Module 1

Data Mining & Warehousing

Nithin Kumar
Asst professor
Dept of CS&E
VVCE, Mysuru

* Data Warehouse :-

Data Warehousing provides Architectures & tools for business Executive to Systematically organize, Understand & use their data to make Strategic decisions.

→ A data warehouse refers to a data repository that is Maintained Separately from an organization operational data-base.
Data Warehouse System allows for Integration of Variety of Application Systems.

→ William H. Inmon States "A data Warehouse is a Subject-oriented, Integrated, time-Variant & non-volatile Collection of data in Support of Management's decision making process".

* Key features of Data Warehouse :- The key features of Data Warehouse are as follows.

→ "Subject-Oriented" : Data warehouse is organized around Major Subjects Such as Customers, Suppliers, product & Sales. Rather than Concentrating on day-to-day operations & Transaction processing of organization.
Data Warehouse focuses on Modeling & Analysis of Data for Decision Making.

→ "Integrated" : A data Warehouse is usually Constructed by Integrating Multiple Heterogeneous Sources Such as Relational DB

Flat files & online Transaction Records.

"Data Cleaning" & "Data Integration" techniques are applied to Ensure Consistency in naming Conventions, Encoding Structure attributes, Measures & so on.

→ "Time - Variant": Data are stored to provide information from an historic perspective. for Ex 5-10 years.
Every key Structure in data Warehouse Contains, Either Implicitly or Explicitly a time Element.

→ "Non - volatile": A data Warehouse is always a physically Separate Store of data transformed from the Application data found in operational Environment.
Due to this separation, A data Warehouse does not req -uire Transaction processing, Recovery & Concurrency Control Mechanisms.

→ The Data Warehouse allows "Knowledge Workers" Such as Managers, Analysts & Executives to use the Warehouse to Quickly & obtain an overview of the data, & make decis -ions based on Info in Warehouse.
The Construction of Data Warehouse Requires "Data cleaning "Data Integration" & "Data Consolidation".

→ Data Warehousing is very useful from the point of View of "Heterogenous Database Integration".
In Traditional Database Approach to heterogenous Database Integration is to build "Wrappers & Integrators" on top of Multiple Heterogenous Databases.

→ When a Query is posed to a Client Site, a Metadata dictionary is used to Translate the Query into queries Appropriate for the individual heterogenous Sites Involved.

* "Query-driven Approach" requires complex information filter-ing & Integration process & competes with local sites for processing resources.
"Traditional Database" Makes use of Query-driven Approach.

* "Update-driven Approach" in which information from Multiple heterogenous sources is integrated in advance & Stored in a warehouse for direct Querying & Analysis.
"Data Warehousing" makes use of Update-Driven Approach.

* Differences between Operational Database Systems & Data Warehouses
The Major task of online operational database systems is to perform online Transaction & Query processing. These Systems are called "Online Transaction processing (OLTP)".
For Example, Most of day-to-day operations of an organization Such as "purchasing", "Manufacturing", "Banking" Etc.

→ "Online Analytical processing (OLAP)" Servers used on knowledge Workers in the role of data Analysis & Decision Making. Such Systems can organize & present the Data in various formats in order to accomodate the diverse needs of different users.

The Major distinguishing features of OLTP & OLAP are

* Users & System Orientation :- An OLTP System is "Customer-oriented" and is used by clerks, clients, & IT professionals.

→ An OLAP System is "Market-Oriented" and is used by knowledge Workers Such as Managers, Executives & Analysts.

* Data Contents :- OLTP System Manages Current data that, typically, are too detailed to be used.

→ OLAP System Manage large Amount historic data, provides facilities for Summarization & Aggregation & Stores & Manage Information at different levels of granularity.

* Database Design :- OLTP usually adopts an Entity - Relationship (ER) data model & Appln oriented database design.

→ OLAP System adopts either "Star" or "Snowflake" data Models.

* View :- An OLTP System Focuses Mainly on Current data within an Enterprise or department, without referring to historic data or data in different organization.

→ OLAP systems Span Multiple Version of database Schema, due to Evolutionary process of an organization.
OLAP sys deal with info that originates from different organization, Integrating info from different Sources.

* Access :- OLTP System consist Mainly of Short atomic transaction so Concurrency Control & Recovery is required.

→ OLAP Systems are Mostly read-only operations for any complex Queries.

Feature	OLTP	OLAP
* Characteristic	* operational processing	* Informational processing
* Orientation	* Transaction	* Analysis
* Users	* DBA, clerk, Database professional	* knowledge workers
* Function	* Day-to-Day operation	* long-term for Decision Support
* DB-Design	* ER-Model, Appln oriented	* Star/Snowflake, Subject oriented
* View	* Detailed, Flat Relational	* Summarized & Multi-dimensional
* Access	* Read/Write	* Read
* Focus	* Data-in	* Information-out
* Unit of Work	* Short, Simple Transaction	* Complex Query

* No of Records Accessed	* Tens	* Millions
* No of Users	* Thousands	* Hundreds
* DB Size	* GB to high-order GB	* ≥ TB
* Priority	* High performance & Availability	* High flexibility, End-user Autonomy
* Metric	* Transaction Throughput	* Query throughput & Response time

* Data Warehousing : A Multitiered Architecture :-

Data warehouses often adopts a 3-tier Architecture such as

* Bottom tier
* Middle tier
* Top tier

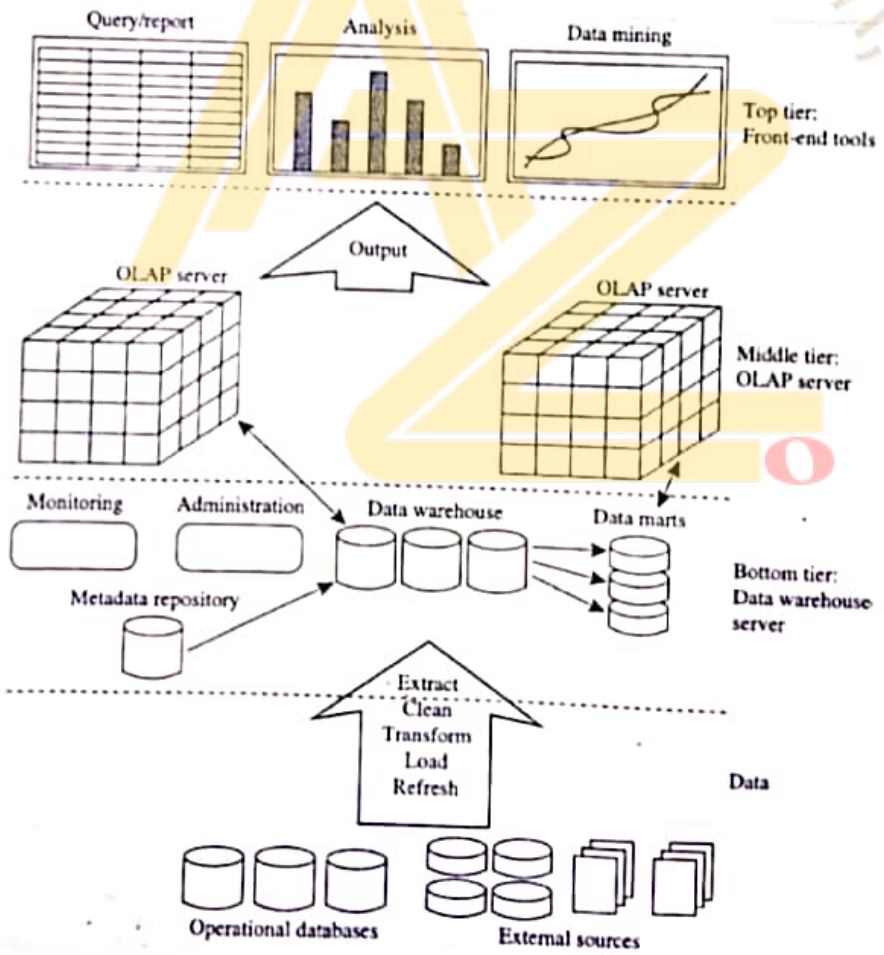


Fig:- Three-tier Data warehouse Architecture

* Bottom tier :- Warehouse Database Server is almost a Relational - database & other External Sources.
Back-End tools & utilities are used to perform Extraction, Cleaning & Transformation (Ex:- to Merge Similar data from different Sources into a unified format).
→ The data are Extracted using Appln program Interface known as "gateways"
Example of gateways include ODBC (Open Database Connection) & OLEDB (Object Linking & Embedding Database) by Microsoft & JDBC.
→ This tier also contains a Metadata Repository, which Stores Information about the Data Warehouse & Contents.

* Middle tier :- Consists of OLAP Servers that is typically Implemented using
→ "Relational OLAP (ROLAP)" : Extended Relational DBMS that Maps Multidimensional data to Standard Relational - operations.
→ "Multidimensional OLAP (MOLAP)" : Direct Implementation of Multidimensional data & operation.

* Top tier :- Consists of Front-End client layer, which contains Query & reporting tools, Analysis tools & Data mining tools. Such as "Trend Analysis", "prediction" & so on.

* Data Warehouse Models :-
There are three Data Warehouse Models
* Enterprise Warehouse
* Data Mart
* Virtual Warehouse

* Enterprise Warehouse :- Collects all the Information about Subjects Spanning Entire organization.

It provide Corporate-wide data with Corporate-wide data -Integration.

→ Enterprise Warehouse Contains detailed data as well as Summarized data & can range in size from of few Gigabytes to hundred -ad or B, Terabyte or beyond.

→ An Enterprise data Warehouse may be Implemented on tradit -ional Mainframe, Computer SuperServers or parallel Architect -ure platforms.
It Requires Extensive Business Modeling & May take years to Design.

* Data Mart :- Contains a Subset of Corporate-wide data that is of Value to Specific Group of users.
For Ex: A Marketing Data Mart May confine its Subject to Customer, Item & Sales.

→ Data Marts are Usually Implemented on "low-cost Departmental Servers" that are Unix/Linux or Window based.
The Implementation Cycle of a Data mart is More likely to be Measured in Weeks rather than Months or Years.

→ Data Marts Can be Categorized as

* Independent → Sourced from operational Systems or External Info providers.

* Dependent → Sourced Directly from Enterprise Data -Warehouse

* Virtual Warehouse :- is a set of Views over operational database for Efficient query processing.
A Virtual Warehouse is Easy to build but requires Excess Capac -ity on operational database Servers.

→ For Ex: Airline ticketing System, collects & displays business data relating to a Specific Moment in time, Creating a Snapshot

of the condition of business at that Moment.

* Recommended Approach for Warehouse development :-

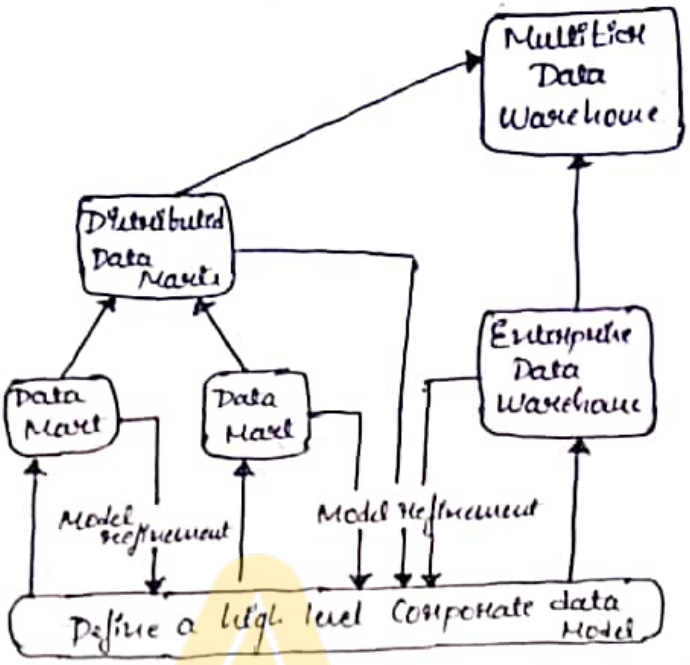


Fig:- A Recommended Approach for Warehouse Development

A Recommended Method for the development of data Warehouse Systems is to Implement the Warehouse in an Incremental & Evolutionary Manner as shown in above figure.

→ Firstly, a high-level corporate data Model is defined within a short period of time, that provides a corporate-wide view of data among different Subject & potential usages.

→ Second, Independent data Marts can be Implemented in parallel with the Enterprise Warehouse based on the Corporate data Model.

→ Third, Distributed data Marts can be Constructed to Integrate different data Marts Via hub Servers.

Finally a "Multiple Data Warehouse" is Constructed, where the Enterprise Warehouse is the Sole Custodian of all Warehouse data, which is then distributed to various dependent Data Marts.

* Data Cube : A Multidimensional Data Model

A Data Cube allows data to be Modeled & Viewed in Multiple Dimensions. It is defined by "Facts" & "Dimensions".

* Dimensions → are the perspectives or Entities with respect to which an organization wants to keep Records.
For Ex:- "All Electronics" may Create a "Sales" data Warehouse in order to keep records of Store Sales with respect to Dimensions "time", "item", "branch" & "location".

* Facts → are numeric Measures. Quantities by which we want to analyze relationship between Dimensions.
For Ex:- Facts for a Sales data Warehouse include "dollars_sold" "Units_Sold", & "Amount_budgeted".

→ Consider a 2D Simple data Cube Example from "All Electronics" Sales data for items Sold per Quarter in City Vancouver.

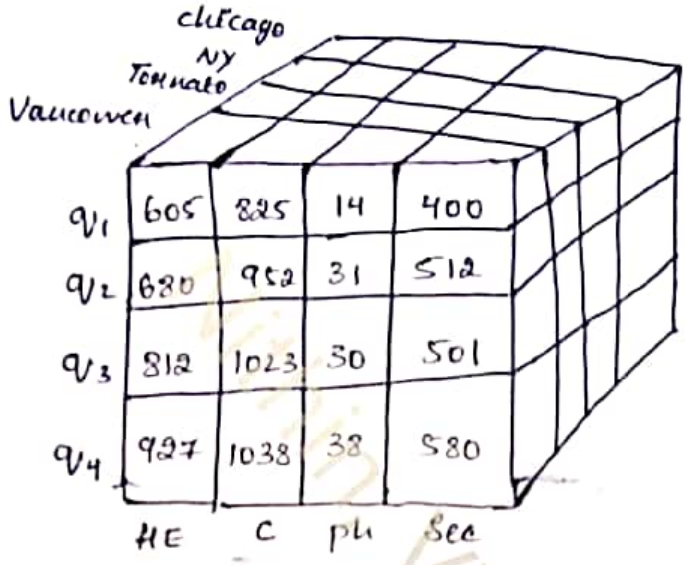
Location = "Vancouver"				
	Item (Type)			
Time (Quarters)	Home Ent	Computer	Phone	Security
q_1	605	825	14	400
q_2	680	952	31	512
q_3	812	1023	30	501
q_4	927	1038	38	580

In this 2-D representation, the Sales for Vancouver are shown w.r.t time dimension & item.

→ Suppose that we would like to view the Sales-data with 3-D representation the data according to time, item & location is as shown below

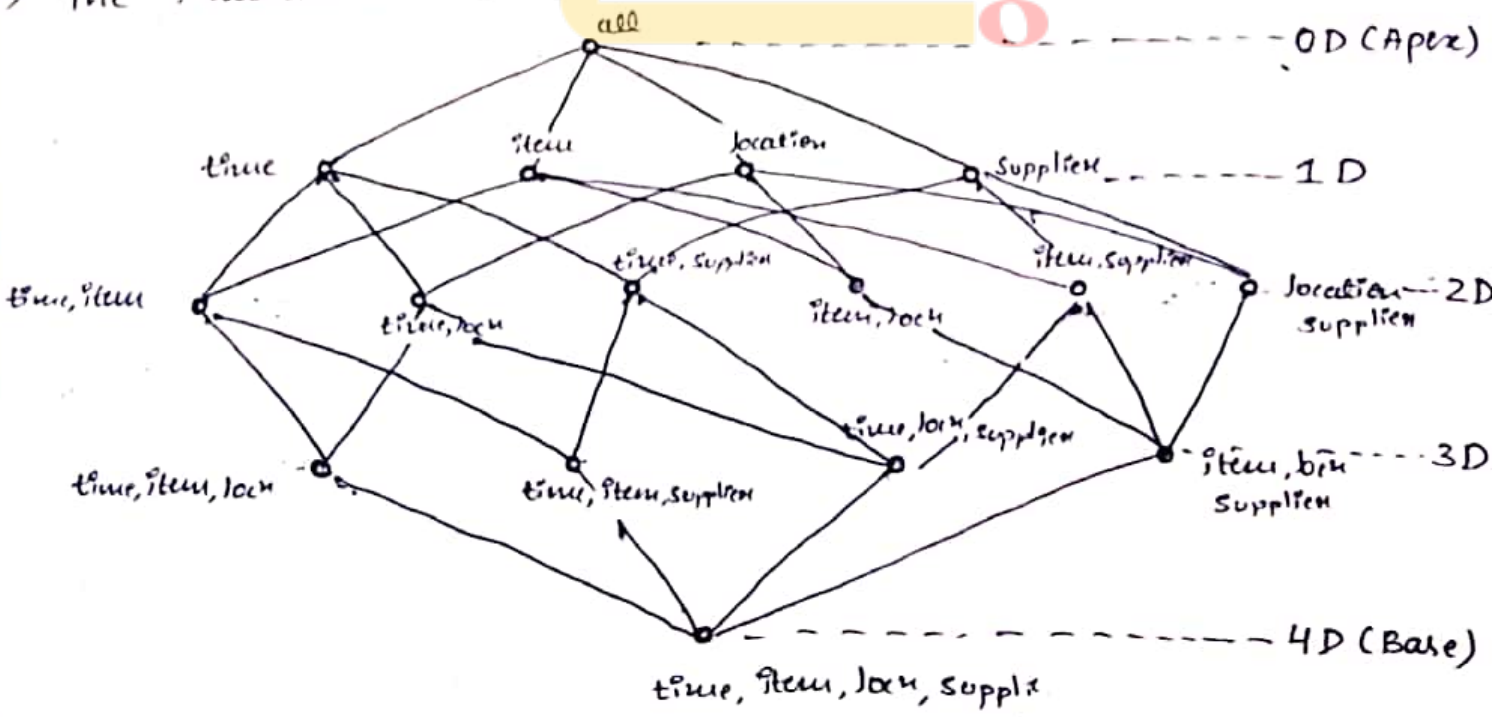
Location = "Chicago"					Location = "New York"				Location = "Toronto"				Location = "Vancouver"			
	item				item				item				item			
time	HE	C	Ph	Sec	HE	C	Ph	Sec	HE	C	Ph	Sec	HE	C	Ph	Sec
q_1	854	882	89	623	1087	720	64	620	1023	760	65	620	1000	620	65	60
q_2	943	890	64	698	1130	841	72	619	1024	820	66	619	1001	619	60	50
q_3	1052	924	59	789	1034	900	63	618	1025	823	67	720	1002	700	55	40
q_4	1189	992	63	876	1142	621	79	617	1026	824	63	723	1003	710	50	30

→ We may also represent the same data in the form of 3D data cube.



→ Suppose that we would like to view our sales data with addition of 4th Dimension such that Supplier. Viewing things in 4-D cube becomes complex. However, we can think of a 4-D cube as being a series of 3D-Cubes.

→ The Multidimensional Data Model can be summarized as show



The 0D cuboid, which holds the highest level of summarizat -ion is called "Apex Cuboid"

In our Example, the "total_sales" or "dollars_sold", Summariz -ed over all four dimensions.

* Star, Snowflake & Fact Constellations : Schema for Multidimen -sional Data Model

The Entity - Relationship data Model is Commonly used to design the Relational database, where database Schema Consists of a Set of Entities & the Relation -ships between them. Such data Model is Appropriate for online Transaction processing.

→ The Most popular data Model for data Warehouse is a Multidimensional Model which can Exist in the form of

* Star Schema
* Snowflake Schema
* Fact Constellation Schema

* Star Schema :- It is the Most Commonly used Multidimens? -ional data Model, in which the data Warehouse Contains

* A large Central table (Fact table) containing the bulk data with no Redundancy.
* A Set of Smaller Attendant - tables (Dimension tables) one for Each dimension.

The Schema Graph resembles a Star burst, with the Dimension tables displayed in Radial pattern around the Central Fact table.

→ A Star Schema for "all Electronics" is shown in the foll figure.

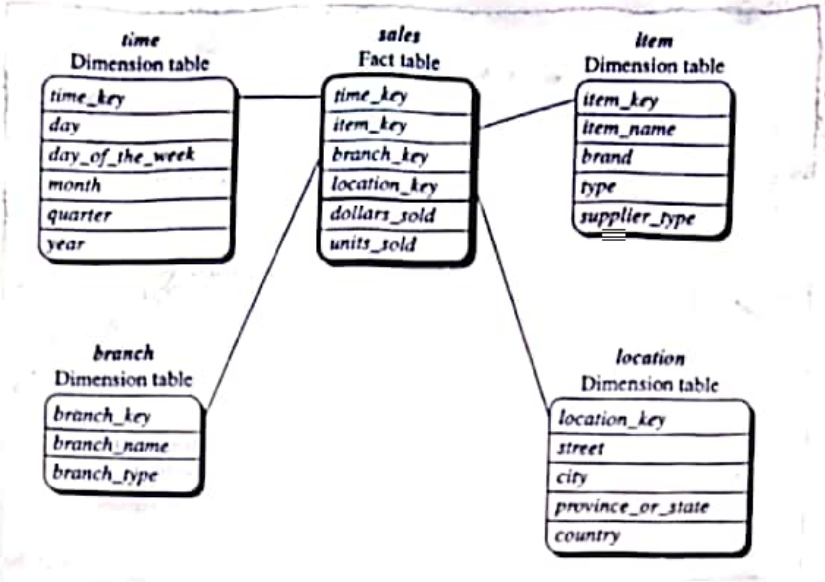


Fig:- Star Schema of Sales Data Warehouse

In the above Schema, Sales is considered along four Dimensions "Time, Item, branch & location". The Schema contains a central Fact table for Sales that contains keys to each 4 Dimensions with two Measures "dollars_sold" & "units_sold".

To Minimize the size of fact table, dimension_Identifiers (time_key item_key Etc) are System generated Identifiers.

→ In Star Schema, Each dimension is represented by only one table & Each table contains a set of attributes.

For Ex:- the "Location" Dimension table contains the attribute set [location_key, Street, City, State, Country]. This constraint may introduce some redundancy.

* Snowflake Schema :- Is a variant of Star-Schema Model, where some dimension tables are Normalized, thereby Splitting the data into additional tables.

→ The Major difference between the Star & Snowflake Schema Model is that the Dimension tables of the Snowflake Model may be kept in Normalized form to reduce redundancies. Such a table is Easy to Maintain & Saves Storage Space.

→ The Snowflake Structure can reduce the Effectiveness of

Browsing, Since More Joins will be needed to Execute a Query. The System performance May be adversely Impacted. Hence, alth-ough the Snowflake Schema Reduces Redundancy, It is not as popular as Star Schema in Data Warehouse Design.

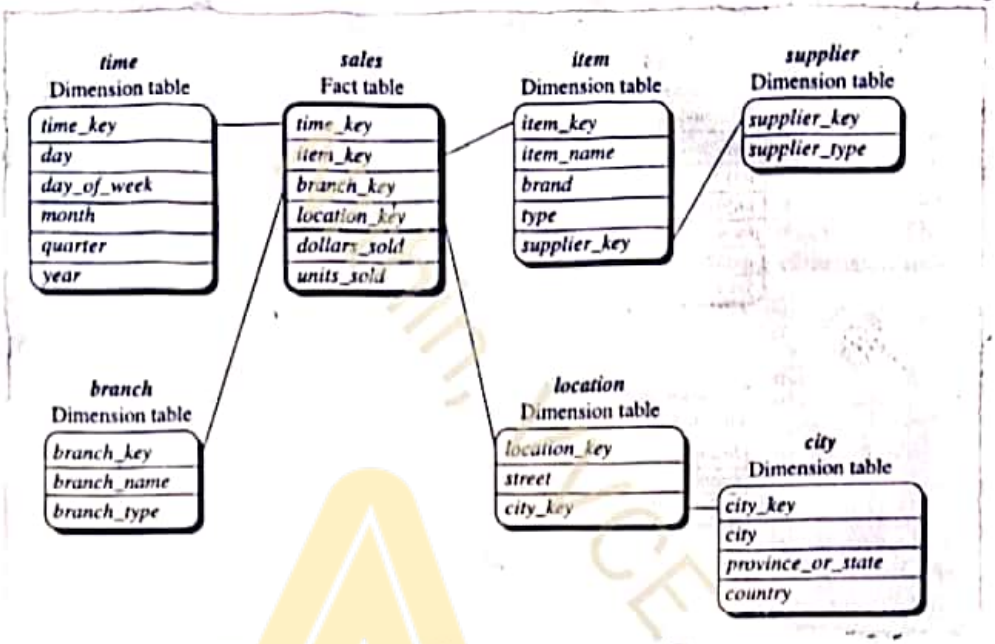


Fig:- Snowflake Schema of Sales

As Shown in above Snowflake Schema for "All Electronics", the Sales fact table is identical to the Star Schema.

→ The Main difference between Star & Snowflake Schema is Definition of Dimension table.

* Dimension table of "Item" in Star Schema is Normalized in Snowflake Schema, resulting table forms "Item" and "Supplier" Dimension tables.
* Similarly Dimension table of "Location" in Star Schema is Normalized in Snowflake Schema, resulting table forms "Location" & "City" Dimension tables.

* Fact Constellation :-

For Each Star Schema or Snowflake Schema, it is possible to Construct "Fact Constellation Schema".

→ The fact Constellation Architecture Contains Multiple Fact tables that Share Many Dimensions.

→ Consider the foll Example of Fact Constellation

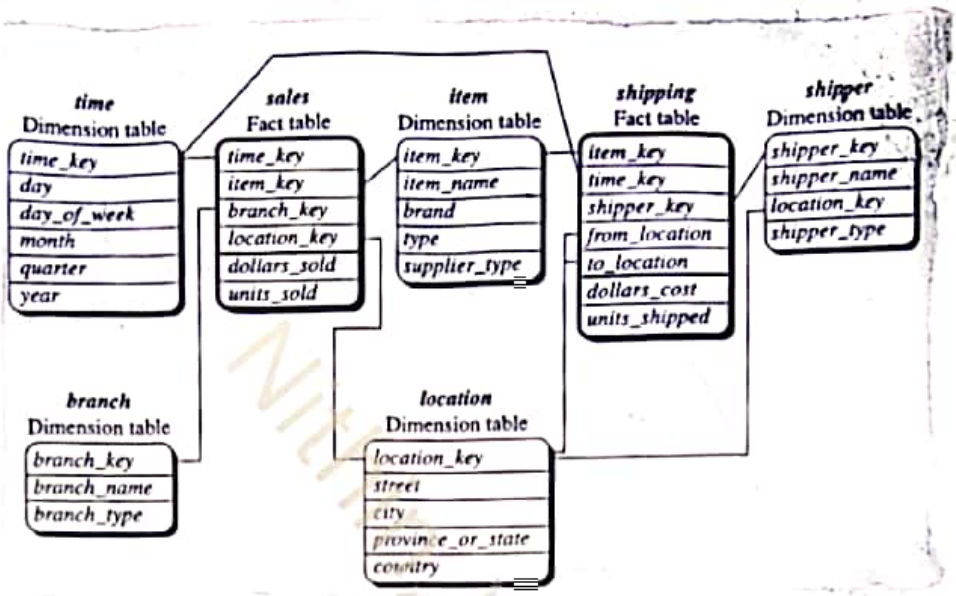


Fig: Fact Constellation Schema of Sales

The above Schema Specifies two Fact tables "Sales" & "Shipping"

→ The Shipping table Consists of 5 Dimensions or keys & Two Measures

* Dimensions → Item_key, Time_key, Shipper_key, to_location From_Location

* Measures → Dollars_Cost, Units_Shipped.

A Fact Constellation Schema allows Dimension tables to be Shared between Fact tables.

→ A "Data Warehouse" Collects Information about Subject, that Spa Whole organization Such as Customers, Sales, Shipping Etc.

For data Warehouse, the fact constellation Schema is Com -only used to Model Multiple Interrelated Subjects.

→ A "Data Mart" is a Subset of Data Warehouse that Focu on Selected Subjects.

For Data Mart, the Star or Snowflake Schema is Commo Used Since both are used to Model Single Subjects.

* Role of Concept Hierarchies :-

A "concept hierarchy" defines a sequence of mappings from a set of low-level concepts to high-level more general concepts.

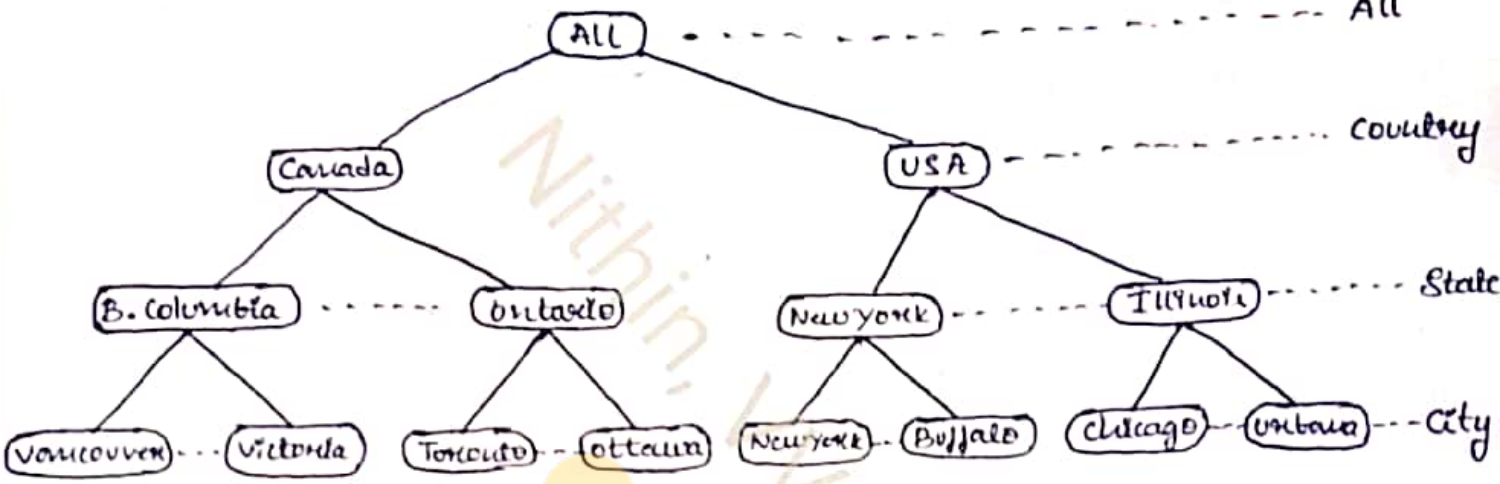


fig:- Concept Hierarchy of location

Mapping a set of low-level concepts (cities) to high-level more general concepts (countries).

→ Many concept hierarchies are implicit within the database schema

→ Due to space limitation, not all of the hierarchy nodes are shown we can use lattice structures of attributes in warehouse dimensions.

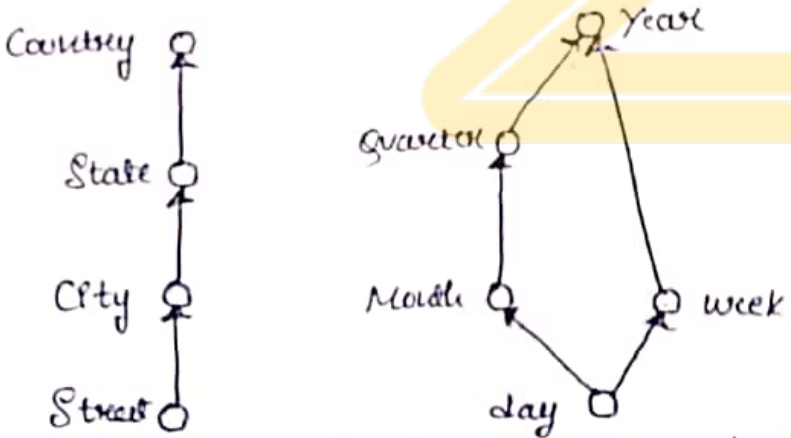


fig:- lattice structures of location & time

→ Concept hierarchies may also be defined by discretizing or grouping values for a given dimension or attribute, resulting in a "Set-Grouping Hierarchy".

For the dimension "price" with interval ($x ... $y) denotes the range from $x (exclusive) to $y (inclusive)

→ There May be More than one Concept Hierarchy for a given attribute or Dimension based on different user View-points.

Concept Hierarchies May be provided Manually by System-Users, Domain Experts or Knowledge Engineers or May be Auto-matically generated based on Statistical Analysis of the Data Distribution.

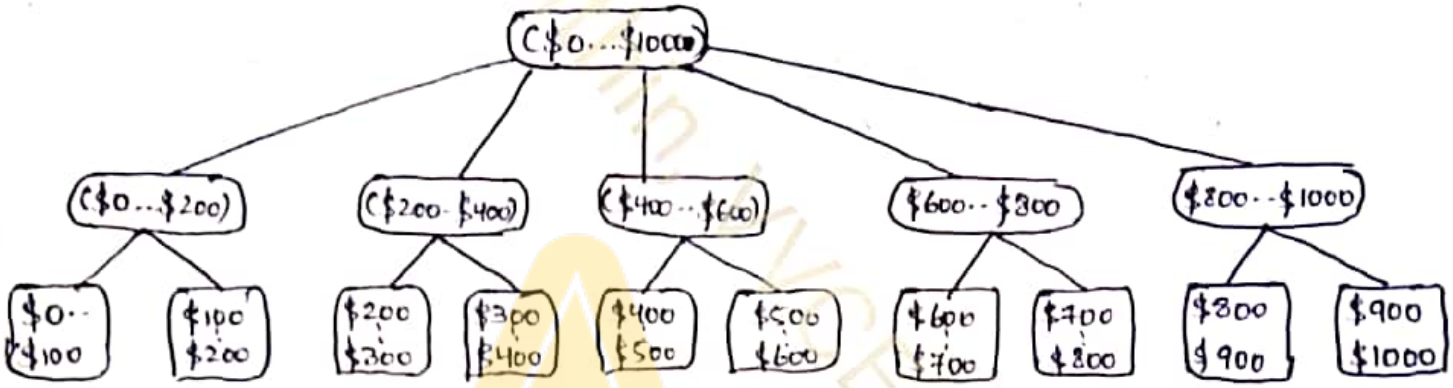


fig:- Set Grouping Hierarchy of price

* Measures: Their Categorization & Computation :-

Measures can be organized into three categories such as

* Distributive
* Algebraic
* Holistic

Based on the kind of Aggregate Functions used.

* Distributive :- An Aggregate Function is distributed, if it can be computed in a Distributed Manner.
Suppose the data are partitioned into 'n' sets, we apply the Function which results in n Aggregate Values.
→ If the result derived by Applying the Function to the 'n' Aggregate values is the same as that derived by applying the Function to entire Data Set in Distributed Manner.
→ Some of good Examples of Distributive Aggregate Functions are

"count()", "Min()", & "Max()".

* Algebric :- An Aggregate Function is Algebric, if it can be computed by an Algebric function with M Arguments, each of which is obtained by applying a distributive Aggregate Functions.

→ A Measure is "Algebric" if it is obtained by Applying an Algebric Aggregate Function.
For Ex:- "Avg()" can be computed by Sum()/count(), where both Sum() & count() are distributive Aggregate Functions.

* Holistic :- An Aggregate Function is Holistic, if there is no constant bound.
There does not Exist any Algebric function that characterizes the computation. Such Measures are called "Holistic".

→ Some of good Examples of Holistic Functions include "Median()" "Mode()" & "Rank()".

More large data cube Application requires Efficient computation of distributive & Algebric Measures. Many efficient techniques for this Exist.
In contrast, it is difficult to compute Holistic Measures Efficiently.

* Typical OLAP operations :-

Number of operations May be applied to Data-Cubes are

* Roll-up
* Drill-Down
* Slice & Dice
* Pivot or Rotate

Consider the foll Example of "ALL Electronics" Sales Data cube which

Contains the Dimensions [location, Time & Item].

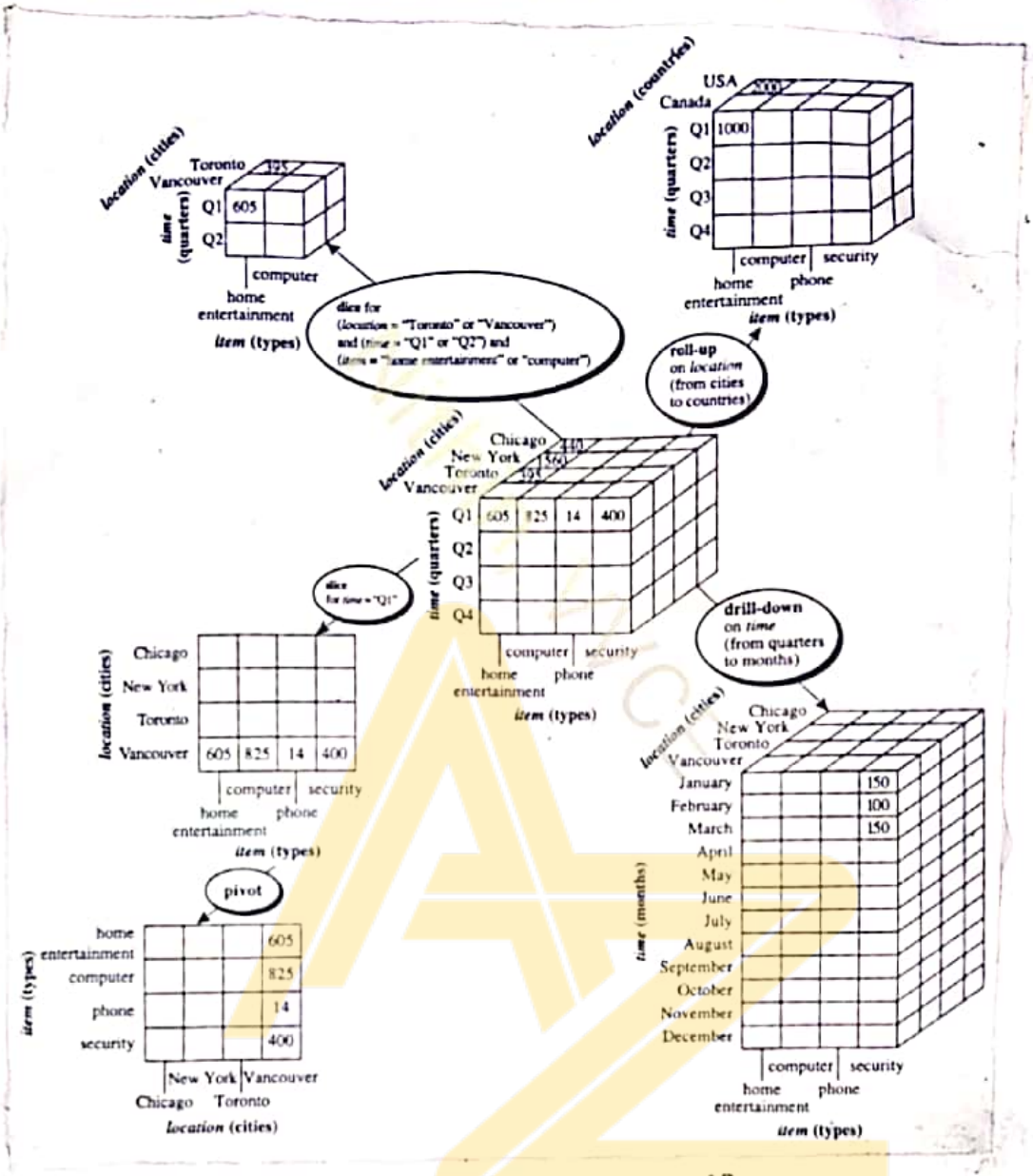


Fig: Typical OLAP operations

* Roll-up :- Is like "Zooming-out" on the Data Cube, It is required when the user needs further Abstraction or Less Detail.
→ This operation performs Further Aggregation on the Data.
For Ex:- Roll-up on location from "cities" to "Countries"

* Drill-Down :- Is like "Zooming-In" on the Data Cube, i.e reverse of roll-up.
→ It is an Appropriate operation when the User needs further detail or when user wants to partition More Finely on

wants to focus on some particular values of certain Dimension

→ For Ex, if the time in data cube is represented in quarter then it can be drilled down or zoomed-in the time to Month wise.

* Slice & Dice :- Slice & Dice are operations for Browsing the data in the Cube. the terms refers to the ability to look at information from different viewpoints.
  * A "Slice" is a Subset of Cube corresponding to a Single value for one or More Members of Dimension.
  * A "Dice" operation is Similar to Slice but dicing does not involve reducing the no of dimension. A dice is obtained by performing a Selection on two or More Dimensions.

* Pivot or Rotate :- Pivot operation is used when the user wishes to re-orient the view of Data Cube. It May involve Swapping the rows & Columns. or Moving one of row dimension into Column dimension.