information**:

    - When does data become information?

      - Data on its own has no meaning. It only takes on meaning and becomes information when it is interpreted. Data consists of raw facts and figures. When that data is processed into sets according to context, it provides information.

      - Data refers to raw input that when processed or arranged makes meaningful output. Information is usually the processed outcome of data. When data is processed into information, it becomes interpretable and gains significance.

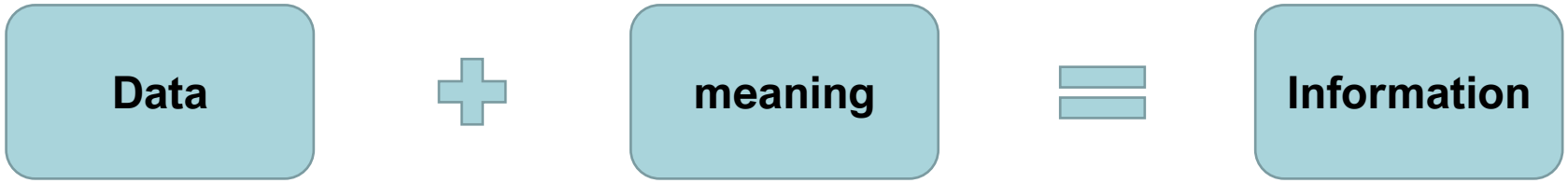
Introduction to data, information and Knowledge

- **Terminologies**:

2. Information:

- When does data become information?
  - In IT, symbols, characters, images, or numbers are data. These are the inputs an IT system needs to process in order to produce a meaningful interpretation.
  - In other words, data in a meaningful form becomes information. Information can be about facts, things, concepts, or anything relevant to the topic concerned. It may provide answers to questions like who, which, when, why, what, and how.

`



Introduction to data, information and Knowledge

- **Terminologies**:
  2. **Information**:
     - **Example**:

Look at the following examples given for data:

- 3, 6, 9, 12
- cat, dog, gerbil, rabbit
- 161.2, 175.3, 166.4, 164.7, 169.3

Only when we assign a context or meaning does the data become information. It all becomes meaningful when we are told:

- 3, 6, 9 and 12 are the first four answers in the 3x table.
- cat, dog, gerbil, rabbit is a list of household pets.
- 161.2, 175.3, 166.4, 164.7, 169.3 are the heights of 15-year-old students.

Introduction to data, information and Knowledge

- **Terminologies**:

3. **Knowledge**:

When someone memorizes information, this is often referred to as '*rote-learning*' or '*learning by heart*'. We can then say that they have acquired some knowledge. Another form of knowledge is produced as a result of understanding information that has been given to us and using that information to gain knowledge of how to solve problems.

Knowledge can therefore be:

- Acquiring and remembering a set of facts or
- The use of information to solve problems.

Introduction to data, information and Knowledge

- **Terminologies**:

  3. Knowledge:

  The first type is often called ***explicit knowledge***. This is knowledge that can be easily passed on to others. Most forms of explicit knowledge can be stored in certain media. The information contained in encyclopedias and textbooks are good examples of ***explicit knowledge.***

  The second type is called ***tacit (implicit) knowledge***. It is the kind of knowledge that is difficult to pass on to another person just by writing it down. For example, saying that Paris is the capital of France is an **explicit knowledge** that can be written down, passed on, and understood by someone else.

  However, the ability to speak a foreign language, bake bread, program a computer or use complicated machinery requires additional pieces of knowledge (such as that gained through experience) that are not always known explicitly and are difficult to pass on to other users.

Introduction to data, information and Knowledge

- **Terminologies**:
  3. **Knowledge**:

  How are data, information and Knowledge linked?

If we put Knowledge into an equation it would look like this

Introduction to data, information and Knowledge

- **Terminologies**:
  3. **Knowledge**:
  - **Example:**

    Look at the following examples given for **data**:
    - 3, 6, 9, 12
    - cat, dog, gerbil, rabbit
    - 161.2, 175.3, 166.4, 164.7, 169.3

    Only when we assign a context or meaning does the data become **information**. It all becomes meaningful when we are told:

    - 3, 6, 9 and 12 are the first four answers in the 3x table.
    - cat, dog, gerbil, rabbit is a list of household pets.
    - 161.2, 175.3, 166.4, 164.7, 169.3 are the heights of 15-year-old students.

Introduction to data, information and Knowledge

- **Terminologies**:
  3. **Knowledge**:
  - **Example:**

    If we now apply this information to gain further **knowledge**, we could say that:

    - 4, 8, 12 and 16 are the first four answers in the 4x table (because the 3x table starts at three and goes up in threes the 4x table must start at four and go up in fours)
    - The tallest student is 175.3cm.
    - A lion is not a household pet as it is not in the list, and it lives in the wild.

Introduction to data, information and Knowledge

- **Activities:**

1. Differentiate data, information and Knowledge.
2. Describe how data becomes knowledge.
3. 5, 10, 15, 20 are items of data. Explain how these could become information and what knowledge could be gained from them.

Design and implementation aspect of OLTP

- **OLTP: (online transaction processing)**

  - OLTP or online transactional processing is a software program or operating system that supports transaction-oriented applications in a three-tier architecture. It facilitates and supports the execution of many real-time transactions in a database.

  - OLTP monitors daily transactions and is typically done over an internet-based multi-access environment. It handles query processing and, at the same time, ensures and protects data integrity.

  - The efficacy of OLTP is determined by the number of transactions per second that it can process. OLTP systems are optimized for transactional superiority hence, suitable for most monetary transactions.

Design and implementation aspect of OLTP

- **OLTP:**
  - OLTP systems activities consist of gathering input data, processing the data, and updating it using the collected data. OLTP is usually supported by a database management system (DBMS) and operates in a client-server system. It also relies on advanced transaction management systems to facilitate multiple concurrent updates.

Design and implementation aspect of OLTP

- **OLTP transaction real-time applications:**

  - OLTP systems facilitate many types of financial and non-financial transactions such as:

    •Automated teller machines (ATMs)
    •Online banking applications
    •Online bookings for airline ticketing, hotel reservations, etc.
    •Online and in-store credit card payment processing
    •Order entry
    •E-commerce and in-store purchases
    •Password changes and sending text messages

Design and implementation aspect of OLTP

- **OLTP transaction real-time applications:**

  - Let's consider the ATM example here:

Assume that a couple has a joint account with a bank. One day both simultaneously reach different ATM centers at precisely the same time and want to withdraw the total amount present in their bank account.

However, the person that completes the authentication process first will be able to get money. In this case, the OLTP system makes sure that the withdrawn amount will be never more than the amount present in the bank. That means, OLTP systems are optimized for transactional superiority instead of data analysis.

Design and implementation aspect of OLTP

- **Characteristics of OLTP :**

1. **Short response time**

   OLTP systems maintain very short response times to be effective for users. For example, responses from an ATM operation need to be quick to make the process effective, worthwhile, and convenient

2. **Process Small Transactions:**

   OLTP systems support numerous small transactions with a small amount of data executed simultaneously over the network. It can be a mixture of queries and Data Manipulation Language (DML) overload. The queries normally include insertions, deletions, updates, and related actions. Response time measures the effectiveness of OLTP transactions, and millisecond responses are becoming common.

Design and implementation aspect of OLTP

- **Characteristics of OLTP :**

3. **Data Maintenance Operations**

   Data maintenance operations are data-intensive computational reporting and data update programs that run alongside OLTP systems without interfering with user queries.

4. **High-level transaction volume and multi-user access**

   OLTP systems are synonymous with many users accessing the same data at the same time. Online purchases of a popular or trending gadget such as an iPhone may involve an enormous number of users all vying for the same product. The system is built to handle such situations expertly.

Design and implementation aspect of OLTP

- **Characteristics of OLTP :**

5. **Very high concurrency**

   An OLTP environment experiences very high concurrency due to the large user population, small transactions, and very short response times. However, data integrity is maintained by a concurrency algorithm, which prevents two or more users from altering the same data at the same time. It prevents double bookings or allocations in online ticketing and sales, respectively.

   A mobile money transfer application is a good example where concurrency is very high as thousands of users can be making transfers simultaneously on the platform at every time of the day.

Design and implementation aspect of OLTP

- **Characteristics of OLTP :**

6. **Round-the-clock availability**

   OLTP systems often need to be available round the clock, 24/7, without interruption. A small period of unavailability or offline operations can significantly impact many people and an equally huge transaction quantity.

   Downtimes can also pose potential losses to organizations, e.g., an online banking system downtime has adverse consequences to the bank's bottom line. Therefore, an OLTP system requires frequent, regular, and incremental backup.

7. **Indexed data sets**

   Index data sets are used to facilitate rapid query, search, and retrieval.

Design and implementation aspect of OLTP

- **Characteristics of OLTP :**

8. **Data usage patterns**

   OLTP systems experience periods of both high data usage and low data usage. Finance-related OLTP systems typically see high data usage during month ends when financial obligations are settled.

9. **Normalized Schema**

   OLTP systems utilize a fully normalized schema for database consistency. That means it helps us to reduce data redundancy and improve data integrity.

10. **Storage**

    OLTP stores data records for the past few days or about a week. It supports sophisticated data models and tables.

Design and implementation aspect of OLTP

- **Architecture of OLTP**

1. **Business Strategy**

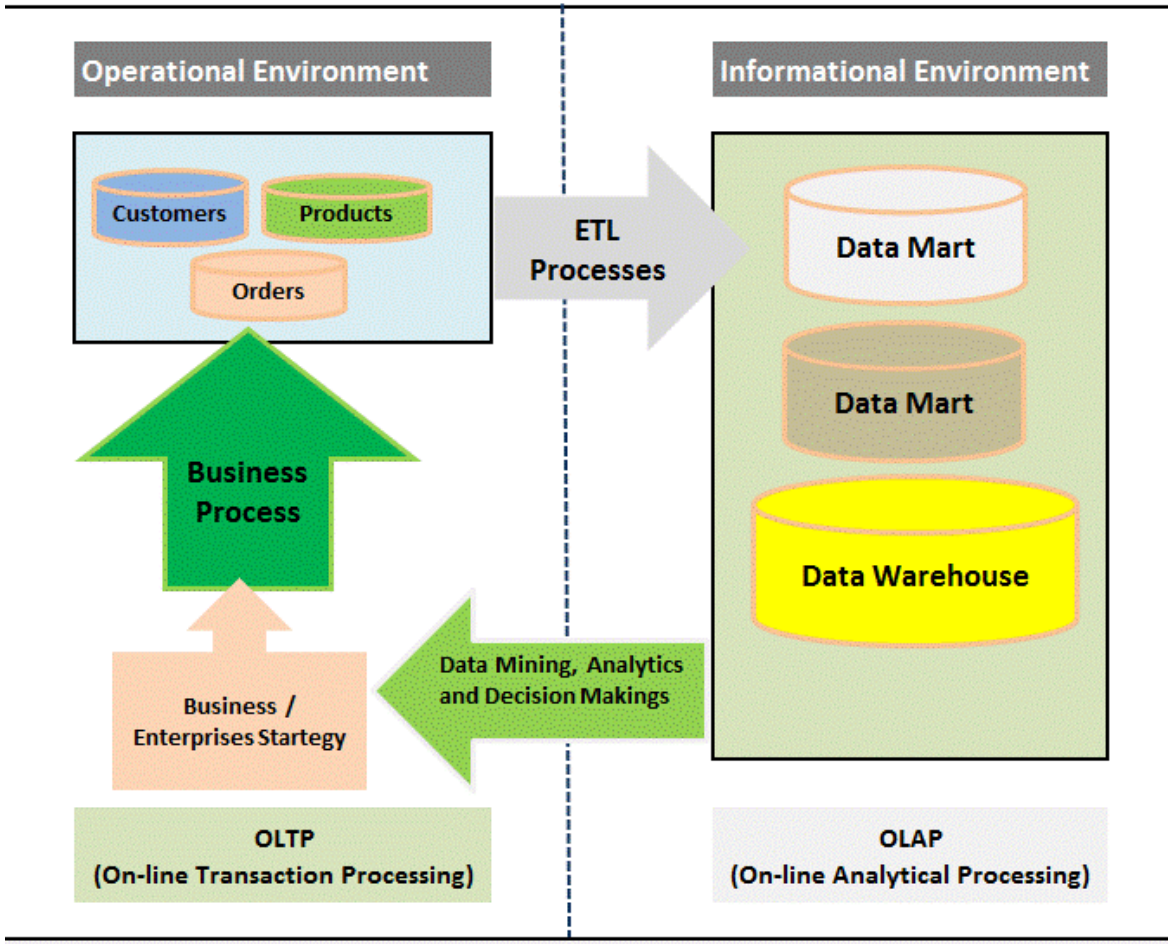
   The business strategy influences the OLTP systems design. The strategy is formulated at the senior management and the level of the board of directors.

2. **Business Process**

   They are processes by the OLTP system that will accomplish the goals set by the business strategy. The processes comprise a set of activities, tasks, and actions.

3. **Product, customer, Orders, transactions**

   The OLTP database contains information on products, transactions, employees, and customers, and suppliers.



Basic OLTP architecture in collaboration with OLAP

Design and implementation aspect of OLTP

- **Architecture of OLTP :**

4. **Extract, Transform, Load (ETL) process**

   The ETL process extracts data from the OLTP database and transforms it into the staging area, which includes data cleansing and optimizing the data for analysis. The transformed data is then loaded into the online analytical processing (OLAP) database, which is synonymous with the data warehouse environment.

5. **Data Warehouse and Data Mart**

   Data warehouses are central repositories of integrated data from one or more incongruent sources. A data mart is an access layer of the data warehouse that is used to access specific/summarized information of a unit or department.

6. **Data mining, analytics and decision Making**

   The data stored in the data warehouse and data mart is used for analysis, data mining, and decision making.

Design and implementation aspect of OLTP

- **System design aspect of OLTP:**

  The defining characteristics of OLTP transactions are atomicity and concurrency. Concurrency prevents multiple users from changing the same data simultaneously. Atomicity (or indivisibility) ensures that all transactional steps are completed for the transaction to be successful. If one step fails or is incomplete, the entire transaction fails.

  Atomic statefulness is a computing condition in which database changes are permanent, requiring transactions to be completed successfully. OLTP systems enable inserting, deleting, changing, and querying data in a database.

Design and implementation aspect of OLTP

- **System design aspect of OLTP:**

  Therefore, designing an OLTP system requires knowing its base characteristics such as atomicity, concurrency, and integrity and avoiding excessive use of clusters and indexes. The following factors should be considered in OLTP design.

Elements	Comment
Rollback segment	It is a part of DB that record transactions in case one is rolled back. They provide read consistency, database recovery and transaction rollback.
Discrete transaction	They defer changes to the data until a transaction is committed.
Clusters	A cluster is a schema that contains one or more tables with columns.
Data block size	It should be within the maximum limit, it avoids unnecessary I/O
Buffer cache size	To avoid unnecessary resource consumption, SQL statements should be tuned to use the database buffer cache
Dynamic allocation	Dynamic allocation of space to tables and rollback segments.
Database partitioning	Partitioning the database increases performance of sites that have frequent transactions, at the same time maintaining security and availability.
Database tuning	OLTP can maximize performance swiftly and effectively.

Design and implementation aspect of OLTP

- **Activity**

    - Define the Benefits of OLTP systems.
    - Define the Drawbacks of OLTP systems.

Introduction to BI and Business Models

- **What is Business Intelligence?**

  - If you want to track the term back in years, well it goes way back in 1989 when Howard Dresner defined it.

  - According to Mr. Dresner of the Gartner group, a set of concepts and methodologies to improve decision making in business trough use of facts and fact-based systems.

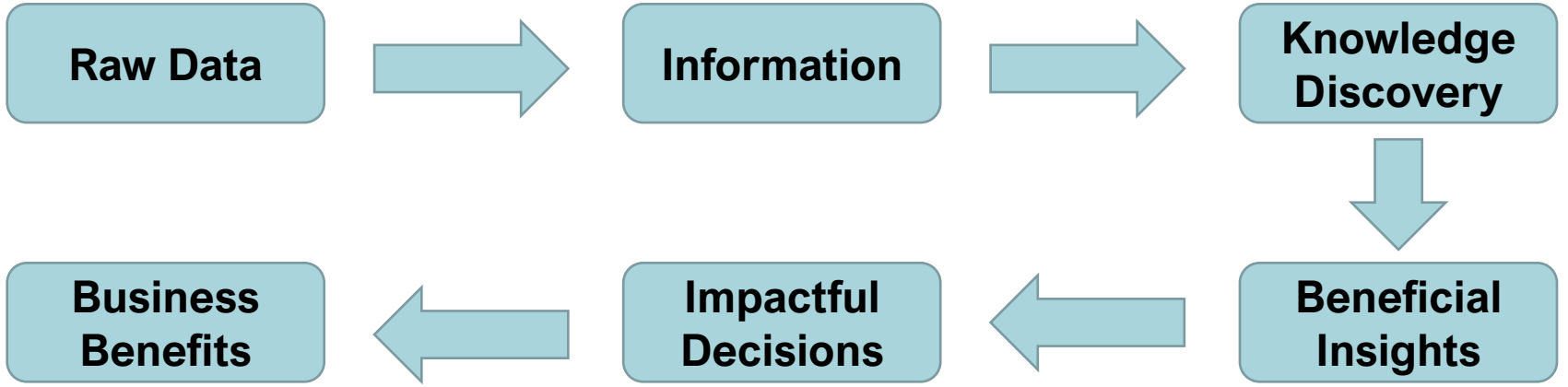
  - The main goal of BI is to improve decision making.
  - And it is a fact-based systems, that means decisions are no longer made on gut feeling or purely on guesses or on hunch. It must be backed by facts.

Introduction to BI and Business Models

- **What is Business Intelligence?**

  - The other task of BI is to mine ***information*** to provide some ***knowledge*** and apply that knowledge to provide beneficial insights.

  - the insights then lead to impactful decision making which in turn provides business benefits such as increased profitability, increased productivity, reduced costs, improved operations, etc.



Introduction to BI and Business Models

- **What was there before BI?**

  - Even before the BI evolution, the decision makers invest in obtaining the market facts and internal functions such as finance and marketing sales to evolve business strategies and plans.

  - They would usually invest in **Market research** (some companies still do invest in market research). Market research would help them to understand better marketplace in which the enterprise is operating or would want to see a future growth.

  - Enterprises also employ third-party agencies to conduct **market surveys or consumer surveys** and competitive analysis. They also use benchmark data to understand their strengths, and specific market opportunities they could exploit as well as risks that might their revenue or market share.

Introduction to BI and Business Models

- **Characteristics of Business Intelligence**
  - **Single version of truth**
  - **Multiple perspective on same problem**
  - **Geographical independence**
  - **Fact-based decision.**

Introduction to BI and Business Models

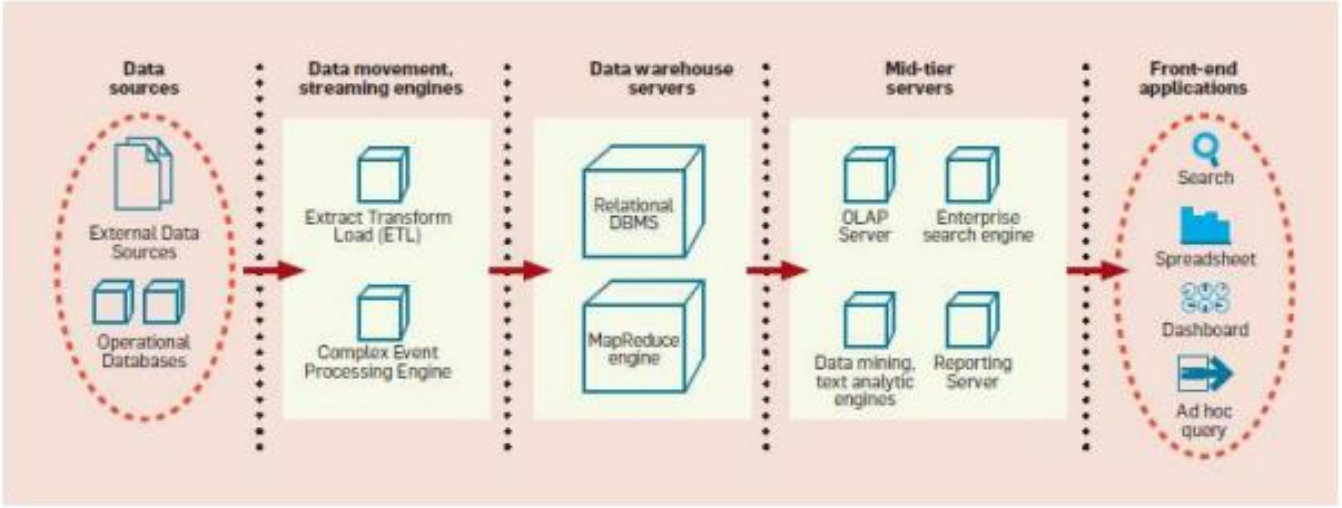
How Business Intelligence Works?

- A business intelligence architecture includes more than just a BI software. Business intelligence data is typically stored in a data warehouse built for an entire organization or in smaller data marts that hold subsets of business information for individual departments and business units, often with ties to an enterprise data warehouse.

- **After having the data in our hands, BI process usually includes**
  - Data preparation
  - Analytical solutions
  - Distribution of key performance indicators and related findings
  - Use of the information to help influence and drive business decisions.

Introduction to BI and Business Models

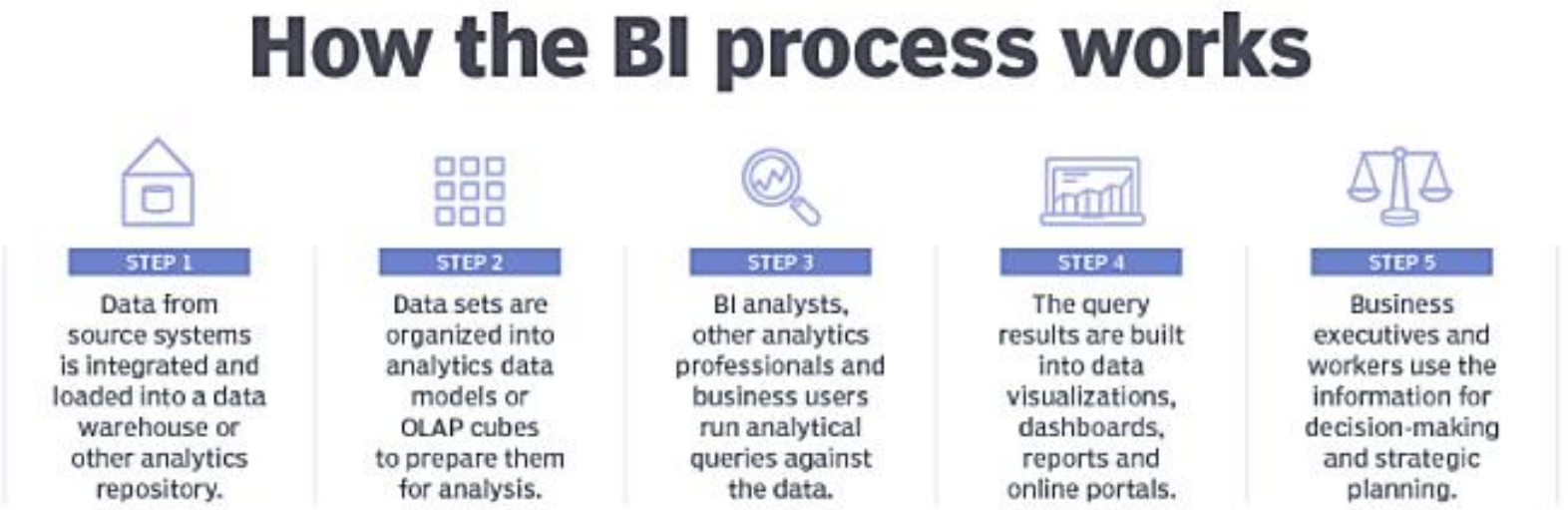
How Business Intelligence Works?



A typical BI Architecture

Introduction to BI and Business Models

How Business Intelligence Works?



Introduction to BI and Business Models

Business Models

- It is important to understand business model while you try to solve a business problem. If you understand the business model it will help you to ask the right questions to the stakeholders, to design a systematic planning and research and help you to discover some meaningful insights.

- **What is a business model?**

  A business model usually answers the following questions:

  - What product/service a company will sell?
  - How it intends to market that product or service?
  - What kind of expenses it will face?
  - How it expects to turn a profit?

Introduction to BI and Business Models

Business Models

- While choosing a business model, following key points should be considered:
  - **Value proposition :** Product should be attractive to customers
  - **Target market:** A group of consumers such as particular firms or product specific organization would be interested in the product.
  - **Competitive advantage:** Immutable product's characteristics.
  - **Cost structure:** Expenditure required to function the business.
  - **Key metrics:** Measurement for Success.
  - **Resources:** The physical, financial and intellectual assets of the organization.
  - **Problem solving:** Strategies for solving customer's problems.
  - **Revenue model:** The multiple ways your company can generate income.
  - **Profit margin:** The amount your revenue exceeds business costs.

Introduction to BI and Business Models

Business Models

- Most well-known types of business models.
  - Subscription model
  - Freemium model
  - Product to service model
  - Crowdsourcing model
  - Franchise model
  - Distribution model
  - Manufacturer model
  - Retailer model.

Design and Implementation aspects of OLAP/DW

Overview of OLAP

- In 2nd chapter we saw the design aspects of OLTP and within that we also saw how OLTP is linked to OLAP. OLAP is in fact the technology behind many BI applications.

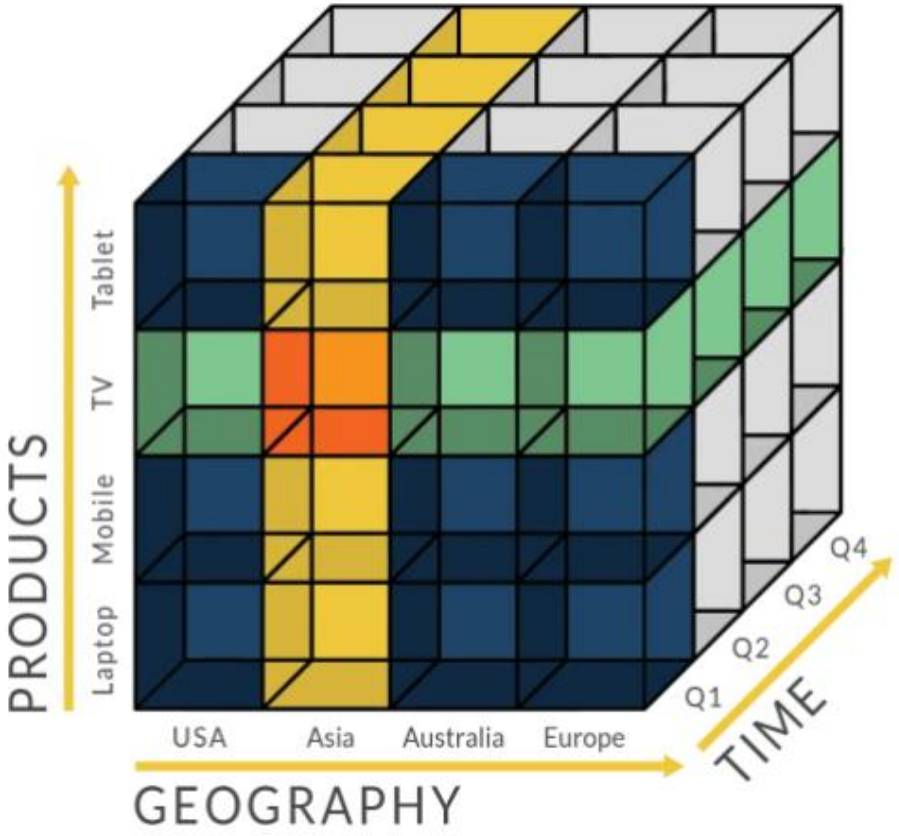
- On-Line Analytical Processing (OLAP) is a decision support tool that allows users to analyze different dimensions of multidimensional data.

- OLAP structures data hierarchically to reflect the real dimensionality of the enterprise as *understood by the users*. Users can pivot, filter, drill down and drill up data and generate numbers of views with simple mouse manipulations. It is commonly used in business reporting for sales, marketing, management, data mining, and similar areas.

Design and Implementation aspects of OLAP/DW

- An OLAP structure created from the operational data is called an OLAP cube. Following figure shows, the cube holds data more like a 3D spreadsheet rather than a relational database, allowing different views of the data to be quickly displayed.

- In multidimensional OLAP (MOLAP) databases, cubes are created and stored physically, whereas in relational OLAP (ROLAP) databases, cubes are virtually created, based on a star or snowflake schema



Design and Implementation aspects of OLAP/DW

- E. F Codd coined the term OLAP and based the principles defined by E. F Codd, following are the 12 rules that OLAP technologies/ implementation must support.

  - ***Multidimensional conceptual view:*** Must support EIS (Executive Information System) slice and dice operations and is usually required in financial modeling.
  - ***Transparency:*** Is part of an open system that supports heterogeneous data sources. Furthermore, the end user should not be concerned about the details of data access or conversions.
  - ***Accessibility***: Presents the user with a single logical schema of the data. OLAP engines act as middleware, sitting between heterogeneous data sources and an OLAP front-end.
  - ***Consistent reporting performance***: Performance should not degrade as the number of dimensions in the model increases.
  - ***Generic dimensionality:*** Not limited to 3-D and not biased toward any dimension. A function applied to one dimension should also be able to be applied to another.

Design and Implementation aspects of OLAP/DW

- ***Client/server architecture:*** Requires open, modular systems. Not only the product should be client/server but the server component of an OLAP product should allow that various clients could be attached with minimum effort and programming for integration.
- ***Dynamic sparse-matrix handling***: Related both to the idea of nulls in relational databases and to the notion of compressing large files, a sparse matrix is one in which not every cell contains data. OLAP systems should accommodate varying storage and data-handling options.
- ***Multiuser support:*** Supports multiple concurrent users, including their individual views or slices of a common database.

- ***Unrestricted cross-dimensional operations***: All dimensions are created equal, so all forms of calculation must be allowed across all dimensions, not just the measures dimension.

Design and Implementation aspects of OLAP/DW

- ***Intuitive data manipulation:*** Users shouldn't have to use menus or perform complex multiple step operations when an intuitive drag and drop action will do.

- ***Flexible reporting***: Users should be able to print just what they need, and any changes to the underlying model should be automatically reflected in reports.

- ***Unlimited dimensional and aggregation levels***: Supports at least 15, and preferably 20, dimensions.

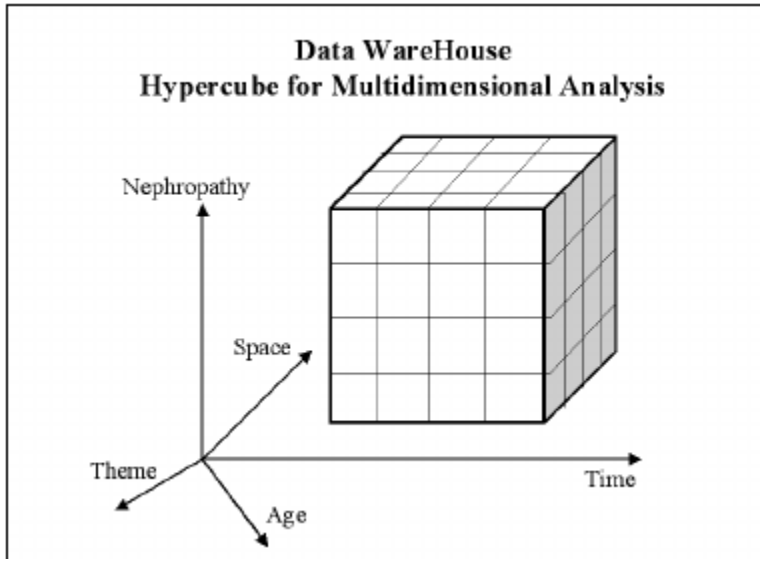
Design and Implementation aspects of OLAP/DW

OLAP Architectures:

- Regardless of the different types, all OLAP architectures involve building a multidimensional data structure (cubes), where dimensions represent business entities such as sales regions and products or natural entities such as time and geography.

- Using multidimensional databases can present their data for an application using two types of cubes**: hypercube and multicubes.**

- In a **hypercube**, as shown in figure, all data appears logically as a single cube. All parts of the manifold represented by this hypercube have identical dimensionality. Each dimension belongs to one cube only. A dimension is ***owned*** by the hypercube. This simplicity makes easy for users to understand.



Design and Implementation aspects of OLAP/DW

OLAP Architectures:

- Designing **a hypercube** model is a top-down process with three major steps:

1. decide which process of the business you want to capture in the model, such as sales activity.
2. identify the values that you want to capture, such as sales amounts. This information is always numeric
3. identify the granularity of the data, meaning the lowest level of detail at which you want to capture. These elements are the dimensions

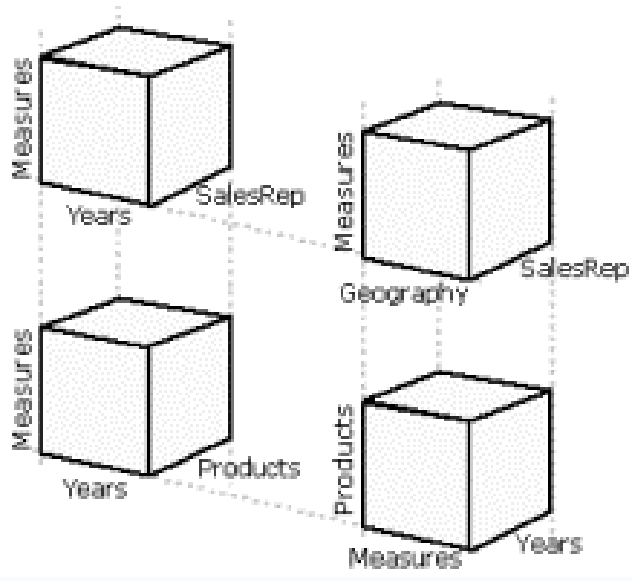
Design and Implementation aspects of OLAP/DW

OLAP Architectures:

- In the **multicube** model, data is segmented into a set of smaller cubes, each of which is composed of a subset of the available dimensions, as shown in Figure. They are used to handle multiple fact tables, each with different dimensionality.

- A dimension can be part of multiple cubes. Dimensions are ***not owned*** by any one cube, like under the hypercube model. Rather, they are available to all cubes, or there can be some dimensions that do not belong to any cube.

- This makes it much more efficient and versatile. It is also a more efficient way of storing very sparse data, and it can reduce the pre-calculation database explosion effect.



Design and Implementation aspects of OLAP/DW

OLAP Architectures:

- ***When and how is the multidimensional data structure constructed?***
  - Some OLAP technologies require an ETL (extract, transform, and load) process, which typically runs at off-peak usage times to build and update a persistent multidimensional data structure.
  - On the other hand, other OLAP technologies access source data directly to build and present multidimensional data on the fly as the user performs analysis.

Design and Implementation aspects of OLAP/DW

OLAP Architectures:

- ***Where does the multidimensional data structure live and how persistent is it?***

The multidimensional data structure may reside in a persistent, dedicated multidimensional database, a hypercube temporarily cached in memory, or a star schema/snowflake schema stored in a relational database.

Design and Implementation aspects of OLAP/DW

OLAP Architectures:

- **OLAP comes in various types or format such as**
  1. ***MOLAP (Multidimensional OLAP).***

  Multidimensional On-Line Analytical Processing (MOLAP) is the more traditional way of OLAP analysis, in which, data is stored in a multidimensional cube. This allows users to view different aspects of data aggregates such as sales by time period, geography, or product. (A relational database is not used for storage)

  MOLAP processes data that is already stored in a multidimensional array in which all possible combinations of data are reflected, each in a cell that can be accessed directly.

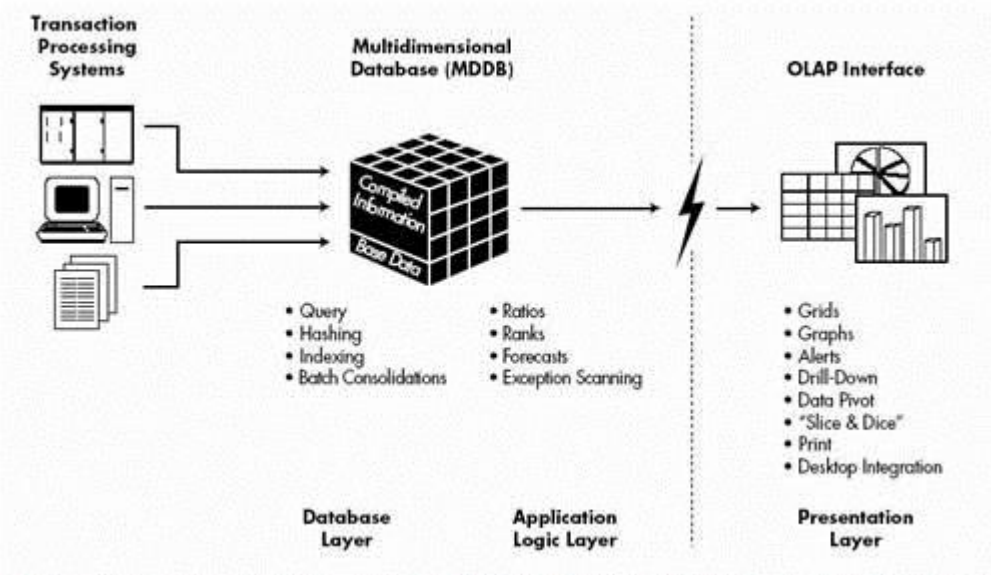
  MOLAP is a good choice when data is needed frequently and for rapid query response.

Design and Implementation aspects of OLAP/DW

OLAP Architectures:

1. MOLAP (Multidimensional OLAP)

MOLAP is a two-tier, client/server architecture. The multidimensional database serves as both the **database layer and the application logic layer**.



database layer is responsible for all data storage, access, and retrieval processes while the **application logic layer** is responsible for the execution of all OLAP requests.

The presentation layer integrates with the application logic layer and provides an interface through which the users view and request OLAP analyses.

The client/server architecture allows multiple users to access the same multidimensional database.

Design and Implementation aspects of OLAP/DW

OLAP Architectures:

1. MOLAP (Multidimensional OLAP)

In MOLAP, the data must be stored *multidimensionally* to be viewed multidimensionally.

Data from various operational systems is loaded into a ***multidimensional database*** through a series of batch routines. Once this atomic data has been loaded into the multidimensional database, the general approach is to perform a series of calculations in batch to aggregate along the dimensions and fill the multidimensional array structures. Then indices are created, and hashing algorithms are used to improve query access time.

When users request OLAP reports through the interface, the application logic layer of the multidimensional database retrieves the stored data. It reads the pre-compiled data and has limited capabilities to dynamically create aggregations or to calculate metrics that have not been pre-calculated and stored.

Design and Implementation aspects of OLAP/DW

OLAP Architectures:

2. ROLAP (Relational OLAP)

Relational On-Line Analytical Processing (ROLAP) performs dynamic multidimensional analysis of data stored in a relational database, rather than in a multidimensional database.

The design may be structured in the form of a star or its variations. A typical use of ROLAP is for large data size that is infrequently queried, such as historical data.

ROLAP accesses data stored in a data warehouse (relational database) to provide OLAP analyses.

Design and Implementation aspects of OLAP/DW

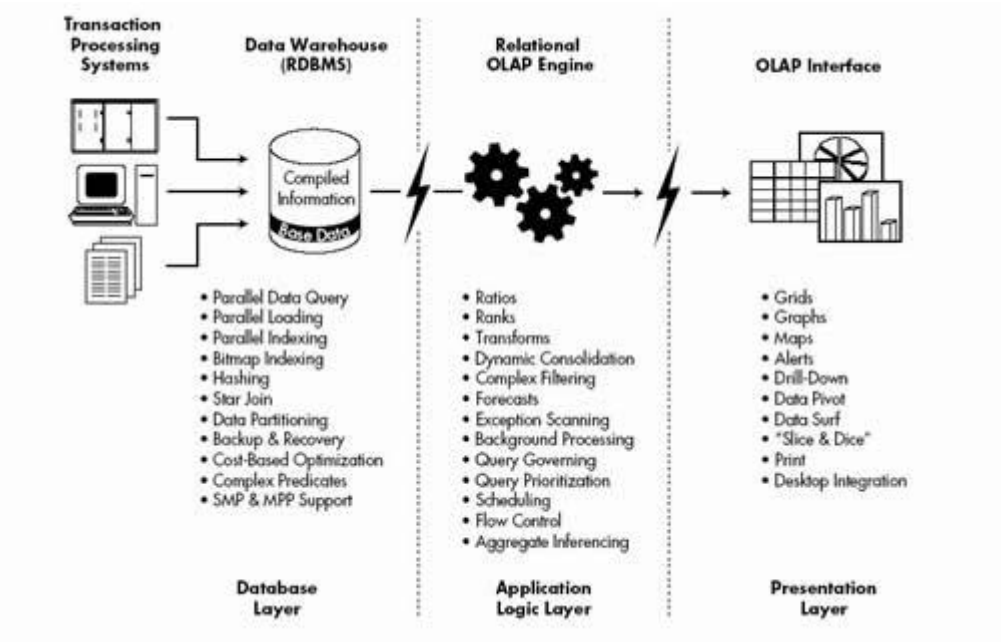
OLAP Architectures:

2. ROLAP (Relational OLAP)

ROLAP is a three-tier, client/server architecture. **The database layer** utilizes relational databases for data storage, access, and retrieval processes.

The application logic layer is the ROLAP engine which executes the multidimensional reports from multiple users.

The ROLAP engine integrates with a variety of **presentation layers**, through which users perform OLAP analyses.



Design and Implementation aspects of OLAP/DW

OLAP Architectures:

2. ROLAP (Relational OLAP)

After defining the data model for the data warehouse, data from OLTP systems is loaded into the database. Database routines are run to aggregate the data, if required by the data model. Then indices are created to optimize query access time. Users submit multidimensional analyses to the ROLAP engine, which then dynamically transforms the requests into SQL execution plans.

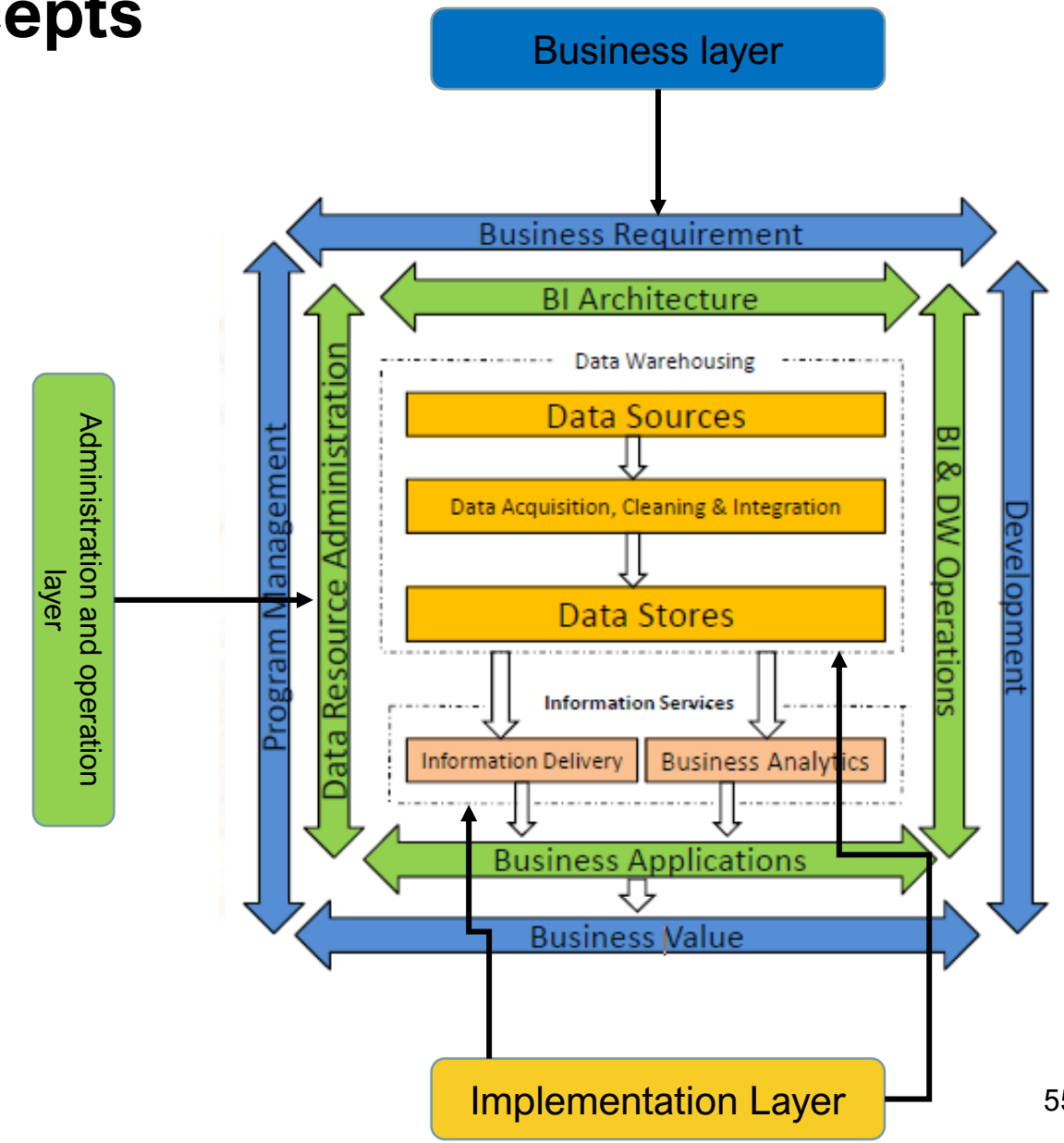
The SQL is submitted to the relational database for processing, the relational query results are cross-tabulated. A multidimensional result set is returned to the users. It can utilize pre-calculated results when they are available, or dynamically generating results from atomic information when necessary.

The ROLAP architecture was invented to directly access data from data warehouses, thereby supporting optimization techniques to meet batch window requirements and providing fast response time. These optimization techniques include application-level table partitioning, aggregate inferencing, denormalization support, and multiple fact table joins.

BI system components and concepts

In BI framework there are usually **four components:**

1. **Business layer**
2. **Administration and operation layer**
3. **Implementation layer**



BI system components and concepts

1. Business Layer

In Business layer there are four components included

a) **Business Requirements:** The requirements of a business are mapped using following three processes.

Business Drivers : Responsible for initiating the need to act. ***For instance***, changing workforce, changing labor laws, change in economy, rules and regulations and technology

Business goals: Responsible for making sure that the targets to be achieved in response to the drivers. For instance, increased productivity, improved market share, improved profit margins, improved customer satisfaction and cost reduction.

Business Strategies: The planned course of actions that will help achieve the defined goals For instance, global deliver model, finding proper partnerships, customer and employee retention programs etc.

BI system components and concepts

1. Business Layer

In Business layer there are four components included

b) Business Value: When a strategy is implemented against certain business goals, then certain costs (monetary, time, effort, information produced by data integration and analysis, application of knowledge from experience, etc.) are involved. However, the final output of the process should create such value for the business whose ratio to the costs involved should be a feasible ratio.

The business value can be measured in the terms of ROI (Return on Investment), ROA (Return on

Assets), TCO (Total Cost of Ownership), TVO(Total Value of Ownership),

BI system components and concepts

1. Business Layer

In Business layer there are four components included

b) Business Value:

Return on Investment (ROI) : Consider you are working for a company which has an online community platform that allows their prospective clients to engage with their users and for this operation your company are using social media platforms to help get news clients and to increase the number of prospects leads. They attribute 10% of their daily revenue to social media. So, this will be the return on investment from social media.

Return on Asset (ROA) : ROA is the earning form the invested capital or asset. Consider your company has a net income of $1 million and has total assets of $5 million. Then its ROA is 20%. ROAs over 5% are generally considered good and over 20% excellent.

BI system components and concepts

1. Business Layer

In Business layer there are four components included

b) Business Value:

Total Cost of Ownership (TCO): TCO defines the cost of owning a business from the time of purchase / formation by the owner, through its operation and maintenance to the time it leaves the possession of the owner.

Total Value of Ownership (TVO) : TVO has replaced the simple concept of Owner's Equity in some companies. It could include a variety of subcategories such as stock, undistributed dividends, retained earnings or profit, or excess capital contributed. (benefits of alternative investments)

BI system components and concepts

1. Business Layer

In Business layer there are four components included

c) Program Management

Project management ensures that people, projects and priorities work in a way individual processes are compatible with each other to ensure seamless integration and smooth functioning of the entire program. It mainly focuses on the following:

- Business priorities
- Mission and goals
- Strategies and risks
- Multiple projects
- Dependencies
- Coast and values
- Business rules
- Infrastructure.

BI system components and concepts

1. Business Layer

In Business layer there are four components included

d) Development

The process of development consists of

- Database/data-warehouse development which consist of ETL, data profiling, data cleansing and database tools.
- Data integration system development which contains data quality tools and data integration tools.
- Business analytics development which contains processes and various technologies used.

BI system components and concepts

2. Administration and Operation Layer

In Administration and operation layer there four components :

a) Bi Architecture:

Bi architecture defines that **data** should follow design standards, it must have a logically suitable data model and it’s metadata should be of high standards

During **Integration** process, certain processing standards have to be followed and should perform according to business semantics and rules.

Data must be consistent.

BI system components and concepts

2. Administration and Operation Layer

In Administration and operation layer there four components :

a) Bi Architecture:

Information derived from the data that has been integrated should be usable, findable and as per the requirements.

Technology used for deriving information must be accessible, it should have a good UI and should support analysis, decision support, data and storage management.

Organization must consist of different roles and responsibilities, like management, development, support and usage roles.

BI system components and concepts

2. Administration and Operation Layer

In Administration and operation layer there four components :

b) BI and DW operations: Data Warehouse (DW) administration requires the usage of various tools to monitor the performance and usage of the warehouse, and perform administrative tasks on it. Some of these tools would be:

- Backup and restore
- Security
- Configuration management
- Database management

BI system components and concepts

2. Administration and Operation Layer

In Administration and operation layer there four components :

c) Data resource administration

In general data resource administration involves:
data governance and metadata management.

BI system components and concepts

2. Administration and Operation Layer

c) Data resource administration

Data Governance : In data governance, techniques for controlling data quality, which is used to assess, improve, manage and maintain information are defined. In helps to define standards that are required to maintain data quality. Data governance includes following roles:

- **Data ownership**: Data owners are either individuals or teams who make decisions such as who has the right to access and edit data and how it is used. Owners may not work with their data every day but are responsible for overseeing and protecting data domain.

BI system components and concepts

2. Administration and Operation Layer

c) Data resource administration

Data Governance :

- **Data stewardship**: It is the collection of practices that ensures, and organization's data is accessible, usable , safe and trusted. It includes overseeing every aspect of the *data lifecycle*: creating, preparing, using, storing, archiving, and deleting data, in accordance with an organization's established data governance principles for promoting data quality and integrity.

BI system components and concepts

2. Administration and Operation Layer

c) Data resource administration

Data Governance :

- **Data custodianship:** Data custodianship is responsible for identifying topics, mange discussions, recommend solutions and strategies and assess effectiveness regarding data and information custodianship.

BI system components and concepts

2. Administration and Operation Layer

c) Data resource administration

Metadata Management: Metadata Management is an organization-wide agreement on how to describe information assets. Through metadata management, organizations can link, use and discover their data. This makes them more transparent and better prepared to evaluate the value and risks associated with data and its usage.

Few examples of metadata are timestamp at which the data was extracted, the data sources from where metadata has been extracted, and the missing fields/columns that have been added by data cleaning or integration processes. Metadata management involves tracking, assessment, and maintenance of metadata.

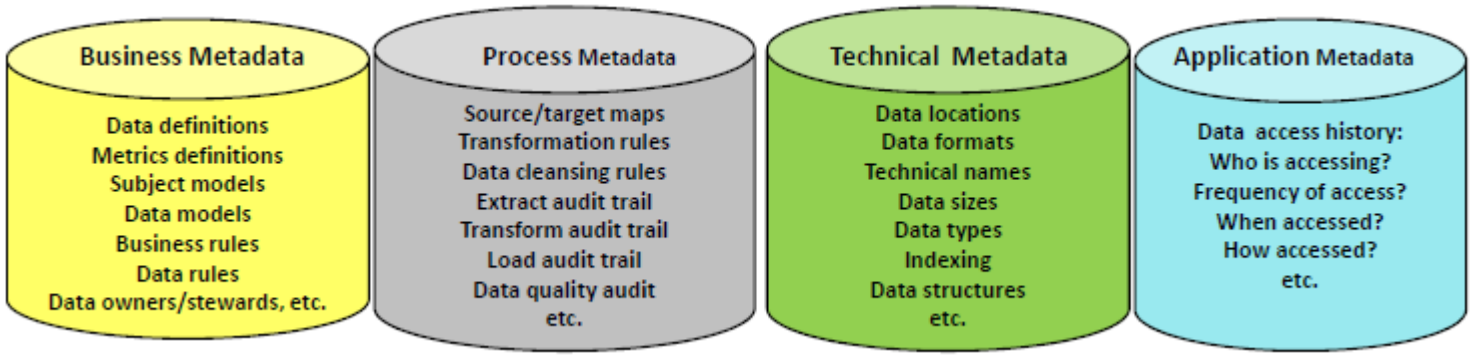
BI system components and concepts

2. Administration and Operation Layer

c) Data resource administration

Metadata Management: Further can be divided into four groups:

1. Business metadata
2. Process metadata
3. Technical metadata
4. Application metadata



BI system components and concepts

2. Administration and Operation Layer

d) **Business Applications:** The application of technology to produce value for the business refers to the generation of information or intelligence from data assets like data warehouses/data marts. Using BI tools, we can generate strategic, financial, customer, or risk intelligence. This information can be obtained through various BI applications, such as DSS (decision support system), EIS (executive information system), OLAP(On-line analytical processing), data mining and discovery, etc.

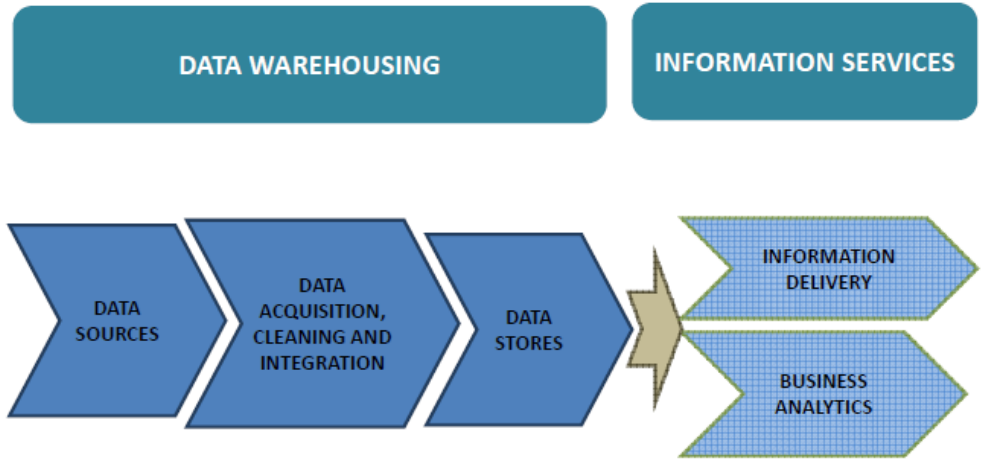
BI system components and concepts

3. Implementation Layer

The implementation layer of the BI component framework consists of technical components that are required for data capture, transformation and cleaning, data into information, and finally delivering that information to leverage business goals and produce value for the organization.

It can be divided into two :

1. **Data warehousing**
2. **Information services**



BI system components and concepts

3. **Implementation Layer**

1. **Data warehousing**: In data warehousing process, we prepare the basic repository of data that becomes the data sources where we extract information form. It is a structured multidimensional data model, which is optimized for data retrieval rather than update.

2. **Information services**: It is not only the process of producing information; rather it involves ensuring that the information produced is aligned with business requirements and can be acted upon to produce value for the company. Information is delivered in the form of KPI (key performance indicator), reports charts, dashboards etc., in the form of analytics.

Applications of BI

Homework Activity:

What are the applications of BI and how are they being applied onto the real-world problems?

At least 3 applications of BI

Role of Data Warehouse in Business Intelligence

Roles of DW are as follows:

- ***The DW system must make information easily accessible***. Data must be understandable and intuitive and obvious to the business user as well as to the developer.

- The structural integrity of data and labels should mimic the business users' thought processes and vocabulary.

- The BI tools that access the data must be simple and easy to use and must return query results to the user with minimal wait times.
- In short, data should be easy to access through BI applications and simple to understand for both users and developers and should return fast results.

Role of Data Warehouse in Business Intelligence

- ***The DW system must present information consistently.*** Data stored in DW must be credible and carefully assembled from a variety of trustworthy sources, cleansed, quality assured and released only when it is fit for user consumption.

- Consistency also implies to the labels and definitions for the DW systems' contents are used across data sources.

- If two performance measures have the same name, then they must represent the same meaning and if they are different then they should be labeled differently.

Role of Data Warehouse in Business Intelligence

- ***The DW system must adopt the changes.*** User's needs, conditions, data and technology are dynamic in nature meaning they are subject to change. The DW system must be designed to handle this inevitable change gracefully so that it does not invalidate the existing data or applications.

- Often the organization would ask new questions and assign new task or new data is added to the warehouse, in such cases the existing data and applications should not be changed or disrupted.

- If you are required to make any changes to descriptive data, you must appropriately account for the changes and make these changes transparent to the users. (maintain a change log)

Role of Data Warehouse in Business Intelligence

- ***The DW system must adopt the changes.*** User's needs, conditions, data and technology are dynamic in nature meaning they are subject to change. The DW system must be designed to handle this inevitable change gracefully so that it does not invalidate the existing data or applications.

- Often the organization would ask new questions and assign new task or new data is added to the warehouse, in such cases the existing data and applications should not be changed or disrupted.

- If you are required to make any changes to descriptive data, you must appropriately account for the changes and make these changes transparent to the users. (maintain a change log)

Role of Data Warehouse in Business Intelligence

- ***The DW system must present information in a timely way.*** As the DW system is used more intensively for operational decisions, raw data may need to be converted into actionable information within hours, minutes, or even seconds.

- The DW/BI team and business users need to have realistic expectations for what it means to deliver data when there is little time to clean or validate it.

Role of Data Warehouse in Business Intelligence

- ***The DW/BI system must be a secure bastion that protects the information assets*.** An organization's informational crown jewels are stored in the data warehouse. At a minimum, the warehouse likely contains information about what you're selling to whom at what price—potentially harmful details in the hands of the wrong people.

- The DW/BI system must effectively control access to the organization's confidential information.

Role of Data Warehouse in Business Intelligence

- ***The DW must serve as the authoritative and trustworthy foundation for improved decision making.*** The data warehouse must have the right data to support decision making. The most important outputs from a DW system are the decisions that are made based on the analytic evidence presented.

- These decisions deliver the business impact and value attributable to the DW/BI system. The original label that predates DW/BI is still the best description of what you are designing: a decision support system.

Role of Data Warehouse in Business Intelligence

- ***The business community must accept the DW to deem it successful.*** It doesn't matter that you built an elegant solution using best-of-breed products and platforms. If the business community does not embrace the DW environment and actively use it, you have failed the acceptance test.

- Unlike an operational system implementation where business users have no choice but to use the new system, DW usage is sometimes optional. Business users will embrace the DW if it is the "simple and fast" source for actionable information.

Components of Data warehouse architecture

Following architecture properties are essential for a data warehouse system:

Separation: Analytical and transactional processing should be kept apart as much as possible.

Scalability: Hardware and software architectures should be easy to upgrade as the data volume, which has to be managed and processed, and the number of users' requirements, which have to be met, progressively increase.

Extensibility: The architecture should be able to host new applications and technologies without redesigning the whole system

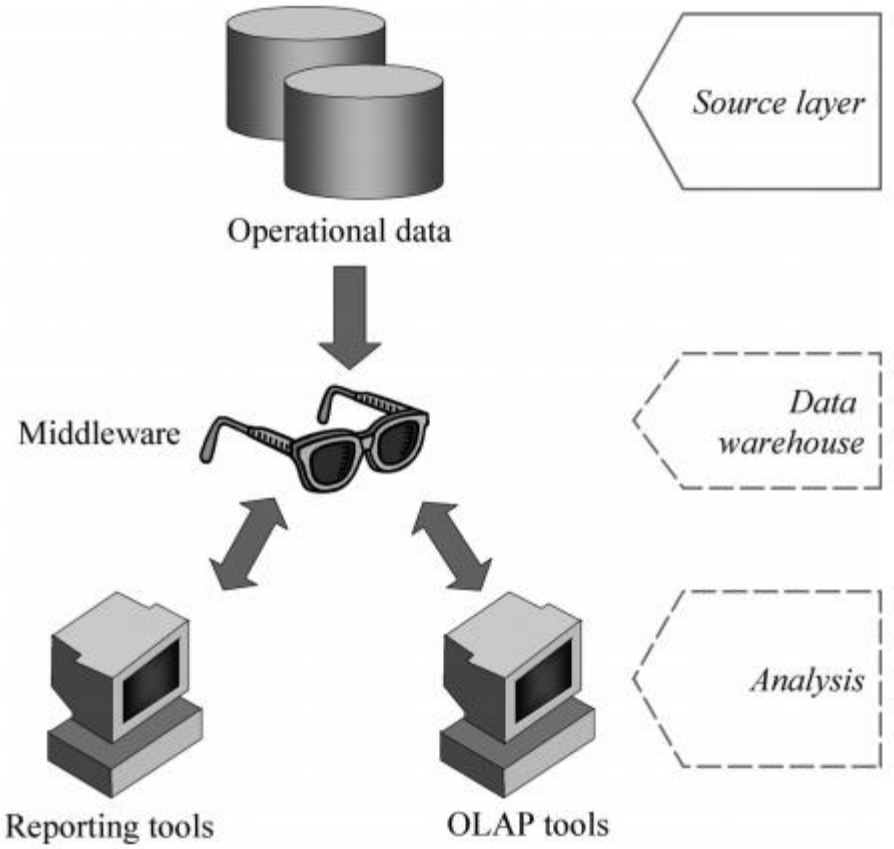
Security: Monitoring accesses is essential because of the strategic data stored in data warehouses.

Administrability: Data warehouse management should not be overly difficult

Components of Data warehouse architecture

1. **Single layer Architecture**: A single-layer architecture is not frequently used in practice. Its goal is to minimize the amount of data stored; to reach this goal, it removes data redundancies.

   here, the only layer physically available: the source layer. In this case, data warehouses are virtual.



Components of Data warehouse architecture

1. Single Layer Architecture

This means that a data warehouse is implemented as a multidimensional view of operational data created by specific middleware, or an intermediate processing layer.

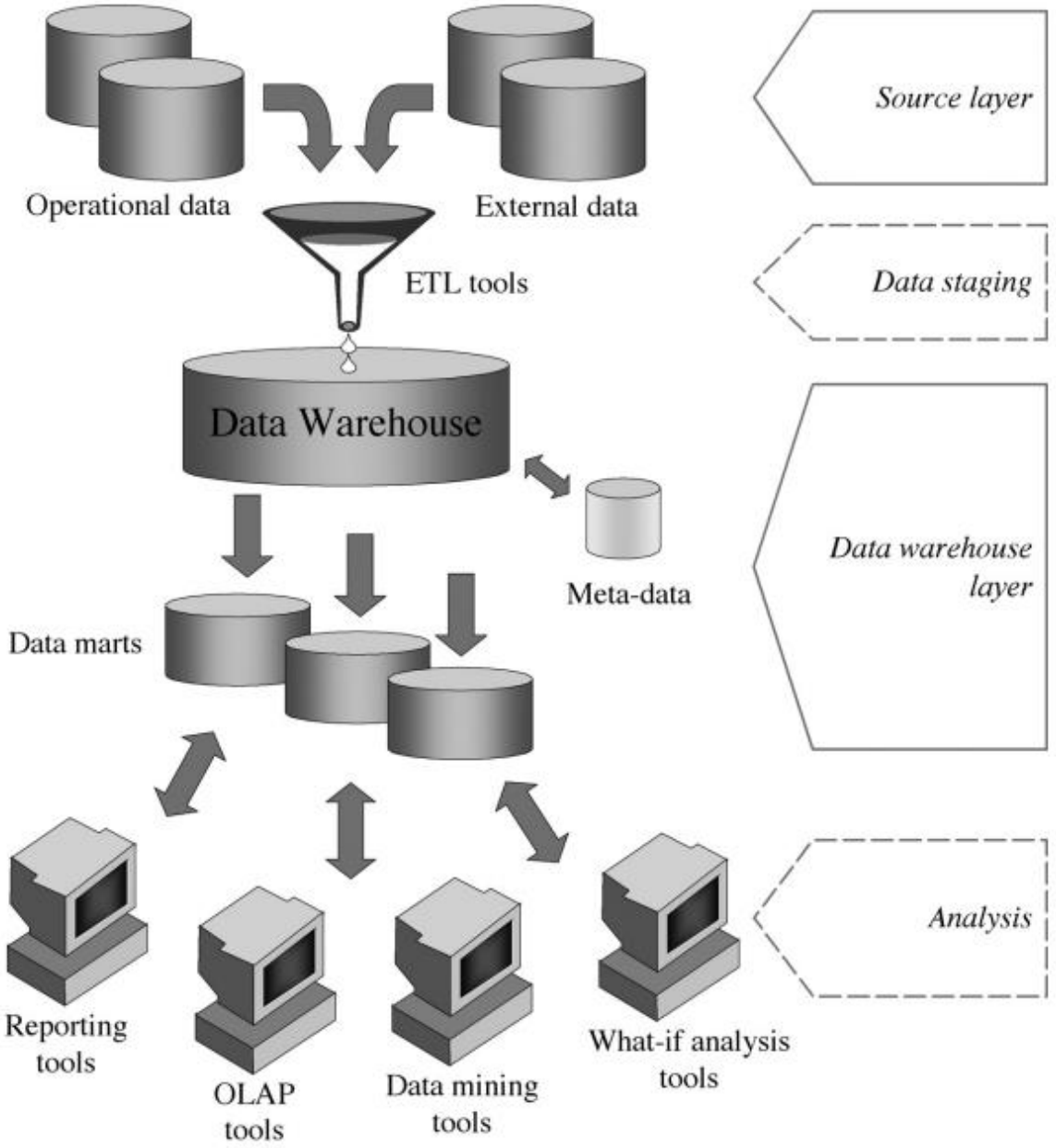
The weakness of this architecture lies in its failure to meet the requirement for separation between analytical and transactional processing. Analysis queries are submitted to operational data after the middleware interprets them. It this way, the queries affect regular transactional workloads

Components of Data warehouse

2. Two Layer Architecture

Two-layer architecture is consisted of four subsequent data flow stages.

1. **Source layer**: A data warehouse system uses heterogeneous sources of data. That data is originally stored to corporate relational databases or legacy databases, or it may come from information systems outside the corporate walls.



Components of Data warehouse architecture

2. Two-Layer Architecture

2. **Data staging**: The data stored to sources should be extracted, cleansed to remove inconsistencies and fill gaps, and integrated to merge heterogeneous sources into one common schema.

The so-called Extraction, Transformation, and Loading tools (ETL) can merge heterogeneous schemata, extract, transform, cleanse, validate, filter, and load source data into a data warehouse.

Technologically speaking, this stage deals with problems that are typical for distributed information systems, such as inconsistent data management and incompatible data structures.

Components of Data warehouse architecture

2. Two-Layer Architecture

3. Data warehouse layer: Information is stored to one logically centralized single repository: a data warehouse. The data warehouse can be directly accessed, but it can also be used as a source for creating data marts, which partially replicate data warehouse contents and are designed for specific enterprise departments.

4. **Analysis:** In this layer, integrated data is efficiently and flexibly accessed to issue reports, dynamically analyze information, and simulate hypothetical business scenarios. Technologically speaking, it should feature aggregate data navigators, complex query optimizers, and user-friendly GUIs

Components of Data warehouse architecture

2. Two-Layer Architecture

The component marked as a data warehouse in the figure is also often called ***the primary data warehouse or corporate data warehouse***. It acts as a centralized storage system for all the data being summed up.

Data marts can be viewed as small, local data warehouses replicating (and summing up as much as possible) the part of a primary data warehouse required for a specific application domain.

(***data marts:*** A data mart is a subset, or an aggregation of the data stored to a primary data warehouse. It includes a set of information pieces relevant to a specific business area, corporate department, or category of users.)

Components of Data warehouse architecture

2. Two-Layer Architecture

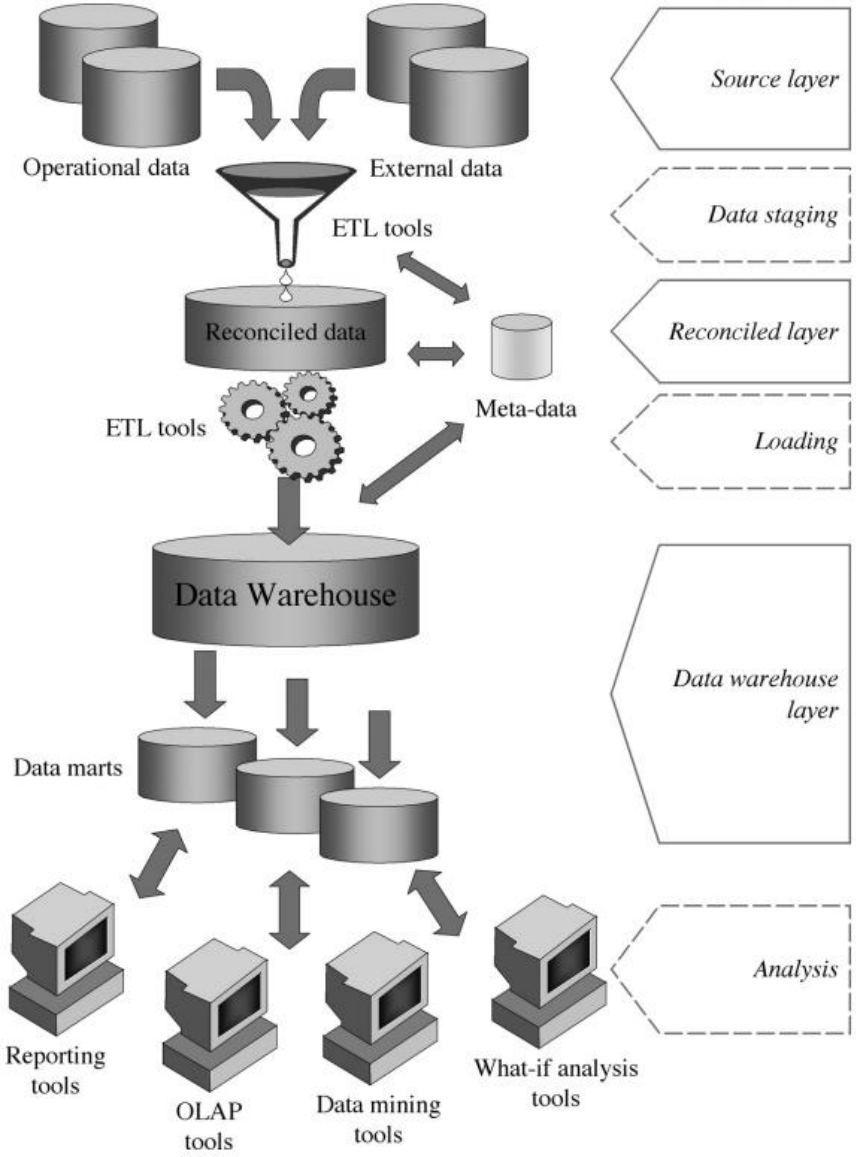
The data marts populated from a primary data warehouse are often called dependent. Although data marts are not strictly necessary, they are very useful for data warehouse systems in midsize to large enterprises because:

- they are used as building blocks while incrementally developing data warehouses;
- they mark out the information required by a specific group of users to solve queries
- they can deliver better performance because they are smaller than primary data warehouses

Components of Data warehouse architecture

3. **Three-Layer Architecture**

In this architecture, the third layer is the ***reconciled data layer or operational data store***. This layer materializes operational data obtained after integrating and cleansing source data. As a result, those data are integrated, consistent, correct, current, and detailed. The figure shows a data warehouse that is not populated from its sources directly, but from reconciled data.



Components of Data warehouse architecture

3. **Three-Layer Architecture**

The main advantage of the reconciled data layer is that it creates a common reference data model for a whole enterprise. At the same time, it sharply separates the problems of source data extraction and integration from those of data warehouse population.

However, reconciled data leads to more redundancy of operational source data. Note that we may assume that even two-layer architectures can have a reconciled layer that is not specifically materialized, but only virtual, because it is defined as a consistent integrated view of operational source data.

The end