CHENNAI INSTITUTE OF TECHNOLOGY
(Autonomous)
(Affiliated to Anna University, Approved by AICTE, Accredited by NAAC & NBA)
Sarathy Nagar, Kundrathur, Chennai – 600069, India.

UNIT II - LECTURE NOTES

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CCS341 DATA WAREHOUSING

III CSE/VI SEMESTER

UNIT II ETL AND OLAP TECHNOLOGY

What is ETL – ETL Vs ELT – Types of Data warehouses - Data warehouse Design and Modeling - Delivery Process - Online Analytical Processing (OLAP) - Characteristics of OLAP - Online Transaction Processing (OLTP) Vs OLAP - OLAP operations- Types of OLAP- ROLAP Vs MOLAPVs HOLAP.

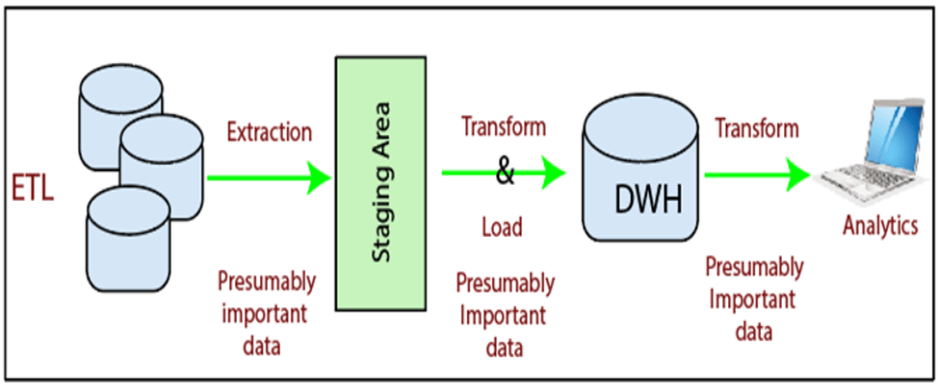
ETL (Extract, Transform, and Load) Process

What is ETL?

The mechanism of extracting information from source systems and bringing it into the data warehouse is commonly called **ETL**, which stands for **Extraction, Transformation and Loading**.

The ETL process requires active inputs from various stakeholders, including developers, analysts, testers, top executives and is technically challenging.

To maintain its value as a tool for decision-makers, Data warehouse technique needs to change with business changes. ETL is a recurring method (daily, weekly, monthly) of a Data warehouse system and needs to be agile, automated, and well documented.



How ETL Works?

ETL consists of three separate phases:

- Extraction is the operation of extracting information from a source system for further use in a data warehouse environment. This is the first stage of the ETL process.
- Extraction process is often one of the most time-consuming tasks in the ETL.
- The source systems might be complicated and poorly documented, and thus determining which data needs to be extracted can be difficult.
- The data has to be extracted several times in a periodic manner to supply all changed data to the warehouse and keep it up-to-date.

Cleansing

The cleansing stage is crucial in a data warehouse technique because it is supposed to improve data quality. The primary data cleansing features found in ETL tools are rectification and homogenization. They use specific dictionaries to rectify typing mistakes and to recognize synonyms, as well as rule-based cleansing to enforce domain-specific rules and defines appropriate associations between values.

The following examples show the essential of data cleaning:

If an enterprise wishes to contact its users or its suppliers, a complete, accurate and up-to-date list of contact addresses, email addresses and telephone numbers must be available.

If a client or supplier calls, the staff responding should be quickly able to find the person in the enterprise database, but this need that the caller's name or his/her company name is listed in the database.
If a user appears in the databases with two or more slightly different names or different account numbers, it becomes difficult to update the customer's information.

Transformation

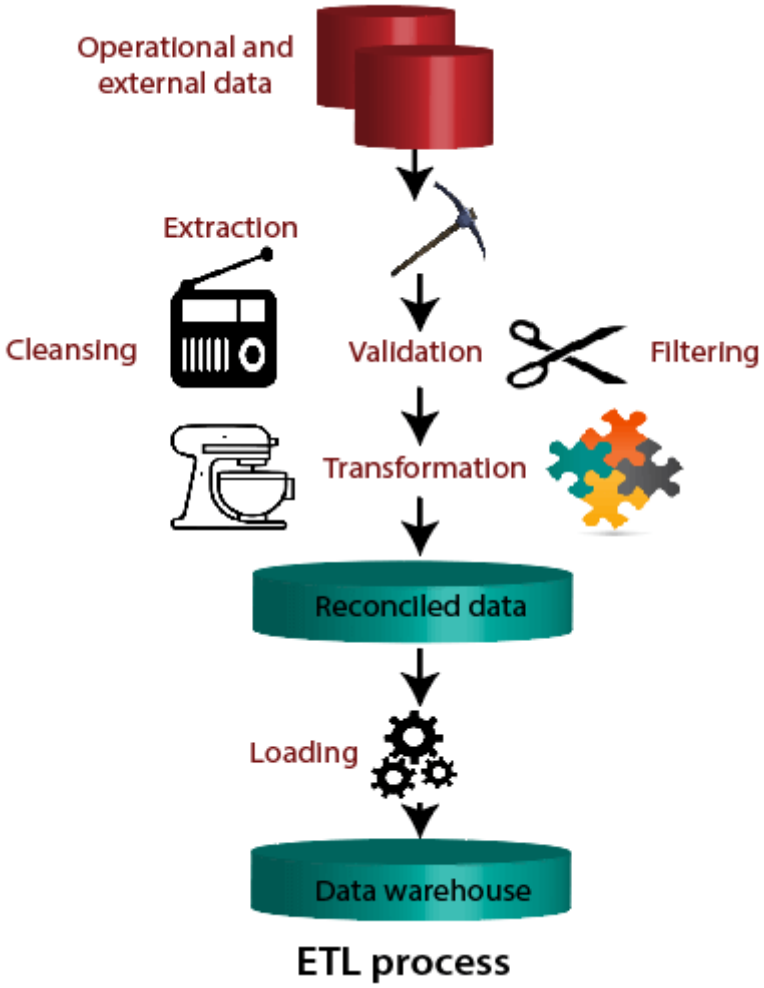
Transformation is the core of the reconciliation phase. It converts records from its operational source format into a particular data warehouse format. If we implement a three-layer architecture, this phase outputs our reconciled data layer.
The following points must be rectified in this phase:

- Loose texts may hide valuable information. For example, XYZ PVT Ltd does not explicitly show that this is a Limited Partnership company.
- Different formats can be used for individual data. For example, data can be saved as a string or as three integers.

Following are the main transformation processes aimed at populating the reconciled data layer:

- Conversion and normalization that operate on both storage formats and units of measure to make data uniform.
- Matching that associates equivalent fields in different sources.
- Selection that reduces the number of source fields and records.

Cleansing and **Transformation** processes are often closely linked in ETL tools.



Loading

The **Load** is the process of writing the data into the target database. During the load step, it is necessary to ensure that the load is performed correctly and with as little resources as possible.

Loading can be carried in two ways:

1. **Refresh:** Data Warehouse data is completely rewritten. This means that older file is replaced. Refresh is usually used in combination with static extraction to populate a data warehouse initially.
2. **Update:** Only those changes applied to source information are added to the Data Warehouse. An update is typically carried out without deleting or modifying preexisting data. This method is used in combination with incremental extraction to update data warehouses regularly.

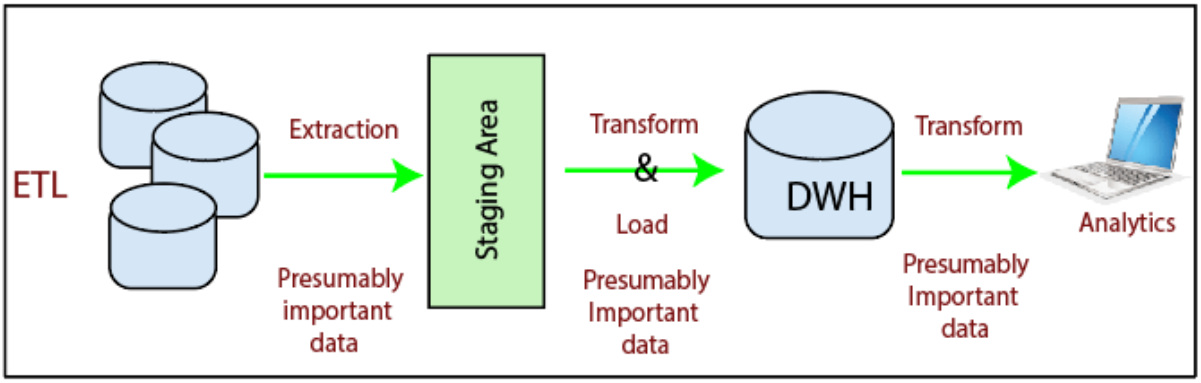
Selecting an ETL Tool

Selection of an appropriate ETL Tools is an important decision that has to be made in choosing the importance of an ODS or data warehousing application. The ETL tools are required to provide coordinated access to multiple data sources so that relevant data may be extracted from them. An ETL tool would generally contains tools for data cleansing, re-organization, transformations, aggregation, calculation and automatic loading of information into the object database.

An ETL tool should provide a simple user interface that allows data cleansing and data transformation rules to be specified using a point-and-click approach. When all mappings and transformations have been defined, the ETL tool should automatically generate the data extract/transformation/load programs, which typically run in batch mode.

ETL (Extract, Transform, and Load)

Extract, Transform and Load is the technique of extracting the record from sources (which is present outside or on-premises, etc.) to a staging area, then transforming or reformatting with business manipulation performed on it in order to fit the operational needs or data analysis, and later loading into the goal or destination databases or data warehouse.



Strengths

Development Time: Designing from the output backwards provide that only information applicable to the solution is extracted and processed, potentially decreasing development, delete, and processing overhead.

Targeted data: Due to the targeted feature of the load process, the warehouse contains only information relevant to the presentation. Reduced warehouse content simplify the security regime enforce and hence the administration overheads.

Tools Availability: The number of tools available that implement ETL provides the flexibility of approach and the opportunity to identify the most appropriate tool. The proliferation of tools has to lead to a competitive functionality war, which often results in loss of maintainability.

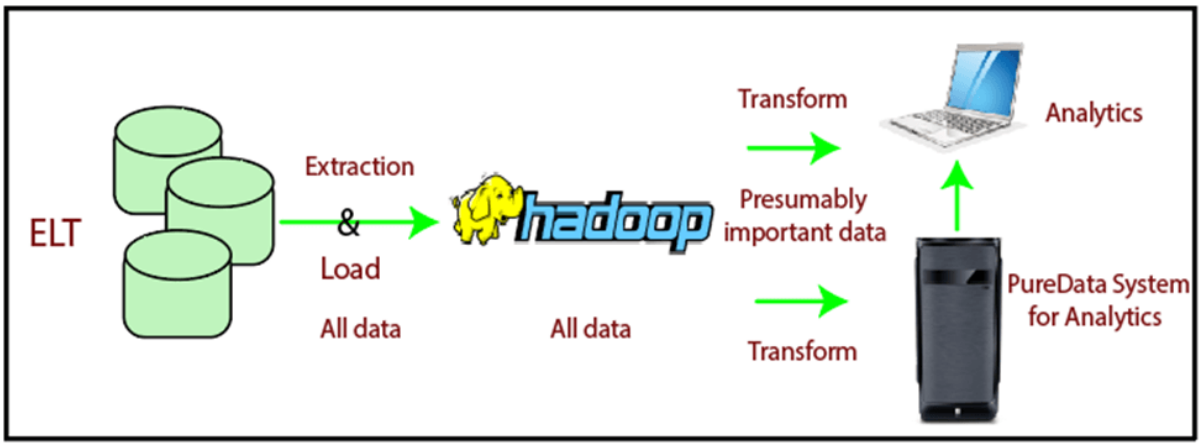
Weaknesses

Flexibility: Targeting only relevant information for output means that any future requirements that may need data that was not included in the original design will need to be added to the ETL routines. Due to the nature of tight dependency between the methods developed, this often leads to a need for fundamental redesign and development. As a result, this increase the time and cost involved.

Hardware: Most third-party tools utilize their engine to implement the ETL phase. Regardless of the estimate of the solution, this can necessitate the investment in additional hardware to implement the tool's ETL engine. The use of third-party tools to achieve the ETL process compels the information of new scripting languages and processes.
Learning Curve: Implementing a third-party tools that uses foreign processes and languages results in the learning curve that is implicit in all technologies new to an organization and can often lead to consecutive blind alleys in their use due to shortage of experience.

ELT (Extract, Load and Transform)

ELT stands for Extract, Load and Transform is the various sight while looking at data migration or movement. ELT involves the extraction of aggregate information from the source system and loading to the target method instead of transformation between the extraction and loading phase. Once the data is copied or loaded into the target method, then change takes place.



The **extract** and **load** step can be isolated from the transformation process. Isolating the load phase from the transformation process delete an inherent dependency between these phases. In addition to containing the data necessary for the transformations, the extract and load process can include components of data that may be essential in the future. The load phase could take the entire source and loaded it into the warehouses.
Separating the phases enables the project to be damaged down into smaller chunks, thus making it more specific and manageable.
Performing the data integrity analysis in the staging method enables a further phase in the process to be isolated and dealt with at the most appropriate point in the process. This method also helps to ensure that only cleaned and checked information is loaded into the warehouse for transformation.
Isolating the transformations from the load steps helps to encourage a more staged way to the warehouse design and implementation.

Strengths

Project Management: Being able to divide the warehouse method into specific and isolated functions, enables a project to be designed on a smaller function basis, therefore the project can be broken down into feasible chunks.
Flexible & Future Proof: In general, in an ELT implementation, all record from the sources are loaded into the data warehouse as part of the extract and loading process. This, linked with the isolation of the transformation phase, means that future requirements can easily be incorporated into the data warehouse architecture.
Risk minimization: Deleting the close interdependencies between each technique of the warehouse build system enables the development method to be isolated, and the individual process design can thus also be separated. This provides a good platform for change, maintenance and management.

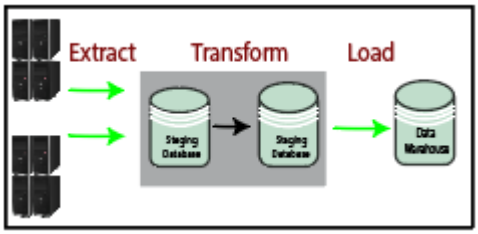
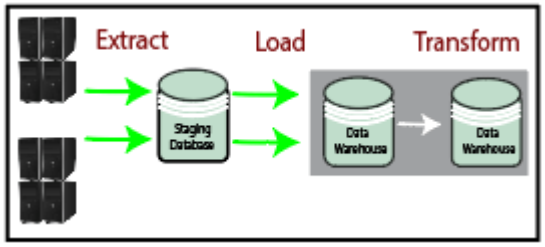
Utilize Existing Hardware: In implementing ELT as a warehouse build process, the essential tools provided with the database engine can be used.
Utilize Existing Skill sets: By using the functionality support by the database engine, the existing investment in database functions are re-used to develop the warehouse. No new skills need to be learned, and the full weight of the experience in developing the engines technology is utilized, further reducing the cost and risk in the development process.

Weaknesses

Against the Norm: ELT is a new method to data warehouse design and development. While it has proven itself many times over through its abundant use in implementations throughout the world, it does require a change in mentality and design approach against traditional methods.
Tools Availability: Being an emergent technology approach, ELT suffers from the limited availability of tools.

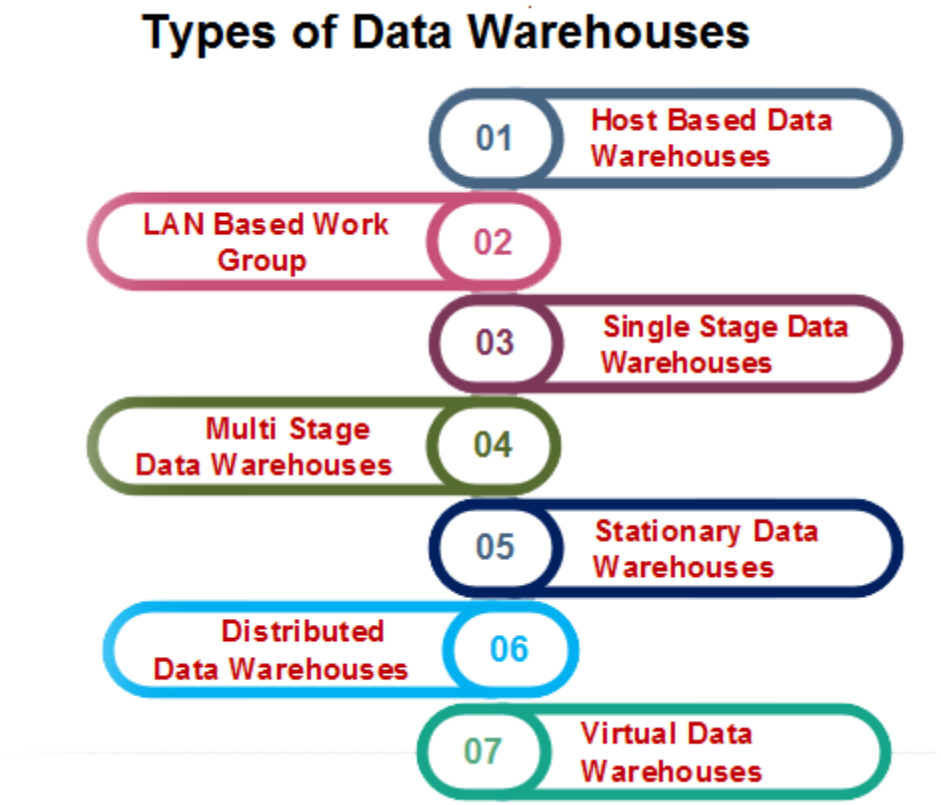
Difference between ETL vs. ELT

Basics	ETL	ELT
Process	Data is transferred to the ETL server and moved back to DB. High network bandwidth required.	Data remains in the DB except for cross Database loads (e.g. source to object).
Transformation	Transformations are performed in ETL Server.	Transformations are performed (in the source or) in the target.
Code Usage	Typically used for o Source to target transfer o Compute-intensive Transformations o Small amount of data	Typically used for o High amounts of data
Time-Maintenance	It needs highs maintenance as you need to select data to load and transform.	Low maintenance as data is always available.
Calculations	Overwrites existing column or Need to append the dataset and push to the target platform.	Easily add the calculated column to the existing table.

Analysis		

Types of Data Warehouses

There are different types of data warehouses, which are as follows:



Host-Based Data Warehouses

There are two types of host-based data warehouses which can be implemented:

- Host-Based mainframe warehouses which reside on a high volume database. Supported by robust and reliable high capacity structure such as IBM system/390, UNISYS and Data General sequent systems, and databases such as Sybase, Oracle, Informix, and DB2.
- Host-Based LAN data warehouses, where data delivery can be handled either centrally or from the workgroup environment. The size of the data warehouses of the database depends on the platform.

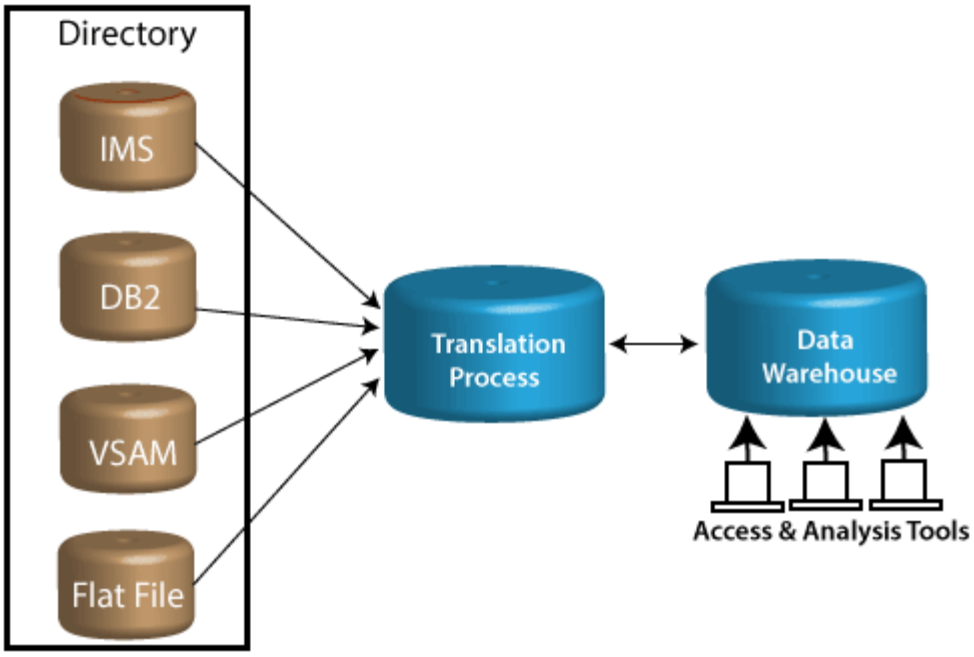
Data Extraction and transformation tools allow the automated extraction and cleaning of data from production systems. It is not applicable to enable direct access by query tools to these categories of methods for the following reasons:

1. A huge load of complex warehousing queries would possibly have too much of a harmful impact upon the mission-critical transaction processing (TP)-oriented application.

2. These TP systems have been developing in their database design for transaction throughput. In all methods, a database is designed for optimal query or transaction processing. A complex business query needed the joining of many normalized tables, and as result performance will usually be poor and the query constructs largely complex.
3. There is no assurance that data in two or more production methods will be consistent.

Host-Based (MVS) Data Warehouses

Those data warehouse uses that reside on large volume databases on MVS are the host-based types of data warehouses. Often the DBMS is DB2 with a huge variety of original source for legacy information, including VSAM, DB2, flat files, and Information Management System (IMS).



Host Based (MVS) Data Warehouse

Before embarking on designing, building and implementing such a warehouse, some further considerations must be given because

1. Such databases generally have very high volumes of data storage.
2. Such warehouses may require support for both MVS and customer-based report and query facilities.
3. These warehouses have complicated source systems.
4. Such systems needed continuous maintenance since these must also be used for mission-critical objectives.

To make such data warehouses building successful, the following phases are generally followed:

1. **Unload Phase:** It contains selecting and scrubbing the operation data.
2. **Transform Phase:** For translating it into an appropriate form and describing the rules for accessing and storing it.
3. **Load Phase:** For moving the record directly into DB2 tables or a particular file for moving it into another database or non-MVS warehouse.

An integrated Metadata repository is central to any data warehouse environment. Such a facility is required for documenting data sources, data translation rules, and user areas to the warehouse. It provides a dynamic network between the multiple data source databases and the DB2 of the conditional data warehouses.

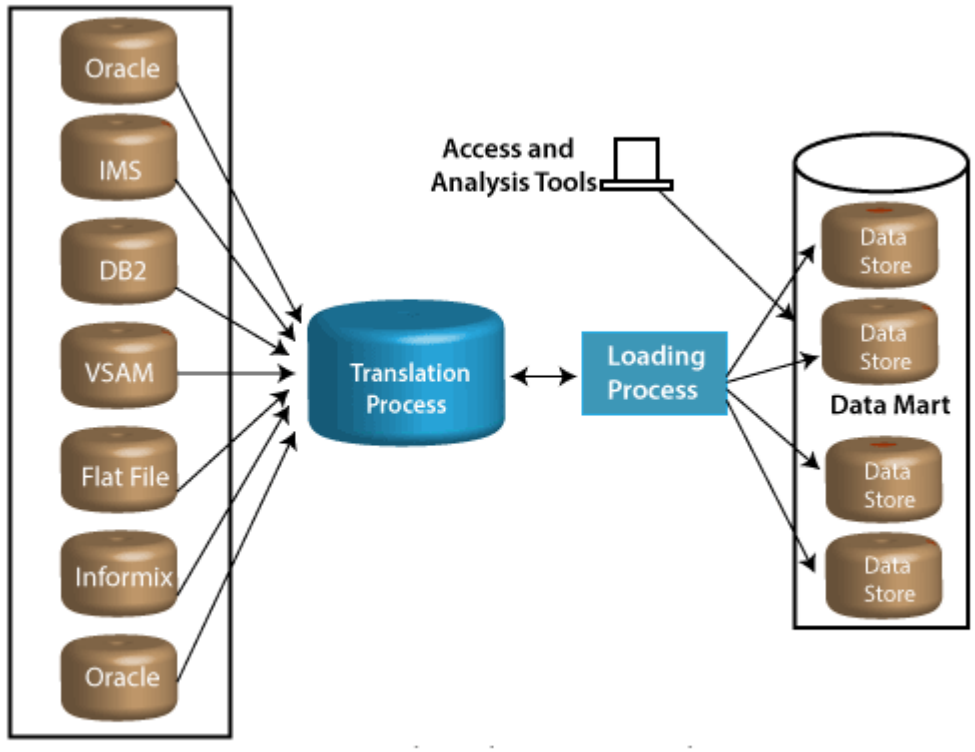
A metadata repository is necessary to design, build, and maintain data warehouse processes. It should be capable of providing data as to what data exists in both the operational system and data warehouse, where the data is located. The mapping of the operational data to the warehouse fields and end-user access techniques. Query, reporting, and maintenance are another indispensable method of such a data warehouse. An MVS-based query and reporting tool for DB2.

Host-Based (UNIX) Data Warehouses

Oracle and Informix RDBMSs support the facilities for such data warehouses. Both of these databases can extract information from MVS¬ based databases as well as a higher number of other UNIX¬ based databases. These types of warehouses follow the same stage as the host-based MVS data warehouses. Also, the data from different network servers can be created. Since file attribute consistency is frequent across the inter-network.

LAN-Based Workgroup Data Warehouses

A LAN based workgroup warehouse is an integrated structure for building and maintaining a data warehouse in a LAN environment. In this warehouse, we can extract information from a variety of sources and support multiple LAN based warehouses, generally chosen warehouse databases to include DB2 family, Oracle, Sybase, and Informix. Other databases that can also be contained through infrequently are IMS, VSAM, Flat File, MVS, and VH.



LAN Based Work Group Warehouse

Limitations

Both DBMS and hardware scalability methods generally limit LAN based warehousing solutions.

Many LAN based enterprises have not implemented adequate job scheduling, recovery management, organized maintenance, and performance monitoring methods to provide robust warehousing solutions.

Often these warehouses are dependent on other platforms for source record. Building an environment that has data integrity, recoverability, and security require careful design, planning, and implementation. Otherwise, synchronization of transformation and loads from sources to the server could cause innumerable problems.

A **LAN based warehouse** provides data from many sources requiring a minimal initial investment and technical knowledge. A LAN based warehouse can also work replication tools for populating and updating the data warehouse. This type of warehouse can include business views, histories, aggregation, versions in, and heterogeneous source support, such as

- DB2 Family
- IMS, VSAM, Flat File [MVS and VM]

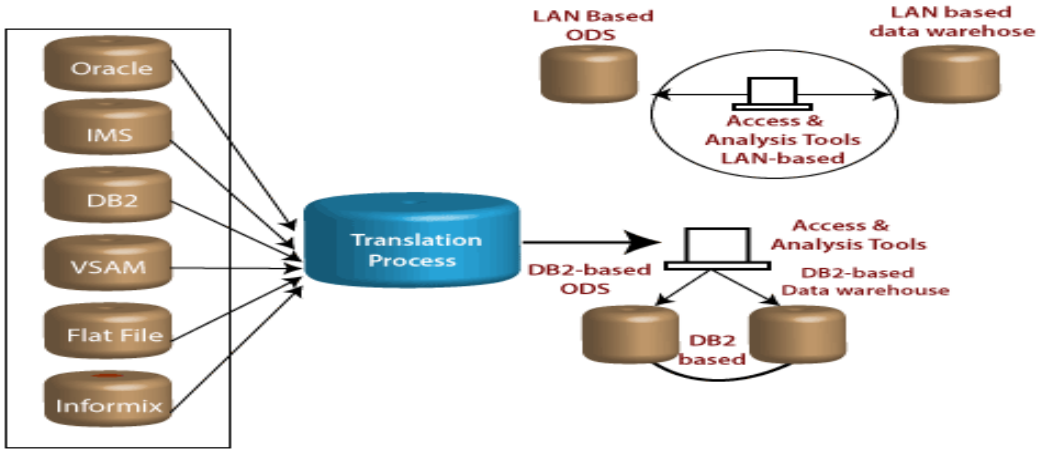
A single store frequently drives a LAN based warehouse and provides existing DSS applications, enabling the business user to locate data in their data warehouse. The LAN based warehouse can support business users with complete data to information solution. The LAN based warehouse can also share metadata with the ability to catalog business data and make it feasible for anyone who needs it.

Multi-Stage Data Warehouses

It refers to multiple stages in transforming methods for analyzing data through aggregations. In other words, staging of the data multiple times before the loading operation into the data warehouse, data gets extracted form source systems to staging area first, then gets loaded to data warehouse after the change and then finally to departmentalized data marts.

This configuration is well suitable to environments where end-clients in numerous capacities require access to both summarized information for up to the minute tactical decisions as well as summarized, a commutative record for long-term strategic decisions. Both the Operational Data Store (ODS) and the data warehouse may reside on host-based or LAN Based databases, depending on volume and custom requirements. These contain DB2, Oracle, Informix, IMS, Flat Files, and Sybase.

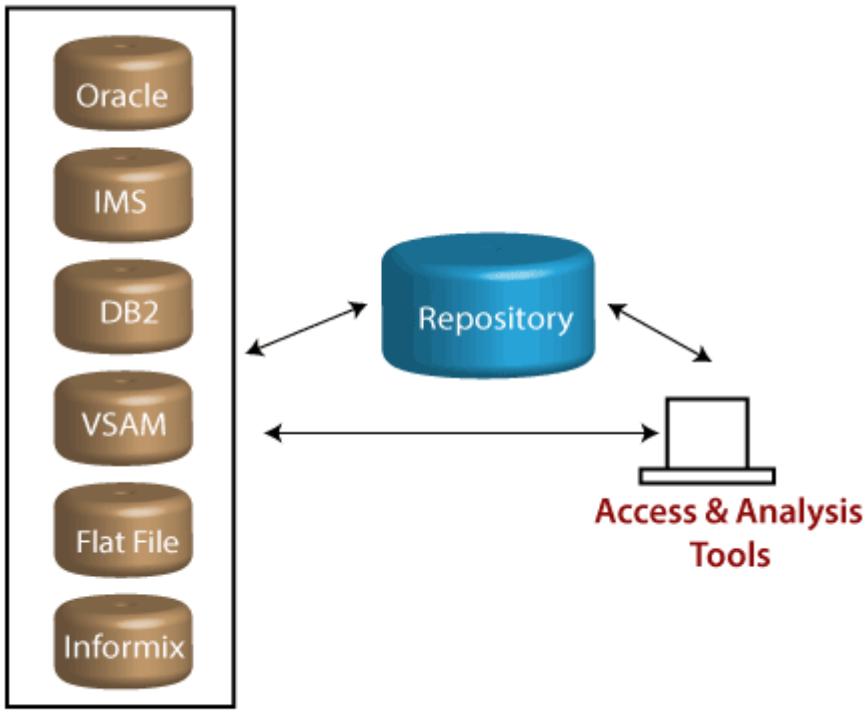
Usually, the ODS stores only the most up-to-date records. The data warehouse stores the historical calculation of the files. At first, the information in both databases will be very similar. For example, the records for a new client will look the same. As changes to the user record occur, the ODs will be refreshed to reflect only the most current data, whereas the data warehouse will contain both the historical data and the new information. Thus the volume requirement of the data warehouse will exceed the volume requirements of the ODS overtime. It is not familiar to reach a ratio of 4 to 1 in practice.



Multistage Data Warehouse

Stationary Data Warehouses

In this type of data warehouses, the data is not changed from the sources, as shown in fig:



Stationary Data Warehouse

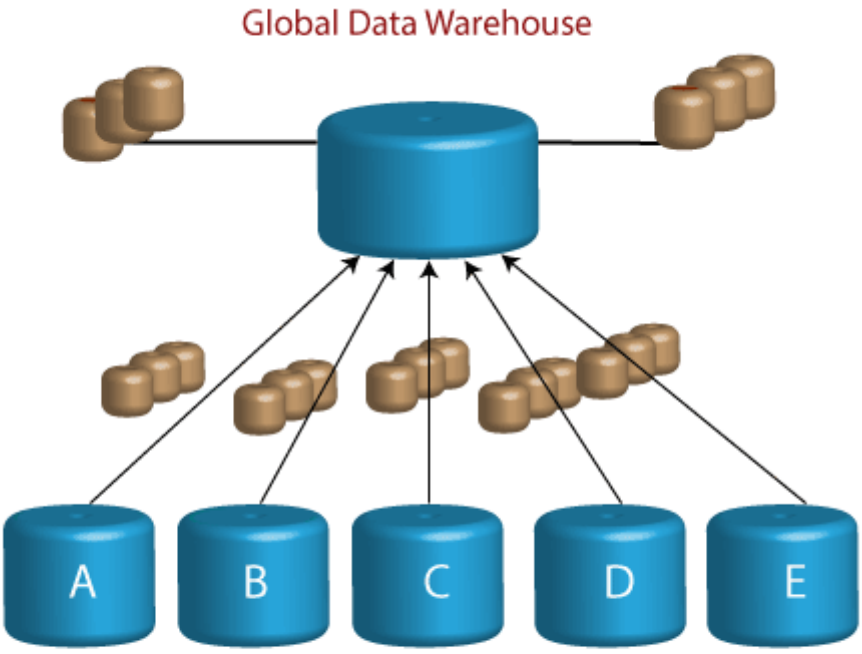
Instead, the customer is given direct access to the data. For many organizations, infrequent access, volume issues, or corporate necessities dictate such as approach. This schema does generate several problems for the customer such as

- Identifying the location of the information for the users
- Providing clients the ability to query different DBMSs as is they were all a single DBMS with a single API.
- Impacting performance since the customer will be competing with the production data stores.

Such a warehouse will need highly specialized and sophisticated 'middleware' possibly with a single interaction with the client. This may also be essential for a facility to display the extracted record for the user before report generation. An integrated metadata repository becomes an absolute essential under this environment.

Distributed Data Warehouses

The concept of a distributed data warehouse suggests that there are two types of distributed data warehouses and their modifications for the local enterprise warehouses which are distributed throughout the enterprise and a global warehouses as shown in fig:



Distributed Data Warehouse

Characteristics of Local data warehouses

- Activity appears at the local level
- Bulk of the operational processing
- Local site is autonomous
- Each local data warehouse has its unique architecture and contents of data
- The data is unique and of prime essential to that locality only
- Majority of the record is local and not replicated
- Any intersection of data between local data warehouses is circumstantial
- Local warehouse serves different technical communities
- The scope of the local data warehouses is finite to the local site
- Local warehouses also include historical data and are integrated only within the local site.

Virtual Data Warehouses

Virtual Data Warehouses is created in the following stages:

1. Installing a set of data approach, data dictionary, and process management facilities.
2. Training end-clients.
3. Monitoring how DW facilities will be used
4. Based upon actual usage, physically Data Warehouse is created to provide the high-frequency results

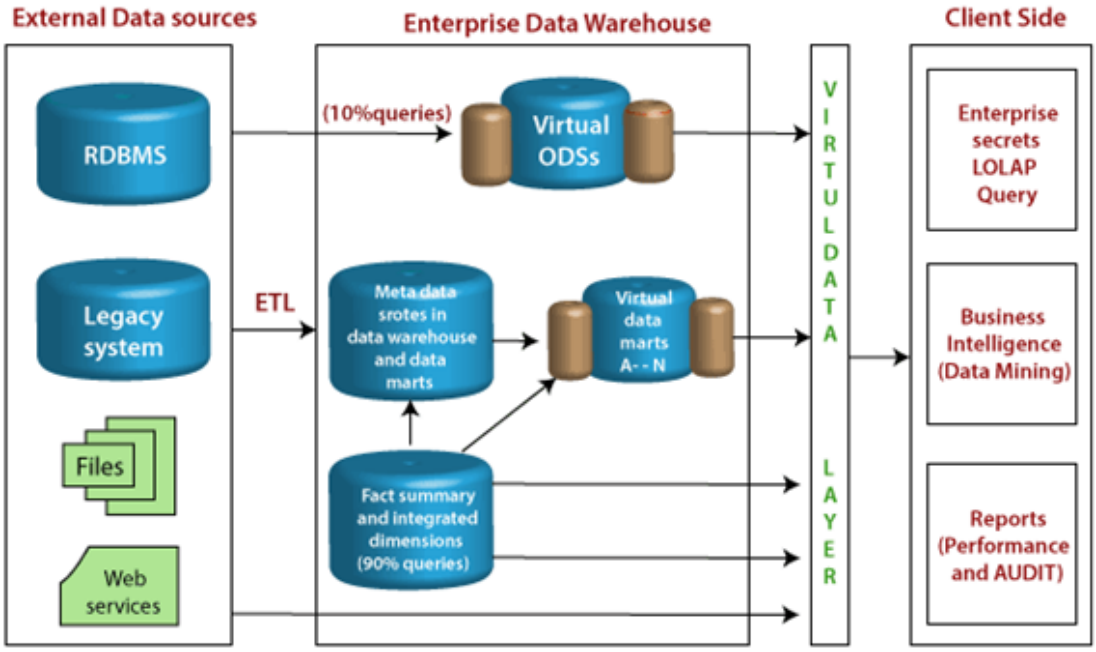
This strategy defines that end users are allowed to get at operational databases directly using whatever tools are implemented to the data access network. This method provides ultimate flexibility as well as the minimum amount of redundant information that must be loaded and maintained. The data warehouse is a great idea, but it is difficult to build and requires investment. Why not use a cheap and fast method by eliminating the transformation phase of repositories for metadata and another database. This method is termed the '**virtual data warehouse**.'

To accomplish this, there is a need to define four kinds of data:

1. A data dictionary including the definitions of the various databases.
2. A description of the relationship between the data components.
3. The description of the method user will interface with the system.
4. The algorithms and business rules that describe what to do and how to do it.

Disadvantages

1. Since queries compete with production record transactions, performance can be degraded.
2. There is no metadata, no summary record, or no individual **DSS** (Decision Support System) integration or history. All queries must be copied, causing an additional burden on the system.
3. There is no refreshing process, causing the queries to be very complex.



10% of user queries are fired on fact summary & 90% of user queries are fired on ODSs

Virtual Data Warehouse

Data Warehouse Design

A data warehouse is a single data repository where a record from multiple data sources is integrated for online business analytical processing (OLAP). This implies a data warehouse needs to meet the requirements from all the business stages within the entire organization. Thus, data warehouse design is a hugely complex, lengthy, and hence error-prone process. Furthermore, business analytical functions change over time, which results in changes in the requirements for the systems. Therefore, data warehouse and OLAP systems are dynamic, and the design process is continuous.

Data warehouse design takes a method different from view materialization in the industries. It sees data warehouses as database systems with particular needs such as answering management related queries. The target of the design becomes how the record from multiple data sources should be extracted, transformed, and loaded (ETL) to be organized in a database as the data warehouse.

There are two approaches

1. "top-down" approach

2. "bottom-up" approach

Top-down Design Approach

In the "Top-Down" design approach, a data warehouse is described as a subject-oriented, time-variant, non-volatile and integrated data repository for the entire enterprise data from different sources are validated, reformatted and saved in a normalized (up to 3NF) database as the data warehouse. The data warehouse stores "atomic" information, the data at the lowest level of granularity, from where dimensional data marts can be built by selecting the data required for specific business subjects or particular departments. An approach is a data-driven approach as the information is gathered and integrated first and then business requirements by subjects for building data marts are formulated. The advantage of this method is which it supports a single integrated data source. Thus data marts built from it will have consistency when they overlap.

Advantages of top-down design

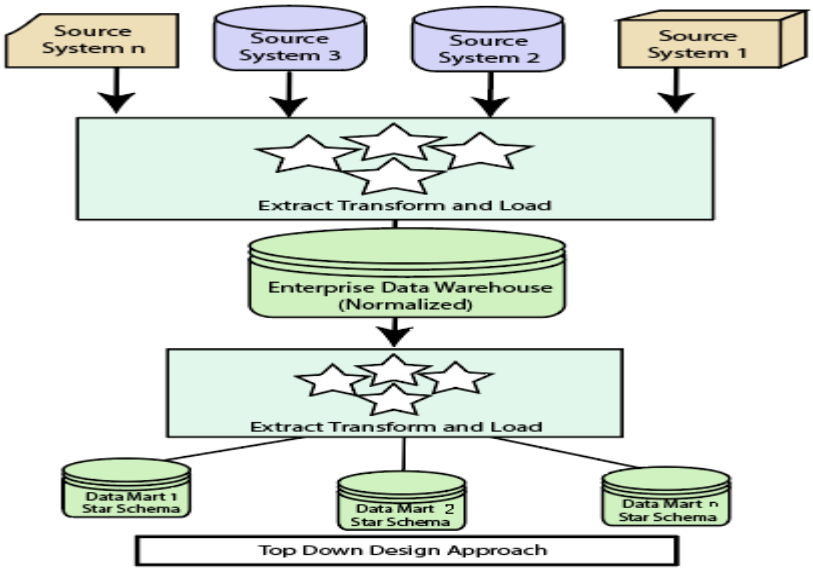
Data Marts are loaded from the data warehouses.

Developing new data mart from the data warehouse is very easy.

Disadvantages of top-down design

This technique is inflexible to changing departmental needs.

The cost of implementing the project is high.



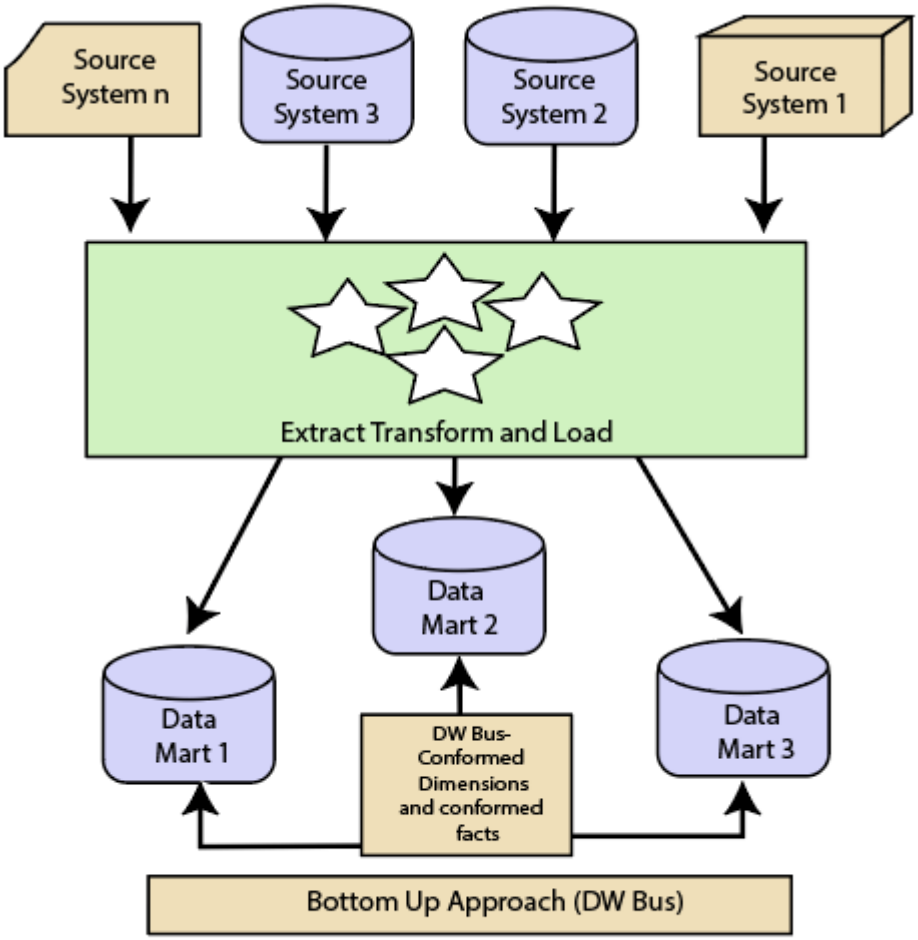
Top Down Design Approach

Bottom-Up Design Approach

In the "Bottom-Up" approach, a data warehouse is described as "a copy of transaction data specifical architecture for query and analysis," term the star schema. In this approach, a data mart is created first to necessary reporting and analytical capabilities for particular business processes (or subjects). Thus it is needed to be a business-driven approach in contrast to Inmon's data-driven approach.

Data marts include the lowest grain data and, if needed, aggregated data too. Instead of a normalized database for the data warehouse, a denormalized dimensional database is adapted to meet the data delivery requirements of data warehouses. Using this method, to use the set of data marts as the enterprise data warehouse, data marts should be built with conformed dimensions in mind, defining that ordinary objects are represented the same in different data

marts. The conformed dimensions connected the data marts to form a data warehouse, which is generally called a virtual data warehouse.
The advantage of the "bottom-up" design approach is that it has quick ROI, as developing a data mart, a data warehouse for a single subject, takes far less time and effort than developing an enterprise-wide data warehouse. Also, the risk of failure is even less. This method is inherently incremental. This method allows the project team to learn and grow.



Bottom Up Design Approach

Advantages of bottom-up design
Documents can be generated quickly.
The data warehouse can be extended to accommodate new business units.
It is just developing new data marts and then integrating with other data marts.
Disadvantages of bottom-up design
The locations of the data warehouse and the data marts are reversed in the bottom-up approach design.

Differentiate between Top-Down Design Approach and Bottom-Up Design Approach

Top-Down Design Approach	Bottom-Up Design Approach
Breaks the vast problem into smaller subproblems.	Solves the essential low-level problem and integrates them into a higher one.
Inherently architected- not a union of several data marts.	Inherently incremental; can schedule essential data marts first.
Single, central storage of information about the content.	Departmental information stored.
Centralized rules and control.	Departmental rules and control.
It includes redundant information.	Redundancy can be removed.
It may see quick results if implemented with repetitions.	Less risk of failure, favorable return on investment, and proof of techniques.

Data Warehouse Modeling

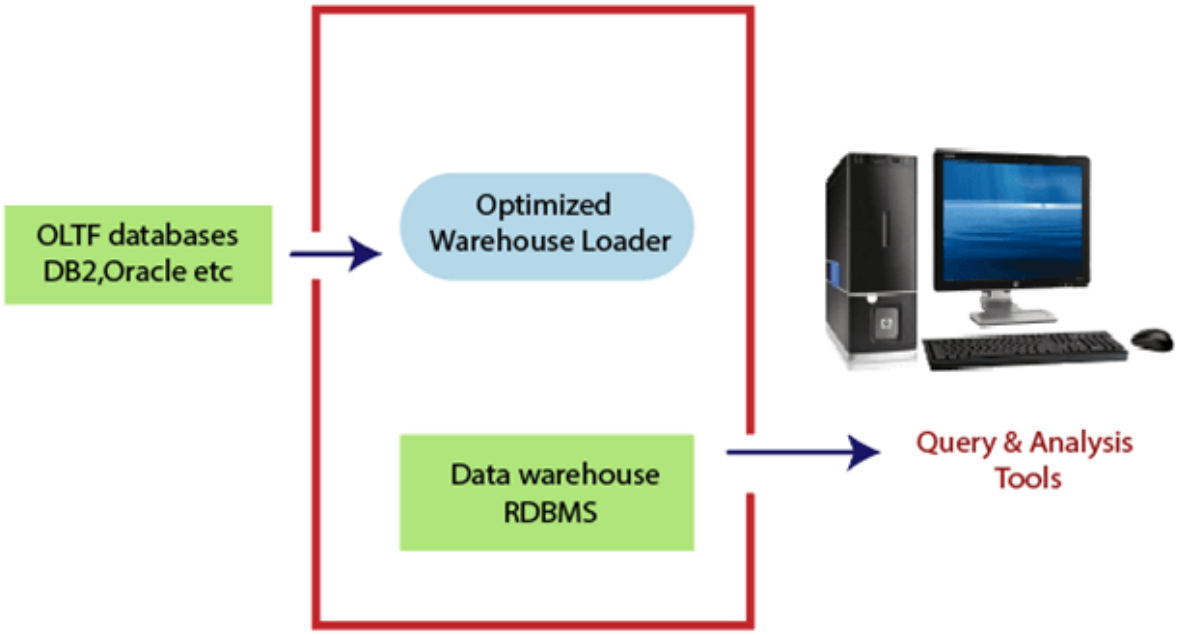
Data warehouse modeling is the process of designing the schemas of the detailed and summarized information of the data warehouse. The goal of data warehouse modeling is to develop a schema describing the reality, or at least a part of the fact, which the data warehouse is needed to support.

Data warehouse modeling is an essential stage of building a data warehouse for two main reasons. Firstly, through the schema, data warehouse clients can visualize the relationships among the warehouse data, to use them with greater ease. Secondly, a well-designed schema allows an effective data warehouse structure to emerge, to help decrease the cost of implementing the warehouse and improve the efficiency of using it.

Data modeling in data warehouses is different from data modeling in operational database systems. The primary function of data warehouses is to support DSS processes. Thus, the objective of data warehouse modeling is to make the data warehouse efficiently support complex queries on long term information.

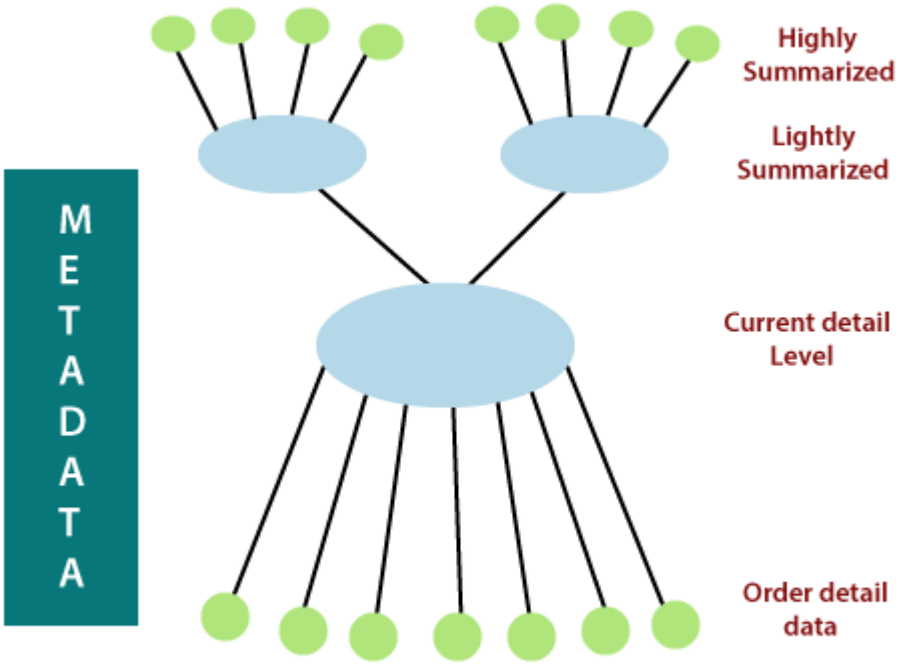
In contrast, data modeling in operational database systems targets efficiently supporting simple transactions in the database such as retrieving, inserting, deleting, and changing data. Moreover, data warehouses are designed for the customer with general information knowledge about the enterprise, whereas operational database systems are more oriented toward use by software specialists for creating distinct applications.

Data Warehouse model is illustrated in the given diagram.



Data Warehouse Model

The data within the specific warehouse itself has a particular architecture with the emphasis on various levels of summarization, as shown in figure:



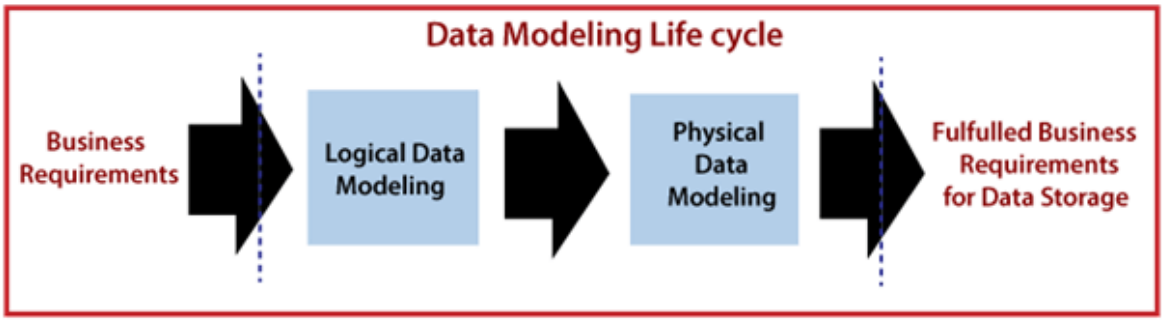
The Structure of data inside the data warehouse

- Reflects the most current happenings, which are commonly the most stimulating.
- It is numerous as it is saved at the lowest method of the Granularity.
- It is always (almost) saved on disk storage, which is fast to access but expensive and difficult to manage.

Older detail data is stored in some form of mass storage, and it is infrequently accessed and kept at a level detail consistent with current detailed data.
Lightly summarized data is data extract from the low level of detail found at the current, detailed level and usually is stored on disk storage. When building the data warehouse have to remember what unit of time is summarization done over and also the components or what attributes the summarized data will contain.
Highly summarized data is compact and directly available and can even be found outside the warehouse.

Data Modeling Life Cycle

In this section, we define a data modeling life cycle. It is a straight forward process of transforming the business requirements to fulfill the goals for storing, maintaining, and accessing the data within IT systems. The result is a logical and physical data model for an enterprise data warehouse.
The objective of the data modeling life cycle is primarily the creation of a storage area for business information. That area comes from the logical and physical data modeling stages, as shown in Figure:



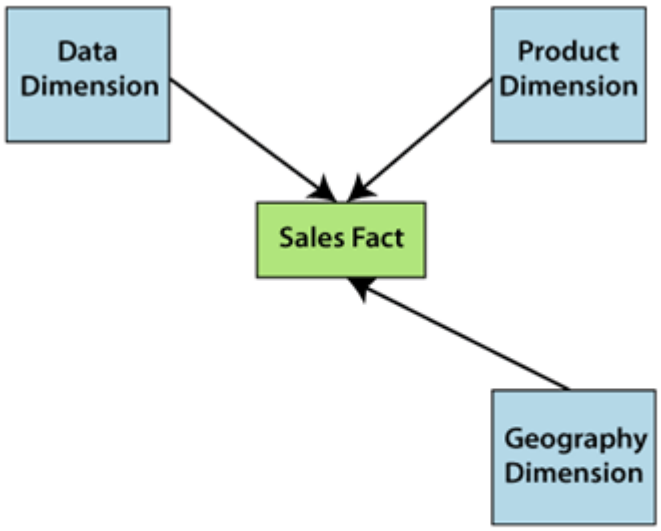
A generic data modeling life cycle

Conceptual Data Model

A conceptual data model recognizes the highest-level relationships between the different entities.
Characteristics of the conceptual data model

- It contains the essential entities and the relationships among them.
- No attribute is specified.
- No primary key is specified.

We can see that the only data shown via the conceptual data model is the entities that define the data and the relationships between those entities. No other data, as shown through the conceptual data model.



Example of Conceptual Data Model

Logical Data Model

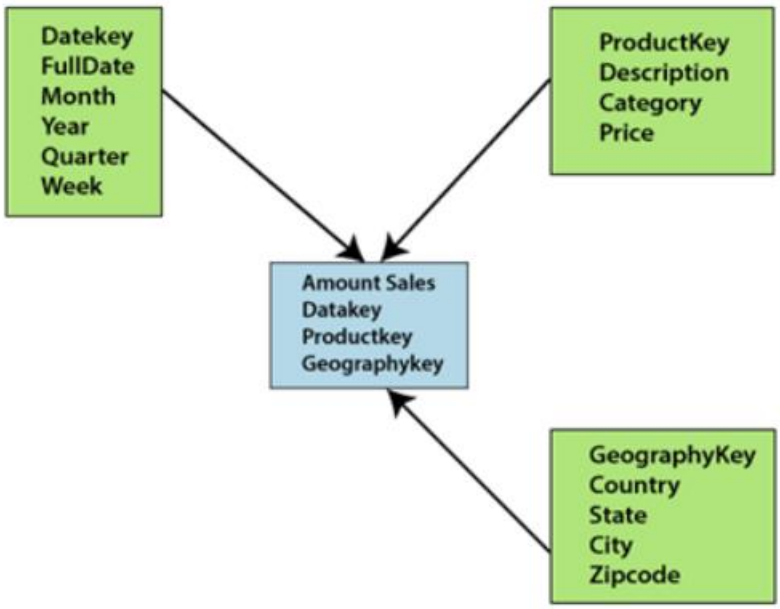
A logical data model defines the information in as much structure as possible, without observing how they will be physically achieved in the database. The primary objective of logical data modeling is to document the business data structures, processes, rules, and relationships by a single view - the logical data model.

Features of a logical data model

- It involves all entities and relationships among them.
- All attributes for each entity are specified.
- The primary key for each entity is stated.
- Referential Integrity is specified (FK Relation).

The phase for designing the logical data model which are as follows:

- Specify primary keys for all entities.
- List the relationships between different entities.
- List all attributes for each entity.
- Normalization.
- No data types are listed



Example of Logical Data Model

Physical Data Model

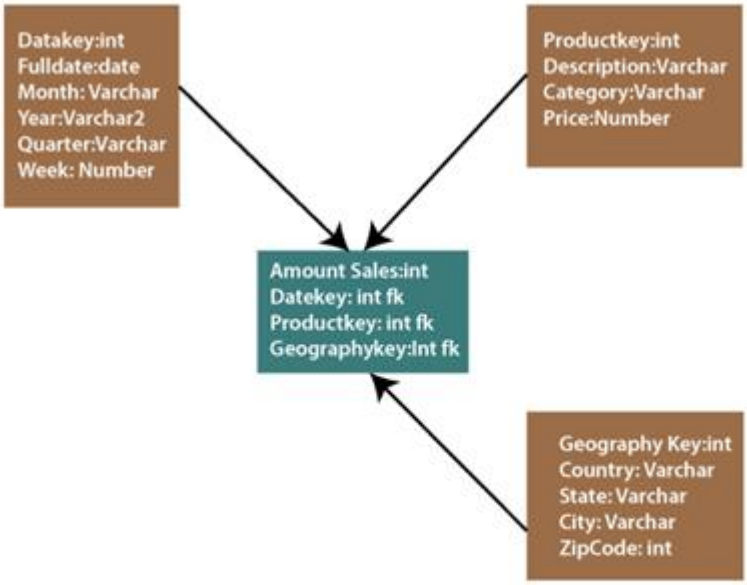
Physical data model describes how the model will be presented in the database. A physical database model demonstrates all table structures, column names, data types, constraints, primary key, foreign key, and relationships between tables. The purpose of physical data modeling is the mapping of the logical data model to the physical structures of the RDBMS system hosting the data warehouse. This contains defining physical RDBMS structures, such as tables and data types to use when storing the information. It may also include the definition of new data structures for enhancing query performance.

Characteristics of a physical data model

- Specification all tables and columns.
- Foreign keys are used to recognize relationships between tables.

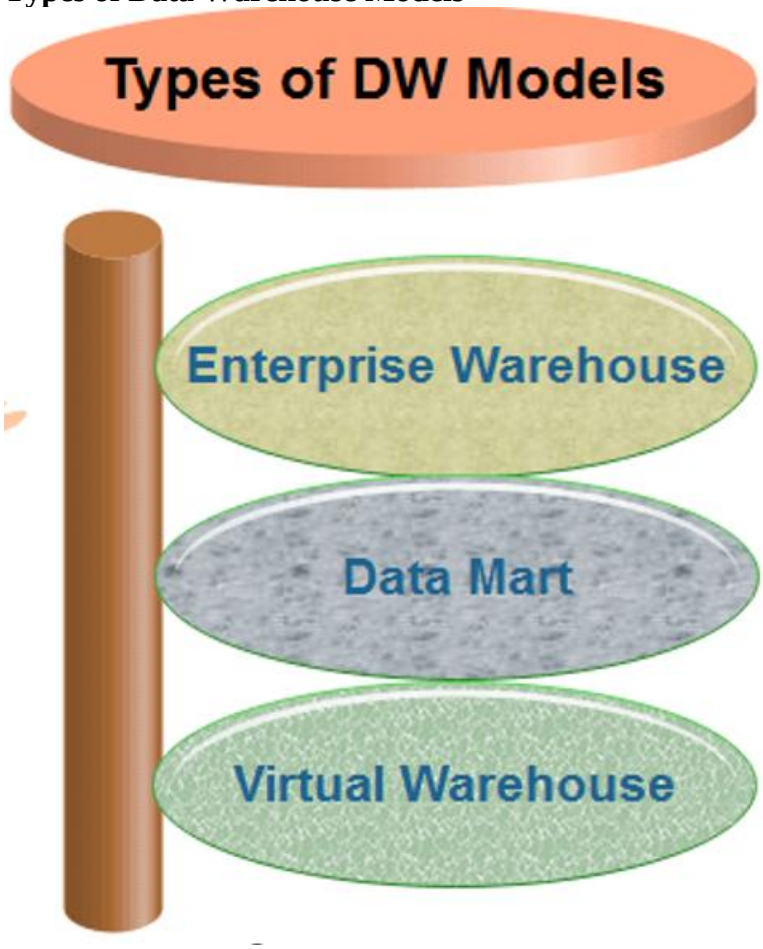
The steps for physical data model design which are as follows:

- Convert entities to tables.
- Convert relationships to foreign keys.
- Convert attributes to columns.



Example of Physical Data Model

Types of Data Warehouse Models



Enterprise Warehouse

An Enterprise warehouse collects all of the records about subjects spanning the entire organization. It supports corporate-wide data integration, usually from one or more operational systems or external data providers, and it's cross-functional in scope. It generally contains detailed information as well as summarized information and can range in estimate from a few gigabyte to hundreds of gigabytes, terabytes, or beyond.

An enterprise data warehouse may be accomplished on traditional mainframes, UNIX super servers, or parallel architecture platforms. It required extensive business modeling and may take years to develop and build.

Data Mart

A data mart includes a subset of corporate-wide data that is of value to a specific collection of users. The scope is confined to particular selected subjects. For example, a marketing data mart may restrict its subjects to the customer, items, and sales. The data contained in the data marts tend to be summarized.

Data Marts is divided into two parts:

Independent Data Mart: Independent data mart is sourced from data captured from one or more operational systems or external data providers, or data generally locally within a different department or geographic area.

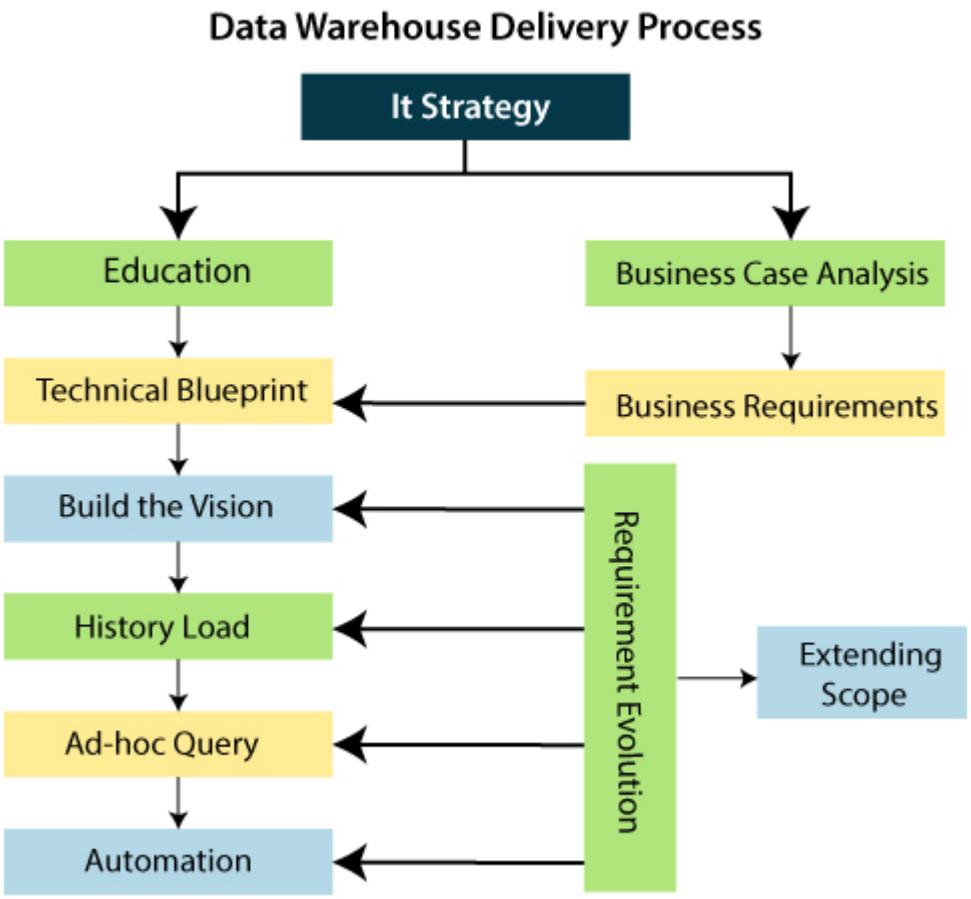
Dependent Data Mart: Dependent data marts are sourced exactly from enterprise data-warehouses.

Virtual Warehouses

Virtual Data Warehouses is a set of perception over the operational database. For effective query processing, only some of the possible summary vision may be materialized. A virtual warehouse is simple to build but required excess capacity on operational database servers.

Data Warehouse Delivery Process:

The delivery process of the data warehouse. Main steps used in data warehouse delivery process which are as follows:



IT Strategy: DWH project must contain IT strategy for procuring and retaining funding.
Business Case Analysis: After the IT strategy has been designed, the next step is the business case. It is essential to understand the level of investment that can be justified and to recognize the projected business benefits which should be derived from using the data warehouse.
Education & Prototyping: Company will experiment with the ideas of data analysis and educate themselves on the value of the data warehouse. This is valuable and should be required if this is the company first exposure to the benefits of the DS record. Prototyping method can progress the growth of education. It is better than working models. Prototyping requires business requirement, technical blueprint, and structures.
Business Requirement: It contains such as
The logical model for data within the data warehouse.
The source system that provides this data (mapping rules)
The business rules to be applied to information.
The query profiles for the immediate requirement
Technical blueprint: It arranges the architecture of the warehouse. Technical blueprint of the delivery process makes an architecture plan which satisfies long-term requirements. It lays server and data mart architecture and essential components of database design.
Building the vision: It is the phase where the first production deliverable is produced. This stage will probably create significant infrastructure elements for extracting and loading information but limit them to the extraction and load of information sources.
History Load: The next step is one where the remainder of the required history is loaded into the data warehouse. This means that the new entities would not be added to the data warehouse, but additional physical tables would probably be created to save the increased record volumes.
AD-Hoc Query: In this step, we configure an ad-hoc query tool to operate against the data warehouse.

These end-customer access tools are capable of automatically generating the database query that answers any question posed by the user.

Automation: The automation phase is where many of the operational management processes are fully automated within the DWH. These would include:

Extracting & loading the data from a variety of sources systems

Transforming the information into a form suitable for analysis

Backing up, restoring & archiving data

Generating aggregations from predefined definitions within the Data Warehouse.

Monitoring query profiles & determining the appropriate aggregates to maintain system performance.

Extending Scope: In this phase, the scope of DWH is extended to address a new set of business requirements. This involves the loading of additional data sources into the DWH i.e. the introduction of new data marts.

Requirement Evolution: This is the last step of the delivery process of a data warehouse. As we all know that requirements are not static and evolve continuously. As the business requirements will change it supports to be reflected in the system.

Concept hierarchy:

Concept hierarchy is directed acyclic graph of ideas, where a unique name identifies each of the theories.

An arc from the concept a to b denotes which is a more general concept than b. We can tag the text with ideas.

Each text report is tagged by a set of concepts which corresponds to its content.

Tagging a report with a concept implicitly entails its tagging with all the ancestors of the concept hierarchy. It is, therefore desired that a report should be tagged with the lowest concept possible.

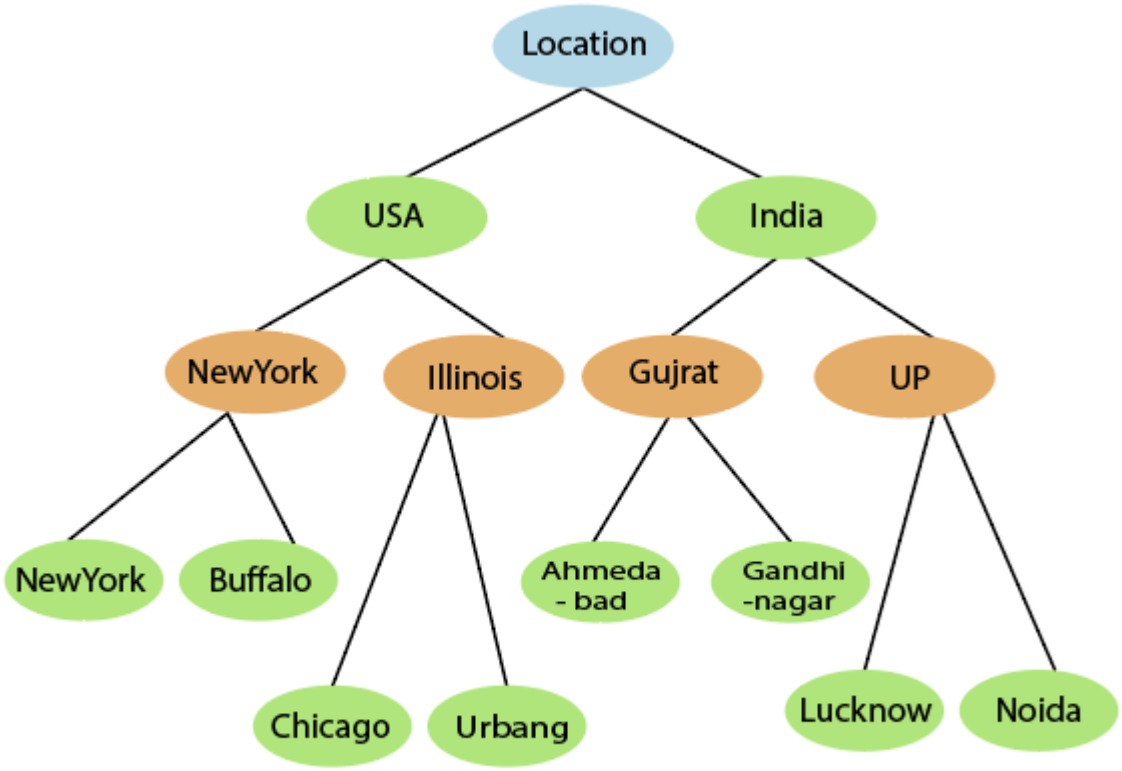
The method to automatically tag the report to the hierarchy is a top-down approach. An evaluation function determines whether a record currently tagged to a node can also be tagged to any of its child nodes.

If so, then then the tag moves down the hierarchy till it cannot be pushed any further.

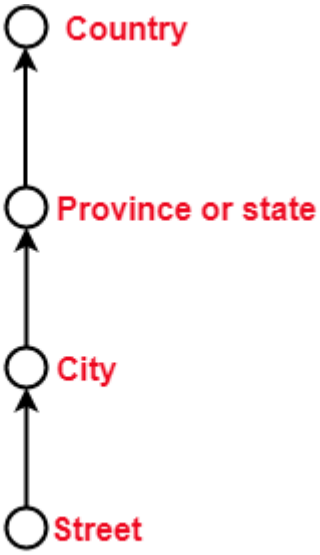
The outcome of this step is a hierarchy of report and, at each node, there is a set of the report having a common concept related to the node.

The hierarchy of reports resulting from the tagging step is useful for many texts mining process.

It is assumed that the hierarchy of concepts is called a priori. We can even have such a hierarchy of documents without a concept hierarchy, by using any hierarchical clustering algorithm, which results in such a hierarchy.



Concept hierarchy for the dimension location



Hierarchical structure for dimension location.

Concept hierarchy defines a sequence of mapping from a set of particular, low-level concepts to more general, higher-level concepts.

In a data warehouse, it is usually used to express different levels of granularity of an attribute from one of the dimension tables.

Concept hierarchies are crucial for the formulation of useful OLAP queries. The hierarchies allow the user to summarize the data at various levels.

For example, using the location hierarchy, the user can retrieve data which summarizes sales for each location, for all the areas in a given state, or even a given country without the necessity of reorganizing the data.

What is OLAP(On-Line Analytical Processing):

OLAP stands for **On-Line Analytical Processing**. OLAP is a classification of software technology which authorizes analysts, managers, and executives to gain insight into information through fast, consistent, interactive access in a wide variety of possible views of data that has been transformed from raw information to reflect the real dimensionality of the enterprise as understood by the clients.

OLAP implement the multidimensional analysis of business information and support the capability for complex estimations, trend analysis, and sophisticated data modeling. It is rapidly enhancing the essential foundation for Intelligent Solutions containing Business Performance Management, Planning, Budgeting, Forecasting, Financial Documenting, Analysis, Simulation-Models, Knowledge Discovery, and Data Warehouses Reporting. OLAP enables end-clients to perform ad hoc analysis of record in multiple dimensions, providing the insight and understanding they require for better decision making.

Who uses OLAP and why?

OLAP applications are used by a variety of the functions of an organization.

Finance and accounting:

- Budgeting
- Activity-based costing
- Financial performance analysis
- And financial modeling

Sales and Marketing

- Sales analysis and forecasting
- Market research analysis
- Promotion analysis
- Customer analysis
- Market and customer segmentation

Production

- Production planning
- Defect analysis

OLAP cubes have two main purposes. The first is to provide business users with a data model more intuitive to them than a tabular model. This model is called a Dimensional Model.

The second purpose is to enable fast query response that is usually difficult to achieve using tabular models.

How OLAP works?

Fundamentally, OLAP has a very simple concept. It pre-calculates most of the queries that are typically very hard to execute over tabular databases, namely aggregation, joining, and grouping. These queries are calculated during a process that is usually called 'building' or 'processing' of the OLAP cube. This process happens overnight, and by the time end users get to work - data will have been updated.

OLAP Guidelines (Dr.E.F.Codd Rule)

Dr E.F. Codd, the "father" of the relational model, has formulated a list of 12 guidelines and requirements as the basis for selecting OLAP systems:

1) Multidimensional Conceptual View: This is the central features of an OLAP system. By needing a multidimensional view, it is possible to carry out methods like slice and dice.

2) Transparency: Make the technology, underlying information repository, computing operations, and the dissimilar nature of source data totally transparent to users. Such transparency helps to improve the efficiency and productivity of the users.

3) Accessibility: It provides access only to the data that is actually required to perform the particular analysis, present a single, coherent, and consistent view to the clients. The OLAP system must map its own logical schema to the heterogeneous physical data stores and perform any necessary transformations. The OLAP operations should be sitting between data sources (e.g., data warehouses) and an OLAP front-end.

4) Consistent Reporting Performance: To make sure that the users do not feel any significant degradation in documenting performance as the number of dimensions or the size of the database increases. That is, the performance of OLAP should not suffer as the number of dimensions is increased. Users must observe consistent run time, response time, or machine utilization every time a given query is run.

5) Client/Server Architecture: Make the server component of OLAP tools sufficiently intelligent that the various clients to be attached with a minimum of effort and integration programming. The server should be capable of mapping and consolidating data between dissimilar databases.

6) Generic Dimensionality: An OLAP method should treat each dimension as equivalent in both is structure and operational capabilities. Additional operational capabilities may be allowed to selected dimensions, but such additional tasks should be grantable to any dimension.

7) Dynamic Sparse Matrix Handling: To adapt the physical schema to the specific analytical model being created and loaded that optimizes sparse matrix handling. When encountering the sparse matrix, the system must be easy to dynamically assume the distribution of the

information and adjust the storage and access to obtain and maintain a consistent level of performance.

8) Multiuser Support: OLAP tools must provide concurrent data access, data integrity, and access security.

9) Unrestricted cross-dimensional Operations: It provides the ability for the methods to identify dimensional order and necessarily functions roll-up and drill-down methods within a dimension or across the dimension.

10) Intuitive Data Manipulation: Data Manipulation fundamental the consolidation direction like as reorientation (pivoting), drill-down and roll-up, and another manipulation to be accomplished naturally and precisely via point-and-click and drag and drop methods on the cells of the scientific model. It avoids the use of a menu or multiple trips to a user interface.

11) Flexible Reporting: It implements efficiency to the business clients to organize columns, rows, and cells in a manner that facilitates simple manipulation, analysis, and synthesis of data.

12) Unlimited Dimensions and Aggregation Levels: The number of data dimensions should be unlimited. Each of these common dimensions must allow a practically unlimited number of customer-defined aggregation levels within any given consolidation path.

Characteristics of OLAP

In the **FASMI characteristics of OLAP methods**, the term derived from the first letters of the characteristics are:

Fast

It defines which the system targeted to deliver the most feedback to the client within about five seconds, with the elementary analysis taking no more than one second and very few taking more than 20 seconds.

Analysis

It defines which the method can cope with any business logic and statistical analysis that is relevant for the function and the user, keep it easy enough for the target client. Although some preprogramming may be needed we do not think it acceptable if all application definitions have to be allow the user to define new Adhoc calculations as part of the analysis and to document on the data in any desired method, without having to program so we excludes products (like Oracle Discoverer) that do not allow the user to define new Adhoc calculation as part of the analysis and to document on the data in any desired product that do not allow adequate end user-oriented calculation flexibility.

Share

It defines which the system tools all the security requirements for understanding and, if multiple write connection is needed, concurrent update location at an appropriated level, not all functions need customer to write data back, but for the increasing number which does, the system should be able to manage multiple updates in a timely, secure manner.

Multidimensional

This is the basic requirement. OLAP system must provide a multidimensional conceptual view of the data, including full support for hierarchies, as this is certainly the most logical method to analyze business and organizations.

Information

The system should be able to hold all the data needed by the applications. Data sparsity should be handled in an efficient manner.

The main characteristics of OLAP are as follows:

1. **Multidimensional conceptual view:** OLAP systems let business users have a dimensional and logical view of the data in the data warehouse. It helps in carrying slice and dice operations.
2. **Multi-User Support:** Since the OLAP techniques are shared, the OLAP operation should provide normal database operations, containing retrieval, update, adequacy control, integrity, and security.
3. **Accessibility:** OLAP acts as a mediator between data warehouses and front-end. The OLAP operations should be sitting between data sources (e.g., data warehouses) and an OLAP front-end.
4. **Storing OLAP results:** OLAP results are kept separate from data sources.
5. **Uniform documenting performance:** Increasing the number of dimensions or database size should not significantly degrade the reporting performance of the OLAP system.
6. OLAP provides for distinguishing between zero values and missing values so that aggregates are computed correctly.
7. OLAP system should ignore all missing values and compute correct aggregate values.
8. OLAP facilitate interactive query and complex analysis for the users.
9. OLAP allows users to drill down for greater details or roll up for aggregations of metrics along a single business dimension or across multiple dimension.
10. OLAP provides the ability to perform intricate calculations and comparisons.
11. OLAP presents results in a number of meaningful ways, including charts and graphs.

Benefits of OLAP

OLAP holds several benefits for businesses: -

1. OLAP helps managers in decision-making through the multidimensional record views that it is efficient in providing, thus increasing their productivity.
2. OLAP functions are self-sufficient owing to the inherent flexibility support to the organized databases.
3. It facilitates simulation of business models and problems, through extensive management of analysis-capabilities.
4. In conjunction with data warehouse, OLAP can be used to support a reduction in the application backlog, faster data retrieval, and reduction in query drag.

Motivations for using OLAP

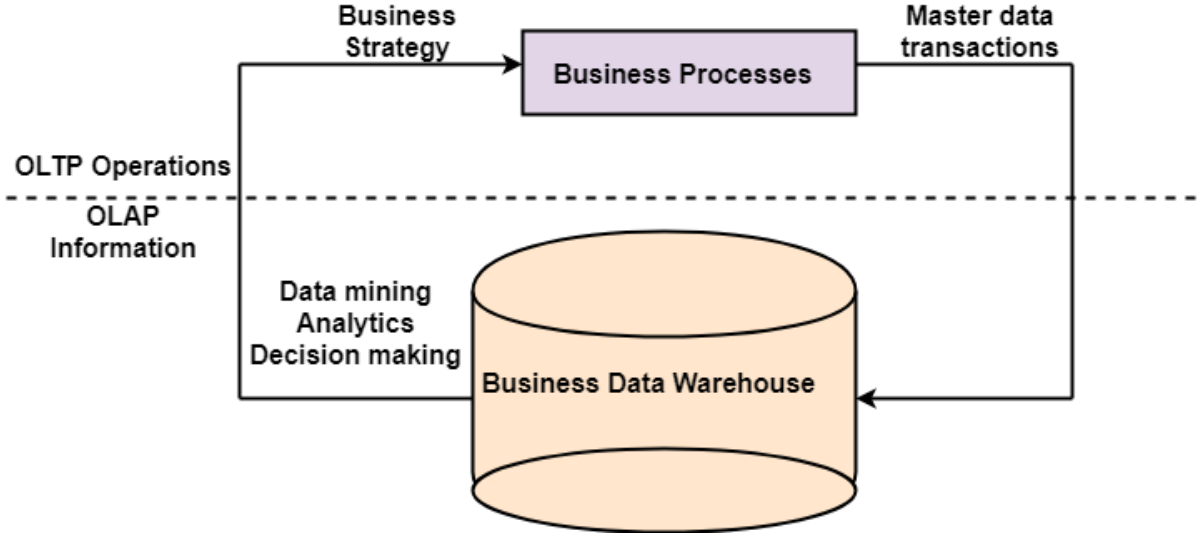
1) Understanding and improving sales: For enterprises that have much products and benefit a number of channels for selling the product, OLAP can help in finding the most suitable products and the most famous channels. In some methods, it may be feasible to find the most profitable users. **For example,** considering the telecommunication industry and considering only one product, communication minutes, there is a high amount of record if a company want to analyze the sales of products for every hour of the day (24 hours), difference between weekdays and weekends (2 values) and split regions to which calls are made into 50 region.

2) Understanding and decreasing costs of doing business: Improving sales is one method of improving a business, the other method is to analyze cost and to control them as much as suitable without affecting sales. OLAP can assist in analyzing the costs related to sales. In some methods, it may also be feasible to identify expenditures which produce a high return on investments (ROI). **For example,** recruiting a top salesperson may contain high costs, but the revenue generated by the salesperson may justify the investment.

Difference between OLTP and OLAP

OLTP (On-Line Transaction Processing) is featured by a large number of short on-line transactions (INSERT, UPDATE, and DELETE). The primary significance of OLTP operations is put on very rapid query processing, maintaining record integrity in multi-access environments, and effectiveness consistent by the number of transactions per second. In the OLTP database, there is an accurate and current record, and schema used to save transactional database is the entity model (usually 3NF).

OLAP (On-line Analytical Processing) is represented by a relatively low volume of transactions. Queries are very difficult and involve aggregations. For OLAP operations, response time is an effectiveness measure. OLAP applications are generally used by Data Mining techniques. In OLAP database there is aggregated, historical information, stored in multi-dimensional schemas (generally star schema).



Following are the difference between OLAP and OLTP system.

1) Users: OLTP systems are designed for office worker while the OLAP systems are designed for decision-makers. Therefore while an OLTP method may be accessed by hundreds or even thousands of clients in a huge enterprise, an OLAP system is suitable to be accessed only by a select class of manager and may be used only by dozens of users.

2) Functions: OLTP systems are mission-critical. They provide day-to-day operations of an enterprise and are largely performance and availability driven. These operations carry out simple repetitive operations. **OLAP** systems are management-critical to support the decision of enterprise support tasks using detailed investigation.

3) Nature: Although **SQL** queries return a set of data, OLTP methods are designed to step one record at the time, for example, a data related to the user who may be on the phone or in the store. OLAP system is not designed to deal with individual customer records. Instead, they include queries that deal with many data at a time and provide summary or aggregate information to a manager. OLAP applications include data stored in a data warehouses that have been extracted from many tables and possibly from more than one enterprise database.

4) Design: OLTP database operations are designed to be application-oriented while **OLAP** operations are designed to be subject-oriented. OLTP systems view the enterprise record as a collection of tables (possibly based on an entity-relationship model). **OLAP** operations view enterprise information as multidimensional).

5) Data: OLTP systems usually deal only with the current status of data. For example, a record about an employee who left three years ago may not be feasible on the Human Resources System. The old data may have been achieved on some type of stable storage media and may not be accessible online. On the other hand, OLAP systems needed historical data over several years since trends are often essential in decision making.

6) Kind of use: OLTP methods are used for reading and writing operations while OLAP methods usually do not update the data.
7) View: An **OLTP** system focuses primarily on the current data within an enterprise or department, which does not refer to historical data or data in various organizations. In contrast, an **OLAP** system spans multiple version of a database schema, due to the evolutionary process of an organization. OLAP system also deals with information that originates from different organizations, integrating information from many data stores. Because of their huge volume, these are stored on multiple storage media.
8) Access Patterns: The access pattern of an OLTP system consist primarily of short, atomic transactions. Such a system needed concurrency control and recovery techniques. However, access to OLAP systems is mostly read-only operations because these data warehouses store historical information.

OLAP Operations in the Multidimensional Data Model

In the multidimensional model, the records are organized into various dimensions, and each dimension includes multiple levels of abstraction described by concept hierarchies. This organization support users with the flexibility to view data from various perspectives. A number of OLAP data cube operation exist to demonstrate these different views, allowing interactive queries and search of the record at hand. Hence, OLAP supports a user-friendly environment for interactive data analysis.
Consider the OLAP operations which are to be performed on multidimensional data. The figure shows data cubes for sales of a shop. The cube contains the dimensions, location, and time and item, where the **location** is aggregated with regard to city values, **time** is aggregated with respect to quarters, and an **item** is aggregated with respect to item types.

Roll-Up

The roll-up operation **(also known as drill-up or aggregation operation)** performs aggregation on a data cube, by climbing down concept hierarchies, i.e., dimension reduction. Roll-up is like **zooming-out** on the data cubes. Figure shows the result of roll-up operations performed on the dimension location. The hierarchy for the location is defined as the Order Street, city, province, or state, country. The roll-up operation aggregates the data by ascending the location hierarchy from the level of the city to the level of the country.
When a roll-up is performed by dimensions reduction, one or more dimensions are removed from the cube. For example, consider a sales data cube having two dimensions, location and time. Roll-up may be performed by removing, the time dimensions, appearing in an aggregation of the total sales by location, relatively than by location and by time.

Example

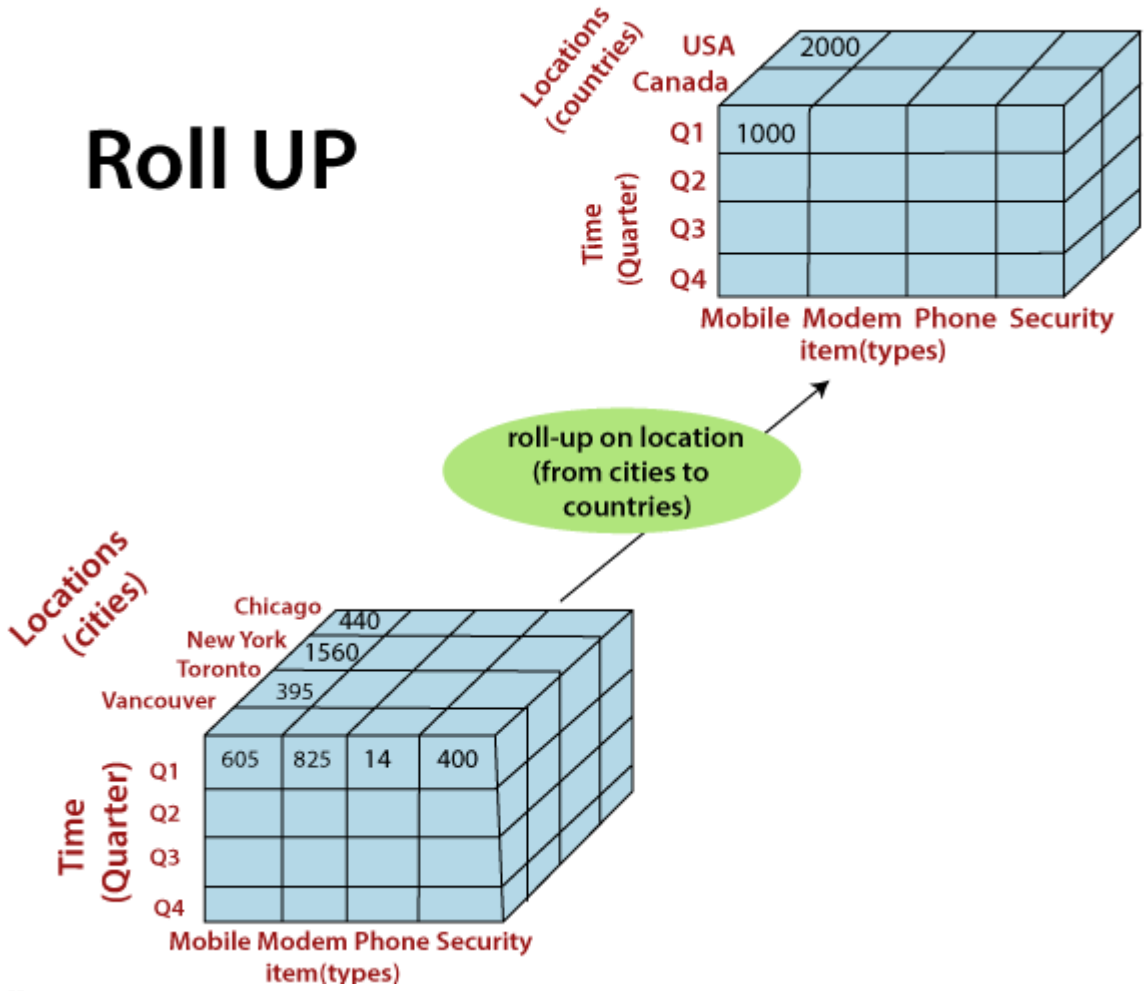
Consider the following cubes illustrating temperature of certain days recorded weekly:

Temperature	64	65	68	69	70	71	72	75	80	81	83	85
Week1	1	0	1	0	1	0	0	0	0	0	1	0
Week2	0	0	0	1	0	0	1	2	0	1	0	0

Consider that we want to set up levels (hot (80-85), mild (70-75), cool (64-69)) in temperature from the above cubes.
To do this, we have to group column and add up the value according to the concept hierarchies. This operation is known as a roll-up.
By doing this, we contain the following cube:

Temperature	cool	mild	hot
Week1	2	1	1
Week2	2	1	1

The roll-up operation groups the information by levels of temperature.
The following diagram illustrates how roll-up works.



Drill-Down:
The drill-down operation **(also called roll-down)** is the reverse operation of **roll-up**. Drill-down is like **zooming-in** on the data cube. It navigates from less detailed record to more detailed data. Drill-down can be performed by either **stepping down** a concept hierarchy for a dimension or adding additional dimensions.
Figure shows a drill-down operation performed on the dimension time by stepping down a concept hierarchy which is defined as day, month, quarter, and year. Drill-down appears by descending the time hierarchy from the level of the quarter to a more detailed level of the month.

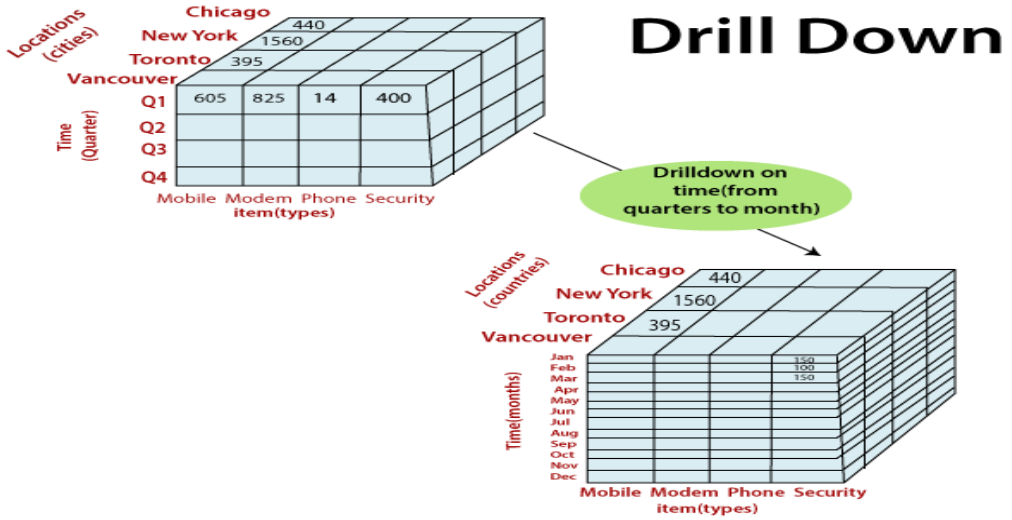
Because a drill-down adds more details to the given data, it can also be performed by adding a new dimension to a cube. For example, a drill-down on the central cubes of the figure can occur by introducing an additional dimension, such as a customer group.

Example

Drill-down adds more details to the given data

Temperature	cool	mild
Day 1	0	0
Day 2	0	0
Day 3	0	0
Day 4	0	1
Day 5	1	0
Day 6	0	0
Day 7	1	0
Day 8	0	0
Day 9	1	0
Day 10	0	1
Day 11	0	1
Day 12	0	1
Day 13	0	0
Day 14	0	0

The following diagram illustrates how Drill-down works.

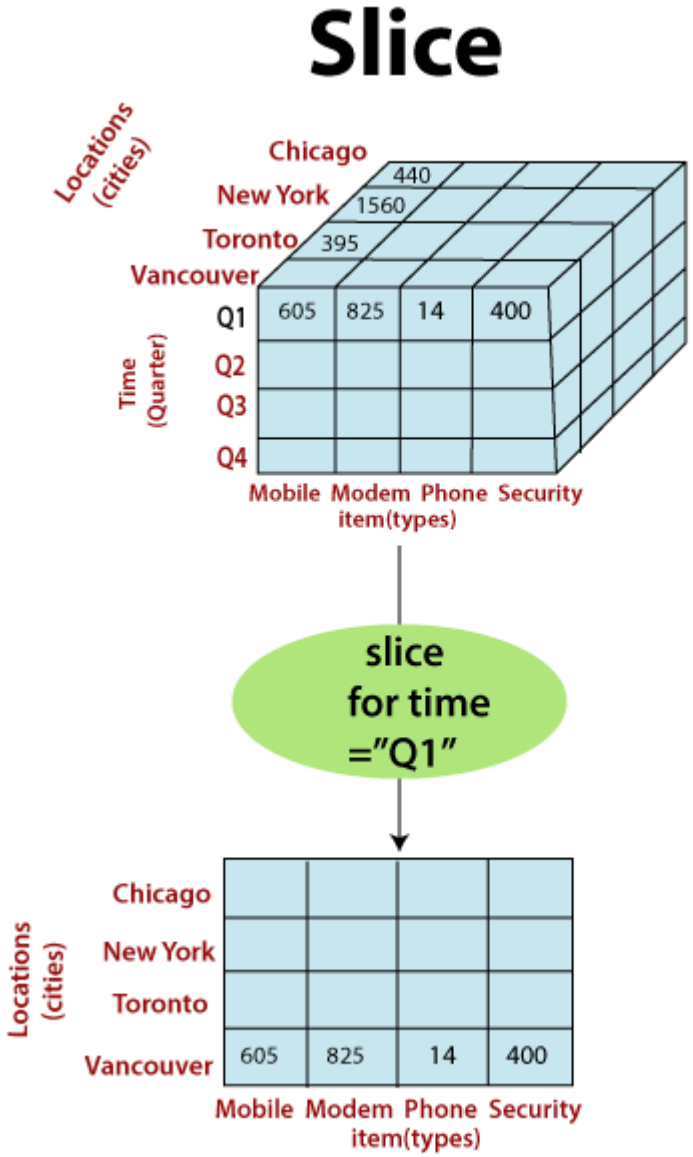


Slice

A **slice** is a subset of the cubes corresponding to a single value for one or more members of the dimension. For example, a slice operation is executed when the customer wants a selection on one dimension of a three-dimensional cube resulting in a two-dimensional site. So, the Slice operations perform a selection on one dimension of the given cube, thus resulting in a subcube. For example, if we make the selection, temperature=cool we will obtain the following cube:

Temperature	**cool**
Day 1	0
Day 2	0
Day 3	0
Day 4	0
Day 5	1
Day 6	1
Day 7	1
Day 8	1
Day 9	1
Day 11	0
Day 12	0
Day 13	0
Day 14	0

The following diagram illustrates how Slice works.



Here Slice is functioning for the dimensions "time" using the criterion time = "Q1".
It will form a new sub-cubes by selecting one or more dimensions.

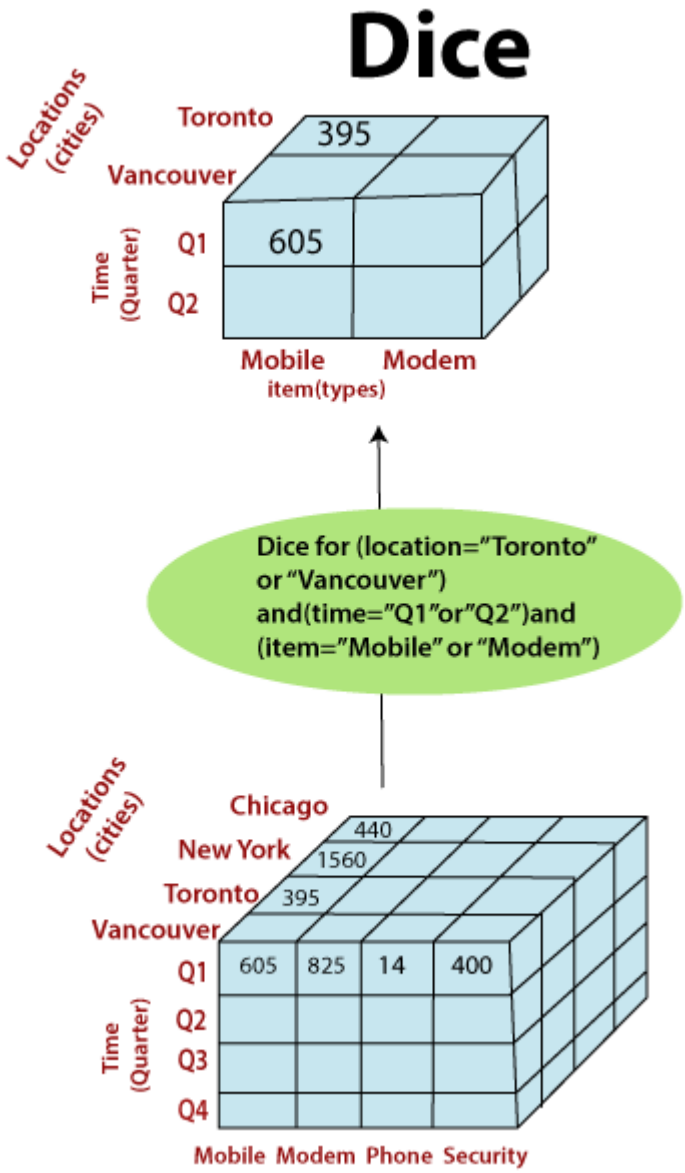
Dice

The dice operation describes a subcube by operating a selection on two or more dimension.

For example, Implement the selection (time = day 3 OR time = day 4) AND (temperature = cool OR temperature = hot) to the original cubes we get the following subcube (still two-dimensional)

Temperature	cool	hot
Day 3	0	1
Day 4	0	0

Consider the following diagram, which shows the dice operations.

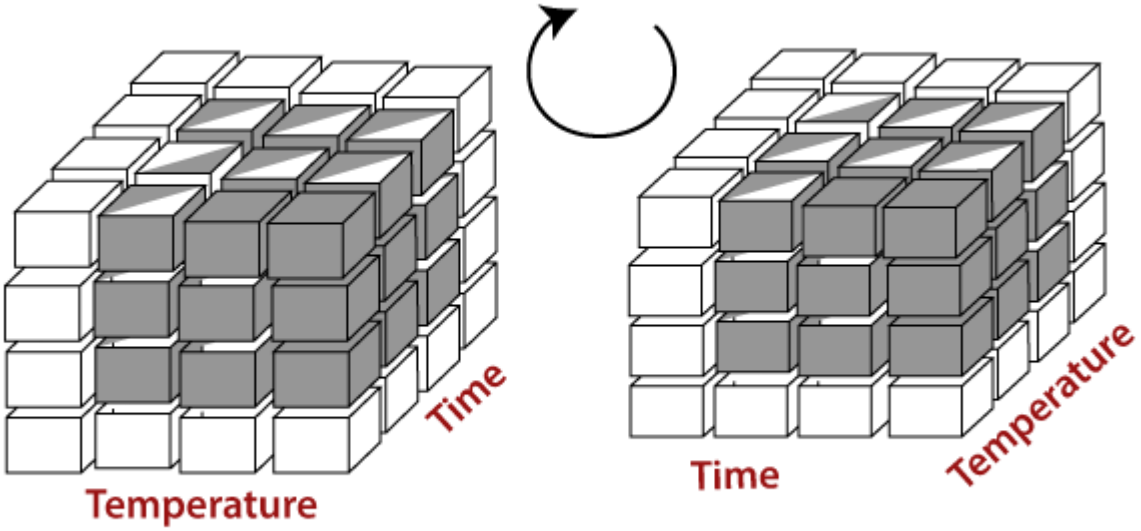


The dice operation on the cubes based on the following selection criteria involves three dimensions.

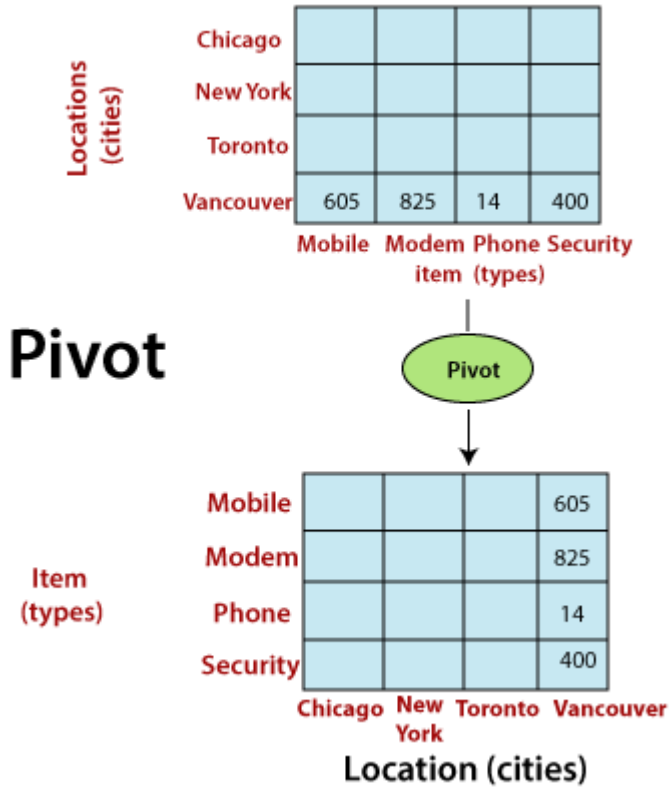
- (location = "Toronto" or "Vancouver")
- (time = "Q1" or "Q2")
- (item =" Mobile" or "Modem")

Pivot

The pivot operation is also called a rotation. Pivot is a visualization operations which rotates the data axes in view to provide an alternative presentation of the data. It may contain swapping the rows and columns or moving one of the row-dimensions into the column dimensions.



Consider the following diagram, which shows the pivot operation.



Other OLAP Operations

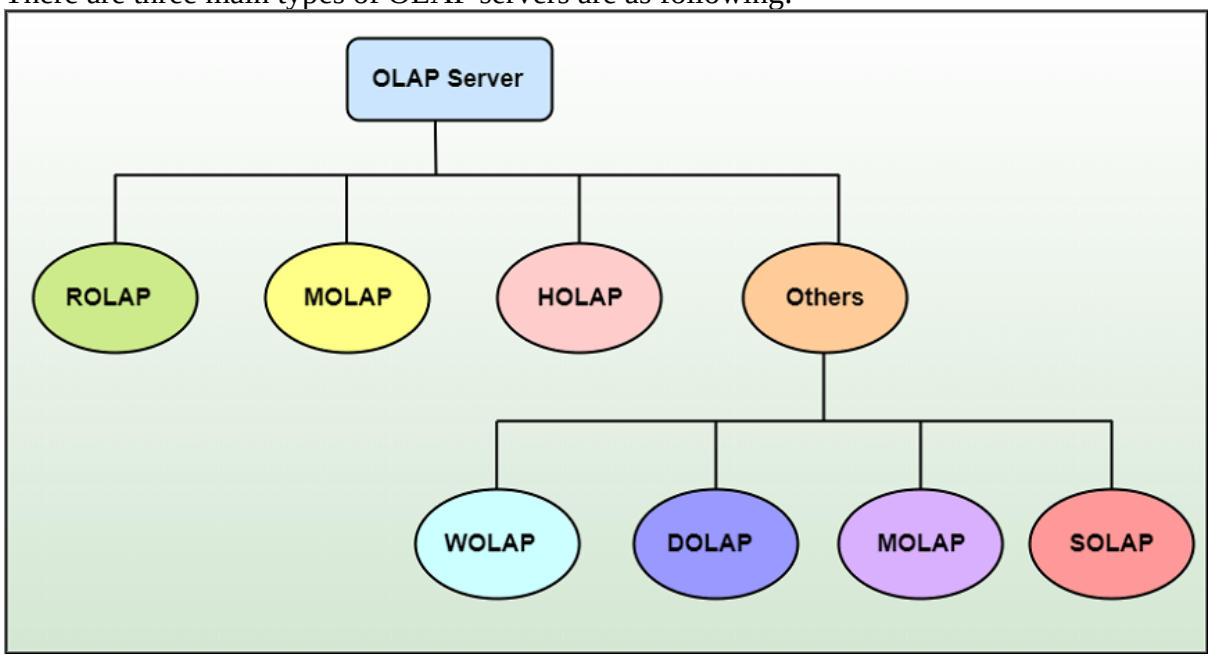
executes queries containing more than one fact table. The drill-through operations make use of relational SQL facilitates to drill through the bottom level of a data cubes down to its back-end relational tables.

Other OLAP operations may contain ranking the top-N or bottom-N elements in lists, as well as calculate moving average, growth rates, and interests, internal rates of returns, depreciation, currency conversions, and statistical tasks.

OLAP offers analytical modeling capabilities, containing a calculation engine for determining ratios, variance, etc. and for computing measures across various dimensions. It can generate summarization, aggregation, and hierarchies at each granularity level and at every dimensions intersection. OLAP also provide functional models for forecasting, trend analysis, and statistical analysis. In this context, the OLAP engine is a powerful data analysis tool.

Types of OLAP

There are three main types of OLAP servers are as following:



ROLAP stands for Relational OLAP, an application based on relational DBMSs.

MOLAP stands for Multidimensional OLAP, an application based on multidimensional DBMSs.

HOLAP stands for Hybrid OLAP, an application using both relational and multidimensional techniques.

Relational OLAP (ROLAP) Server

These are intermediate servers which stand in between a relational back-end server and user frontend tools.

They use a relational or extended-relational DBMS to save and handle warehouse data, and OLAP middleware to provide missing pieces.

ROLAP servers contain optimization for each DBMS back end, implementation of aggregation navigation logic, and additional tools and services.

ROLAP technology tends to have higher scalability than MOLAP technology.

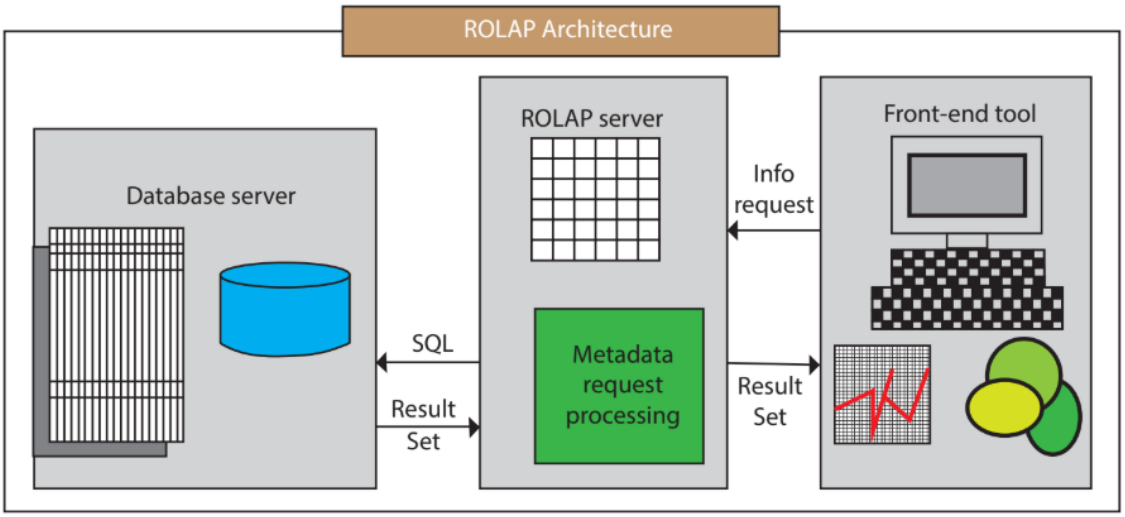
ROLAP systems work primarily from the data that resides in a relational database, where the base data and dimension tables are stored as relational tables. This model permits the multidimensional analysis of data.

This technique relies on manipulating the data stored in the relational database to give the presence of traditional OLAP's slicing and dicing functionality. In essence, each method of slicing and dicing is equivalent to adding a "WHERE" clause in the SQL statement.

Relational OLAP Architecture

ROLAP Architecture includes the following components

- Database server.
- ROLAP server.
- Front-end tool.



Relational OLAP (ROLAP) is the latest and fastest-growing OLAP technology segment in the market. This method allows multiple multidimensional views of two-dimensional relational tables to be created, avoiding structuring record around the desired view.

Some products in this segment have supported reliable SQL engines to help the complexity of multidimensional analysis. This includes creating multiple SQL statements to handle user requests, being 'RDBMS' aware and also being capable of generating the SQL statements based on the optimizer of the DBMS engine.

Advantages

Can handle large amounts of information: The data size limitation of ROLAP technology is depends on the data size of the underlying RDBMS. So, ROLAP itself does not restrict the data amount.

<="" strong="">RDBMS already comes with a lot of features. So ROLAP technologies, (works on top of the RDBMS) can control these functionalities.

Disadvantages

Performance can be slow: Each ROLAP report is a SQL query (or multiple SQL queries) in the relational database, the query time can be prolonged if the underlying data size is large.

Limited by SQL functionalities: ROLAP technology relies on upon developing SQL statements to query the relational database, and SQL statements do not suit all needs.

Multidimensional OLAP (MOLAP) Server

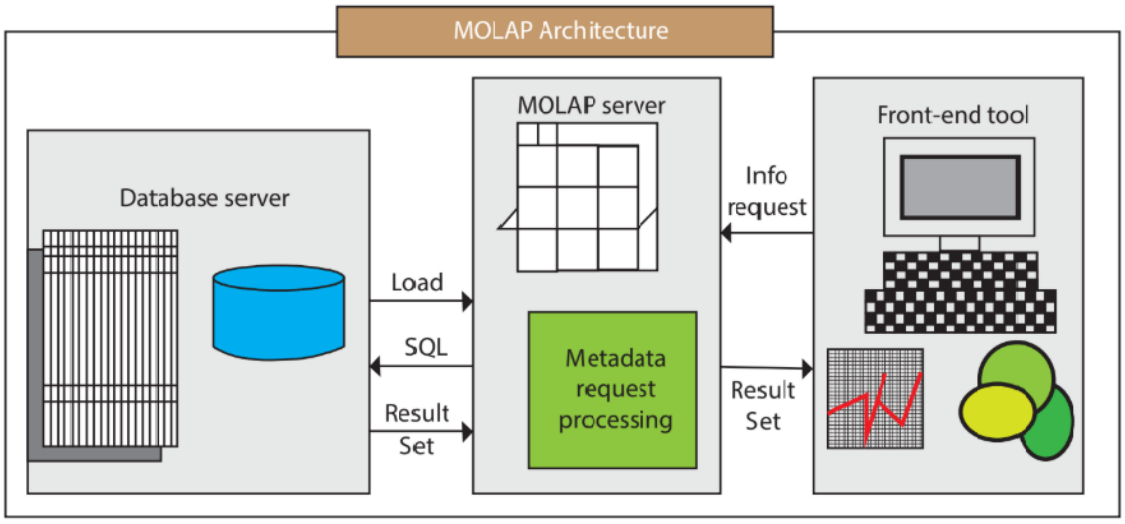
A MOLAP system is based on a native logical model that directly supports multidimensional data and operations. Data are stored physically into multidimensional arrays, and positional techniques are used to access them.

One of the significant distinctions of **MOLAP** against a **ROLAP** is that data are summarized and are stored in an optimized format in a multidimensional cube, instead of in a relational database. In MOLAP model, data are structured into proprietary formats by client's reporting requirements with the calculations pre-generated on the cubes.

MOLAP Architecture

MOLAP Architecture includes the following components

- Database server.
- MOLAP server.
- Front-end tool.



MOLAP structure primarily reads the precompiled data. MOLAP structure has limited capabilities to dynamically create aggregations or to evaluate results which have not been pre-calculated and stored.

Applications requiring iterative and comprehensive time-series analysis of trends are well suited for MOLAP technology (e.g., financial analysis and budgeting).

Examples include Arbor Software's Essbase. Oracle's Express Server, Pilot Software's Lightship Server, Sniper's TM/1. Planning Science's Gentium and Kenan Technology's Multiway.

Some of the problems faced by clients are related to maintaining support to multiple subject areas in an RDBMS. Some vendors can solve these problems by continuing access from MOLAP tools to detailed data in and RDBMS.

This can be very useful for organizations with performance-sensitive multidimensional analysis requirements and that have built or are in the process of building a data warehouse architecture that contains multiple subject areas.

An example would be the creation of sales data measured by several dimensions (e.g., product and sales region) to be stored and maintained in a persistent structure. This structure would be provided to reduce the application overhead of performing calculations and building aggregation during initialization. These structures can be automatically refreshed at predetermined intervals established by an administrator.

Advantages

Excellent Performance: A MOLAP cube is built for fast information retrieval, and is optimal for slicing and dicing operations.

Can perform complex calculations: All evaluation have been pre-generated when the cube is created. Hence, complex calculations are not only possible, but they return quickly.

Disadvantages

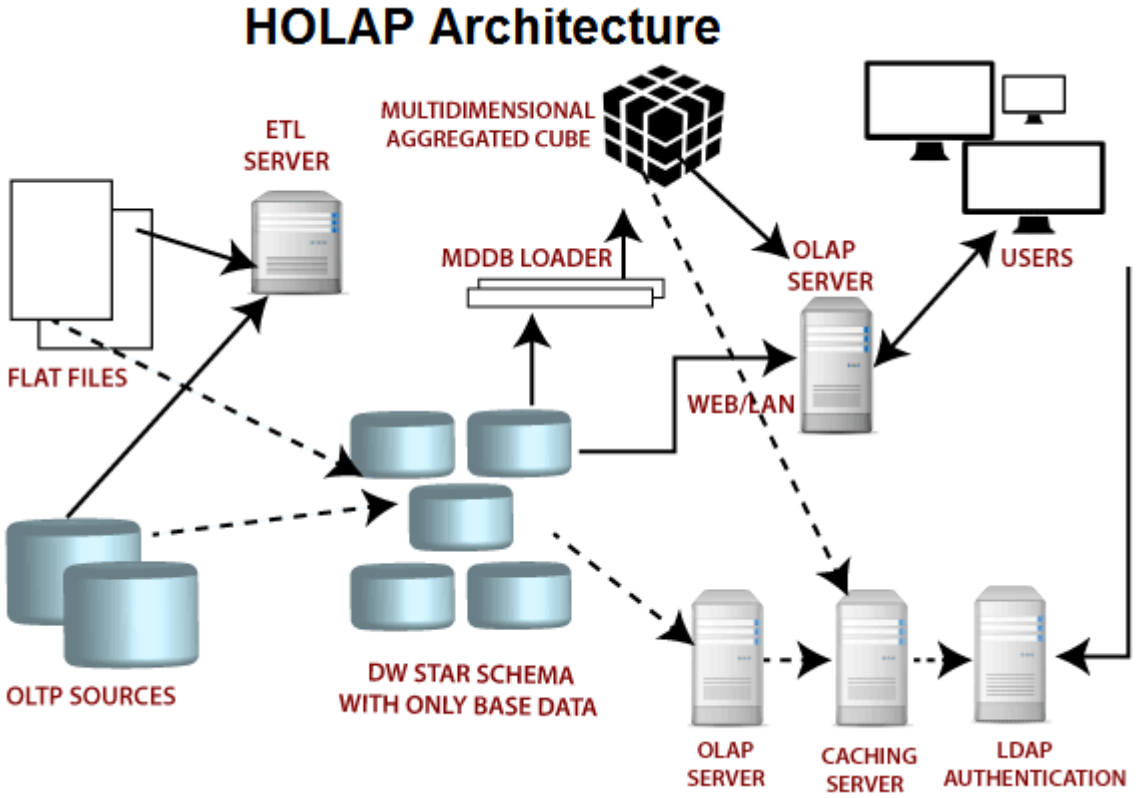
Limited in the amount of information it can handle: Because all calculations are performed when the cube is built, it is not possible to contain a large amount of data in the cube itself.

Requires additional investment: Cube technology is generally proprietary and does not already exist in the organization. Therefore, to adopt MOLAP technology, chances are other investments in human and capital resources are needed.

Hybrid OLAP (HOLAP) Server

HOLAP incorporates the best features of **MOLAP** and **ROLAP** into a single architecture. HOLAP systems save more substantial quantities of detailed data in the relational tables while the aggregations are stored in the pre-calculated cubes. HOLAP also can drill through from the

cube down to the relational tables for delineated data. The **Microsoft SQL Server 2000** provides a hybrid OLAP server.



Advantages of HOLAP

1. HOLAP provide benefits of both MOLAP and ROLAP.
2. It provides fast access at all levels of aggregation.
3. HOLAP balances the disk space requirement, as it only stores the aggregate information on the OLAP server and the detail record remains in the relational database. So no duplicate copy of the detail record is maintained.

Disadvantages of HOLAP

1. HOLAP architecture is very complicated because it supports both MOLAP and ROLAP servers.

Other Types

There are also less popular types of OLAP styles upon which one could stumble upon every so often. We have listed some of the less popular brands existing in the OLAP industry.

Web-Enabled OLAP (WOLAP) Server

WOLAP pertains to OLAP application which is accessible via the web browser. Unlike traditional client/server OLAP applications, WOLAP is considered to have a three-tiered architecture which consists of three components: a client, a middleware, and a database server.

Desktop OLAP (DOLAP) Server

DOLAP permits a user to download a section of the data from the database or source, and work with that dataset locally, or on their desktop.

Mobile OLAP (MOLAP) Server

Mobile OLAP enables users to access and work on OLAP data and applications remotely through the use of their mobile devices.

Spatial OLAP (SOLAP) Server

SOLAP includes the capabilities of both Geographic Information Systems (GIS) and OLAP into a single user interface. It facilitates the management of both spatial and non-spatial data

Difference between ROLAP, MOLAP, and HOLAP

ROLAP	MOLAP	HOLAP
ROLAP stands for Relational Online Analytical Processing.	MOLAP stands for Multidimensional Online Analytical Processing.	HOLAP stands for Hybrid Online Analytical Processing.
The ROLAP storage mode causes the aggregation of the division to be stored in indexed views in the relational database that was specified in the partition's data source.	The MOLAP storage mode principle the aggregations of the division and a copy of its source information to be saved in a multidimensional operation in analysis services when the separation is processed.	The HOLAP storage mode connects attributes of both MOLAP and ROLAP. Like MOLAP, HOLAP causes the aggregation of the division to be stored in a multidimensional operation in an SQL Server analysis services instance.
ROLAP does not because a copy of the source information to be stored in the Analysis services data folders. Instead, when the outcome cannot be derived from the query cache, the indexed views in the record source are accessed to answer queries.	This MOLAP operation is highly optimize to maximize query performance. The storage area can be on the computer where the partition is described or on another computer running Analysis services. Because a copy of the source information resides in the multidimensional operation, queries can be resolved without accessing the partition's source record.	HOLAP does not causes a copy of the source information to be stored. For queries that access the only summary record in the aggregations of a division, HOLAP is the equivalent of MOLAP.
Query response is frequently slower with ROLAP storage than with the MOLAP or HOLAP storage mode. Processing time is also frequently slower with ROLAP.	Query response times can be reduced substantially by using aggregations. The record in the partition's MOLAP operation is only as current as of the most recent processing of the separation.	Queries that access source record for example, if we want to drill down to an atomic cube cell for which there is no aggregation information must retrieve data from the relational database and will not be as fast as they would be if the source information were stored in the MOLAP architecture.