

Module 1

Data Mining & Warehousing

Nithin Kumar
Asst professor
Dept of CSE
VUCE, Myosum

* Data Warehouse :-

Data Warehousing provides Architecture & tools for business Executive to Systematically organize, understand & use these 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 Systems allow for Integration of Variety of Application Systems.

→ William H. Inmon State "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 Customer, Supplier, product & Sales. Rather than concentrating on day-to-day operations & Transaction processing of organization.

Data Warehouse focus 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 structures, 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 require transaction processing, recovery & concurrency control mechanism.

→ The Data Warehouse allows "knowledge workers" such as managers, analysts & executives to use the warehouse to quickly & obtain an overview of the data, & make decisions 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 "Heterogeneous Database Integration".

In traditional database approach to heterogeneous database integration is to build "Wrappers & Integrators" on top of multiple heterogeneous databases.

→ When a query is posed to a client site, a metadata dictionary is used to translate the query into query appropriate for the individual heterogeneous sites involved.

* "Query - driven Approach" Requires complex Information filtering & 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 heterogeneous 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 & Data Warehouse

The major task of online operational database 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)" serves users or knowledge workers in the role of data analysis & decision making.

Such systems can organize & present the data in various forms in order to accommodate the diverse needs of different users.

The major distinguishing features of OLTP & OLAP are

* User & 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 Content :- OLTP system manages current data that, typically, are too detailed to be used.

→ OLAP System Manage Large Amount historic data , provide facilities for Summarization & Aggregation & Store & 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 Model.

* 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 Sims deal with Info that originate from different organization , Integrating Info from different sources.

* Access- OLTP System consist Mainly of Short atomic trauma - So Concurrency Control & Recovery is Required.

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

<u>Feature</u>	<u>OLTP</u>	<u>OLAP</u>
* <u>Characteristics</u>	* operational processing	* Informational processing
* <u>Orientation</u>	* Transaction	* Analysis
* <u>User</u>	* DBA, Clerk, Database professionals	* Knowledge Workers
* <u>Function</u>	* Day-to-Day operation	* Long-term for Decision Support
* <u>DB-Design</u>	* ER-Model, Appln oriented	* Star/SnowFlake, Subject oriented
* <u>View</u>	* Detailed, Flat Relational	* Summarized & Multi-dimensional
* <u>Access</u>	* Read / Write	* Read
* <u>Focus</u>	* Data-in	* Information-out
* <u>Unit of Work</u>	* Short, Simple Transactions	* Complex Query

* Operations	* Index / Hash	* Lots of Scan
* No of Records Accessed	* Tens	* Millions
* No of Users	* Thousands	* Hundreds
* DB Size	* 1GB 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 warehouse often

adopts a 3-tier Architecture Such as

- * Bottom tier
- * Middle tier
- * Top tier

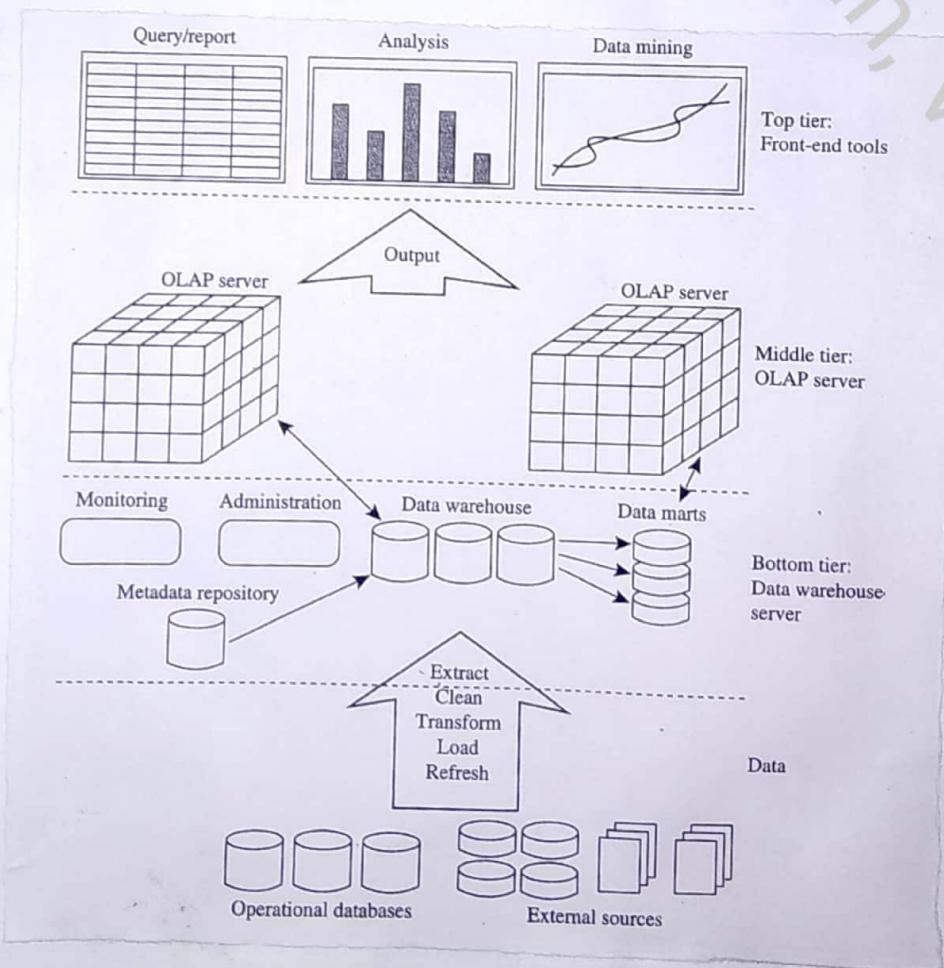


Fig:- Three - Tier Datawarehouse Architecture

* 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 Application program Interface known as "gateway"

Example of gateway 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 & Content.

* Middle tier :- Consists of OLAP Server that is typically implemented using

→ "Relational OLAP (ROLAP)": Extended Relational DBMS that maps Multidimensional data to Standard Relational-operation.

→ "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 Model :-

These are three Data Warehouse Model

* Enterprise Warehouse

* Data Mart

* Virtual Warehouse

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

It provides 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 Petabytes, Terabytes or beyond.
- An Enterprise data Warehouse may be implemented on traditional Mainframes, Computer SuperServers or parallel Architectural 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 contain subjects to Customer, Item & Sales.

- Data Marts are usually implemented on "low-cost Department Servers" that are Unix/Linux or Windows 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 Warehouses - Is a set of views over operational database for efficient query processing.

A Virtual Warehouse is easy to build but requires Excel capacity on operational database servers.

- For ex:- Air 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 :-

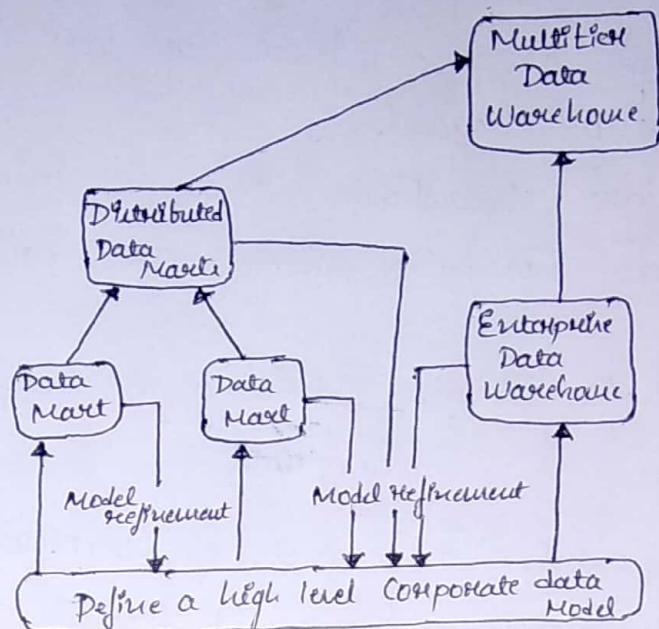


Fig: A Recommended Approach for Warehouse Development

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 subjects & potential users.
- Second, Independent data Mart can be implemented in parallel with the Enterprise warehouse based on the corporate data Model.
- Third, Distributed data Mart can be constructed to inter-link different data Mart via hub Server.

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 "Fact" & "Dimensions".

* Dimensions → are the perspectives on 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 "dollar-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 (Quarter)	Home Ent	Computer	Phone	Security
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

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

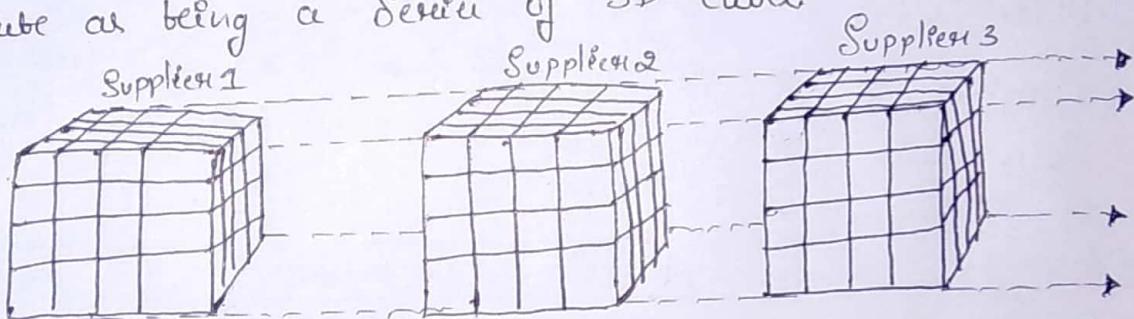
→ Suppose that we would like to view the Sales-data with 3-D Report the data according to time, Item & Location as shown below

Loc = "Chicago"		Loc = "Newyork"		Loc = "Toronto"		Loc = "Vancouver"		
Item	Item	Item	Item	Item	Item	Item	Item	
time	HE	C	Ph	Sec	HE	C	Ph	Sec
Q1	854	882	89	623	1087	720	64	620
Q2	943	890	64	698	1130	841	72	619
Q3	1032	924	59	789	1034	900	63	618
Q4	1129	992	63	876	1142	621	79	617

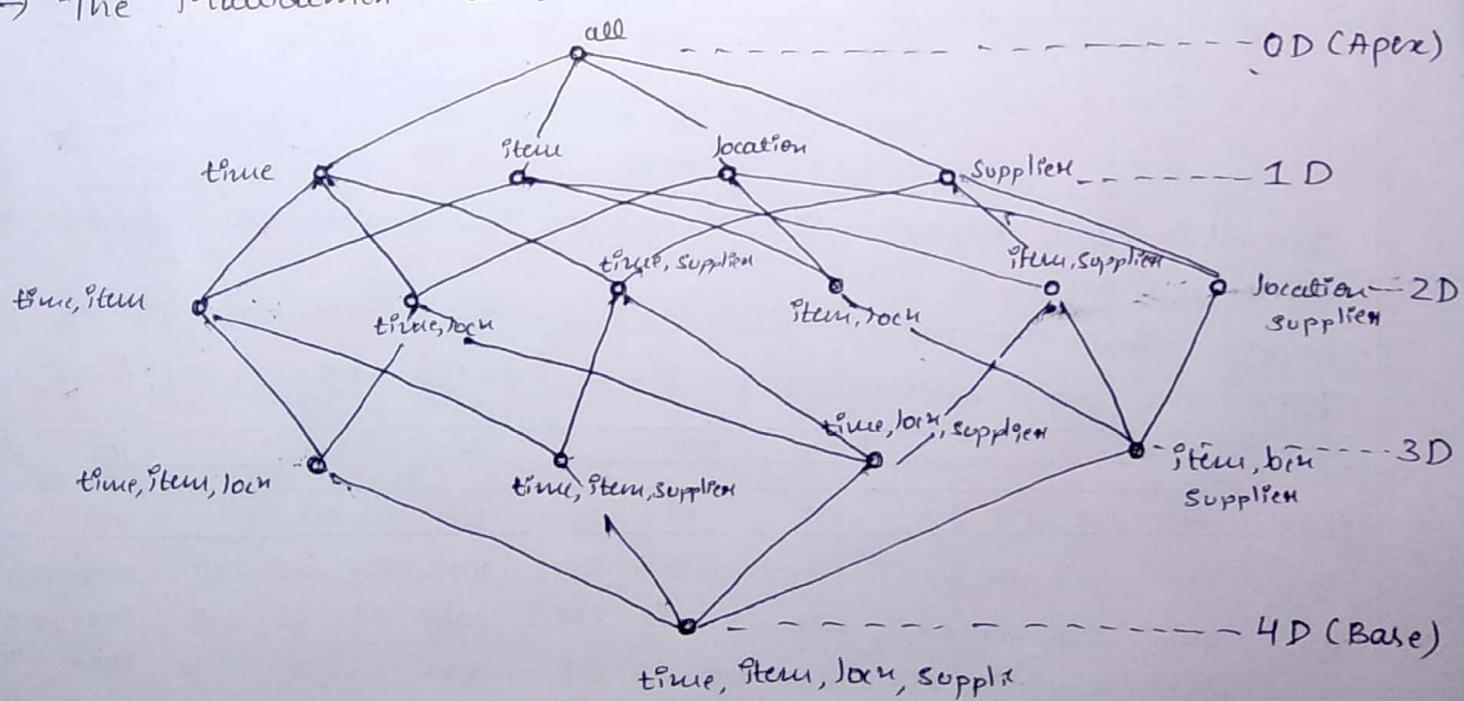
→ We may also represent the same data in the form of 3D data cube.

	chicago	Ny	Toronto	Vancouver
q1	605	825	14	400
q2	680	952	31	512
q3	812	1023	30	501
q4	927	1038	38	580
	AE	C	PH	SEA

→ 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 shown



The OD cuboid, which holds the highest level of summarization is called "Apex Cuboid". In our Example, the "total_sales" or "dollar_sold", summarized over all four dimensions.

* Star, Snowflake & Fact Constellations : Schema for Multidimensional 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 Relationships 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 :- Is the Most Commonly used Multidimensional 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 Consists of a Starburst, with the Dimension tables displayed in radial pattern around the Central Fact table.

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

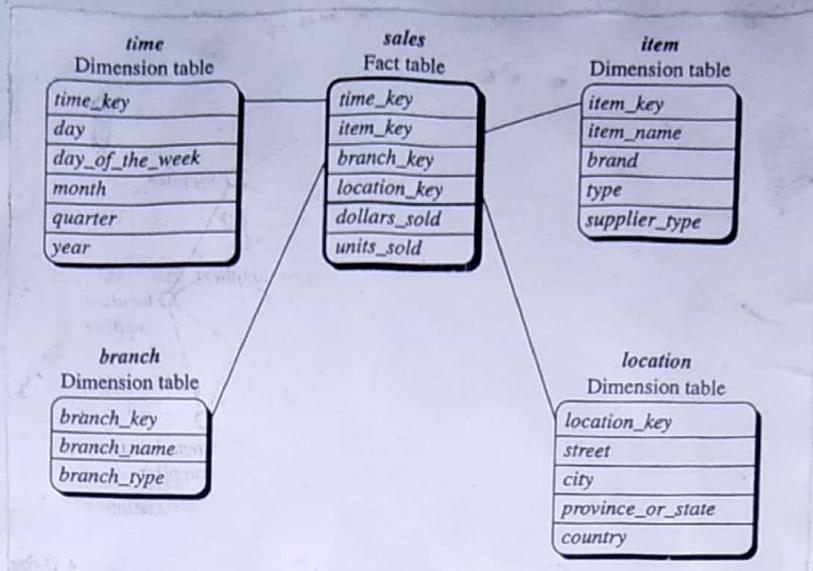


Fig:- Star Schema of Sales Data Warehouse

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-1, 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 Models is that the Dimension tables of the Snowflake Model May be kept in Normalized form to reduce Redundancy. Such a table is Easy to Maintain & Save 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, although the Snowflake Schema Reduces Redundancy, it is not as popular as Star Schema in Data Warehouse Design.

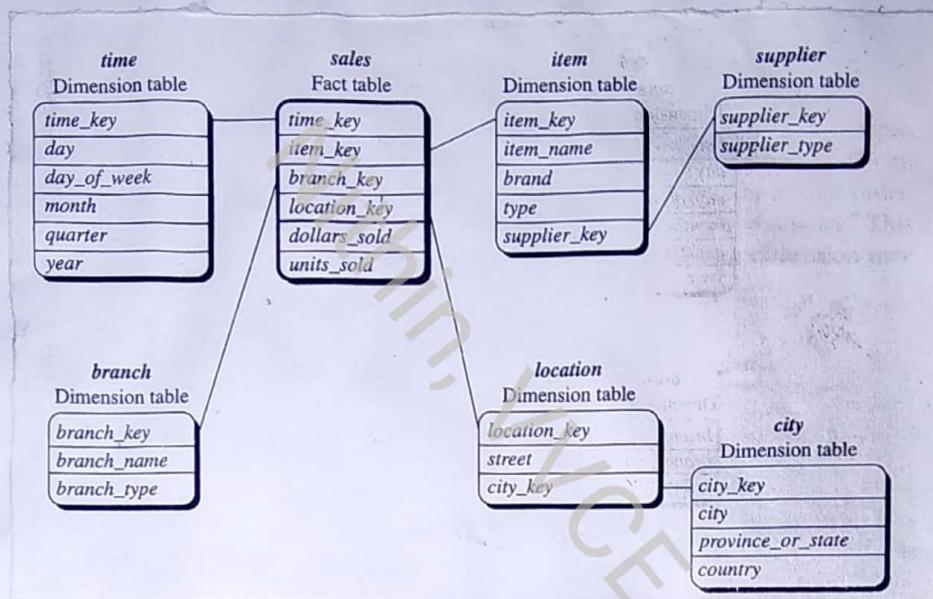


Fig:- Snowflake Schema of Sales

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 form "Item" and "Supplier" Dimension tables.
 - * Similarly Dimension table of "Location" in Star Schema is Normalized in Snowflake Schema, resulting table from "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 full Example of Fact Constellation

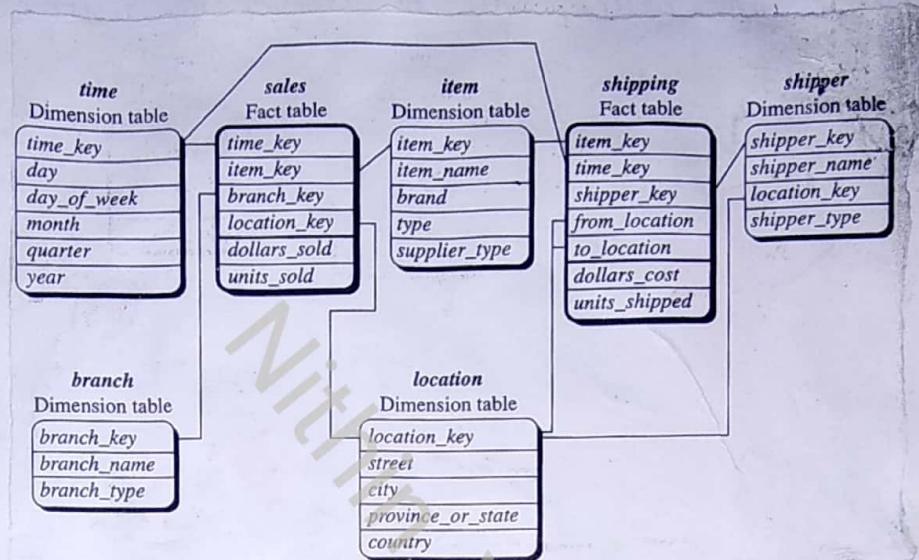


Fig: Fact Constellation Schema of Sales

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 Subjects that Span Whole organization Such as Customer, Sales, Shipping Etc.

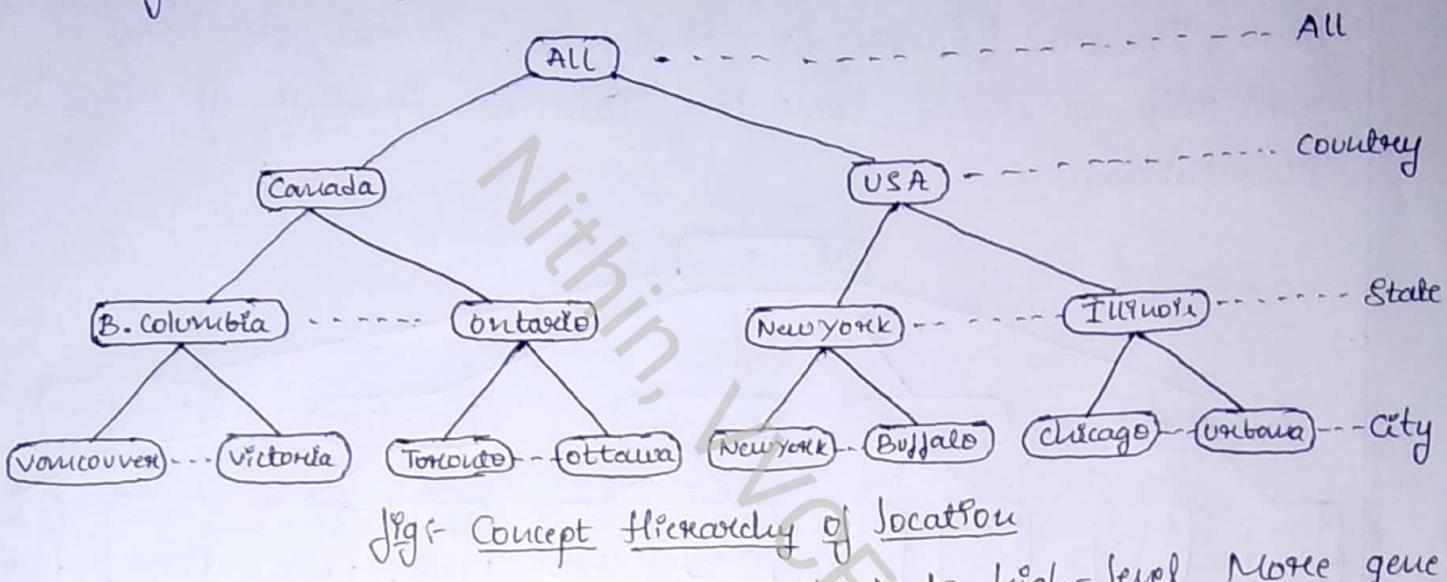
For data warehouses, the fact constellation Schema is commonly used to Model Multiple Interrelated Subjects.

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

For Data Mart, the Star or Snowflake Schema is commonly used since both are used to Model Single Subjects.

* Role of Concept Hierarchy:-

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



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 limitations, not all of the hierarchy nodes are shown we can use lattice structure of attribute in warehouse dimension

- One,

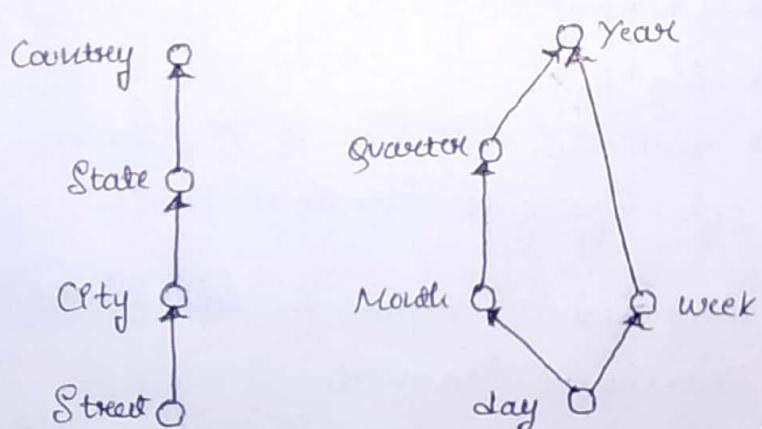


fig:- lattice structure of location & time

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

For the dimension "price" with interval $(\$x \dots \$y)$ 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 viewpoints.

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

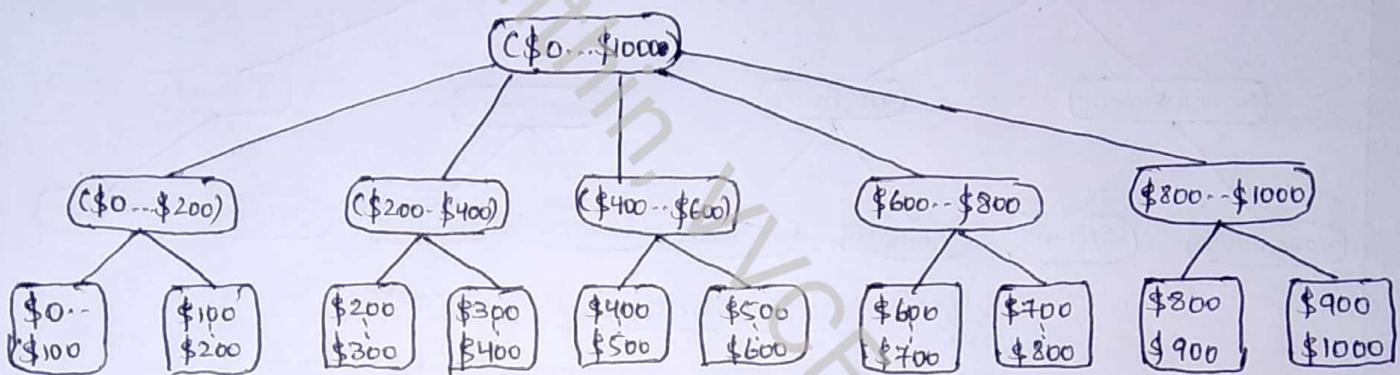


Fig: Set Categorizing hierarchy of price

* 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()", "Sum()", & "Max()".

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

→ A Measure is "Algebraic" if it is obtained by Applying an Algebraic Aggregate Functions.

For ex - "Avg()" can be computed by $\text{Sum}() / \text{Count}()$, where both $\text{Sum}()$ & $\text{Count}()$ are distributive Aggregate Functions.

* Holistic :- An Aggregate Function is Holistic, if there is no constant bound.

There does not exist any Algebraic 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 & Algebraic 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].

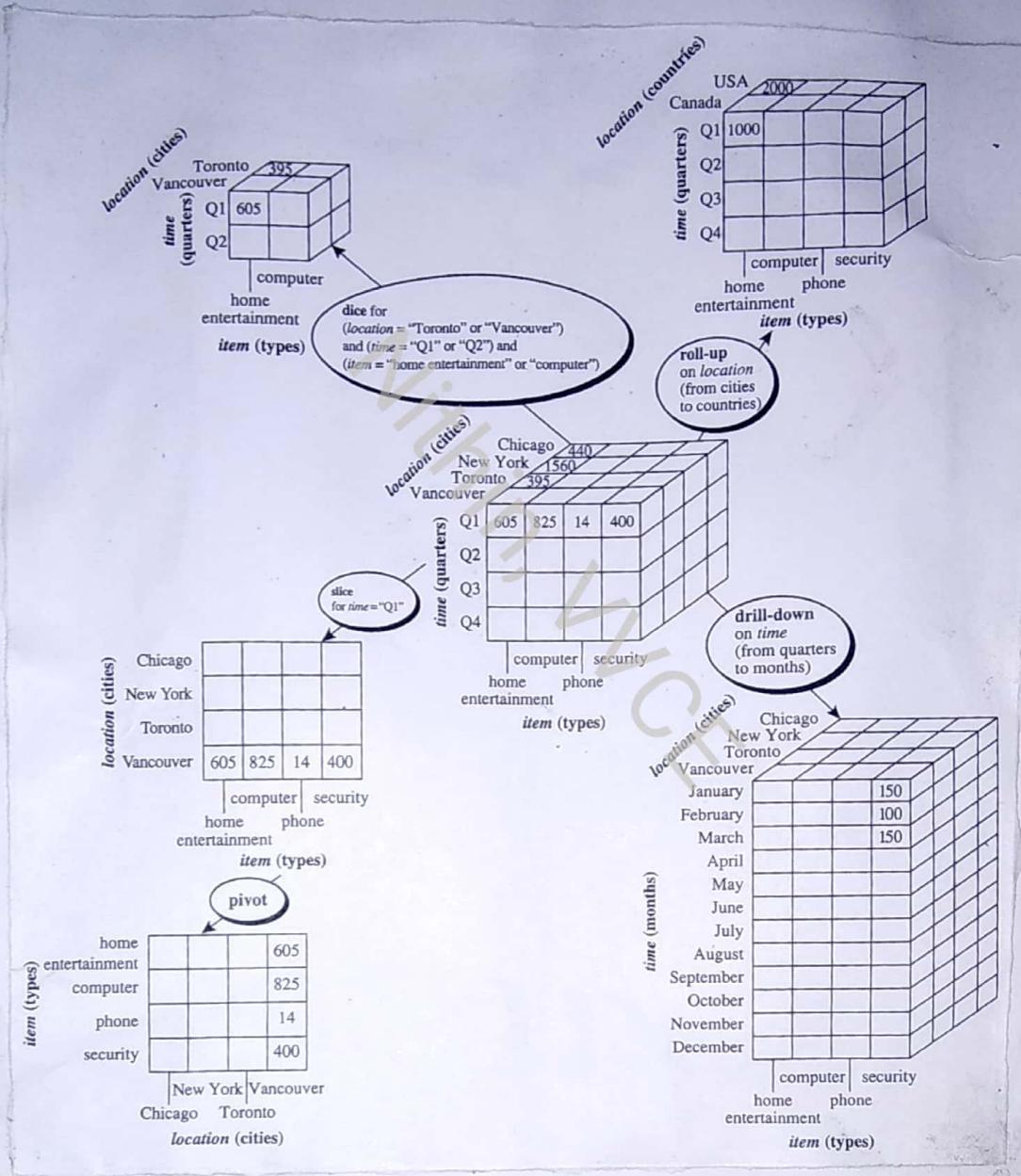


Fig Typical OLAP operations

* 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 dimensions.

→ 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 slicing does not involve reducing the no. of dimensions. 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-align the view of data cube. It may involve swapping the rows & columns, or moving one of row dimension into column dimension.