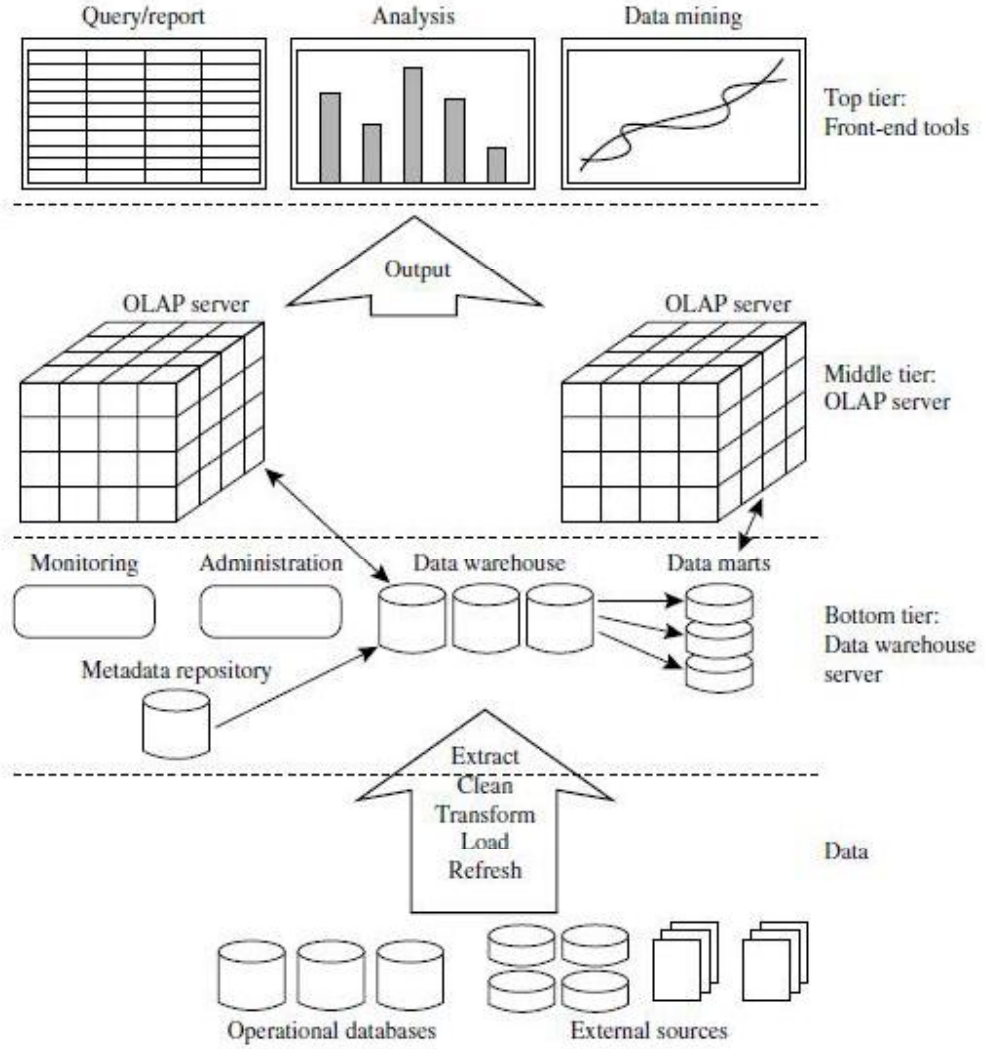
OLAP vs OLTP

**Table 4.1** Comparison of OLTP and OLAP Systems

| *Feature* | *OLTP* | *OLAP* |
|---|---|---|
| Characteristic | operational processing | informational processing |
| Orientation | transaction | analysis |
| User | clerk, DBA, database professional | knowledge worker (e.g., manager, executive, analyst) |
| Function | day-to-day operations | long-term informational requirements decision support |
| 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, 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 |
| Number of records accessed | tens | millions |
| Number of users | thousands | hundreds |
| DB size | GB to high-order GB | ≥ TB |
| Priority | high performance, high availability | high flexibility, end-user autonomy |
| Metric | transaction throughput | query throughput, response time |

*Note:* Table is partially based on Chaudhuri and Dayal [CD97].

# Data Warehouse : Three – tier Architecture




BOTTOM TIER

- The bottom tier is a warehouse database server that is almost always a relational database system.
- Back-end tools and utilities are used to add data into the bottom tier from operational databases or other external sources.
- These tools and utilities perform data extraction, cleaning, transformation, loading and refresh.
- The data are extracted using application program interfaces known as gateways.
- E.g.) ODBC (Open Database Connection), OLEDB (Object Linking and Embedding Database) by Microsoft & JDBC (Java Database Connectivity)
- This tier also contains a metadata repository, which stores information about the data warehouse and its contents

MIDDLE TIER

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

TOP TIER

- 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 Cube & Dimension Analysis

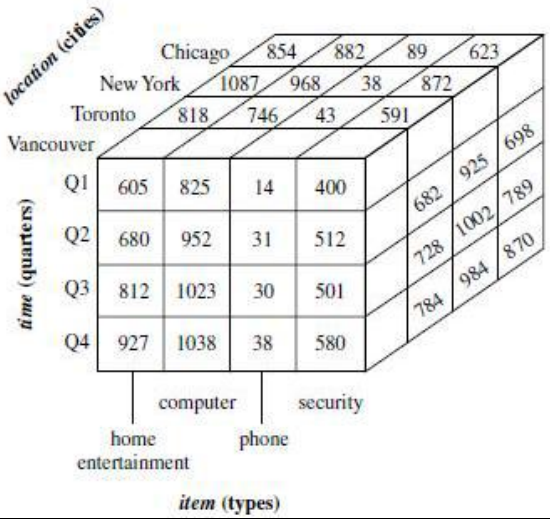
- Data warehouses and OLAP tools are based on a multidimensional data model.
- This model views data in the form of a data cube.
  - A data cube allows data to be modeled and viewed in multiple dimensions. It is defined by dimensions and facts.
  - In general terms, dimensions are the perspectives or entities with respect to which an organization wants to keep records.
  - E.g.) Store's sales with respect to the dimensions time, item, branch, and location.
  - Each dimension may have a table associated with it, called a dimension table.
  - A multidimensional data model is typically organized around a central theme
  - The fact table contains the names of the facts, or measures, as well as keys to each of the related dimension tables.
  - In data warehousing the data cube may contains n-dimensional.
- Let's have a 2-D data view for sales with respect to
  - TIME(4 - Quarters) & ITEM (4-types) at Location = "Vancouver"

| *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 |

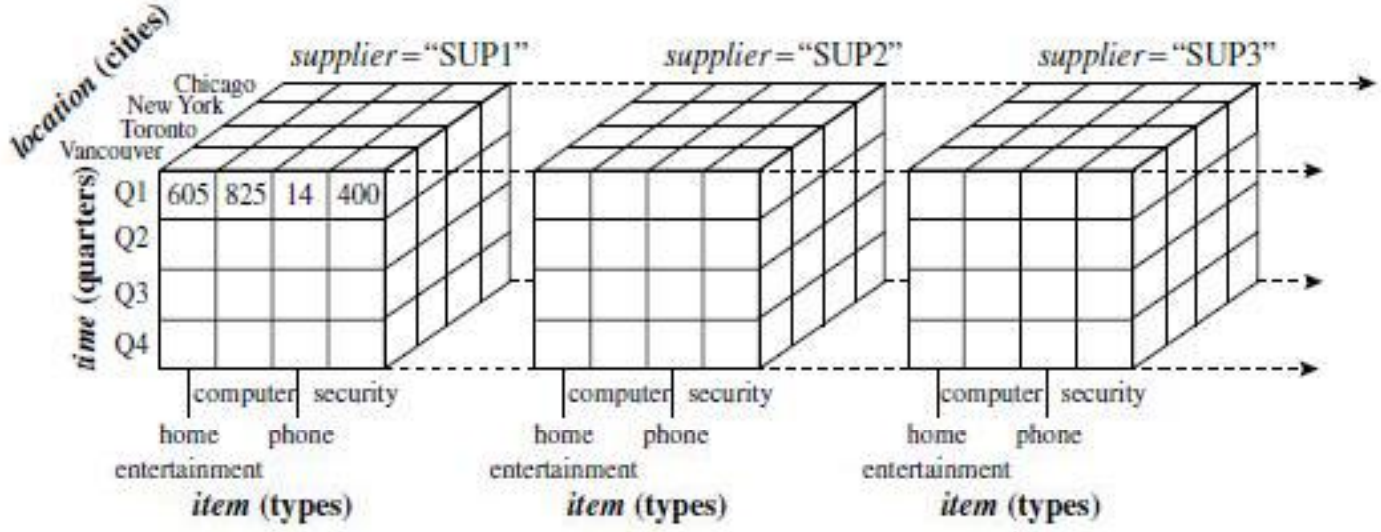
- Now 3 – D data View for the sales where locations and items are increased

| | *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 |

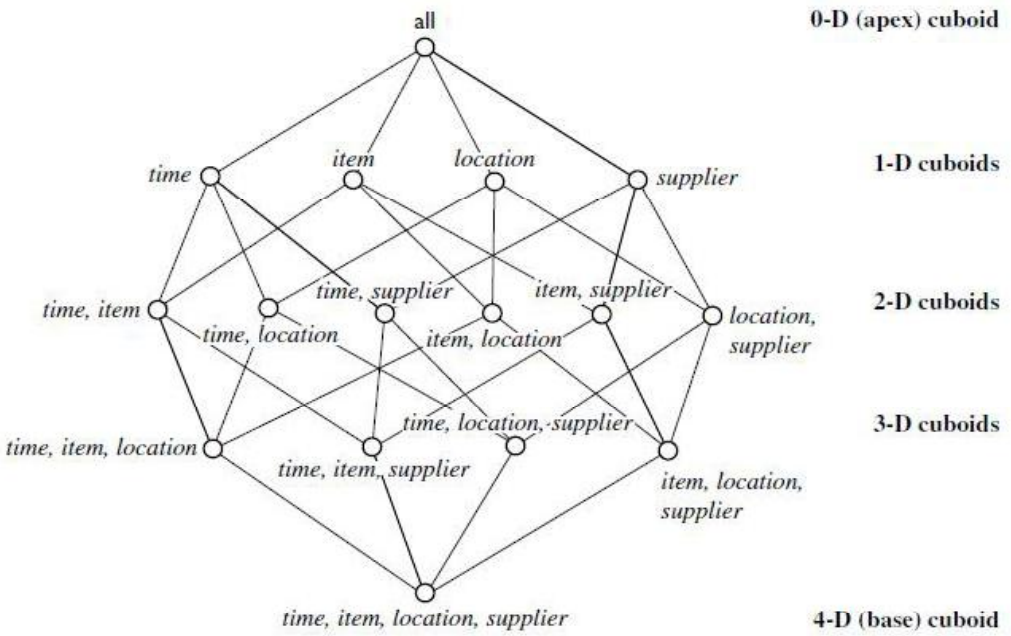
- Now 3 – D Data Cube View




- If another dimension is added as suppliers then 4 – D view is required as below.



- Given a set of dimensions, we can generate a cuboid for each of the possible subsets of the given dimensions.
- The result would form a lattice of cuboids, each showing the data at a different level of summarization.
- Figure shows a lattice of cuboids forming a data cube for the dimensions time, item, location, and supplier.




# Stars, Snowflakes, and Fact Constellations

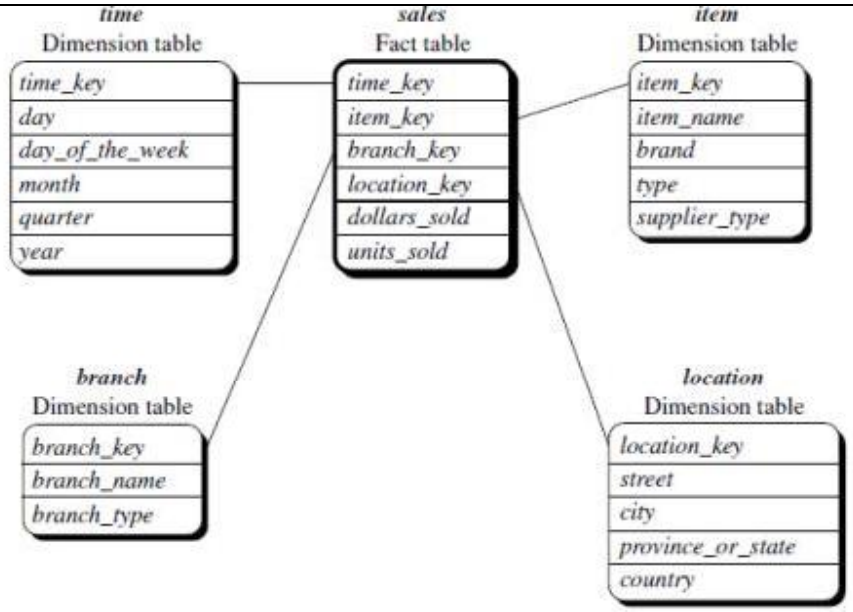
# (Schemas for Multidimensional Data Models)

- Entity – Relationship data model is usually used for Operational database System.
- For Data Warehouse, it requires aconcise, subject – oriented schema that facilitates data analysis.

STAR SCHEMA

- In this data warehouse contains
  1) a large central table (fact table) containing the bulk of the data, with no redundancy, and
  2) a set of smaller attendant tables (dimension tables), one for each dimension.
- It is a relational model where there is a One –to-Many Relationship between each dimension table and fact table.

- E.g.) A star schema for AllElectronics sales is




- Fact table contains dimension identifiers (item_key, item_key, etc) , these are system generated identifiers.
- Fact table also contains some attributes (like dollars_sold, units_sold).
- Each dimension Table is connected to the fact table using dimension identifiers.
- Each dimension table contains set of attributes.

- In this schema , constraint may introduce the redundancy.
  E.g.) in "location table" , one attribute is "country",
  - Each country has no. of states,
  - If data contains "Gujarat " & "Goa" as state attribute's value in this case country is "india". So it generates Redundancy in "Star Schema".
- ADVANTAGES

1) Query Performance
   Queries run faster against a star schema database than an OLTP system because the star schema has fewer tables and clear join paths. This design feature enforces accurate and consistent query results.

2) Load Performance and Administration
   The star schema structure reduces the time required to load large batches of data into a database.
   By defining facts and dimensions and separating them into different tables, the impact of a load operation is reduced.

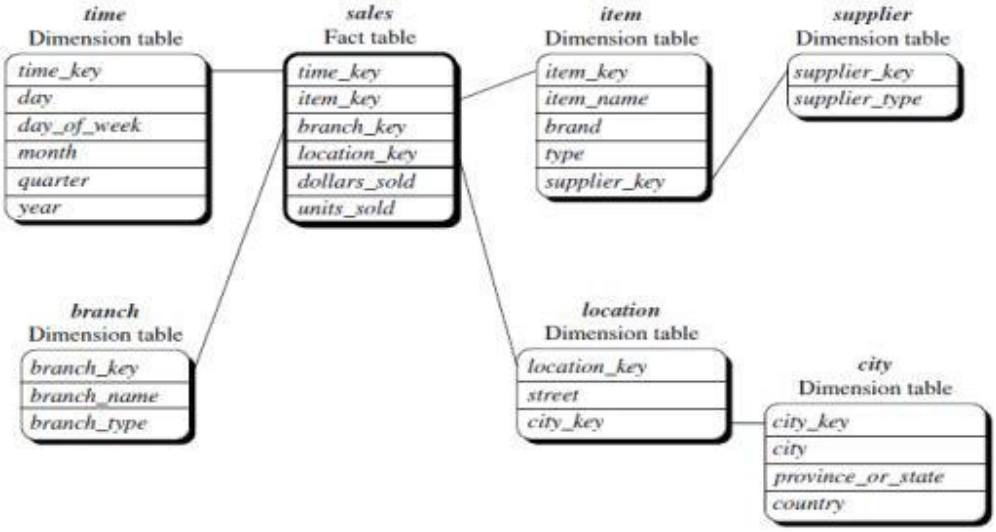
3) Built-in Referential Integrity
   In star schema Referential integrity is enforced by the use of primary and foreign keys.
   Primary keys in dimension tables become foreign keys in fact tables to link each record across dimension and fact tables.

4) Efficient Navigation Through Data
   Navigating through data is efficient because dimensions are joined through fact tables.
   You can browse a single dimension table in order to select attribute values to construct an efficient query.

## 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.

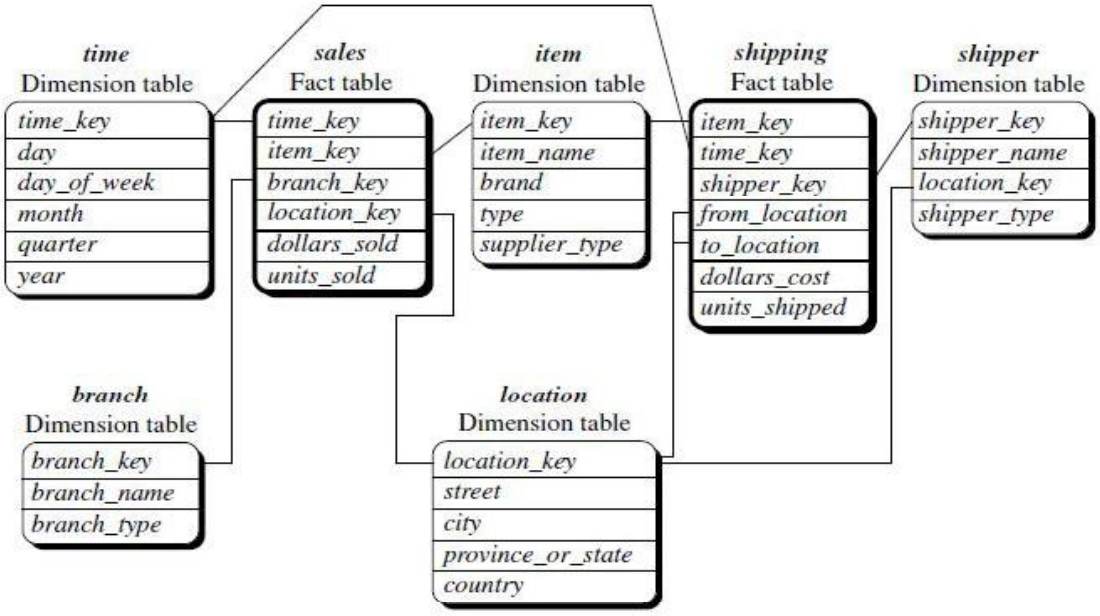
- E.g.) A Snowflake schema for AllElectronics sales is




- From figure, “location” & “item” dimension tables are normalized to additional tables.
- “location” is normalized to “city” and “item” is normalized to “supplier”.
- It means that the Redundancy is reduced in the “Snowflake schema”.

## FACT CONSTELLATION OR GALAXY SCHEMA

- It has more than one Fact Tables.
- In this Dimension tables are shared between different fact tables.
- E.g.) This schema specifies two fact tables, sales and shipping.



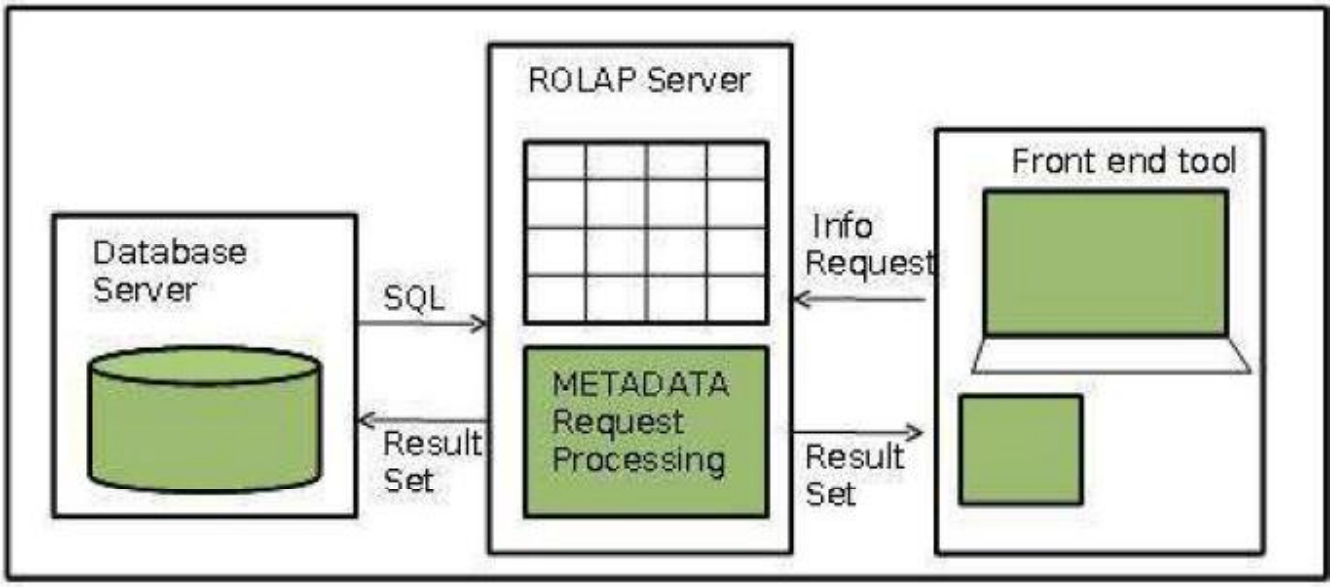
# OLAP Models

# (ROLAP & MOLAP)

## ROLAP

- Relational OLAP servers are placed between relational back-end server and client front-end tools.
- To store and manage the warehouse data, the relational OLAP uses relational or extended relational DBMS.

- ROLAP servers are highly scalable.
- ROLAP tools analyze large volumes of data across multiple dimensions.
- ROLAP tools store and analyze highly volatile and changeable data.

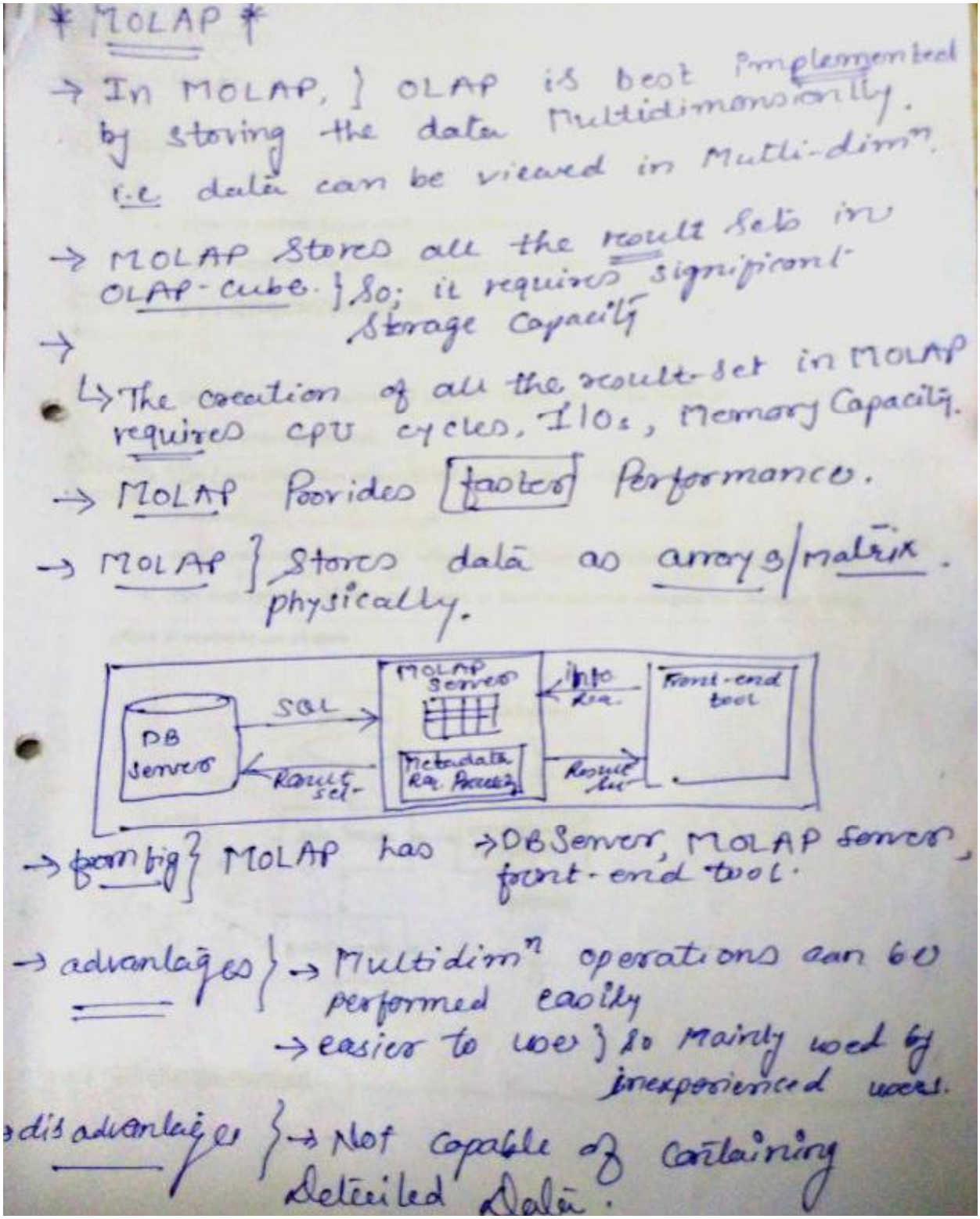



- ROLAP Architecture includes the following components:
  - Database server
  - ROLAP server
  - Front-end tool.

- Advantages
  - ROLAP servers can be easily used with existing RDBMS.
  - Data can be stored efficiently, since no zero facts can be stored.
  - ROLAP tools do not use pre-calculated data cubes.

- Disadvantages
  - Poor query performance.
  - Some limitations of scalability depending on the technology architecture that is utilized.

MOLAP

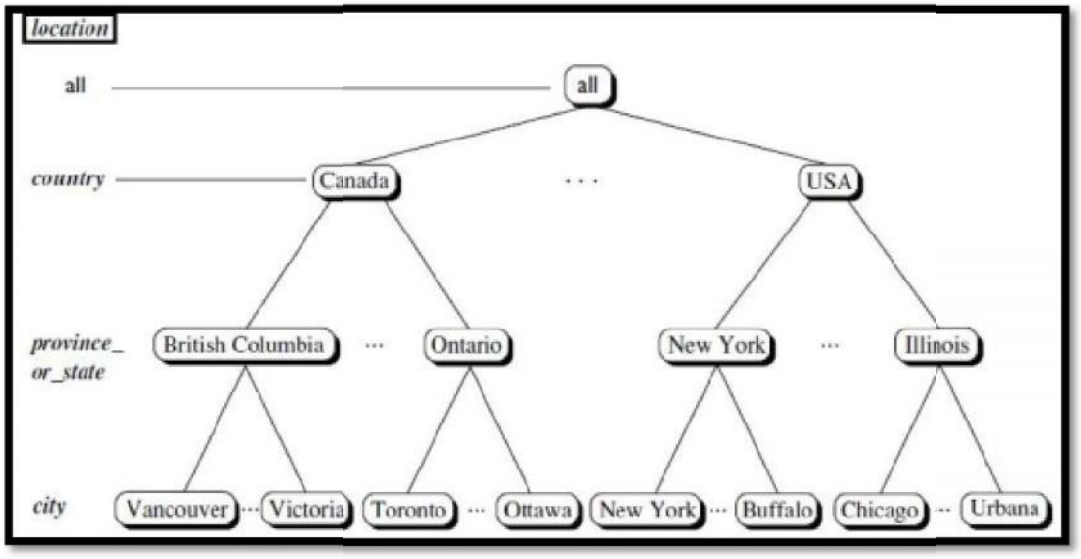

* MOLAP *
→ In MOLAP, } OLAP is best implemented by storing the data multidimensionally. i.e data can be viewed in Multi-dimⁿ.
→ MOLAP Stores all the result sets in OLAP-cube. } So; it requires significant Storage Capacity
→
↳ The creation of all the result-set in MOLAP required CPU cycles, I/Os, Memory Capacity.
→ MOLAP Poorides [faster] Performance.
→ MOLAP } Stores data as arrays/matrix. physically.
DB Server
SQL
Result set
MOLAP Server
Metadata
Front-end tool
→ from big } MOLAP has →DB Server, MOLAP Server, front-end tool.
→ advantages } → Multidimⁿ operations can be performed easily
→ easier to use } So mainly used by inexperienced users.
→ disadvantages } → Not capable of containing detailed Data.

## ROLAP vs MOLAP

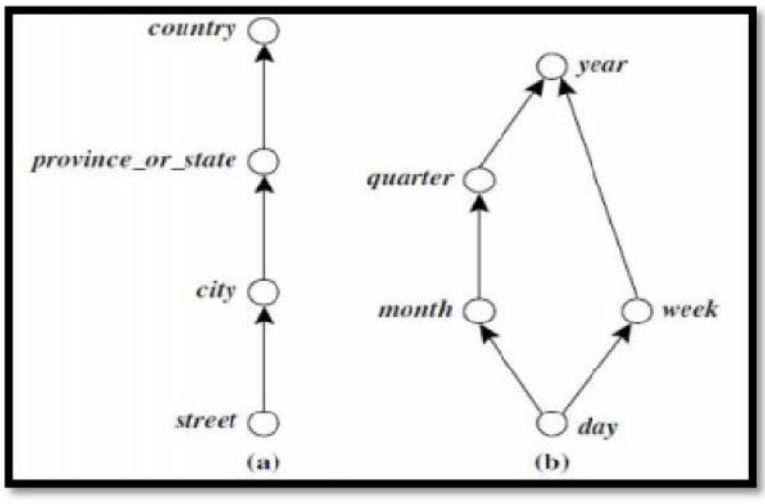
| | ROLAP | MOLAP |
|---|---|---|
| DATA Storage | • Detailed data with less summarization.<br>• Very Large Data Volumes.<br>• All data access from Warehouse storage. | • Summarized data kept in Databases.<br>• Moderate data volumes<br>• Summary Data access from MDDB (Multi Dimensional DBs) and Detailed data access from warehouse |
| Underlying Technologies | • Use of Complex SQL to fetch data from warehouse.<br>• ROLAP Engine creates Data Cubes on the fly and Multidimensional views by presentation Layer. | • Here, Data cubes are created by MOLAP and multidimensional views are stored in arrays. High Matrix data retrieval.<br>• MOLAP Engine creates data cubes and stored in arrays. |
| Functions and Features | • Limitations on complex analysis functions.<br>• Access not faster than MOLAP.<br>• Drill – through to the lowest level easier and drill – across is not easy for ROLAP. | • Large library functions for complex calculations.<br>• Faster Access.<br>• Extensive Drill-Down and slice–Dice Capabilities. |

## Concept Hierarchy

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



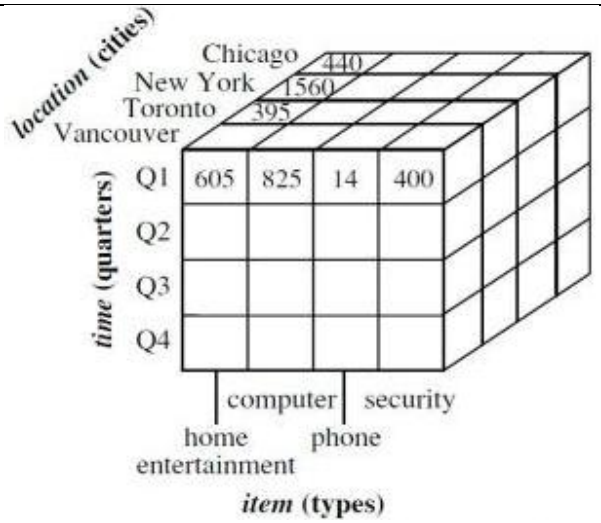

- From Above Figure; The mappings form a concept hierarchy for the dimension location, mapping a set of low-level concepts (i.e., cities) to higher-level, more general concepts (i.e., countries).
- Hierarchical and lattice structures of attributes in warehouse dimensions for :
  a) a hierarchy for location and
  b) a lattice for time.



## OLAP Operations

- 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.

- OLAP Operations are
  1) Roll-up
  2) Drill-down
  3) Slice and dice
  4) Pivot (rotate)
- For OLAP Operations given cube is:

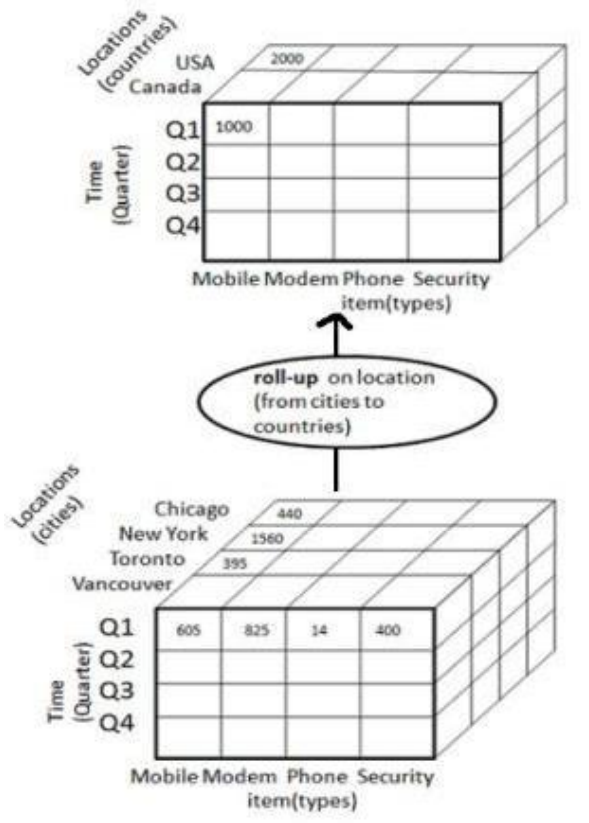



ROLL – UP :

Roll-up performs aggregation on a data cube in any of the following ways:

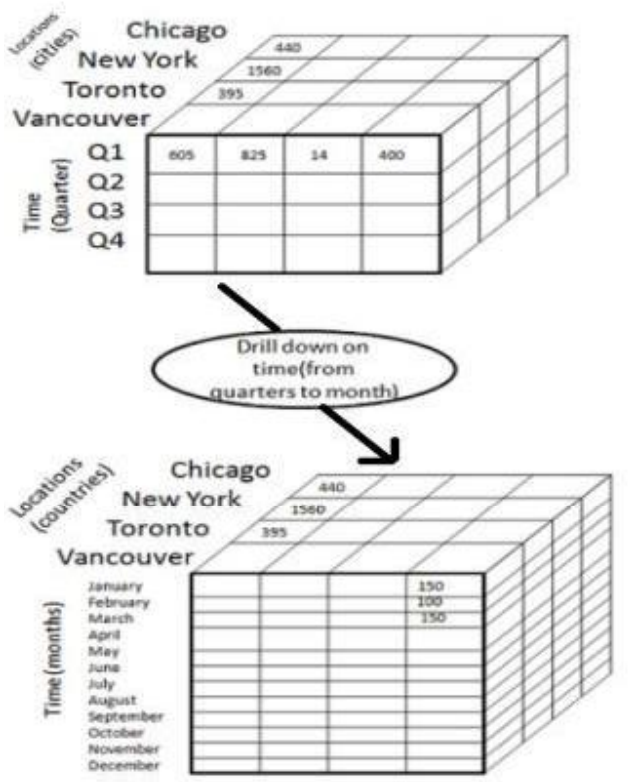
- By climbing up a concept hierarchy for a dimension
- By dimension reduction

- Initially the concept hierarchy was "street < city < province < country".
- On rolling up, the data is aggregated by ascending the location hierarchy from the level of city to the level of country.
- The data is grouped into cities rather than countries.
- When roll-up is performed, one or more dimensions from the data cube are removed.



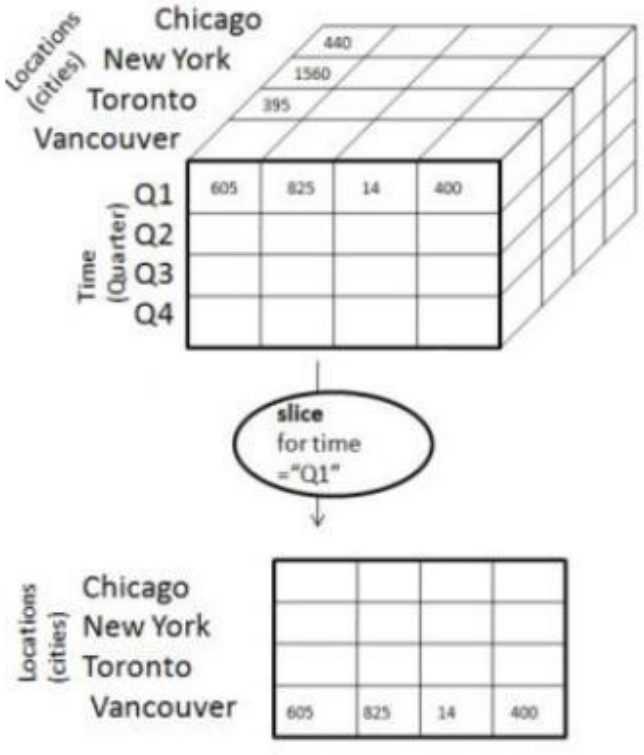
DRILL – DOWN:

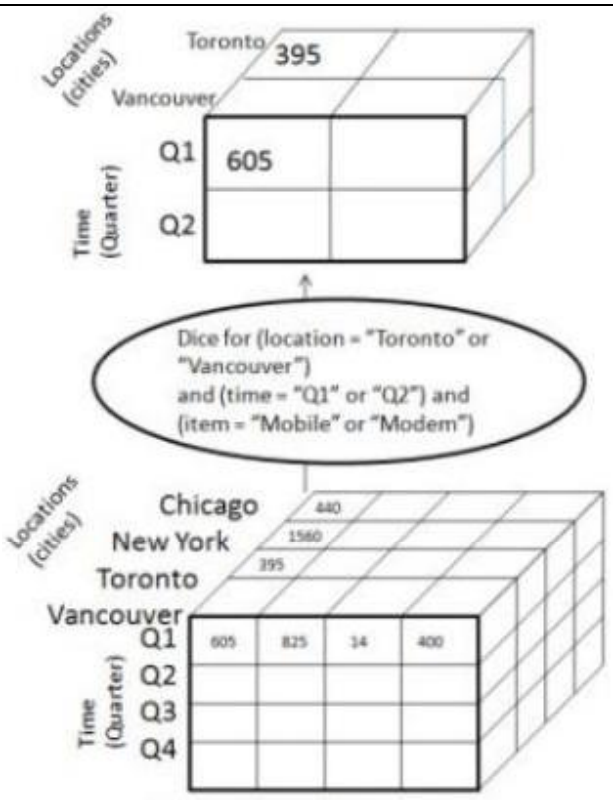
- Drill-down is the reverse operation of roll-up. It is performed by either of the following ways:
  - By stepping down a concept hierarchy for a dimension
  - By introducing a new dimension.
- Initially the concept hierarchy was "day < month < quarter < year."
- On drilling down, the time dimension is descended from the level of quarter to the level of month.
- When drill-down is performed, one or more dimensions from the data cube are added.
- It navigates the data from less detailed data to highly detailed data.




SLICE:

- The slice operation selects one particular dimension from a given cube and provides a new sub-cube.
- Here Slice is performed for the dimension "time" using the criterion time = "Q1".
- It will form a new sub-cube by selecting one or more dimensions.



<table>
<tr>
<td>DICE:<br>• Dice selects two or more dimensions from a given cube and provides a new sub-cube.<br>• The dice operation on the cube based on the following selection criteria involves three dimensions.<br>o (location = "Toronto" or "Vancouver")<br>o (time = "Q1" or "Q2")<br>o (item =" Mobile" or "Modem")</td>
<td> 
</td>
</tr>
<tr>
<td>PIVOT (ROTATE):<br>• The pivot operation is also known as rotation.<br>• It rotates the data axes in view in order to provide an alternative presentation of data.</td>
<td> 
</td>
</tr>
<tr>
<td>"Drill-across" executes queries involving (i.e., across) more than one fact table.</td>
<td>"Drill-through" operation uses relational SQL facilities to drill through the bottom level of a data cube down to its back-end relational tables.</td>
</tr>
</table>