Expt. No. ..... Page No. .....

Unit - II

Data Warehouse and OLAP Technology for DM

Prepared By
P. Mansa Devi
Assistant Professor
CSE Dept. GIU

- Data Warehouse Definition
- Differences between OLTP & OLAP
- Why a Datawarehouse is separated from operational Databases.
- Advantages of Datawarehouse?

- Multi Dimensional Data Model
- Schemas for Multidimensional data models
- Categories of Measures
- 4 types of concept Hierarchies

- Datawarehouse Architecture
- Datawarehouse Models
  - a) Enterprise warehouse
  - b) Data Mart
  - c) Virtual Warehouse
- Extraction, Transformation & Loading
- Metadata Repository.

- Datawarehouse Implementation :-
  - a) Efficient Data cube Computation :- An overview
  - b) Indexing OLAP Data
  - c) Efficient Processing of OLAP Queries
  - d) OLAP server Architectures : ROLAP Vs MOLAP VS HOLA
- Development of Data Cube Technology
- Datawarehousing to Data Mining.

Teacher's Signature : _____

Date .....

Expt. No. ..... Page No. .....

unit-II

Prepared By
P. Mansa Devi (PMD)
Assistant Professor
CSE Dept, GI.U

Data warehouse:-

→ Data warehouse is a form of storage system (database) where large volume of data is stored in such a way that retrieving desirable information from the system is very easy and reliable.

→ Data warehouse is stored in different location so that it doesn't collide with transactional database system, which stores day-to-day information & answers the queries that are pre-recorded in the database.

Advantages of Data ware house:-

1. Data Warehouse is capable of storing and consolidating past information.
2. It provides support for sophisticated multidimensional queries.
3. Data warehouse increases the performance of integrated database system as data from hetero-geneous sources are extracted, pre-processed, cleaned, transformed into one unified data store.
4. It is capable of understanding the current business trends & making better forecasting decissions.

Teacher's Signature : _____

Characteristic (or) Features of data warehouse can be defined as,

① Subject-oriented (Not application-oriented).
② Consolidated data
③ Time-dependent data ④ Non-erasable data.

① Subject-oriented (Not application oriented):-

a) Data warehouse is to support decision makers in making strategic decisions.
b) For making decissions, it perform data analysis & by applying data modelling tools.
c) Data warehouse doesn't maintain information about day-to-day transactions of organization, but concentrate on the subjects which are critical to the organization.

② Consolidated Data:-

a) Data warehouse is capable of retrieving appropriate data from heterogeneous databases (like relational databases, flat files) in order to make efficient decissions.
b) Because of the heterogeneity, data is stored inconsistent manner. To confirm consistency and reliability in naming rules, encoding methods, different techniques such as data cleaning & data integration needs to be applied.
c) Therefore, it is necessary to perform data transformation, data consolation before transferring data from operational system into data warehouse.

(3) Time Dependent Data :-

Data warehouse database not only stores current information, but also stores historic information about a particular transaction.

→ In operational database, though historic information is stored, it generates only present information as these databases are capable of supporting only day-to-day transactions.

→ Data in Data warehouse are achieved as snapshots over historic & present time period.

→ In these databases, every data structures are time dependent ie, they are directly (or) indirec elements of time. This data warehouse approac is non-trivial for design as well as for implementation stages of data warehouse.

Advantages:-

a) It enables analyzing of historic data
b) It associates information of past data with present data.
c) It provides better future prediction.

④ Non-erasable Data:-

This feature confirms that once data enters the data warehouse it remains static until particular event is triggered. Data warehouse contains data which is extracted transformed, integrated from operational database.

→ The data is transferred from operational system to data warehouse at regular intervals of time depending on the specification of the business. There are only 2 operations executed by data warehouse for accessing data. (a) Data Loading method.

(b) Data access method.

Because of the above features, data warehouse can be considered as a consistent storage area that provides support for decision making & for analytical reporting

* Why a data warehouse is separate from operational system.

Operational databases store huge amounts of data why not perform online analytical processing directly on such databases instead of spending additional time and resources to construct a separate data warehouse?

→ The major reason for such a separation is to help promote the high performance of both systems.

* Reasons :-

1) Data warehouses are optimized, to execute select-type queries whereas, operational databases are optimized, to execute insert and update type queries.

2) Data warehouse schemes are simplified and denormalized, whereas the schemes of operational

Teacher's Signature : _____

systems are large and complex.

③ Data warehouse stores historic information whereas operational system stores only day-to-day information.

④ Data in Data warehouse are non-volatile (ie, do not change) whereas, data in operational system is highly volatile.

Feature	OLTP	OLAP
characteristic	operational processing	Informational processing
orientation	Transaction	Analysis
user	clerk, DBA, Database professional	Knowledge worker (eg, manager, executive, analyst)
function	Day-to-day operations	long-term informational requirements decission support.
DB Design	ER-Based, application-oriented current, guaranteed up-to-date.	Star/snowflake, subject-oriented, historic, accuracy maintained over time.

Date .....

Expt. No. ..... Page No. .....

Summarization	primitive, highly detailed	Summarized, Consolidated
view	detailed, flat relational	summarized, multidimensional
unit of work	short, simple transaction	complex query
Access	read/write	mostly read
Focus	data in	information out
operations	index/hash on primary key	lots of scans
no. of records accessed	tens	millions
priority	High performance, high availability	high flexibility, end-user autonomy
Metric	Transaction throughput	query throughput, response time.

* Multi Dimensional Model :-

OLAP :-

→ OLAP comprises of set of standards that are responsible for providing dimensional structure for supporting decision system.

→ The main purpose of OLAP is to perform data analysis and to access data on-line. It provides user-friendly interface for evaluating data interactively.

→ OLAP system consists of more complicated query outcomes than transactional database system. This analytical processing is performed on data warehouses which involve analysis of actual data.

→ OLAP is a type of s/w technology that allows system analysts, managers to understand the data using fast, interactive technologies.

* Multidimensional Data Model :-

① Multidimensional data models are used for representing OLAP tools and data warehouse.

② OLAP contain set of standards that are responsible for providing dimensional structure for supporting decision system.

③ In these models, data is viewed in the form of data cubes, which is defined in terms of dimensions and fact.

④ These data cubes are n-dimensional cube using which it is possible to model and represent data in number of dimensional levels.

Date .....

Expt. No..... Page No.....

→ Multidimensional data model is intuitively analytical and easy to use.

→ It conforms to the way how the users apprehend business issues.

Dimensions :-

→ Organizations store the records in accordance to the dimension, which are collection of modules of similar view type.

→ Let us consider, Automobile - sales database, it consists of sales repository for maintaining the information of sales that took place in a particular division with respect to product, place, time, division dimensions.

→ Individual dimension is represented in the form of a table i.e., a table is associated with every dimension in the database.

→ This table is referred as dimension table, in dimension table, explain each dimension's attributes in detail.

→ For Eg., the attributes for dimension table i.e, time are day, week, month, quarter, half, year.

→ Users (or) a person with good knowledge can define a dimension table.

Facts :-

① Multidimensional data model is usually organized based on specific subject.

② Fact table are used to represent a subject such as Automobile sales in All Automobile database. (or) All Electronic sales.

③ Fact is a group of data item which consist of numerical measures and data content.

④ With the help of fact it is easy to examine the correlation among different dimensions.

⑤ For a Automobile sales repository the facts are sales_in_rupees, item_sold etc.

⑥ Fact table not only consist of fact name but also unique keys associated with every dimension table.

Data cube :-

A data cube allows data to be modeled and viewed in multiple dimensions. It is defined by dimensions and facts.

Date .....

Expt. No. ..... Page No. .....

Example:-

For Eg., All Electronics may create a sales data warehouse in order to keep records of the store's sales with respect to the dimensions time, item, branch and location. These dimensions allows the store to keep track of things like monthly sales of items & the branches and locations at which the items were sold — contd.

Eg:- A 2-D view of sales data for All Electronics according to the dimensions time and item, where the sales are from branches located in the city of Vancouver. The measure displayed is dollars_sold (in thousands).

Table 3.2 A 2-D view of sales data for *AllElectronics* according to the dimensions *time* and *item*, where the sales are from branches located in the city of Vancouver. The measure displayed is *dollars_sold* (in thousands).

location = "Vancouver"				
	item (type)			
time (quarter)	home entertainment	computer	phone	security
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

Contd.

→ Each dimension may have a table associated with it, called a dimension table. ie, it describes the dimension. For Eg., a dimension table for Item may contain the attributes Item_name, Brand, & type.

→ Dimension table can be specified by users (or) experts (or) automatically generated & adjusted based on data distributions.

Teacher's Signature : _____

For understanding the data cubes and the multidimensional data model.

Eg Exploration. → Consider a representation of 2D view i.e., Suppose we want sales data from AllElecteonics. In particular we look at the AllElecteonics sales data for items sold per quarter in the city of Vancouver. These data are shown in tabular 3.2 (previous page).

→ In this 2-D representation, the sales for Vancouver are shown with respect to the (a) time dimension (organized in quarters) (b) item dimension (organized according to the types of items sold).

(c) The fact (or) measure displayed is dollars_sold (in thousands).

3D View

⇒ Now, suppose that we would like to view the sales data with a third dimension. For instance, suppose we would like to view the data according to time & item, as well as location for the cities, Chicago, New york, Toronto, and Vancouver.

→ These 3D data are shown in Table 3.3. are represented as a series of 2-D tables.

3D view

Table 3.3 A 3-D view of sales data for *AllElectronics*, according to the dimensions *time*, *item*, and *location*. The measure displayed is *dollars_sold* (in thousands).

	location = "Chicago"				location = "New York"				location = "Toronto"				location = "Vancouver"			
	item				item				item				item			
time	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.
Q1	854	882	89	623	1087	968	38	872	818	746	43	591	605	825	14	400
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682	680	952	31	512
Q3	1032	924	59	789	1034	1048	45	1002	940	795	58	728	812	1023	30	501
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784	927	1038	38	580

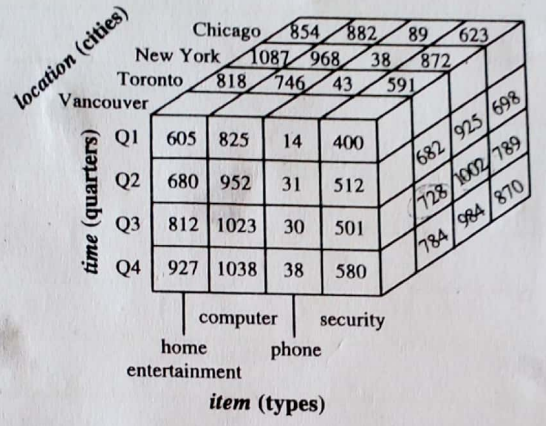


Figure 3.1 A 3-D data cube representation of the data in Table 3.3, according to the dimensions *time*, *item*, and *location*. The measure displayed is *dollars_sold* (in thousands).

Conceptually, we may also represent the same data in the form of a 3D-data cube as shown in Fig. 3.1.

Note:-

⇒ Till now, we have viewed the data in 2D form & 3D-form and now 4D-view.

4D View:-

⇒ Suppose that we would now like to view our sales data with an additional fourth dimension such as supplier.

Teacher's Signature :

→ Viewing things in 4D - becomes tricky. However, we can think of a 4-D cube as being a series of 3-D cubes, as shown Fig 3.2. as shown below.

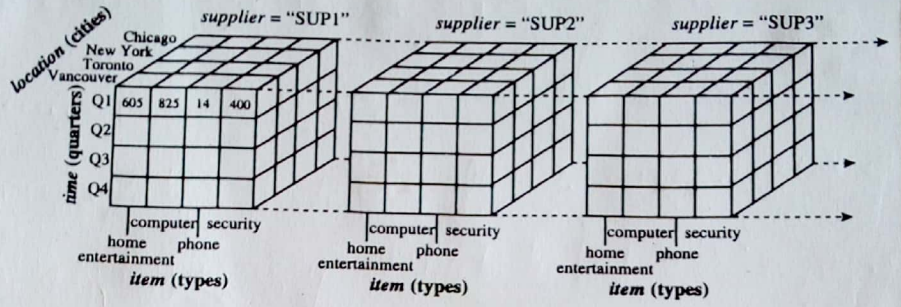


Figure 3.2 A 4-D data cube representation of sales data, according to the dimensions *time*, *item*, *location*, and *supplier*. The measure displayed is *dollars_sold* (in thousands). For improved readability, only some of the cube values are shown.

→ If we continue in this way, we may display any n-D data as a series of (n-1)-D cubes. The data cube is a metaphor for multidimensional data storage. The actual physical storage of such data may differ its logical representation. The important thing to remember is that data cubes are n-dimensional and do not confine data to 3-D.

→ The above tables show the data at different degrees of summarization. In the data warehousing research literature, a data cube such as each of the above is often referred to as a Cuboid.

→ Given a set of dimensions, we can generate a cuboid for each of the possible subsets of the given dimensions.

Date .....

Expt. No..... Page No.....

→ The result would form a lattice of cuboids, each showing the data at a different level of summarization (or) group by.

→ The lattice of cuboids is then referred to as a data cube.

→ Fig 3.3 shows a lattice of cuboids forming a data cube for the dimensions time, item, location and supplier.

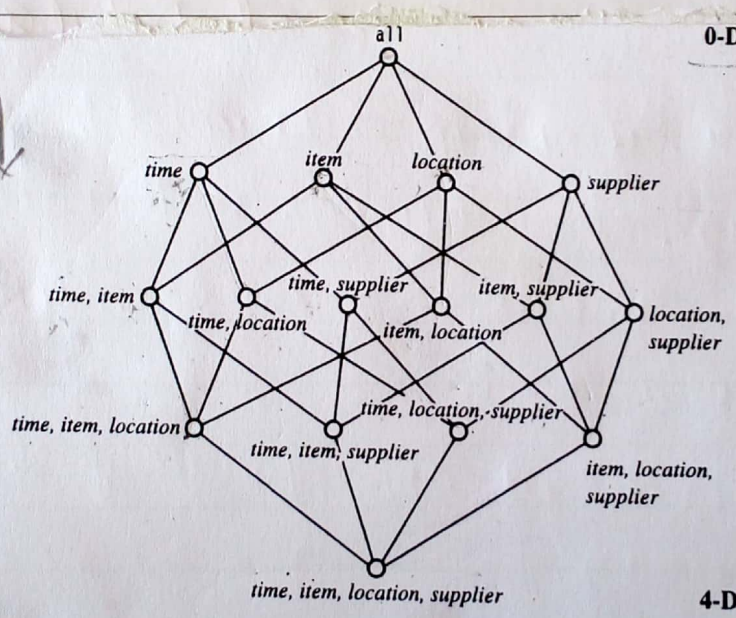


Figure 3.3 Lattice of cuboids, making up a 4-D data cube for the dimensions *time*, *item*, *location*, and *supplier*. Each cuboid represents a different degree of summarization.

→ The cuboid that holds the lowest level of summarization is called Base Cuboid.

→ For Eg., the 4-D cuboid in Fig 3.2 is the Base cuboid for the given time, item, location, and supplier dimensions.

→ Fig 3.1 is a 3-D (non-base) cuboid for time, item, & location, summarized for all suppliers.

→ The 0-D cuboid, which holds the highest level of summarization is called apex cuboid.

In our example, this is the total sales, (or) dollars sold, summarized over all four dimensions. The apex cuboid is typically denoted by all.

* Schemas for Multidimensional Databases :-

① The entity-relationship data model is commonly used in the design of relational databases, where a database schema consists of a set of entities and the relationship between them.

→ Such a data model is appropriate for on-line transaction processing.

② A data warehouse, however, requires a concise, subject-oriented schema that facilates online data analysis.

③ The most popular data model for data warehouse is a multidimensional model. Such a model can exist in the form of a star schema, a snowflake schema; Fact constellation schema.

Date .....

Expt. No. ..... Page No. .....

a) Star Schema :-

(1) The star schema is the basic data design method for data warehouse. It acquires data for querying and analysis.

(2) It is modelling pattern in which data warehouse include single fact table and set of dimension tables, one for each dimension.

(3) The schema graph resembles a star burst, with the dimension tables displayed in a radial pattern around the central fact table.

(4) Therefore, there is one-to-one relationship between the tuple of dimension table and fact table, but one-to-many relationship between fact table and dimension table.

Example :- Star Schema :- A star schema for AllElectronics sales is shown in Fig 3.4 :

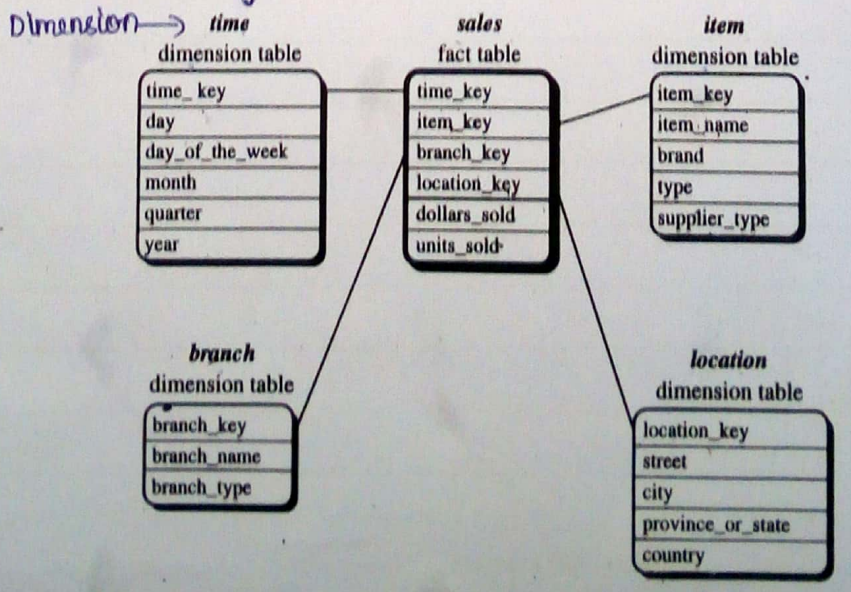


Figure 3.4 Star schema of a data warehouse for sales.

→ Sales are considered along 4 dimensions namely, time, item, branch, and location.

The schema contains a central fact table for sales that contains keys to each of the 4 dimensions, along with two measures :- dollars_sold & units_sold.

Notice, that in the star schema, each dimension is represented by only one table, and each table contains set of attributes.

For Eg, the location dimension table contains the attribute set {location_key, street, city, state, country}. This constraint may introduce some redundancy. For Example, "Vancouver" & "Victoria" are both cities in the candian State (or) province of British Columbia. Entries for such cities in the location dimension table will create redundancy among the attributes province (or) state and country ie, (Vancouver, British Columbia, Cannada) and (Victoria, British columbia, canada).

Moreover, the attributes within a dimension table may form either a hierarchy (total order) (or) a lattice (partial order).

Date .....

Expt. No. ..... Page No. .....

Advantages of Star Schema :-

1) It is easily understandable.
2) Using this schema, concept hierarchies can be defined easily.
3) It requires less maintenance.
4) It is more suitable for query processing.

Disadvantages :-

1) The performance of summary levels is reduced due to presence of summarized data.
2) It is problematic when dimension tables are large.

(6) Snow-Flake Schema :-

(1) The snowflake schema is a variant of the star schema model, where some dimension tables are normalized, thereby further splitting the data into additional tables.

(2) The major difference between the snowflake and star schema models is that the dimension tables of the snowflake model may be kept in normalized form to reduce redundancies.

(3) Such a table is easy to maintain and saves storage space. However, this saving of space.

(4) Snowflake structure can reduce the effectiveness of Browsing, since more joins will be needed to execute a query.

Teacher's Signature : _____

Example :-

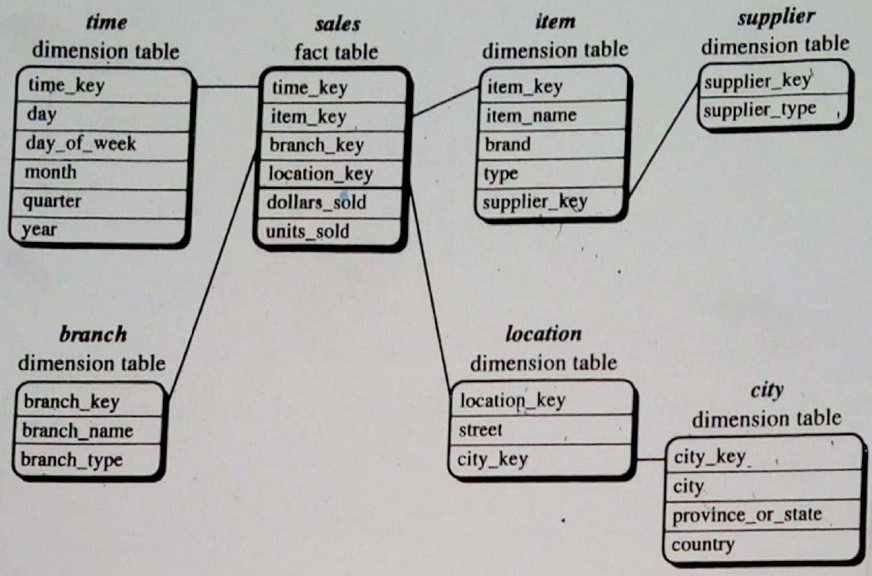


Figure 3.5 Snowflake schema of a data warehouse for sales.

→ A snowflake schema for All Electronics sales is given in Fig 3.5.

→ The main difference between the two schemas is in the definition of dimension tables.

→ Consider the single dimension table for item in the star schema is normalized & in the snowflake schema, resulting in new item & supplier tables. For Eg, the item dimension table now contains the attributes item_key, item_name, brand, type and supplier_key and supplier_type information.

→ Similarly, the single dimension table for location in the star schema can be normalized into ~~two~~ two new tables: Location and city.

Date .....

Expt. No. ..... Page No. .....

The city_key in the new location table links to the city dimension.

→ Notice that further normalization can be performed on province_(or)_state and country in the snowflake schema shown in fig 3.5.

Advantages :-

① It reduces redundancies.

② It is easy to maintain & update normalized tables.

Disadvantages :-

① The performance of the system is unfavourably effected.

② Snowflake schema is not very much intuitive and it is very difficult for end-users to understand.

(c) Fact Constellation Schema :-

Sophisticated applications may require multiple fact tables to share dimension tables. This kind of schema can be viewed as a collection of stars, & hence is called a galaxy schema (or) a fact Constellation.

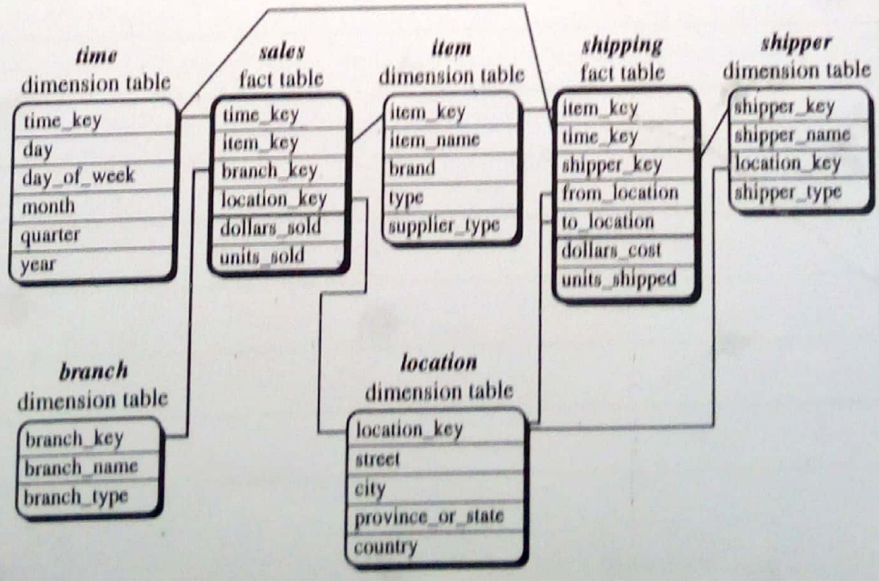


Figure 3.6 Fact constellation schema of a data warehouse for sales and shipping.

Eg:- This schema specifies two fact tables, sales and shipping.

→ The sales table definition is identical to that of the star schema (Fig 3.4).

→ The shipping table has 5 dimensions (or) keys :- item_key, time_key, shipper_key, from_location, and to_location.

→ And two measures : dollars_cost and units_shipped.

→ A fact constellation schema allows dimension tables to be shared between fact tables. ie, for an example, the dimension tables for time, item, and location are shared b/w both the sales and shipping fact tables.

Disadvantages:-

① It is highly developed and complex application.

② When the size of dimension tables are large, it can degrade the performance of the system.

Date .....

Expt. No. ..... Page No. .....

Examples for Defining Star, Snowflake, and Fact Constellation Schemas :-

Data warehouses & data marts can be defined using two language primitives, ⊛

(i) one for cube definition.

(ii) one for dimension definition.

→ The cube definition statement has the following syntax :-

define cube <cube_name>[<dimension_list>]:<measure_list>

→ The dimension definition statement has the following syntax:-

define dimension <dimension_name> as(<attribute_or_dimension_list>)

Note: ① Lets look at examples of how to define the star, snowflake, and fact constellation schemas of Examples 3.1 to 3.3 using DMQL.

② DMQL keywords are displayed in sans serif font.

Eg:- Star Schema Definition. The star schema of example 3.1 and Fig 3.4 is defined in DMQL as follows:-

define cube sales_star [time, item, branch, location]:
dollars_sold = sum(sales_in_dollars), units_sold = count(*)

Con

Teacher's Signature (continue in next page)

contd...

```
define dimension time as (time_key, day, day_of_week, month, quarter, year)
define dimension item as (item_key, item_name, brand, type, supplier_type)
define dimension branch as (branch_key, branch_name, branch_type)
define dimension location as (location_key, street, city, province_or_state,
    country)
```

The define cube statement defines a data cube called *sales_star*, which corresponds to the central *sales* fact table of Example 3.1. This command specifies the dimensions and the two measures, *dollars_sold* and *units_sold*. The data cube has four dimensions, namely, *time*, *item*, *branch*, and *location*. A define dimension statement is used to define each of the dimensions. ■

Example 3.5 **Snowflake schema definition.** The snowflake schema of Example 3.2 and Figure 3.5 is defined in DMQL as follows:

```
define cube sales_snowflake [time, item, branch, location]:
    dollars_sold = sum(sales_in_dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week, month, quarter, year)
define dimension item as (item_key, item_name, brand, type, supplier
    (supplier_key, supplier_type))
define dimension branch as (branch_key, branch_name, branch_type)
define dimension location as (location_key, street, city
    (city_key, city, province_or_state, country))
```

This definition is similar to that of *sales_star* (Example 3.4), except that, here, the *item* and *location* dimension tables are normalized. For instance, the *item* dimension of the *sales_star* data cube has been normalized in the *sales_snowflake* cube into two dimension tables, *item* and *supplier*. Note that the dimension definition for *supplier* is specified within the definition for *item*. Defining *supplier* in this way implicitly creates a *supplier_key* in the *item* dimension table definition. Similarly, the *location* dimension of the *sales_star* data cube has been normalized in the *sales_snowflake* cube into two dimension tables, *location* and *city*. The dimension definition for *city* is specified within the definition for *location*. In this way, a *city_key* is implicitly created in the *location* dimension table definition. ■

Finally, a fact constellation schema can be defined as a set of interconnected cubes. Below is an example.

Example 3.6 **Fact constellation schema definition.** The fact constellation schema of Example 3.3 and Figure 3.6 is defined in DMQL as follows:

```
define cube sales [time, item, branch, location]:
    dollars_sold = sum(sales_in_dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week, month, quarter, year)
define dimension item as (item_key, item_name, brand, type, supplier_type)
define dimension branch as (branch_key, branch_name, branch_type)
define dimension location as (location_key, street, city, province_or_state,
    country)

define cube shipping [time, item, shipper, from_location, to_location]:
    dollars_cost = sum(cost_in_dollars), units_shipped = count(*)
define dimension time as time in cube sales
define dimension item as item in cube sales
define dimension shipper as (shipper_key, shipper_name, location as
    location in cube sales, shipper_type)
define dimension from_location as location in cube sales
define dimension to_location as location in cube sales
```

A define cube statement is used to define data cubes for *sales* and *shipping*, corresponding to the two fact tables of the schema of Example 3.3. Note that the *time*, *item*, and *location* dimensions of the *sales* cube are shared with the *shipping* cube. This is indicated for the *time* dimension, for example, as follows. Under the define cube statement for *shipping*, the statement "define dimension *time* as *time* in cube *sales*" is specified. ■

Date .....

Expt. No. .....

Page No. .....

* **Measures :- Their Categorization & Computation :-**

→ How are measures computed?" To answer this question, we first study how measures can be categorized.

→ Note that a multidimensional point in the data cube space can be defined by a set of dimensions-value pairs, for Eg, <time = "Q1", location = "Vancouver", item = "computer">.

→ A data cube measure is a numerical function that can be evaluated at each point in the data cube space.

→ A measure value is computed for a given point by aggregating the data corresponding to the respective dimension-value pairs defining the given point.

* Measures can be organized into three categories :-

(a) Distributive
(b) Algebraic.
(c) Holistic

Based on the kind of aggregate functions used.

(a) Distributive :-

A measure is said to be distributive, if it is estimated in distributed manner which involve the following steps.

(i) Fragmenting the entire data set into smaller data subsets.

(ii) Applying the aggregate function on each

Teacher's Signature : _____

individual subset such as sum(), count().

(iii) Combining the outcomes of each subset so as to get a value which is equal to the measure value computed on entire data set after applying the same aggregate functions. Some of the distributive measures are sum(), count(), max(), min().

(b) Algebraic Measure:-

(i) A measure is said to be algebraic, if it is estimated by applying functions to atleast one distributive measures.

(ii) These algebraic functions consists of M arguments, where M represents a +ve integer.

(iii) Some of the algebraic measures include average() which is computed using distributive measures like sum() (or) count(), min_M(), max_M() & standard Deviation.

(c) Holistic Measure:-

A measure is said to be holistic, if it is computed on complete data set but not on the fragmented data subset by applying holistic aggregate functions.

Date .....

Expt. No. ..... Page No. .....

* Concept Hierarchies :-

① Concept Hierarchies defines a hierarchical mapping of low-level attributes to higher-level attribute values.

② Concept hierarchies are usually applied before data mining because mining generalized data requires fewer input/output operations & generates efficient results when compared to mining ungeneralized data set.

The 4 major types of concept hierarchies are,

a) Schema Hierarchies
b) Set grouping hierarchies
c) Operation-derived Hierarchies
d) Rule-Based hierarchies.

① Schema Hierarchies :-

a) A schema hierarchy specifies a partial (or) total ordering of attributes at the schema level.

b) Schema hierarchies can be specified by user, to indicate semantic relationships between attributes.

c) For an Example, Consider a student Engineering course. The possible attributes can be course, department, institution, University etc. Thus we can define a hierarchy as.

Course < Department < Institution < University

→ Hence schema hierarchy clearly specifies the relationship between the attribute.

Teacher's Signature : _____

② Set-grouping Hierarchies:-

a) Set-grouping hierarchy categorizes, dimension (or) attribute values into groups by specifying a range for each group.

b) This type of concept hierarchy is used for small set of relationships between objects.

c) Let us consider an set-grouping hierarchy for the attribute percentage that can be specified in terms of ranges of values to organize into groups.

[49-59] < second class
[60 - 74] < First class
[75 - 90] < Distinction.

③ Operation-Derived Hierarchies:-

a) Operation-derived hierarchies specify hierarchy information based on user operations.

b) The information is obtained by decoding operation like a set of URL's visited by user.

c) Information mining by a data mining system ca also be represented using operation derived hierarchies.

Date .....

Expt. No. .....

Page No. .....

For example, an URL specifies the hierarchy of a website on a web server. Let us consider the URL www.gitam.edu/Results/May-2017.

→ Here, the URL specifies the hierarchy of organization of data at GU web servers.
→ An operation can also extract the information from a complex string.

④ Rule-Based Hierarchy :-

a) Here a set of rules are used to specify the concept hierarchy. The hierarchy is verified against the rules and current database information.

b) Let us consider an example of categorizing students based on their total marks:
   a) Distinction (b) First class (c) Second class

c) The rules are as follows :-
   Distinction(x) <= Marks(x) > 700.
   First class(x) <= Marks(x) > 500 & marks(x) < 650
   Second class(x) <= Marks(x) < 400.

* OLAP Operations :-

→ Data in multidimensional model is arranged at different level of granularity of dimensions defined by Concept hierarchies.

Teacher's Signature : _____

→ In the multidimensional model, data are organized into multiple dimensions, and each dimension contains multiple levels of abstraction defined by concept hierarchies. This organization provides users with the flexibility to view data from different perspectives.

→ A number of OLAP data cube operations exist to materialize these different views, allowing interactive querying and analysis of the data at hand. Hence, OLAP provides user-friendly environment for interactive data analysis.

OLAP operations are as follows:-

1) Roll-up
2) Drill-Down
3) Slice and Dice.
4) Pivot (rotate).

Example:-

OLAP operations:- -)

a) Each operation described. Consider the ~~previous~~ Eg. in Fig 3.10 (Sticked in next page).

b) Consider the data cube for AllElectronics sales. The cube contains the dimensions as location,

Date .....

Expt. No. ..... Page No. .....

time, and item.

→ where location is aggregated with respect to city values.

→ Time is aggregated with respect to quarters,

→ And Item is aggregated w.r.to item types.

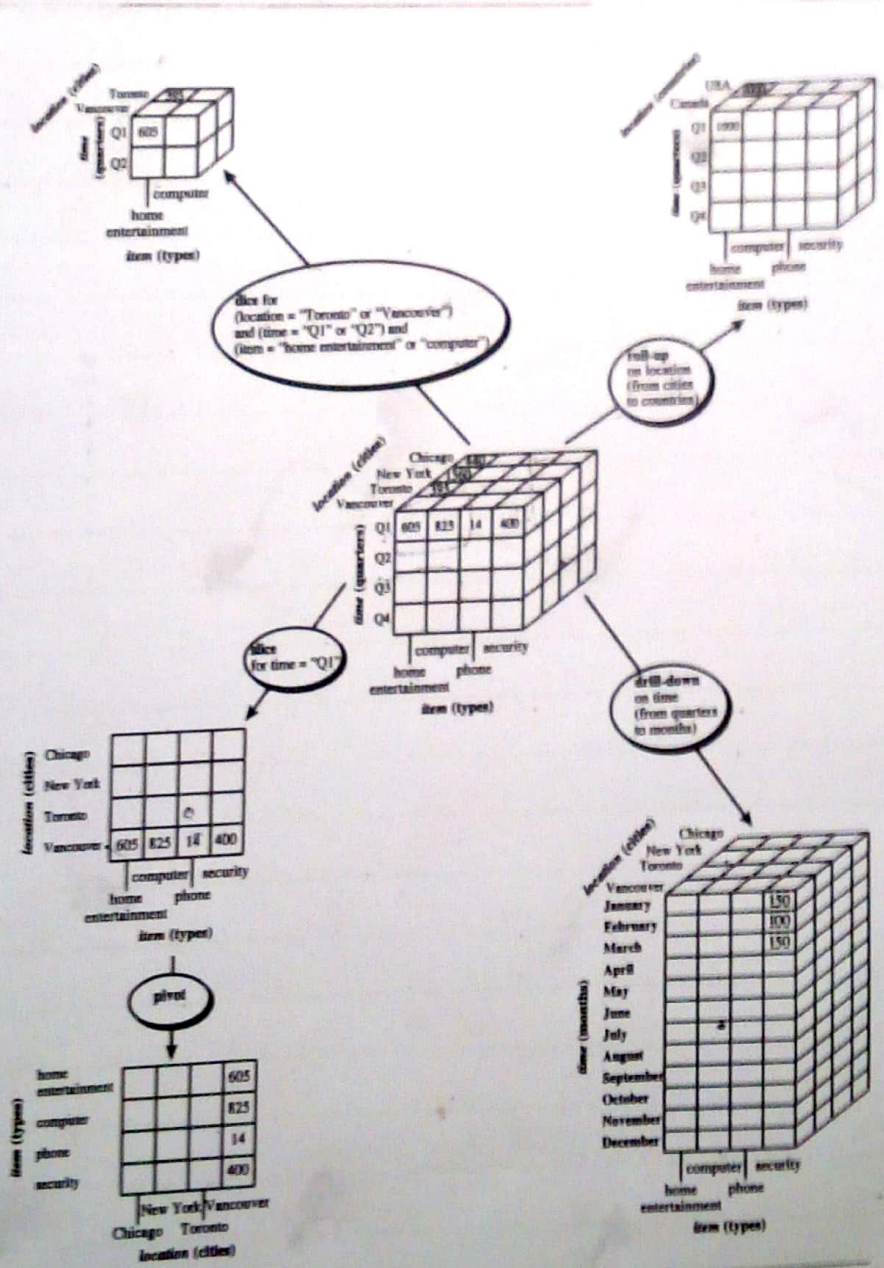


Figure 3.10 Examples of typical OLAP operations on multidimensional data.

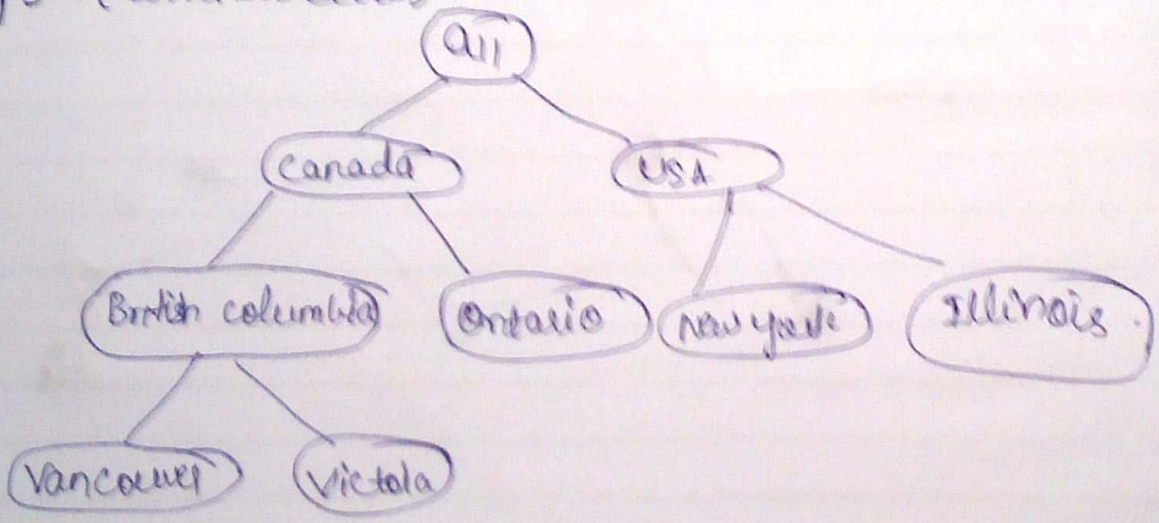
Teacher's Signature :

① **Roll-up :-**

① The Roll-up operation also called the drill-up operation.

② It performs aggregation on a data cube, (or) either by climbing up a concept hierarchy for a dimension (or) by dimension reduction.

③ Fig 3.10 shows the result of roll-up operation performed on the central cube by climbing up the concept hierarchy for location in Fig 3.7 (sticked below).



④ This hierarchy was defined as the total order "street < city < state < country."

⑤ The roll-up operation shown aggregates the data by ascending the location hierarchy from the level of city to the level of country.

Date .....

Expt. No..... Page No.....

② Drill - Down :-

a) Drill - Down is the reverse of roll - up. It navigates from less detailed data to more detailed data.

b) Drill - down can be realized by either stepping down a concept hierarchy for a dimension (or) introducing additional dimensions.

c) Fig 3.10 shows the result of a drill - down operation performed on the central cube by stepping down a concept hierarchy for time. ie., defined as "day < month < quarter < year".

d) Drill - Down occurs by descending the time hierarchy from the level of quarter to the more detailed level of month. The resulting data cube details the total sales per month rather than summarizing them by quarter.

③ Slice and Dice :-

a) The slice operation performs a selection on one dimension of the given cube, resulting in subcube.

b) Fig 3.10 shows a slice operations, where the sales data are selected from the

Teacher's Signature : _____

central cube for the dimension time using the criterion time = "Q_1".

(c) The dice operation defines a sub-cube by performing a selection on two (or) more dimensions.

(d) Fig 3.10 shows a dice operation on the central cube based on the following selection criteria that involve 3 dimensions :-

(location = "Toronto" or "Vancouver") and (time = "Q_1" (or) "Q_2") and ("item = "home entertainment (or) "computer").

) Pivot (Rotate) :-

a) Pivot (also called rotate) is a visualization operation that rotates the data axes in view in order to provide an alternative presentation of the data.

b) Fig 3.10 shows a pivot operation where the item and location axes in a 2-D slice are rotated.

other examples include rotating the axes in a 3-D cube (or) transforming a 3-D cube into a series of 2D planes.

* Starnet Query Model for Querying Multidimensional Databases :-

a) The Querying of multidimensional databases can be based on a starnet model.

b) A starnet model consists of radial lines emanating from a central point, where each line represents a concept hierarchy for a dimension. Each abstraction level in the hierarchy is called a footprint.

Eg: A Starnet query model for the "AllElectronics" datawarehouse :-

Expt. No. ..... Page No. .....

This Starnet consists of four radial lines, ~~represe~~ representing concept hierarchies for the dimensions location, customer, item and Time. Each line consists of footprints representing abstraction levels of the dimension.

For example, the time line has four footprints as follows: day, month, quarter, year. A concept hierarchy may involve a single attribute (like date for the time hierarchy) or several attributes (Eg., the concept hierarchy for location involves the attributes street, city, state and country). In order to examine the item sales at All Electronics, users can roll up along the time dimension from month to quarter, or say drill down along the location dimension from country to city.

Concept hierarchies can be used to generalize data by replacing low-level values (such as 'day' for the time dimension) by higher-level abstractions (such as 'year') or to specialize data by replacing higher-level abstractions with lower-level values.

* Data warehouse Architecture :-

→ Steps for the Design and construction of Data warehouses

→ 3-Tier Data warehouse Architecture

→ Data warehouse BackEnd Tools & utilities

→ Meta data Repository

→ Types of OLAP servers :- ROLAP Vs MOLAP Vs HOLAP

* Steps for the Design and Construction of Data warehouses :-

a) The Design of a Data Warehouse : A Business Analysis Framework.

b) The Process of Data warehouse Design

a) The Design of a Data Warehouse : A Business Analysis Framework :-

"What can Business Analysts gain from having a Data warehouse?

1) Having a Data warehouse may provide a competitive advantage by presenting relevant information from which to measure performance and make critical adjustments in order to help win over competitors.

2) A Data warehouse can enhance business productivity because it is able to quickly and efficiently gather information that accurately describes the organization.

3) A data warehouse facilitates customer relationship management because it provides a consistent view of customers and items across all lines of business, all departments, and all markets.

4) Finally, data warehouse may bring about cost reduction by tracking trends, patterns, and exceptions over long periods in a consistent and reliable manner.

* To design an effective data warehouse we need to understand and analyze business needs and construct a business analysis framework.

→ The construction of a large and complex information system can be viewed as the construction of a large and complex building, for which the owner, architect, and builder have different views.

→ These views are combined to form a complex framework that represents the top-down, business driven & as well as the bottom-up, implementors view of the information system.

* Four different views regarding the design of a data warehouse must be considered :-

a) The top-down view
b) The Data source view
c) The Data warehouse view
d) The Business Query view

Teacher's Signature : _____

① Top-down view:-

It allows the selection of the relevant information necessary for the data warehouse. This information matches the current and future business needs.

② Data source view:-

→ It exposes the information being captured, stored and managed by operational systems.

→ This information may be documented at various levels of detail & accuracy, from individual data source tables to integrated data source tables.

③ Data warehouse view:-

→ It includes fact tables and dimension tables. It represents the information that is stored inside the data warehouse & as well as information regarding the source, data, & time of origin, added to provide historical context.

④ Finally, Business query view is the perspective of data in the Data warehouse from the viewpoint of the end user.

Date .....

Expt. No. ..... Page No. .....

(b) The Process of Data Warehouse Design :-

A data warehouse can be built using a top-down approach, a bottom-up approach, (or) a combination of both.

→ The top-down approach starts with the overall design and planning, where the business problems that must be solved are clear & well understood.

→ The Bottom-up approach starts with experiments and prototypes. This is useful in the early stage of business modeling and technology development.

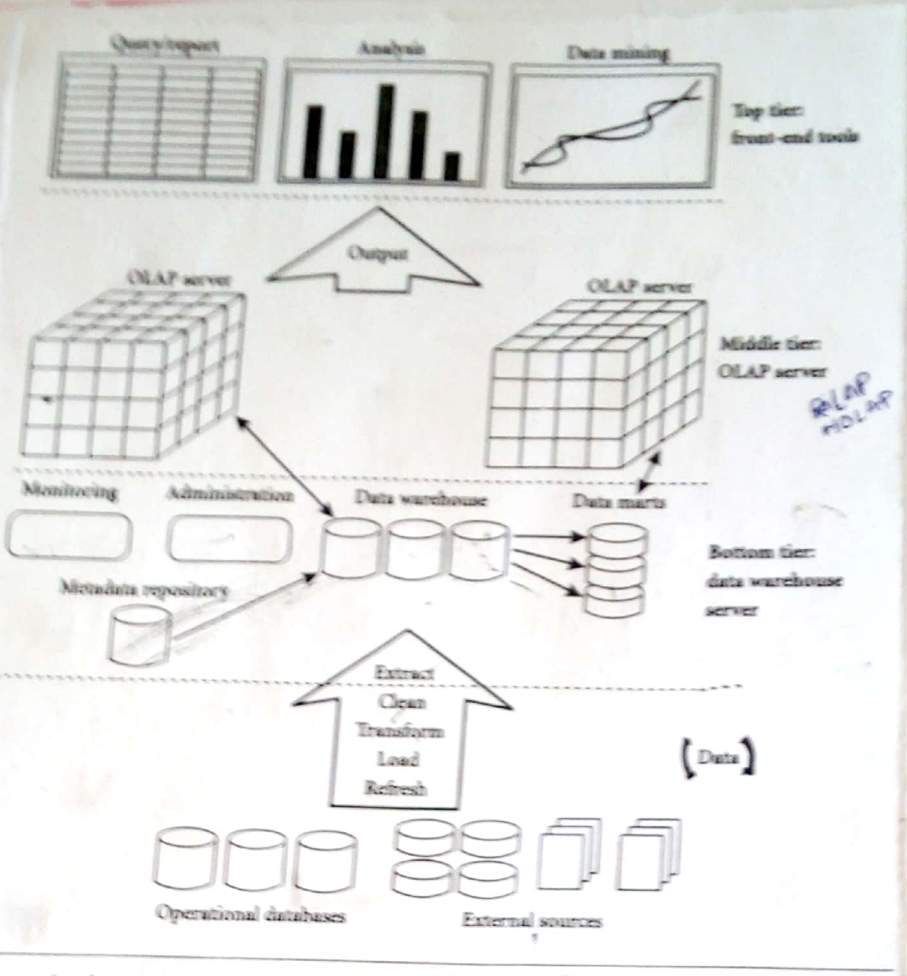
→ In the combined approach, an organization can exploit the planned and strategic nature of the top-down approach while retaining the rapid implementation and opportunistic application of the bottom-up approach.

→ In general, the data warehouse design process consists of the following steps :-

a) choose a business process to model
b) choose the grain of the business process
c) choose the dimensions.
d) choose the measures

Teacher's Signature : _____

* 3-Tier Data Warehouse Architecture :-



A three-tier data warehousing architecture. - Fig : 3.12

(1) → The Bottom tier is a warehouse database server that is almost a relational database system.

→ Back-end tools and utilities are used to feed data into the bottom tier from operational databases (or) other external sources (such as customer profile information provided by external consultants).

→ These tools and utilities perform data extraction, cleaning, and transformation.

Date .....

Expt. No. ..... Page No. .....

→ The data are extracted using application program known as gateways.

→ This tier also contains a metadata repository, which stores information about the data warehouse & its contents.

② Middle-Tier :-

→ The middle tier is an OLAP server that is typically implemented using either

(i) A Relational OLAP (ROLAP) model, ie, extended relational DBMS that maps operations on multidimensional data to standard relational operations.

(ii) Multidimensional OLAP (MOLAP) model ie., special-purpose server that directly implements multidimensional data and operations.

③ The top-tier is a front-end client layer, which contains query and reporting tools, analysis tools and data mining tools.

From the architecture point of view, there are 3 data warehouse models:

① The enterprise warehouse
② The data mart
③ The virtual warehouse

Teacher's Signature : _____

① Enterprise Warehouse :-

a) An enterprise warehouse collects all of the information about subjects spanning the entire organization.

b) It provides corporate-wide data integration, usually from one (or) more operational systems (or) external information.

c) It contains detailed data as well as summarized data, and can range in size from a few gigabytes to hundreds of gigabyte, terabytes.

d) It requires extensive business modelling and may take years to design and build.

② Data Mart :-

a) A data mart contains a subset of corporate-wide data that is of value to a specific group of users. The scope is confined to specific selected subjects. For example, a marketing data mart may confine its subjects to customer, item, and sales.

b) Depending on the source of data, data marts can be categorized as independent (or) dependent.

Date .....

Expt. No. ..... Page No. .....

(c) Independent data marts are sourced from data captured from one (or) more operational systems (or) external information providers.

(d) Dependent data marts are sourced directly from enterprise data warehouses.

(3) Virtual warehouse :-

a) A virtual warehouse is a set of views over operational databases.

b) For efficient query processing, only some of the possible summary views may be materialized. A virtual warehouse is easy to build that requires excess capacity on operational database servers.

* Data Warehouse Back-End Tools and Utilities :-

Data Warehouse systems use back-end tools and utilities to populate and refresh their data. (Fig 3.12). These tools and utilities include the following functions :-

a) Data Extraction - which typically gathers data from multiple, heterogeneous and external sources.

b) Data cleaning - which detects errors in the data and rectifies them when possible.

Teacher's Signature : _____

(c) Data transformation - which converts data from host format to warehouse format.

(d) Load - which sorts, summarizes, consolidates, computes views, checks integrity & builds indices and partitions.

(e) Refresh - which propagates the updates from the data sources to the warehouses.

Besides cleaning, loading, refreshing and metadata definition tools, data warehouse systems usually provide a good set of data warehouse management tools.

* Metadata Repository :-

→ Metadata are data about data. When used in a data warehouse, metadata are the data that define warehouse objects.

→ Fig 3.12 showed a metadata repository within the bottom tier of the data warehousing architecture.

→ Metadata are created for the data names and definitions of the given warehouse.

→ A metadata repository should contain the following

a) A description of the structure of the data warehouse, which includes the warehouse schema, view, dimensions hierarchies, and derived data definitions as well as data mart locations and contents.

b) Operational metadata, which include data lineage (history of migrated data & the sequence of transformations applied to it), and monitoring information (warehouse usage statistics, error reports and audit trails).

c) The algorithms used for summarization, which include measure and dimension definition algorithms, data on granularity, partitions, subject areas, aggregation.

d) Data related to system performance, which include indices and profiles that improve data access and retrieval performance.

* Types of OLAP servers :-

ROLAP	MOLAP	HOLAP
① ROLAP refers to relational online analytical processing server.	① MOLAP refers to multidimensional online analytical processing server.	① HOLAP refers to hybrid online analytical processing server

	ROLAP	MOLAP	HOLAP
(2)	It is based on tuples & relational tables, which are represented in the form of 3NF, star (or) snowflake.	It is based on adhoc logical model, which is represented in the form of datacubes	It is based on both relational data tables & precalculated data cubes.
(3)	It consumes less amount of time for data processing.	It consumes more amount of time for data processing.	It consumes less amount of time for data processing.
(4)	It works with relational databases.	It works with multi-dimensional array based storage.	It works with both relational & multi-dimensional data storage.
(5)	It requires only one-level of storage representation.	It requires two-level of storage repres--ntation.	It requires 2-level of storage representation
(6)	It is less complex	It is more complex	It is more complex
(7)	It is more scalable	It is less scalable	It is more scalable
(8)	The functionality is used in business objects.	The functionality is used in cognos.	The functionality is used in both business objects & cognos

Date

Expt. No. Page No.

ROLAP	MOLAP	HOLAP
⑧ The Query performance is bad.	⑧ The query performance is good.	⑧ The Query performance is good.
⑩ Example of ROLAP is IBM.	⑩ Eg is applix TM1	⑩ Eg of HOLAP is microsoft's OLAP server (which is part of SQL server)

* Data Warehouse Implementation :-

① Data warehouse contain huge volumes of data.
② OLAP servers demand that decision support queries be answered in the order of seconds.
③ It is crucial for DWH systems to support highly efficient cube computation techniques, access methods & query processing techniques.

→ ⓐ Efficient Computation of Data Cubes.
→ The Compute cube Operator & the curse of Dimensionality.
** → Partial Materialization : Selected Computation of Cuboids.

ⓑ Indexing OLAP Data :-
ⓐ Bitmap
ⓑ Join
ⓒ.

ⓒ Efficient Processing of OLAP Queries

Teacher's Signature :

① Efficient Computation of Data Cubes :-

→ At the core of multidimensional data analysis is the efficient computation of aggregations across many sets of dimensions.

→ In SQL terms, these aggregations are referred to as group-bys. Each group-by can be represented by a cuboid, where the set of group-by's forms a lattice of cuboids defining a data cube.

→ The Compute cube Operator and the curse of Dimensionality :-

→ There is one approach to cube computation ie, extends SQL so as to include a compute cube operator.

→ The compute cube operator computes aggregates over all subsets of the dimensions specified in the operation. This can require excessive storage space, especially for large number of data cubes.

Example :- 3.11

A data cube is a lattice of cuboids.

Suppose that you would like to create a data cube for AllElectronics sales that contains the following :-

Date .....

Expt. No. ..... Page No. .....

a) city (b) item (c) year and sales_in_dollars.

Now, you would like to be able to analyze the data, with queries such as the following:

"Compute the sum of sales, grouping by city & item"

"Compute the sum of sales, grouping by city"

"Compute the sum of sales, grouping by item"

→ And now develop the in form of data cube.

* What is the total number of cuboids, (or) group-by's, that can be computed for this data cube?

ns: Consider the 3 attributes :- city, item, and year as the dimensions for the data cube, and sales_in_dollars as the measure, the total number of cuboids, (or) group-by's that can be computed for this data cube is $2^3 = 8$.

The possible group-by's are the following:

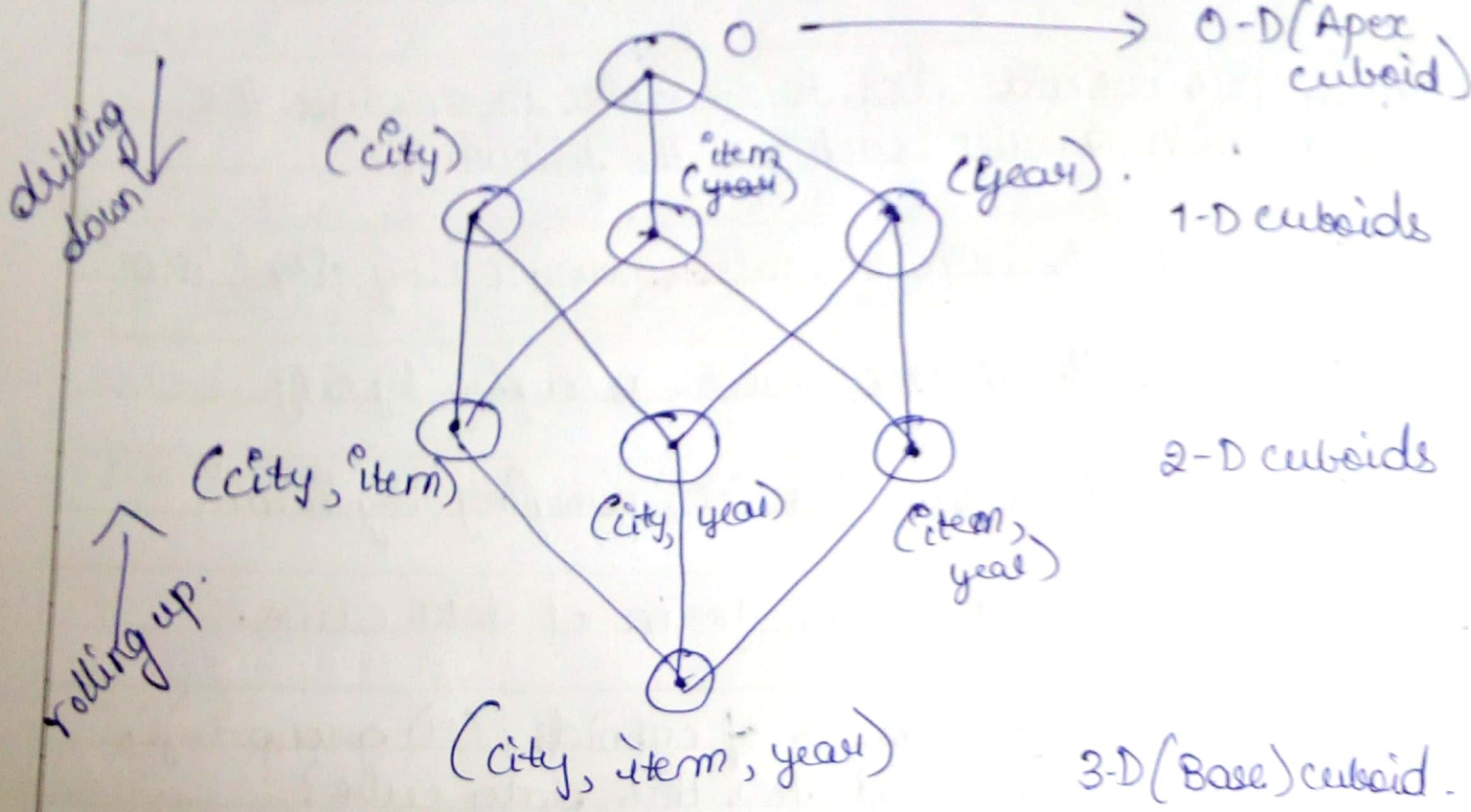
{(city, item, year), (city, item), (city, year), (item, year), (city), (item), (year), ()},

↓

where() means that the group-by empty (i.e., the dimensions are not grouped). These group's-by's

Teacher's Signature : _____

form a lattice of cuboids for the data cube as shown in below figure:-



→ The base cuboid contains all 3 dimensions, city, item, and year. It can return the total sales for any combination of the 3 dimensions.

→ The apex cuboid (or) 0-D cuboid, refers to the case where the group-by is empty. It contains the total sum of all sales.

→ The base cuboid is the least generalized (most specific) of the cuboids.

→ The apex cuboid is the most generalized (least specific) of the cuboids.

Date .....

Expt. No. ..... Page No. .....

→ If we start at the apex cuboid and explore downward in the lattice, this is equivalent to drilling down within the data cube.

→ If we start at the base cuboid and explore upward, this is akin to rolling up.

→ An SQL Query containing no group-by, such as "Compute the sum of total sales", is zero-dimensional operation.

→ An SQL Query containing one group-by, such as "Compute the sum of sales, group by city" is a one-dimensional operation.

→ A cube operator on n dimensions is equivalent to a collection of group-by statements, one for each subset of the n dimensions.

→ By Considering the DMQL, the data cube in Eg 3.11 could be defined as

```
define cube sales_cube [city, item, year]: sum(sale_in
```

→ For a cube with n dimensions, there are total of 2^n cuboids, including the base cuboid. A statement such as

```
Compute cube sales_cube
```

Teacher's Signature : _____

Curse of Dimensionality :-

A major challenge related to this precomputation, however is that required storage space may explode, if all of the cuboids in a data cube are pre-computed, especially when the cube has many dimensions. The storage requirements are even more excessive when many of the dimensions have associated concept hierarchies, each with multiple levels. This problem is referred to as the curse of dimensionality.

"How many cuboids are there in an n-dimensional data cube?". If there were no hierarchies associated with each dimension, then the total number of cuboids for an n-dimensional data cube, is 2^n. However, in practice, many dimensions do have hierarchies. For Example, the dimension time is usually not explored at only one conceptual level, such as year, but rather at multiple conceptual levels, such as in the hierarchy "day < month < quarter < year". For an n-dimensional data cube, the total number of cuboids that

can be generated (including the cuboids generated by climbing up the hierarchies along each dimension) is

$$\text{Total number of cuboids} = \prod_{i=1}^{n} (L_i + 1)$$

where L_i is the number of levels associated with dimension i.

→ This formula is based on the fact that, at most one abstraction level in each dimension will appear in the cuboid.

(2) Partial Materialization : Selected Computation of Cuboids :-

There are 3 choices for the data cube materialization given a base cuboid :-

a) No Materialization
b) Full Materialization
c) Partial Materialization.

a) No Materialization :-

Do not precompute any of the "non-base" cuboids. This leads to computing expensive multidimensional aggregates on the fly, which can be extremely slow.

Teacher's Signature :

② Full Materialization :-

Precompute all of the cuboids. The resulting lattice of computed cuboids is referred to as the full cube. This choice typically requires huge amounts of memory space in order to store of all precomputed cuboids.

③ Partial Materialization :-

Selectively compute a proper subset of the whole set of possible cuboids. Alternatively, we may compute a subset of the cube, which contains only those cells that satisfy some user-specified criterion, such as where the tuple count of each cell is above some threshold.

(5)

* Development of Data cube Technology :-

The development of Data cube Technology describes

a) Discovery - Driven Exploration of Data cubes.

→ Anamolies in the data are automatically detected and marked for the user with visual cues.

b) Multifeature cubes for complex DM Queries :-

→ Involves multiple dependent aggregates at multiple granularities.

5.1

a) Discovery - Driven Exploration of data cubes :-

→ The entire unit talks on how the data is summarized and stored in a multidimensional data cube of an OLAP system.

→ A user/analyst searches for intersecting patterns using various OLAP operations such as roll-up, drill down, slice & dice.

→ The discovery is not automated, instead, it is the user who following his/her own intution/hypothesis to identify the exceptions/anomalies in the data.

Date .....

Expt. No. ..... Page No. .....

→ The above process is hypothesis - driven exploration and has disadvantages.

→ Discovery driven exploration is an alternative approach in which precomputed measures indicating data exceptions are used to guide the user in the data analysis process, at all levels of aggregation.

→ These measures are exception indicators. An exception is a data cube cell value that is significantly different from the value anticipated, based on statistical model.

→ For Eg, if the analysis of item_sales data reveals an increase in sales in December in comparison to all other months, this is exception if time division is considered, since there is a similar increase in sales for other items during December.

→ Visual cues such as background color are used to reflect the degree of exception of each cell, based on the precomputed exception indicators.

→ Three measures are used as exception indicators to help identify data anomalies.

→ These measures are computed and associated with every cell, for all aggregation levels. They are as follows:-

① SelfExp
② InExp
③ Path Exp.

① SelfExp :- Indicates the degree of surprise of the cell value, relative to other cells at the same aggregation level.

② InExp :- Indicates the degree of surprise somewhere beneath the cell, if we were to drill down from it.

③ Path Exp :- Indicates the degree of surprise for each down path from the cell.

How can the data cube be efficiently constructed for discovery-driven exploration

→ This computation consists of 3 phases :-

a) First phase involves computation of the aggregate values such as sum/count over which exceptions will be found.

b) Second phase consists of model fitting, in which the co-efficients are determined & used to

Date .....

Expt. No. ..... Page No. .....

compute the standardized residuals (difference between a given cell value and its expected value).

c) Third phase computes the selfExp, InExp and pathExp values, based on the standardized residuals.

→ Therefore, the computation of data cubes for discovery-driven exploration can be done efficiently.

5.2 Complex aggregation at multiple granularities: Multifeature cubes :-

① Data cubes facilitate the answering of analytical (or) mining-oriented queries as they follow computation of aggregate data at multiple granularity levels.

② Traditional data cubes are typically constructed on commonly used dimensions (Eg., time, location, and product) using simple measures (Eg., count(), average(), and sum()).

③ Multifeature cubes enable more in-depth analysis.

④ This can compute more complex queries of which

Teacher's Signature : _____

the measures depend on groupings of multiple aggregates at varying granularity levels.

(5) Many complex data mining queries can be answered by multifeature cubes without significant increase in computational cost, in comparision to cube computation for simple queries with traditional data cubes.

Eg 1:- A simple data cube query: Let the query be "Find the total sales in 2010, broke down by item, region, and month, with subtotals for each dimension".

Ans:- A traditional data cube is constructed that aggregates the total sales at the following eight different granularity levels:

{(item, region, month), (item, region), (item, month), (month, region), (item), (month), (region), ()}

→ This does not involve any dependent aggregate

Eg2:- A complex query involving dependent aggregates:-

Suppose the query is "Grouping by all subsets of {item, region, month}, find the maximum price in 2010 for each group and the total sales among all maximum price tuples"?

Ans:- The specification of such a query using standard SQL can be long, repetitive and difficult to optimize and maintain. Using extended SQL its difficulty can be reduced.

```
Select item, region, month, max(price), sum(R.sa
from Purchases
where year = 2010
cube by item, region, month: R
such that R.price = max(price)
```

→ The tuples representing purchases in 2010 are first selected.

→ The cube by computes aggregates (group-by's) for all possible combinations of item, region and month.

→ The attributes here are grouping attributes.

→ The variable A is a grouping variable.

→ The modelling cube is a multifeature cube in that it supports complex data mining queries for which multiple dependent aggregates are computed on the set of many pairs triples for each group.

Expt. No. ..... Page No. .....

* From Data Warehousing to Data Mining :-

"How do Data warehousing and OLAP relate to Data mining?"

→ In this section, we study the usage of data warehousing for information processing, Analytical processing and data mining.

→ We will also discuss about on-line Analytical mining (OLAM), a powerful paradigm that integrates OLAP with data mining technology.

a) Data Warehouse Usage :-
b) From Online Analytical processing to on-line Analytical mining.
   ↳ Architecture of OLAM.

a) Data Warehouse Usage :-

→ Data warehouses and data marts are used in wide range of applications.

→ Business Executives use the data in data warehouses & data marts to perform data analysis and make decisions.

→ In many firms, data warehouses are used as an integral part of a plan - execute - assess "closed loop" feedback system for enterprise management.

Teacher's Signature : _____

→ Data warehouses are used extensively in Banking and financial services, consumer goods and retail distribution sectors.

→ Longer a data warehouse has been in use, the more it will have evolved. This evolution takes place throughout a number of phases.

⇒ Initially, the data warehouse is mainly used for generating reports and answering predefined queries.

⇒ Progressively, it is used to analyze summarized & detailed data, where the results are presented in the form of reports and charts.

⇒ Later, the data warehouse is used for strategic purposes, performing multidimensional analysis & slice & dice operations.

⇒ Finally, the data warehouse may be employed for Knowledge discovery & strategic decision making using data mining tools.

→ In this context, the tools for data warehousing can be categorized into access and retrieval tools, database reporting tools, data analysis tools & DM tools.

→ Business users, need to have the means to know ~~what~~ as follows:-

(i) What exists in the DWH.
(ii) How to access the contents of the DWH.
(iii) How to examine the contents using analysis tools.
(iv) How to present the results of such analysis.

→ There are 3 kinds of Data warehouse applications :-

a) Information processing.
b) Analytical processing
c) Data Mining.

a) Information processing :-

→ It supports querying, basic statistical analysis, and reporting using tables, charts,

→ A current trend in data warehouse information processing is to construct low cost web-based accessing tools that are then integrated with web browsers.

b) Analytical processing :-

→ It supports basic OLAP operations, including slice-&-dice, drill-down, roll up, & pivoting.

→ It generally operates on historical data in both summarized & detailed forms.

→ The major strength of online-analytical processing over information processing is the multidimensional data analysis of data warehouse data.

c) Data Mining :-

It supports knowledge discovery by finding hidden patterns and associations, constructing analytical models, performing classification and prediction, & presenting the mining results using visualization tools.

Note:- How does DM relate to information processing and on-line analytical processing?

Ans:- (a) 1) Information processing, based on queries, can find useful information.

2) However, answers to such queries reflect the information directly stored in databases. But they donot reflect the patterns.

3) Therefore, Information processing is not DM.

b) Analytical processing :-

① It is step closer to DM because it can deriv

Information summarized at multiple granularities from user-specified subsets of a data warehouse.

(Q) Note:- "Do OLAP systems perform DM? Are OLAP systems actually DM systems?".

Ans:- The functionalities of OLAP and data mining can be viewed as disjoint:-

→ (i) OLAP is a data summarization/aggregation tool that helps simplify data analysis.

(ii) While, DM allows the automated discovery of implicit patterns and interesting knowledge hidden in large amounts of data.

→ (i) OLAP tools are targeted toward simplifying and supporting interactive data analysis.

(ii) whereas the goal of DM tools is to automate as much of the process as possible, while still allowing users to guide the process.

In above sense, DM goes one step beyond traditional on-line Analytical processing.

Teacher's Signature : _____

(6) From On-line Analytical processing to On-line Analytical mining :-

① In the field of DM, substantial research has been performed for DM on various platforms, including transaction databases, relational databases, spatial databases, text databases, time-series databases, flat files, data warehouses, and so on...

② On-line Analytical mining (OLAM) (also called OLAP mining) integrates OLAP with DM & mining knowledge in multidimensional databases.

③ OLAM is particularly important for the following reasons :-

i) High Quality of Data in DWH :-

(ii) Available Information processing infrastructure surrounding DWH.

(iii) OLAP-based exploratory data analysis

(iv) On-line selection of DM functions.

Architecture of On-line Analytical Mining :-

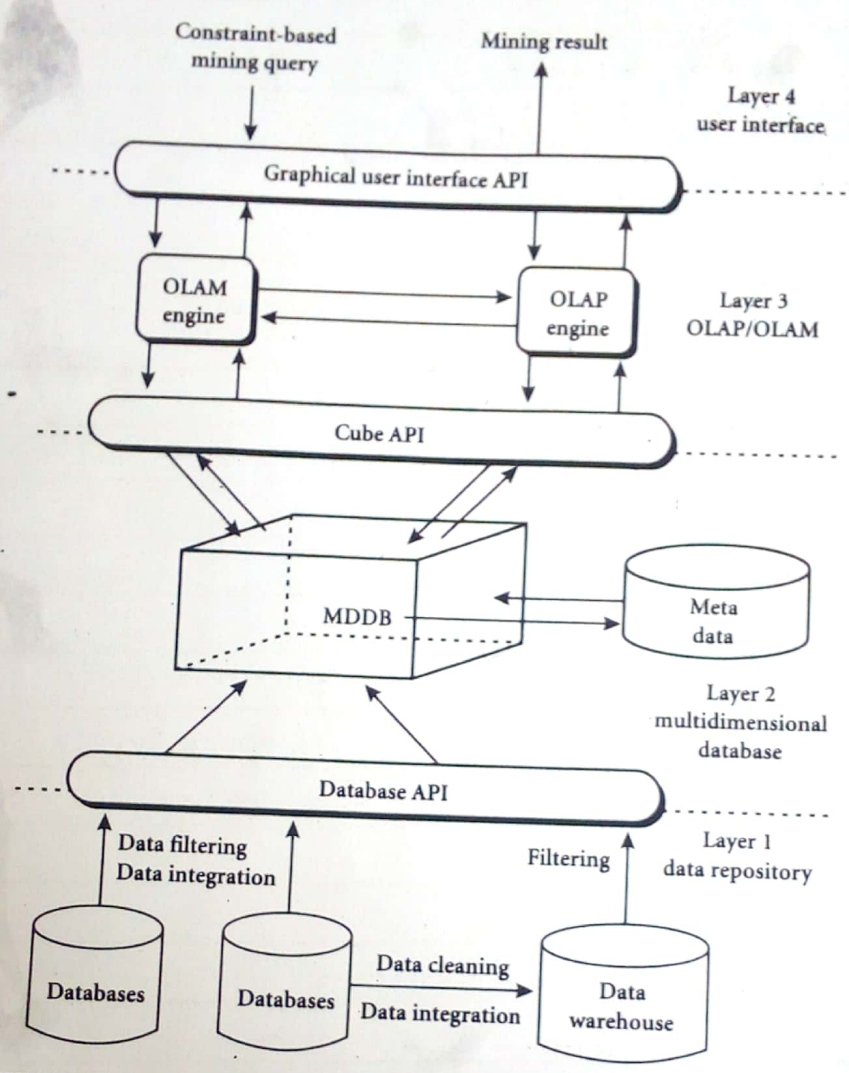


Figure 3.18 An integrated OLAM and OLAP architecture.

① An OLAM server performs analytical mining in data cubes in a similar manner as an OLAP server performs on-line analytical processing.

② An integrated OLAM and OLAP architecture, where the OLAM & OLAP servers both accept user on-line queries via GUI API & work with the data cube in the data analysis via a cube API.

Teacher's Signature : _____

③ A metadata directory is used to guide the access of the data cube.

④ The data cube can be constructed by accessing and/or integrating multiple databases via an MDDB API /(or) by filtering a DWH via a DB API that support ODBC connections.

⑤ Since, an OLAM server may perform multiple DM tasks, such as concept description, association, classification, prediction, clustering, time-series analysis & so..

⑥ It usually consists of multiple integrated DM modules and is more sophisticated than an OLAP server.