

MID-2 UNIT-2: Data warehouse and OLAP Technology

- ⇒ Differences b/w operational database systems and data warehouses
- ⇒ A Multidimensional data model
- ⇒ Data warehouse architecture
- ⇒ " " Implementation - Efficient computation of data cubes, From Data warehousing to data mining.

* Data warehouse: Data warehouse refers to a data repository that is maintained separately from an organization's operational databases.

- ⇒ Data Warehousing provides architectures and tools for business executives to systematically organize, understand, and use their data to make strategic decisions.
- ⇒ A data warehouse is a subject-oriented, integrated, time-variant and nonvolatile collection of data in support of management's decision making process

Subject-oriented: A data warehouse is organized around major subjects such as customer, supplier, product and sales. Rather than concentrating on the day-to-day operations and transaction processing of an organization, a dwh focuses on the modeling and analysis of data for decision makers.

Hence, dwhs typically provide a simple and concise view of particular subject issues by excluding data that are not useful in the decision support syst process.

Integrated : A dwt is usually constructed by integrating multiple heterogeneous sources, such as relational databases, flat files and online transaction records. Data cleaning and data integration techniques are applied to ensure consistency in naming conventions, encoding structures, attribute measures and so on.

Time-variant : Data are stored to provide information from an historic perspective (e.g. the past 5-10 years). Every key structure in the dwt contains, either implicitly or explicitly, a time element.

Nonvolatile : A dwt is always a physically separate store of data transformed from the application data found in the operational environment. Due to this separation, a dwt does not require transaction processing, recovery and concurrency control mechanisms. It usually requires only two operations in data accessing: initial loading of data and access of data.

Operational database :-

The operational database is the source of information for the dwt. It includes detailed information used to run the day-to-day operations of the business. The data frequently changes as updates are made and reflect the current value of the last transactions.

How are organizations using the information from dwt?

(1) increasing customer focus, which includes the analysis of customer buying patterns (such as buying preference, buying time, budget cycles, and appetites for spending)

- (2) repositioning products and managing product portfolios by comparing the performance of sales by quarter, by year, and by geographic regions in order to fine-tune production strategies.
- (3) analyzing operations and looking for sources of profit
- (4) managing customer relationships, making environmental corrections and managing the cost of corporate assets.

Online transaction processing (OLTP) systems:

- ⇒ Operational database Management Systems also called as OLTP (online Transactions Processing Databases) are used to manage dynamic data in real-time.
- ⇒ The major task of online operational database systems is to perform online transaction and query processing. These systems are called online transaction processing (OLTP) systems.
- ⇒ They cover most of the day-to-day operations of an organization such as purchasing, inventory, manufacturing, banking, payroll, registration and accounting.

Online Analytical Processing (OLAP) Systems:

Data warehouse systems serve users or knowledge workers in the purpose of data analysis and decision-making. Such systems can organize and present information in specific formats to accommodate the diverse needs of various users. These systems are called as Online Analytical Processing (OLAP) Systems.

⇒ DWH and OLTP database are both relational databases.
(The major distinguishing features of OLTP and OLAP are summarized as follows:)

* Operational database	Data Warehouse
⇒ Operational systems are designed to support high-volume transaction processing.	⇒ Data warehousing systems are typically designed to support high volume analytical processing.
⇒ Operational systems are usually ^(OS) concerned with current data.	⇒ Data warehousing systems are usually concerned with historical data.
⇒ Data within OS are mainly updated regularly according to need.	⇒ Non-volatile, new data may be added regularly. Once added rarely changed.
⇒ Less no. of data accessed	⇒ Large no. of data accessed.
⇒ It is designed for real time business dealing and processes.	⇒ It is designed for analysis of business measures by subject area, categories and attributes.
⇒ It is optimized for a simple set of transactions, generally adding or retrieving a single row at a time per table.	⇒ It is optimized for extent loads and high, complex, unpredictable queries that access many rows per table.
⇒ OS are widely process oriented.	⇒ Data warehousing systems are widely subject-oriented.
⇒ Relational databases are created for on-line transactional processing (OLTP)	⇒ Data warehouse designed for on-line analytical processing (OLAP)

Users and System Orientation:

OLTP: An OLTP system is customer oriented and is used for transaction and query processing by clerks, clients and information technology professionals.

OLAP: An OLAP system is market-oriented and is used for data analysis by knowledge workers, including managers, executives and analysts.

Data Contents:

OLTP: An OLTP system manages current data that, typically, are too detailed to be easily used for decision making.

OLAP: An OLAP system manages large amounts of historic data, provides facilities for summarization and aggregation and stores and manages information at different levels of granularity.

These features make the data easier to use for informed decision making.

Database Design:

OLTP: An OLTP system usually adopts an entity-relationship (ER) data model and an application-oriented database design.

OLAP: An OLAP system typically adopts either a star or a snowflake model and a subject-oriented database design.

View:

OLTP: An OLTP system focuses mainly on the current data within an enterprise or department, without referring to historic data or data in different organizations.

OLAP: An OLAP system often spans multiple versions of a database schema, due to the evolutionary process of an organization.

OLAP systems also deal with information that originates from different organizations, integrating information from many data stores. Because of their huge volume, OLAP data are stored on multiple storage media.

Access Patterns:

OLTP: The access patterns of an OLTP system consist mainly of short atomic transactions. Such a system requires concurrency control and recovery mechanisms.

OLAP: Accesses to OLAP systems are mostly read-only operations (because most dwellers store historic rather than up-to-date information), although they could be complex queries.

* But, why have a separate Data Warehouse? — 166 pg.
Why not perform OLAP directly on operational databases instead of spending additional time and resources to construct a separate DWH?

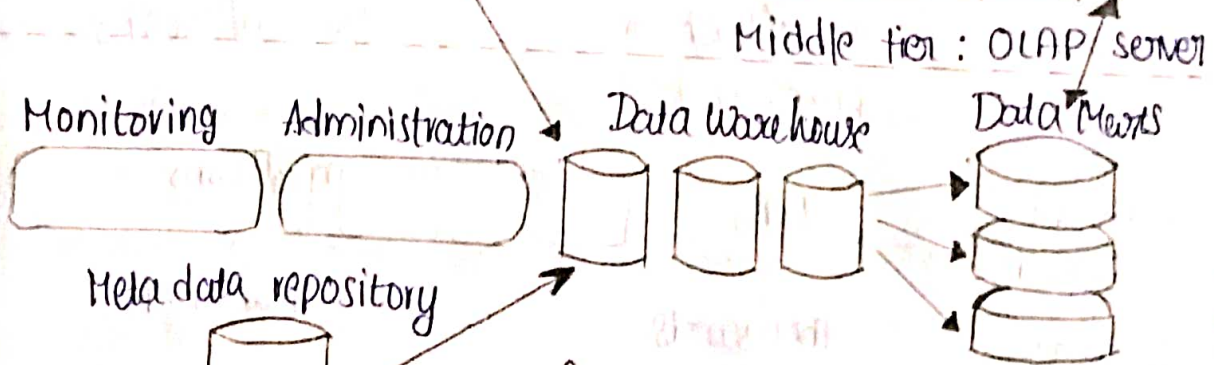
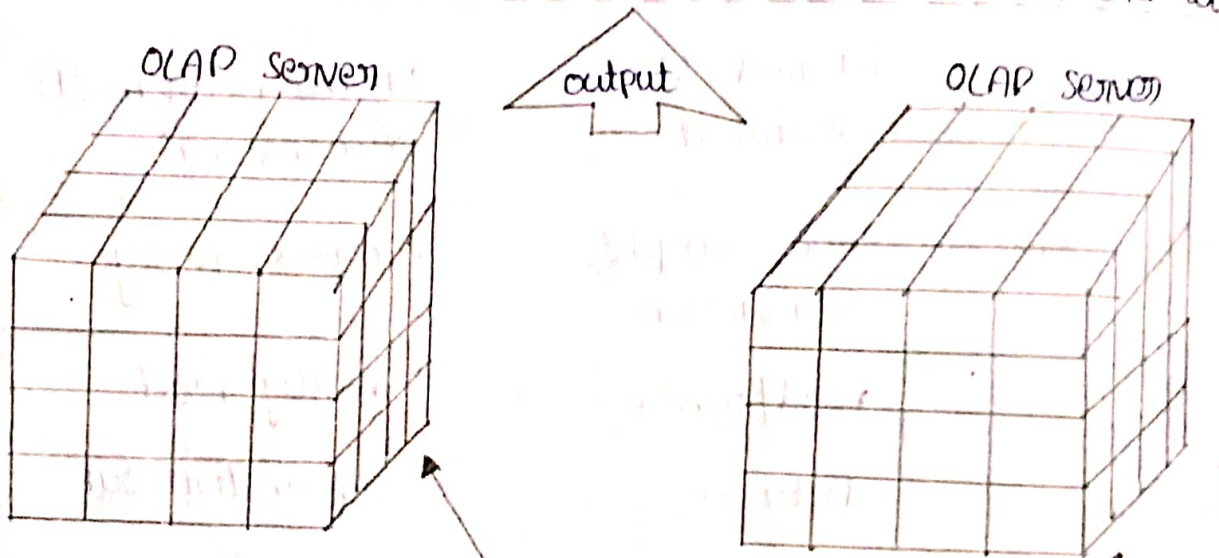
Comparison of OLTP and OLAP systems:

<u>Feature</u>	<u>OLTP</u>	<u>OLAP</u>
characteristic orientation	operational processing transaction	informational processing analysis
User	clerk, DBA, database professional	knowledge worker (e.g. manager, executive analyst)
Function	day-to-day operations	long term informational requirements decision support

Feature	OLTP	OLAP
DB design	ER-based application oriented	star/snowflake, subject-oriented.
Data	current, guaranteed up-to-date	historic, accuracy maintained over time
Summarization	primitive, highly detailed	summarized, consolidated
View	detailed flat relational	summarized, multi-dimensional
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
No. of users	thousands	hundreds
DB size	GB to high-order GB	ZTB
Priority	high performance, high availability	high flexibility, end-user autonomy
Metric	transaction throughput	query throughput, response time.

* Data Warehousing: A Multitiered Architecture:

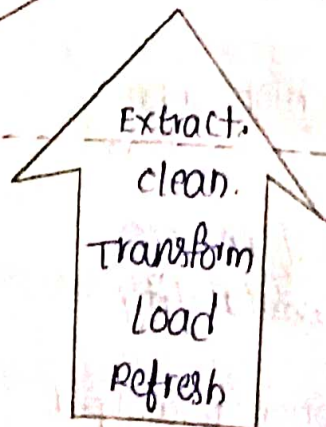
Data warehouses often adopt a three-tier architecture.



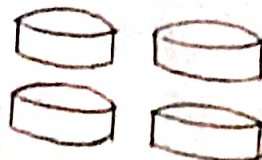
Meta data repository



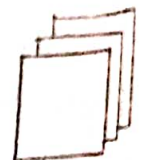
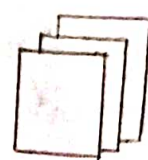
Bottom tier: DWH server



operational databases



External sources



Data

i) $\Rightarrow$ The bottom tier is a warehouse database server
 $\Rightarrow$ that is almost always a relational database system.

$\Rightarrow$ Back-end tools and utilities are used to feed data into the bottom tier from operational databases or other external resources (e.g. customer profile information provided by external consultants.)

$\Rightarrow$ These tools and utilities perform data extraction, cleaning, and transformation (e.g. to merge similar data from different sources into a unified format), as well as load and refresh functions to update the DWH.

$\Rightarrow$ This tier also contains a metadata repository, which stores information about the DWH and its contents.

ii
 $\Rightarrow$ The middle tier is an OLAP server that is typically implemented using either

(1) a relational OLAP (ROLAP) model (i.e., an extended relational DBMS that maps operations on multidimensional data to standard relational operations); or

(2) a multidimensional OLAP (MOLAP) model (i.e., a special-purpose server that directly implements multidimensional data and operations).

iii
 $\Rightarrow$ The top tier is a front-end client layer, which contains query and reporting tools, analysis tools and/or data mining tools (e.g., trend analysis, prediction, and so on.)

* Data Warehouse Models:

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

- The enterprise warehouse
- The data mart
- " virtual warehouse

Enterprise warehouse:

- ⇒ Give the information of all the different types of subjects data (complete data)
- ⇒ An enterprise warehouse collects all of the information about subjects spanning the entire organization.
- ⇒ It provides corporate-wide data integration, usually from one or more operational systems or external information providers and is cross-functional in scope.
- ⇒ It typically contains detailed data as well as summarized data, and can range in size from a few gigabytes to hundreds of gigabytes, terabytes, or beyond.

Data Mart:

- ⇒ A data mart contains a subset of corporate-wide data i.e. of value to a specific group of users. The scope is confined to specific selected subjects.
- ⇒ It is subject-oriented, making specific data available to a defined group of users more quickly, providing them with critical insights.
- ⇒ The availability of specific data ensures that they do not need to waste time searching through an entire data warehouse.

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

Independent: Independent data marts are sourced from data captured from one or more

operational systems or external information providers or from data generated locally within a particular department or geographic data.

Dependent: Dependent data marts are sourced directly from ~~entire~~ enterprise dwll.

virtual warehouse:

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

⇒ For efficient query processing, only some of the possible summary views may be materialized.

⇒ A virtual warehouse is easy to build but requires excess capacity on operational database servers.

Top down, bottom up approaches.

* Extraction, Transformation and Loading:-

DWH systems use back-end tools and utilities to populate and refresh their data. These tools and utilities include the following functions:

Data extraction: which typically gathers data from multiple, heterogeneous and external sources.

Data cleaning: which detects errors in the data and rectifies them when possible.

Data transformation: which converts data from legacy or host format to warehouse format.

Load: which sorts, summarizes, consolidates, computes views, checks integrity, and builds indices and partitions.

Refresh : which propagates the updates from the data sources to the warehouse.

* Metadata Repository :-

⇒ Metadata are data about data. When used in DWH, metadata are the data that define warehouse objects.

A metadata repository should contain the following:

- * A description of the DWH structure, which includes the warehouse schema, view, dimensions, hierarchies and derived data definitions as well as data mart locations and contents.
- * Operational metadata, which include data lineage (history of migrated data and the sequence of transformations applied to it), currency of data (active, archived or purged) and monitoring information (warehouse usage statistics, error reports and audit trails).
- * The algorithms used for summarization, which include measure and dimension definition algorithms, data on granularity, partitions, subject areas, aggregation, summarization and predefined queries and reports.
- * Mapping from the operational environment to the DWH, which includes source databases and their contents, gateway descriptions, data partitions, data extraction, clearing, transformation rules and defaults, data refresh and purging rules, and security (user authorization and access control).
- * Data related to system performance, which include indices and profiles that improve data access and retrieval performance, in addition to rules for the timing and scheduling of refresh, update, and replication cycle.

* Business metadata, which include business terms and definitions, data ownership information and changing policies.

→ A DWH contains different levels of summarization, of which metadata is one.

Data Warehouse Modeling : Data cube and OLAP

* Data Cube : A Multidimensional Data Model :

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

dimension :

dimensions are the perspectives or entities with respect to which an organization wants to keep records. Each dimension may have a table associated with it, called dimension table.

Fact :

A multidimensional data model is typically organized around a central theme, such as sales. This theme is represented by a fact table. Facts are numeric measures. By which we want to analyze relationships b/w dimensions. The fact table contains the names of the facts, or measures as well as keys to each of the related dimension tables.

2-D view of sales data for MIElectronics according to time and item

location = "Vancouver"

time (quarter)	item(type)			
	home entertainment	computer	phone	security
Q1	605	825	143	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	381	580

The sales are from branches located in city Vancouver. The measure displayed

dollars-sold (in thousands).

⇒ This table represents AllElectronics sales data for items sold per Quarter in the city Vancouver.

* The sales for Vancouver are shown w.r.t the time dimension (organized in quarters) and item dimension (organized according to the types of items sold).

* The fact or measure displayed is dollars-sold (in thousands).

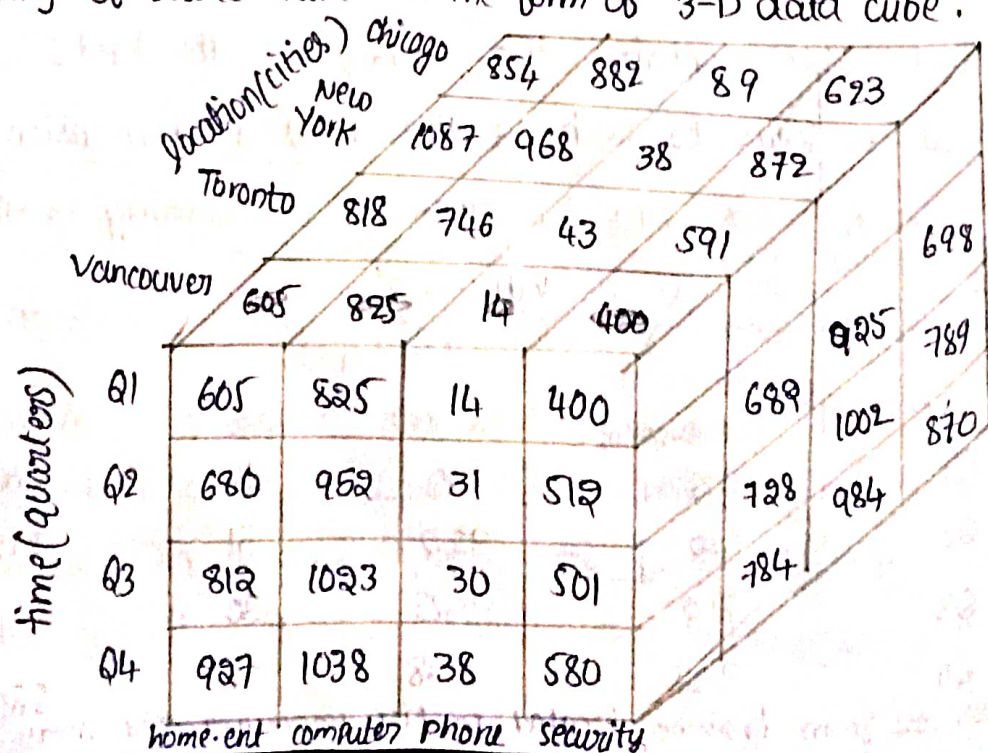
3-D view of sales data for AllElectronics according to time item and location.

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	83	870	1142	1091	54	984	978	864	59	784	927	1038	38	580

⇒ The 3-D data in the table are represented as a series of 2-D tables.

* It represents data according to time and item, as well as location, for the cities Chicago, New York, Toronto and Vancouver.

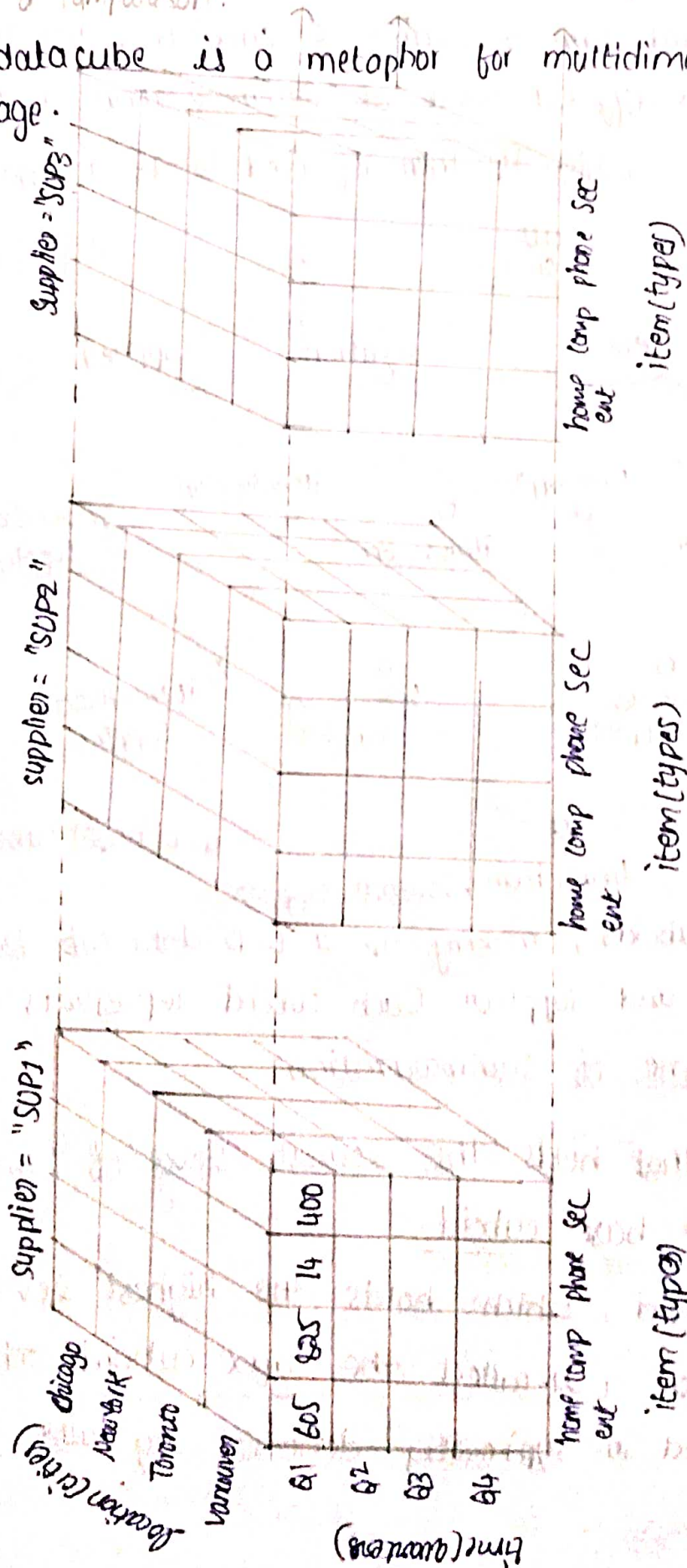
Representing of same data in the form of 3-D data cube.



=> In this way, we may display any n-dimensional data as a series of (n-1)-dimensional "cubes".

A metaphor is a figure of speech that describes an object or action in a way that isn't literally true, but helps explain an idea or make a comparison.

=> A datacube is a metaphor for multidimensional data storage.



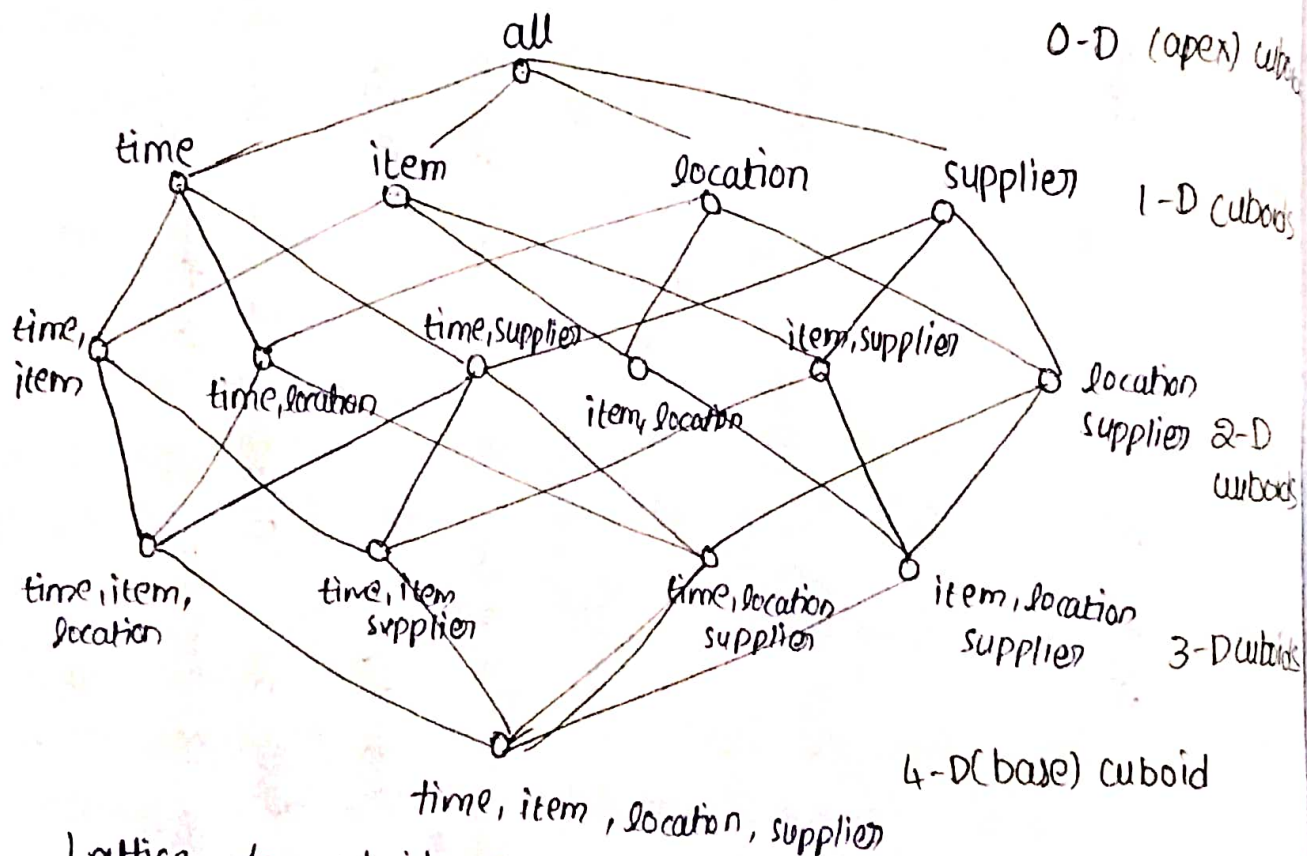
A 4-D data cube representation of sales data, according to time, item, location and supplier. The measure displayed is dollars - sold (in thousands).

⇒ a data cube like those shown in figures are often referred to as a cuboid.

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

⇒ 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.



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

⇒ The cuboid that holds the lowest level of summarization is called the base cuboid.

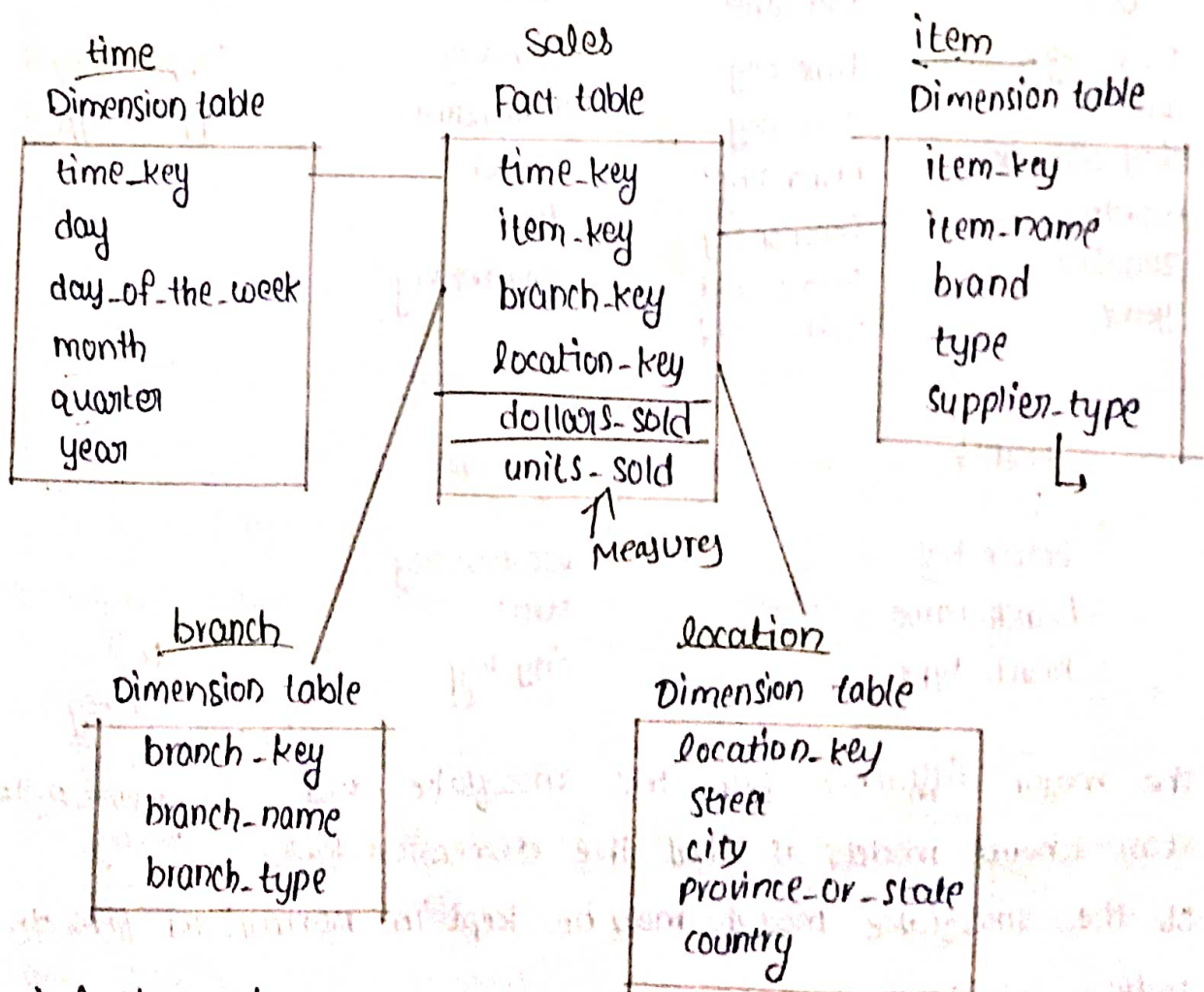
⇒ The 0-D cuboid, which holds the highest level of summarization, is called the apex cuboid. The apex cuboid is typically denoted by all.

* stars, snowflakes and Fact Constellations: Schemas for multidimensional Data Models

The most popular data model for a DWH is a multi-dimensional model, which can exist in the form of a star schema, a snowflake schema or a fact constellation schema.

star schema: The most common modeling paradigm is the star schema, in which the DWH contains

- (1) a large central table (fact table) containing the bulk of the data, with no redundancy
- (2) a set of smaller attendant tables (dimension tables), one for each dimension.

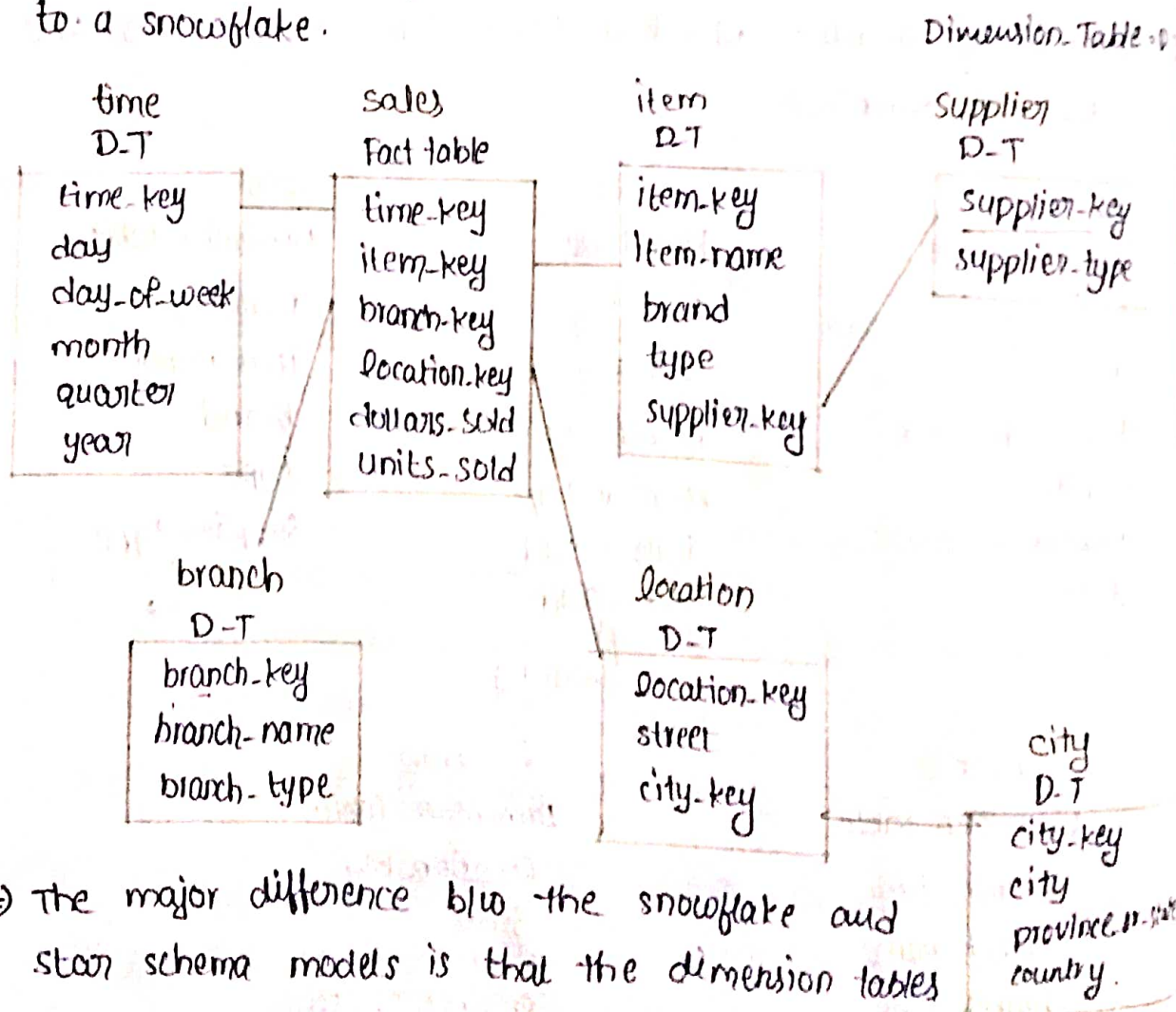


⇒ A star schema for AllElectronics sales contains a central fact table for sales that contains keys to each of the four dimensions, along with two measures: dollars-sold and units-sold.

⇒ To minimize the size of the fact table, dimension identifiers (e.g., time-key and item-key) are system generated identifiers.

⇒ In the star schema, each dimension is represented by only one table, and each table contains a set of attributes.

Snowflake schema: 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. The resulting schema graph forms a shape similar to a snowflake.



⇒ The major difference b/w the snowflake and star schema models is that the dimension tables of the snowflake model may be kept in normalized form to reduce redundancies.

normalized - it means transforming the data, namely converting the source data into another format that allows processing effectively. The main purpose of the data normalization is to minimize or even exclude duplicated data.

⇒ Above figure represents snowflake schema for AllElectronics sales. Here, the sales fact table is identical to that of the star schema.

The main difference b/w two schemas is in the definition of dimension tables. The single dimensional table for "item" in the star schema is normalized in the snowflake schema, resulting in new "item" and "supplier" tables.

Ex:- The "item" dimension table now contains the attributes item-key, item-name, brand, type and supplier-key where supplier-key is linked to "supplier" dimension table, containing supplier-key and supplier-type information.

Fact constellation: Sophisticated applications may require multiple fact tables to share dimension tables. This kind of schema can be viewed as a collection of stars, and hence is called a galaxy schema or a fact constellation.

⇒ The below figure represents fact constellation schema. This schema specifies two fact tables sales and shipping.

* The "sales" table definition is identical to that of the star schema.

* 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 b/w fact tables.

Ex:- The dimension tables for time, item and location are shared b/w the sales and shipping fact tables.

time
D-T

time-key
day
day-of-week
month
quarter
year

branch
D-T

branch-key
branch-name
branch-type

sales

Fact-table

time-key
item-key
branch-key
location-key
dollars-sold
units-sold

item

D-T

item-key
item-name
brand
type
supplier-type

location
D-T

location-key
street
city
province-or-state
country

shipping

fact-table

item-key
time-key
shipper-key
from-location
to-location
dollars-cost
units-shipped

shipper

D-T

shipper-key
shipper-name
location-key
shipper-type

* Dimensions : The Role of Concept Hierarchies

⇒ A concept hierarchy defines a sequence of mappings from a set of low-level concepts to higher-level, more general concepts.

Consider a concept hierarchy for the dimension "location".

* City values for "location" include Vancouver, Toronto, New York and Chicago. Each city can be mapped to the province or state to which it belongs.

* Vancouver can be mapped with British Columbia, Chicago to Illinois.

* The provinces and states can in turn be mapped to the country (e.g. Canada or the United States) to which they belong.

⇒ These mappings form a concept hierarchy for the dimension "location", mapping a set of low-level concepts (i.e. cities) to higher-level, more general concepts (i.e. countries).

