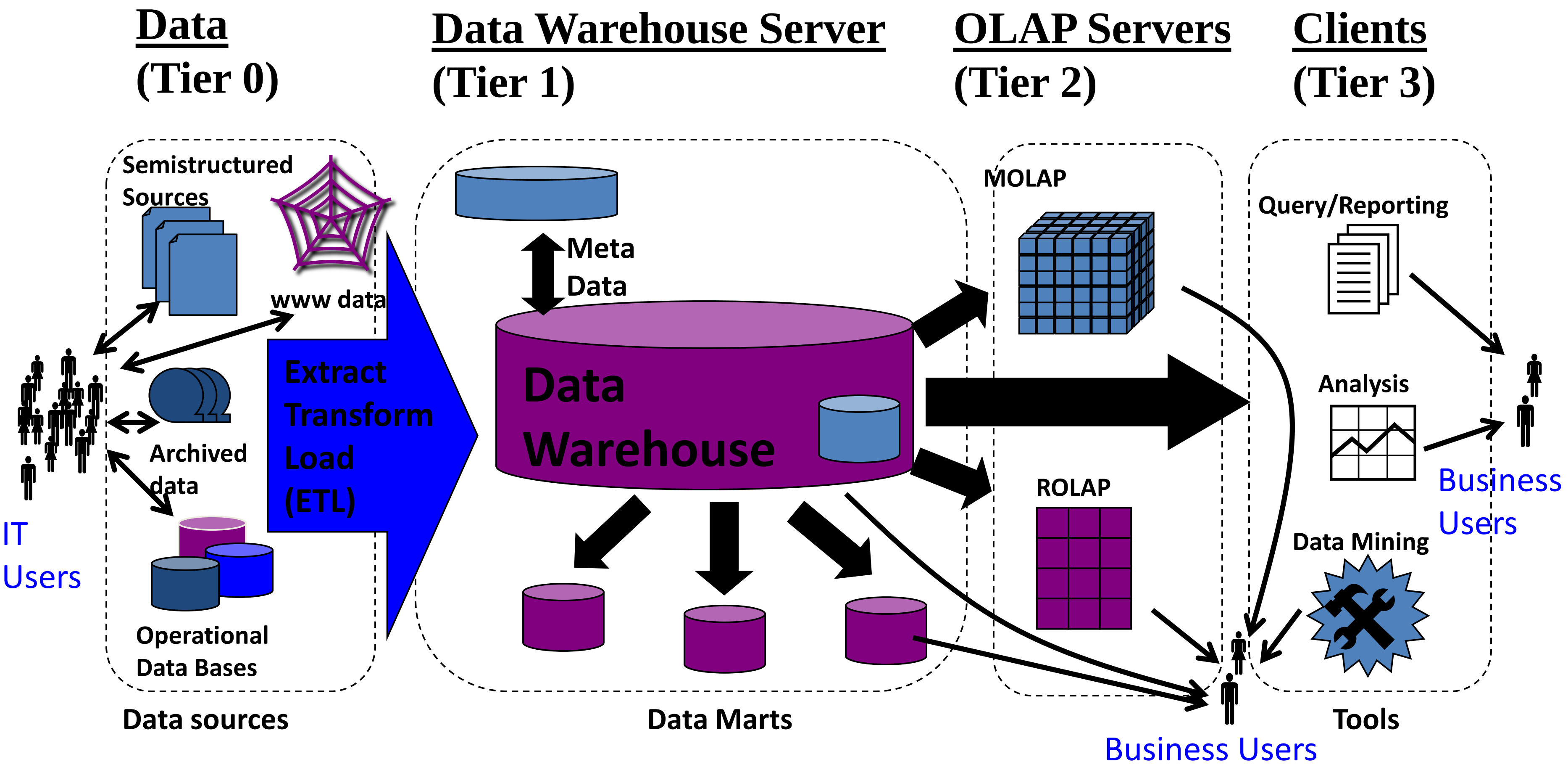
Session 2

Online Analytical Processing (OLAP)

Putting the pieces together



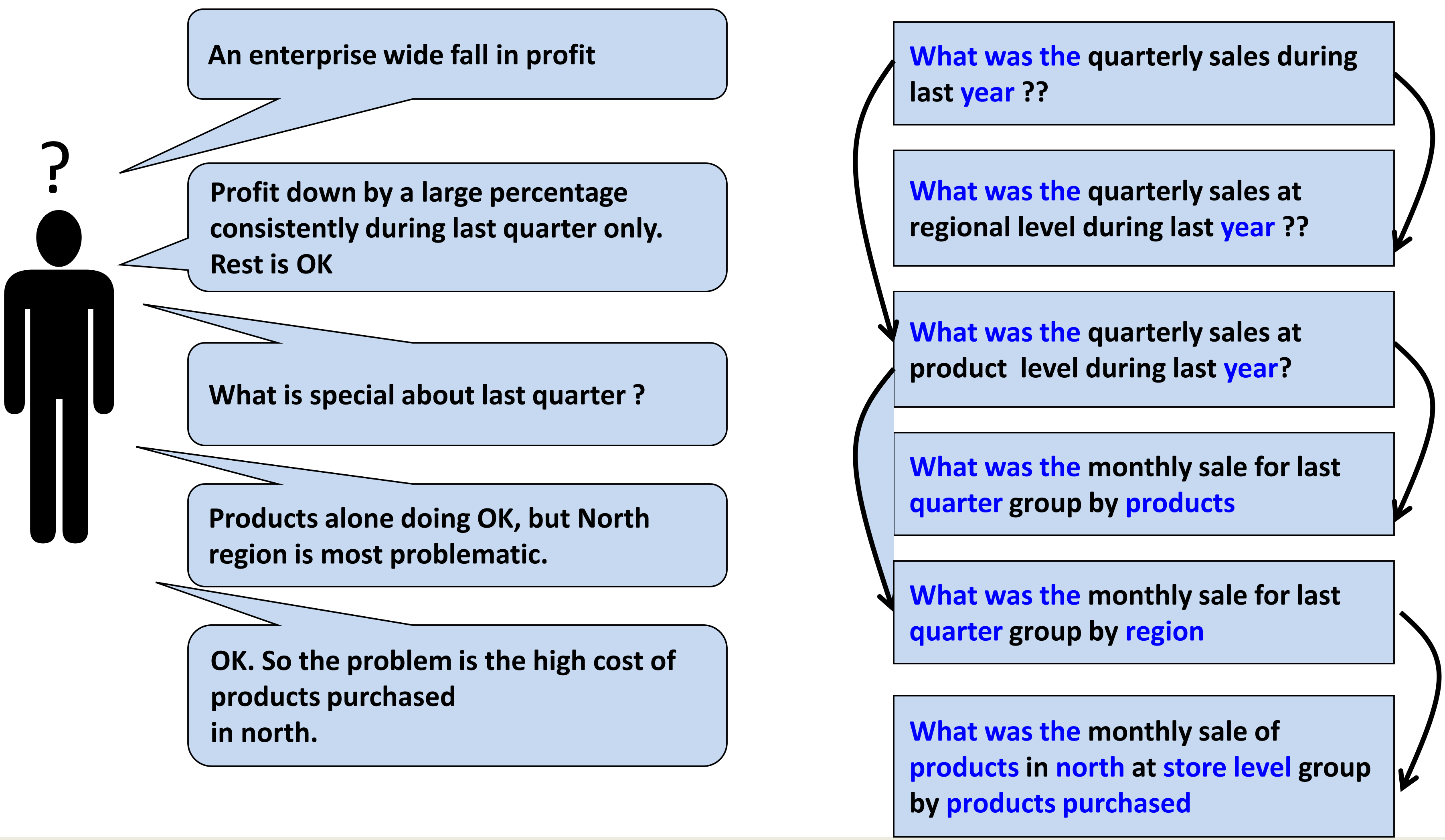
DWH & OLAP

- Relationship between DWH & OLAP

- Data Warehouse & OLAP go together

- Analysis supported by OLAP

Supporting the human thought process



THOUGHT PROCESS

An enterprise wide fall in profit

Profit down by a large percentage consistently during last quarter only. Rest is OK

What is special about last quarter ?

Products alone doing OK, but North region is most problematic.

OK. So the problem is the high cost of products purchased
in north.

QUERY SEQUENCE

What was the quarterly sales during last year ??

What was the quarterly sales at regional level during last year ??

What was the quarterly sales at product level during last year?

What was the monthly sale for last quarter group by products

What was the monthly sale for last quarter group by region

What was the monthly sale of products in north at store level group by products purchased

How many such query sequences can be programmed in advance?

Analysis of last example

- Analysis is Ad-hoc
- Analysis is interactive (user driven)
- Analysis is iterative
  - Answer to one question leads to a dozen more
- Analysis is directional
  - Drill Down
  - Roll Up
  - Pivot

More in subsequent slides

Challenges…

- Not feasible to write predefined queries.
- Enable ad-hoc query support
  - Business user can not build his/her own queries (does not know SQL, should not know it).
  - On_the_go SQL generation and execution too slow because On_the_go SQL can be generated by trapping user's cursor's movement but it will be slow as Olap requires result with in 5 seconds

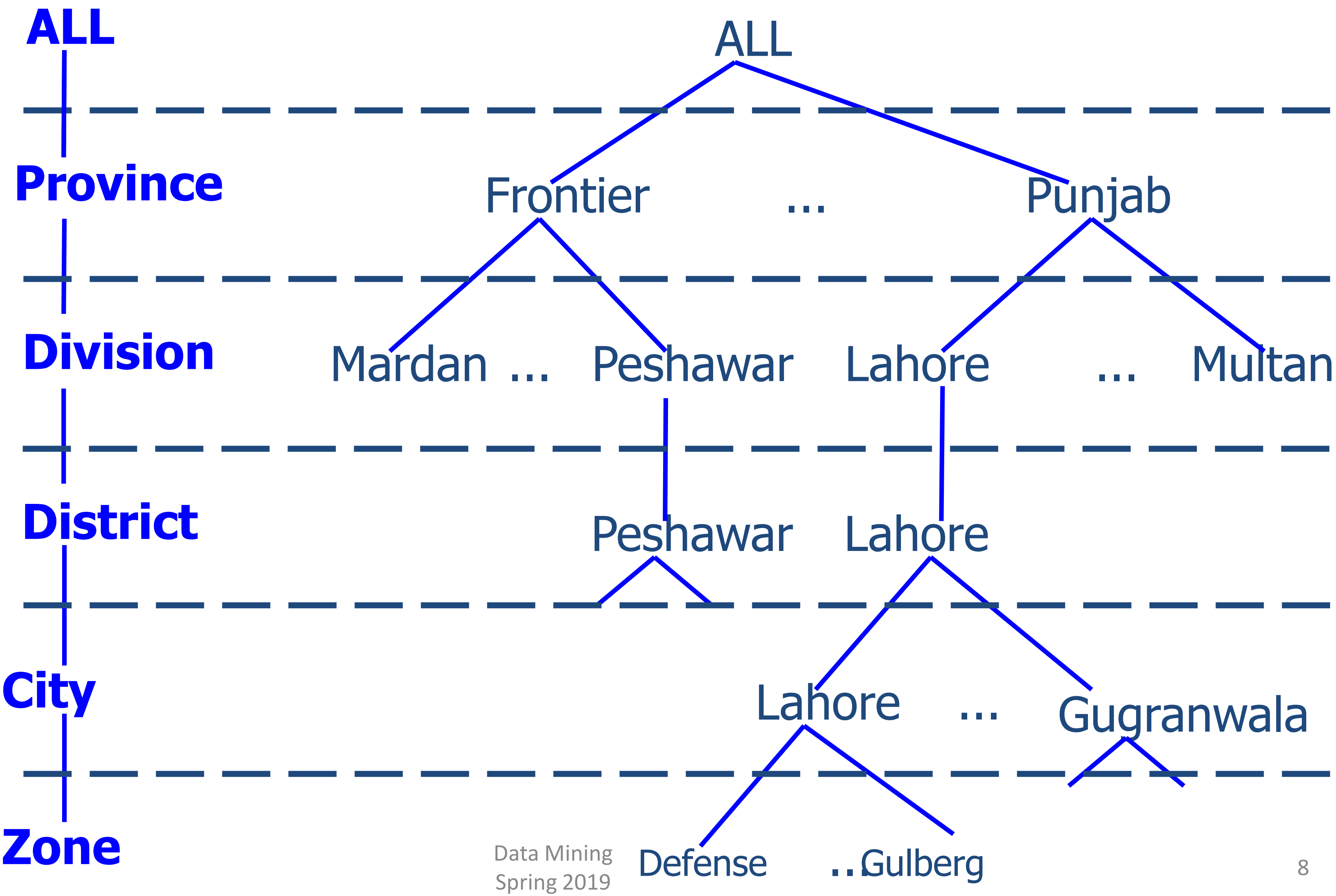
Challenges

- Contradiction
  - Want to compute answers in advance, but don't know the questions as we have seen in CLDS(reverse of SDLC)

- Solution
  - Compute answers to “all” possible “queries”. But how?

  - NOTE: Queries are multidimensional aggregates at some level

“All” possible queries (level aggregates)



OLAP: Facts & Dimensions

- FACTS: Quantitative values (numbers) or “measures.”
  - e.g., units sold, sales $, C^{o}, Kg etc.

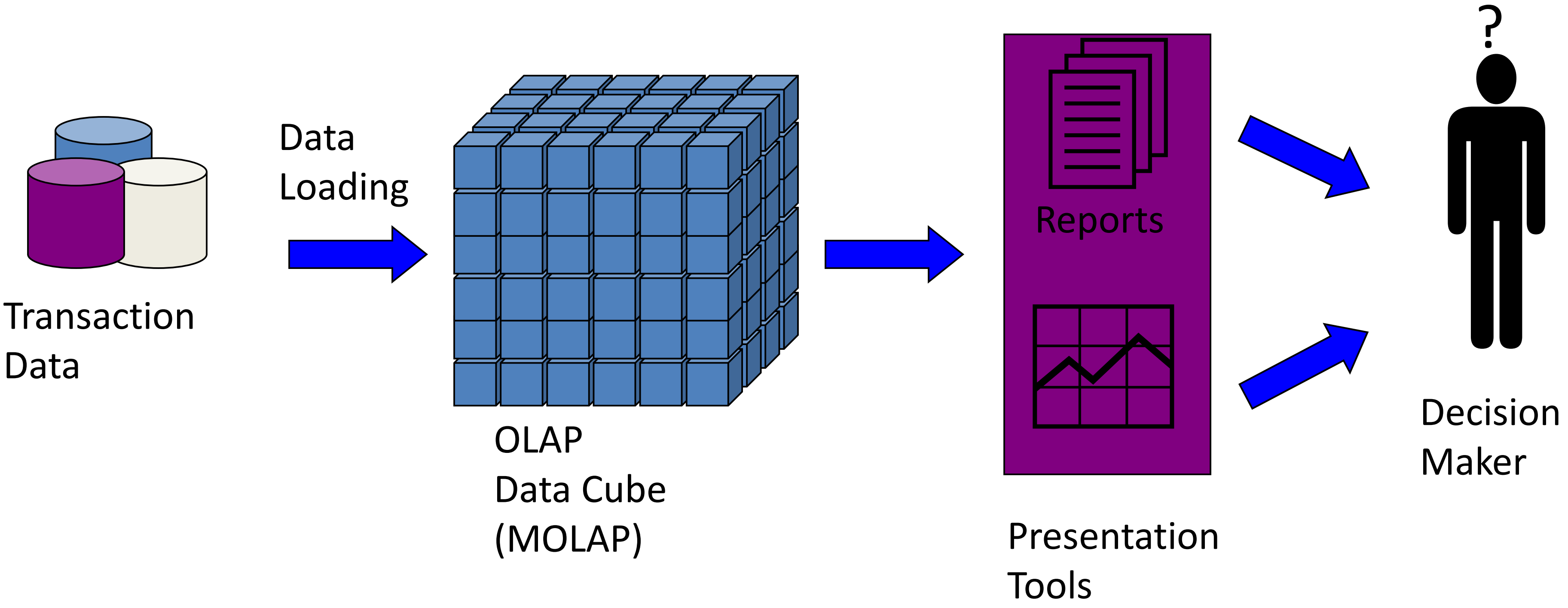
- DIMENSIONS: Descriptive categories use in where clause
  - e.g., time, geography, product etc.

  - DIM often organized in hierarchies representing levels of detail in the data (e.g., week, month, quarter, year, decade etc.).

Where Does OLAP Fit In?

- OLAP is a classification of applications(a framework), NOT a database design technique.
- Analytical processing uses multi-level aggregates, instead of record level access.
- Objective is to support very
  - I. fast
  - II. iterative and
  - III. ad-hoc decision-making.

Where does OLAP fit in?



Transaction Data

Data Loading

OLAP Data Cube (MOLAP)

Presentation Tools

Decision Maker

OLTP vs. OLAP

Feature	OLTP	OLAP
Level of data	**Detailed**	**Aggregated**
Amount of data per transaction	**Small**	**Large**
Views	**Pre-defined**	**User-defined**
Typical write operation	**Update, insert, delete**	**Bulk insert**
“age” of data	**Current (60-90 days)**	**Historical 5-10 years and also current**
Number of users	**High**	**Low-Med**
Tables	**Flat tables**	**Multi-Dimensional tables**
Database size	**Med (10^9 B – 10^{12} B)**	**High (10^{12} B – 10^{15} B)**
Query Optimizing	**Requires experience**	**Already “optimized”**
Data availability	**High**	**Low-Med**

OLAP FASMI Test

Fast: Delivers information to the user at a fairly constant rate. Most queries answered in under five seconds.

Analysis: Performs basic numerical and statistical analysis of the data, pre-defined by an application developer or defined ad-hocly by the user.

Shared: Implements the security requirements necessary for sharing potentially confidential data across a large user population.

Multi-dimensional: The essential characteristic of OLAP.

Information: Accesses all the data and information necessary and relevant for the application, wherever it may reside and not limited by volume.

...from the *OLAP Report* by Pendse and Creeth.

Multidimensional OLAP (MOLAP)

OLAP Implementations

1. **MOLAP:** OLAP implemented with a multi-dimensional data structure.

2. **ROLAP:** OLAP implemented with a relational database.

3. **HOLAP:** OLAP implemented as a hybrid of MOLAP and ROLAP.

4. **DOLAP:** OLAP implemented for desktop decision support environments. Normally use for sales force optimization where we provide a sub part of a MOLAP for specific users so that they can load it in their laptops

MOLAP Implementations

OLAP has historically been implemented using a multi_dimensional data structure or “cube”.

- Dimensions are key business factors for analysis:
  - Geographies (city, district, division, province,...)
  - Products (item, product category, product department,...)
  - Dates (day, week, month, quarter, year,...)

MOLAP Implementations

- No standard query language for querying MOLAP
  - *No SQL !*

- Vendors provide proprietary languages allowing business users to create queries that involve pivots, drilling down, or rolling up.
  - E.g. MDX of Microsoft

  - Languages generally involve extensive visual (click and drag) support.

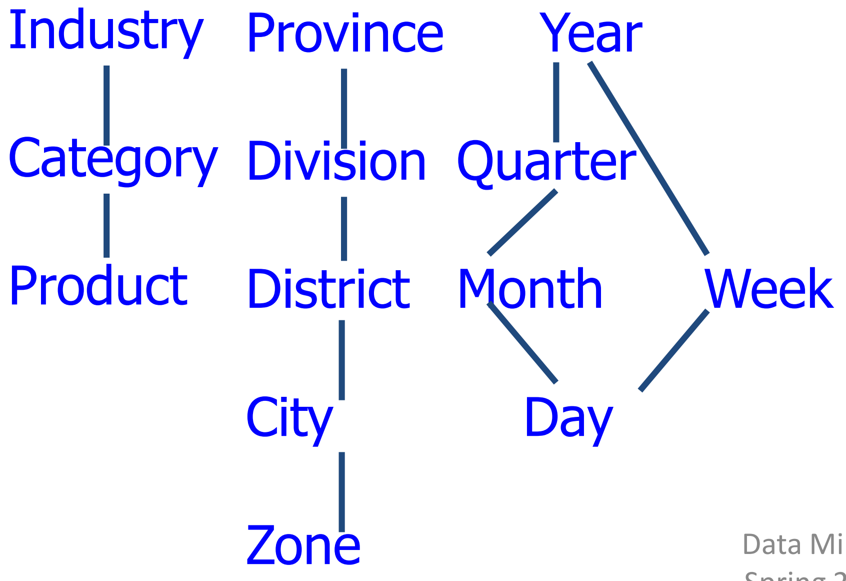
  - Application Programming Interface (API)'s also provided for probing the cubes.

Aggregations in MOLAP

- **Sales volume as a function of (i) product, (ii) time, and (iii) geography**
- **A cube structure created to handle this.**

Dimensions: Product, Geography, Time

Hierarchical summarization paths



Industry

Category

Product

Province

Division

District

City

Zone

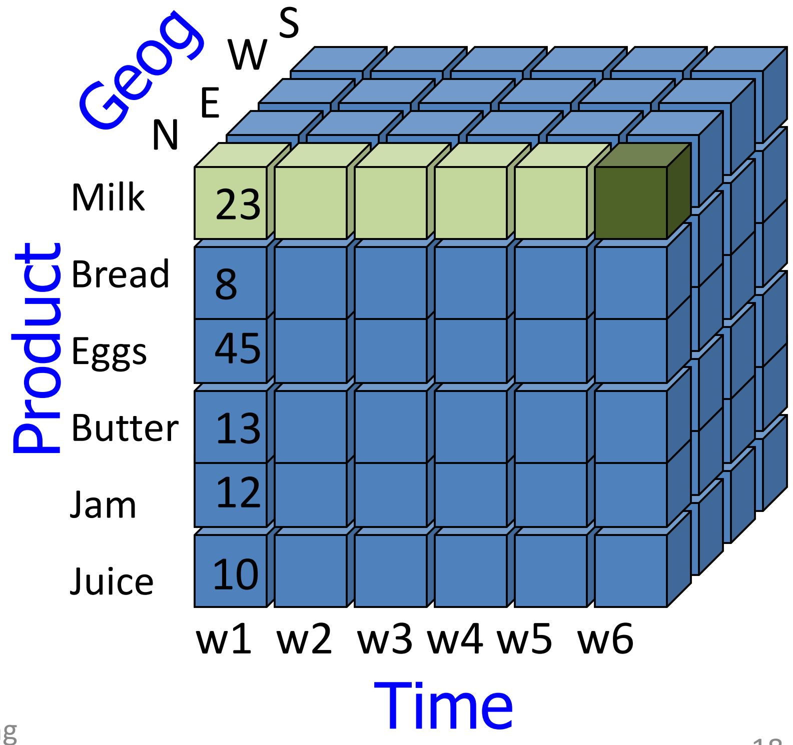
Year

Quarter

Month

Week

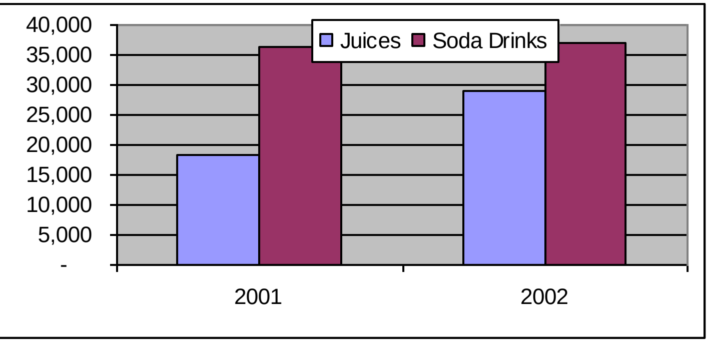
Day



Cube operations

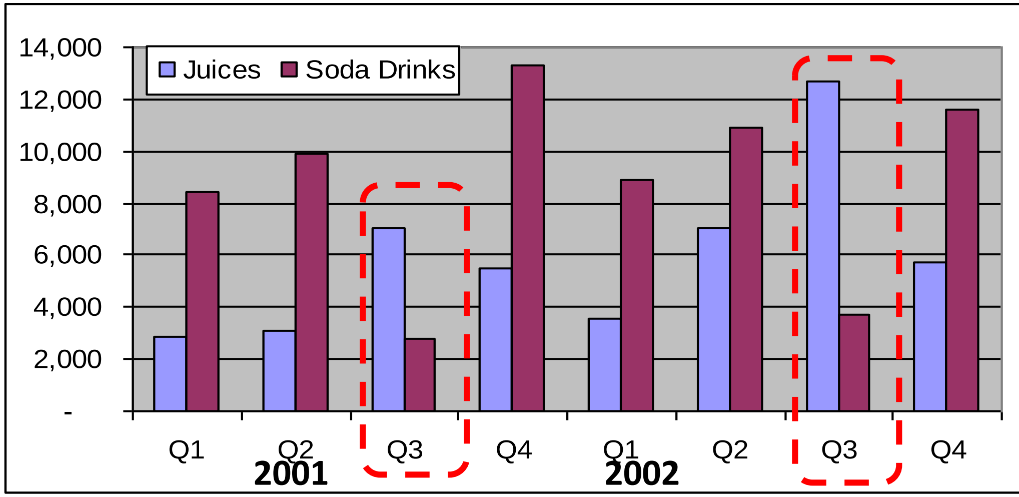
- Drill down: get more details
  - e.g., given summarized sales as above, find breakup of sales by city within each region, or within Sindh
- Rollup: summarize data
  - e.g., given sales data, summarize sales for last year by product category and region
- Slice and dice: select and project
  - e.g.: Sales of soft-drinks in Karachi during last quarter
- Pivot: change the view of data

Querying the cube

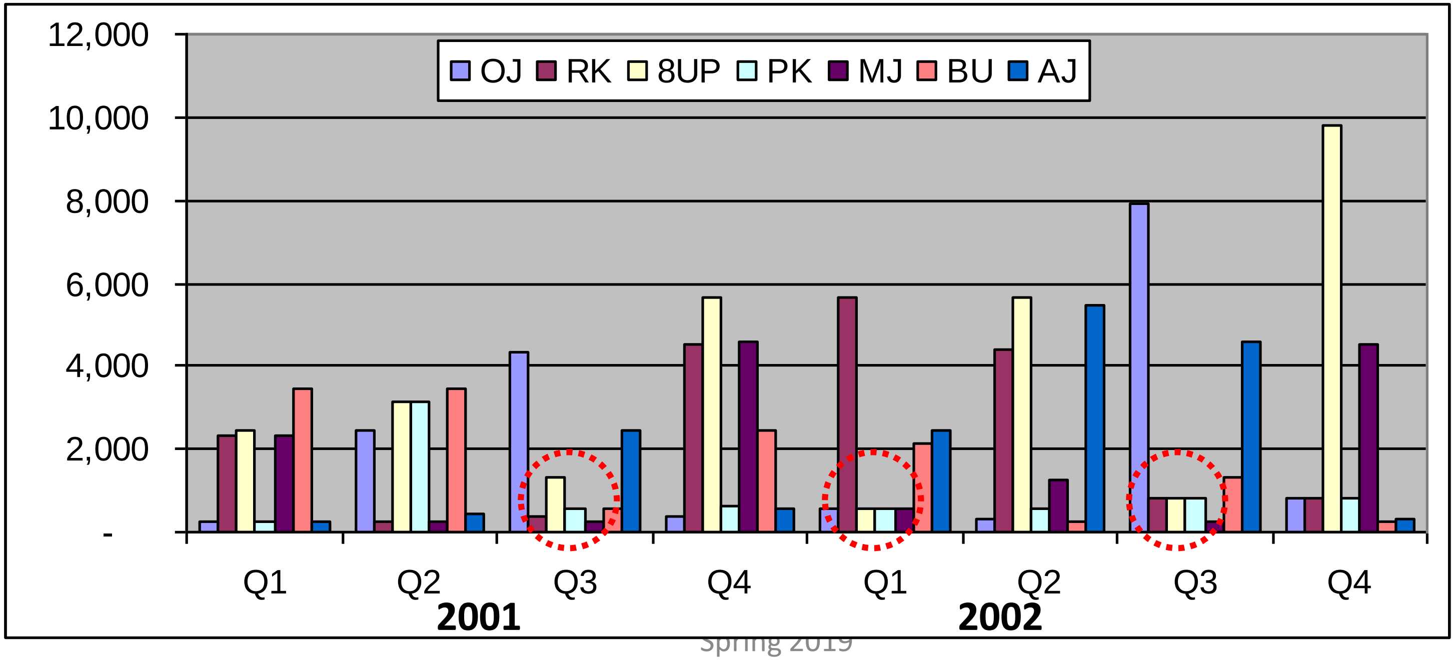


Drill-Down

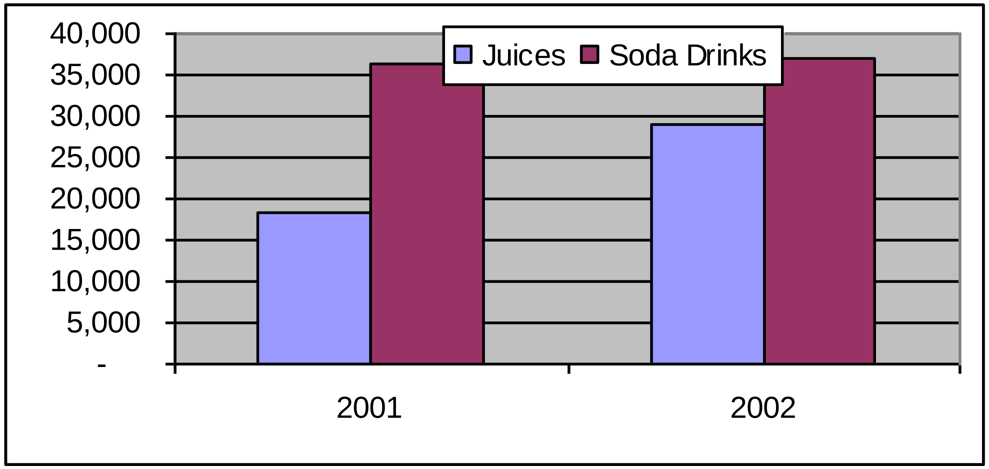
Roll-Up

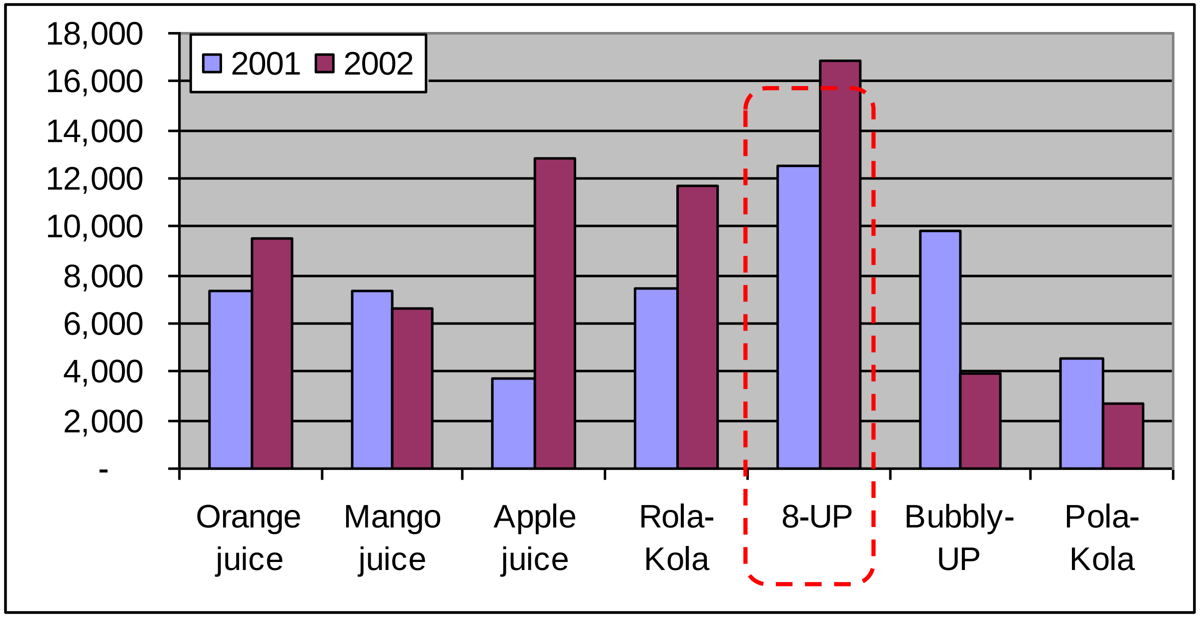


Drill-down



Querying the cube: Pivoting





MOLAP evaluation

Advantages of MOLAP:

- Instant response (pre-calculated aggregates).

- Impossible to ask question without an answer.

- Value added functions sorting the result, getting mean or median etc. (ranking, % change).

MOLAP evaluation

Drawbacks of MOLAP:

- With lots of dimensions aggregate will take **long load time** (pre-calculating the cube may take days!).

- Since here we are considering all items so there might be some empty cells or very sparse cube (wastage of space) for high cardinality (sometimes in small hundreds). e.g. number of heaters sold in Jacobabad or Sibi.

MOLAP Implementation issues

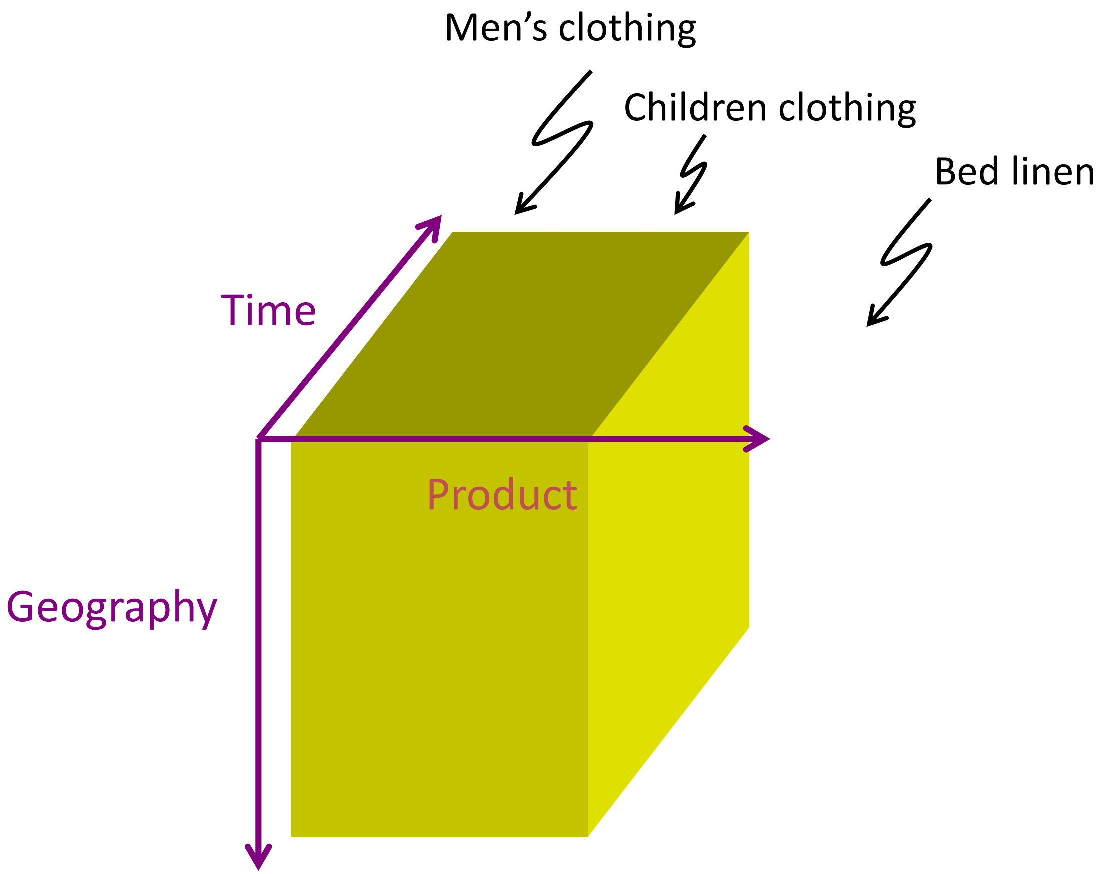
Maintenance issue: Every data item received must be aggregated into *every* cube means in each update/fresh data, we need to re-calculate all aggregates. Lot of work.

Storage issue: As dimensions get less detailed (e.g., year vs. day) cubes get much smaller, but storage consequences for building hundreds of cubes can be significant. Lot of space.

Partitioned Cubes

- To overcome the space limitation of MOLAP, the cube is partitioned.
- The divide&conquer cube partitioning approach helps alleviate the scalability limitations of MOLAP implementation.
- One logical cube of data can be spread across multiple physical cubes on separate (or same) servers.
- Ideal cube partitioning is completely invisible to end users.
- Performance degradation does occurs in case of a join across partitioned cubes.

Partitioned Cubes: How it looks Like?



Sales data cube partitioned at a major cotton products sale outlet can be partitioned into Men’s clothing, Children and Bed linen

Virtual Cubes

Used to query two dissimilar cubes by creating a third “virtual” cube by a join between two cubes.

- Logically similar to a DBMS **view** i.e. linking two (or more) cubes along common dimension(s).

- Biggest advantage is saving in space by eliminating storage of redundant information.

Example: Joining the store cube and the list price cube along the product dimension, to calculate the sale price without redundant storage of the sale price data which means that a company have outlets in different cities and they are buying an item from some manufacturer but its cost varies by adding transportation cost so to calculate its sales price we need additional space for that cube, here we are not keeping them but making them at runtime

Relational OLAP (ROLAP)

Why ROLAP?

Issue of scalability i.e. curse of dimensionality for MOLAP

- Deployment of significantly **large dimension tables** as compared to MOLAP using secondary storage as in ROLAP, we have some normalization.

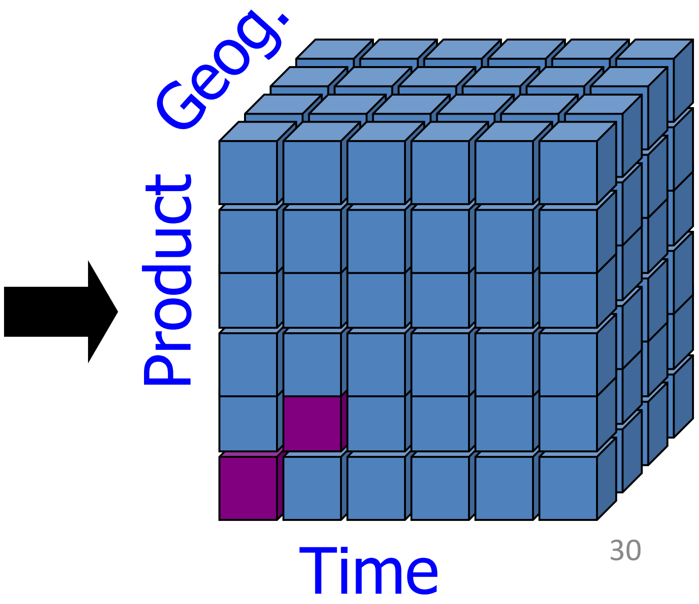
- Star schema designs usually used to facilitate ROLAP querying (in next lecture).

ROLAP as a "Cube"

- OLAP data is stored in a relational database (e.g. a star schema)
- The fact table is a way of *visualizing as* a "un-rolled" cube.
- So where is the cube?
  - It's a matter of perception
  - Visualize the fact table as an elementary cube.

Fact Table

Month	Product	Zone	Sale K Rs.
M1	P1	Z1	250
M2	P2	Z1	500



How to create “Cube” in ROLAP

- Cube is a logical entity containing values of a certain fact at a certain aggregation level at an intersection of a combination of dimensions.

- The following table can be created using 3 queries

Month_ID

Product_ID

SUM (Sales_Amt)	M1	M2	M3	ALL
P1				
P2				
P3				
Total				

How to create “Cube” in ROLAP using SQL

- For the table entries, without the totals

```
SELECT        S.Month_Id, S.Product_Id,
        SUM(S.Sales_Amt)
FROM        Sales
GROUP BYS.Month_Id, S.Product_Id;
```

- For the row totals

```
SELECT        S.Product_Id, SUM (Sales_Amt)
FROM          Sales
GROUP BYS.Product_Id;
```

- For the column totals

```
SELECT     S.Month_Id, SUM (Sales)
FROM       Sales
GROUP BY   S.Month_Id;
```

Problem With Simple Approach

- Number of required queries increases exponentially with the increase in number of dimensions.
  - Its wasteful to compute all queries.
  - In the example, the first query can do most of the work of the other two queries
  - If we could **save that result and aggregate** over Month_Id and Product_Id, we could compute the other queries more efficiently

CUBE Clause

- The CUBE clause is part of SQL:1999
  - GROUP BY CUBE (v1, v2, …, vn)
  - Equivalent to a collection of GROUP BYs, one for each of the subsets of v1, v2, …, vn

CUBE operator

Select

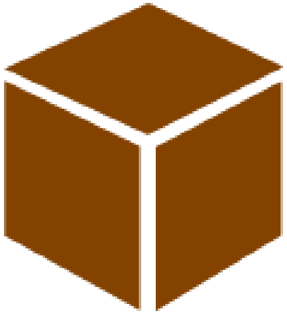
+

Group By → *Row information*

+

WITH CUBE → ***Super Aggregate Row***

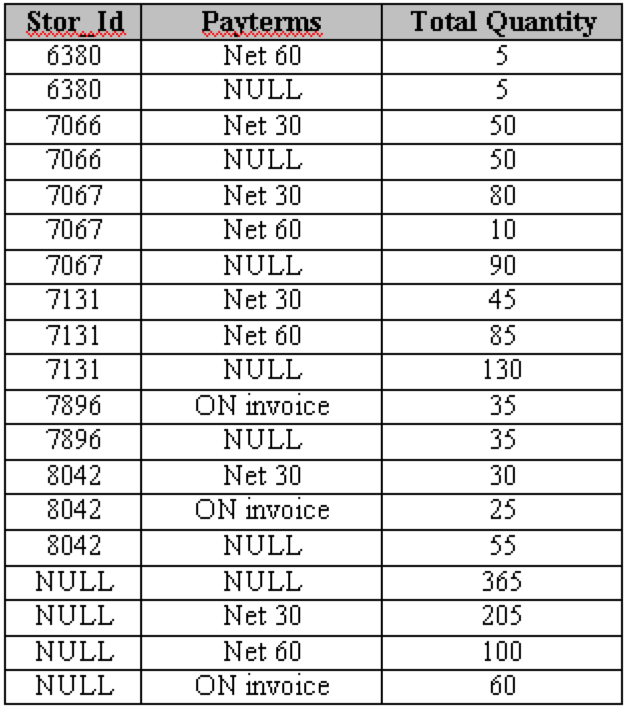
SUMMARY created by all possible groupings of selected columns



Using CUBE - Summarizing Data

Example

```
SELECT Stor_Id, Payterms, SUM(Qty)
AS Total_Quantity
FROM Sales
GROUP BY Stor_id, Payterms
WITH CUBE
```



Stor_Id	Payterms	Total Quantity
6380	Net 60	5
6380	NULL	5
7066	Net 30	50
7066	NULL	50
7067	Net 30	80
7067	Net 60	10
7067	NULL	90
7131	Net 30	45
7131	Net 60	85
7131	NULL	130
7896	ON invoice	35
7896	NULL	35
8042	Net 30	30
8042	ON invoice	25
8042	NULL	55
NULL	NULL	365
NULL	Net 30	205
NULL	Net 60	100
NULL	ON invoice	60

ROLAP & Space Requirement

If one is not careful, with the increase in number of dimensions, the number of summary tables gets very large

Consider the example discussed earlier with the following two dimensions on the fact table...

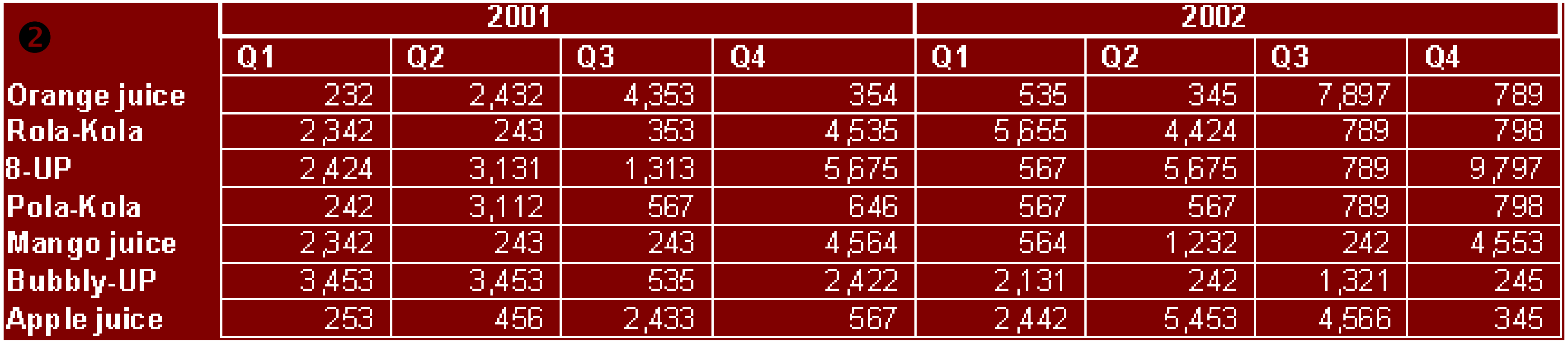
Time: Day, Week, Month, Quarter, Year, All Days

Product: Item, Sub-Category, Category, All Products

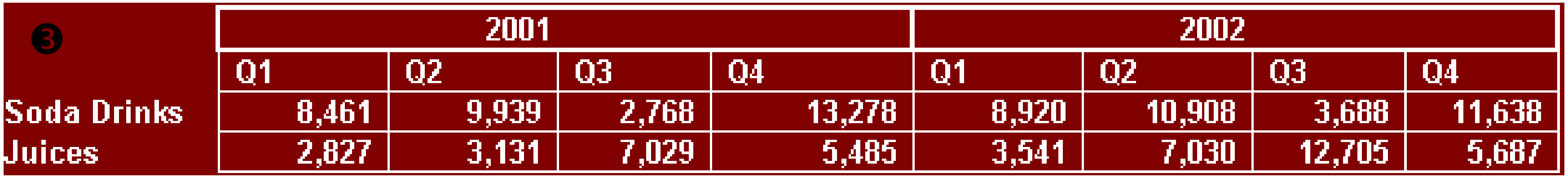
With both time and product the total number of tables will be 24 (6*4)

EXAMPLE: ROLAP & Space Requirement

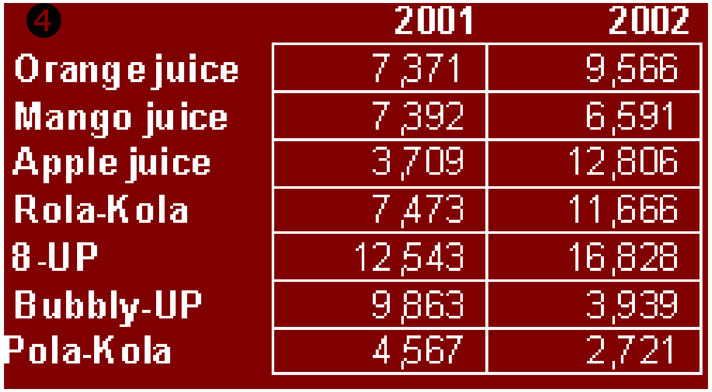
A naïve implementation will require all combinations of summary tables at each and every aggregation level.



❷	2001				2002			
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
Orange juice	232	2,432	4,353	354	535	345	7,897	789
Rola-Kola	2,342	243	353	4,535	5,655	4,424	789	798
8-UP	2,424	3,131	1,313	5,675	567	5,675	789	9,797
Pola-Kola	242	3,112	567	646	567	567	789	798
Mango juice	2,342	243	243	4,564	564	1,232	242	4,553
Bubbly-UP	3,453	3,453	535	2,422	2,131	242	1,321	245
Apple juice	253	456	2,433	567	2,442	5,453	4,566	345

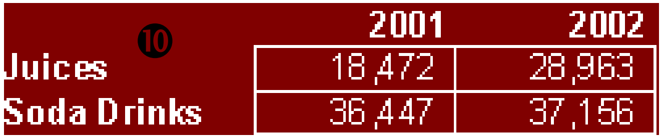


❸	2001				2002			
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
Soda Drinks	8,461	9,939	2,768	13,278	8,920	10,908	3,688	11,638
Juices	2,827	3,131	7,029	5,485	3,541	7,030	12,705	5,687



❹	2001	2002
Orange juice	7,371	9,566
Mango juice	7,392	6,591
Apple juice	3,709	12,806
Rola-Kola	7,473	11,666
8-UP	12,543	16,828
Bubbly-UP	9,863	3,939
Pola-Kola	4,567	2,721

…



❿	2001	2002
Juices	18,472	28,963
Soda Drinks	36,447	37,156

24 summary tables, add in geography, results in 120 tables(6*4*5)

ROLAP Issues

- Maintenance.
- Non standard hierarchy of dimensions.
- Non standard conventions.
- Explosion of storage space requirement.
- Aggregation pit-falls.

ROLAP Issue: Maintenance

Summary tables are mostly a maintenance issue (similar to MOLAP) than a storage issue.

- Notice that summary tables get much smaller as dimensions get less detailed (e.g., year vs. day as seen in previous slide).
- Should plan for twice the size of the un-summarized data for ROLAP summaries in most environments as creating aggregates means creating tables, which need additional space.
- Assuming "to-date" summaries, every detail record that is received into warehouse must aggregate into EVERY summary table.

ROLAP Issue: Hierarchies

Dimensions are NOT always simple hierarchies

Dimensions can be more than simple hierarchies i.e. item, subcategory, category, etc.

The product dimension might also branch off by trade style that cross simple hierarchy boundaries such as:

Looking at sales of air conditioners that cross manufacturer boundaries, such as COY1, COY2, COY3 etc means now we are no more looking w.r.to items, sub-items but w.r.to manufacturers.

Looking at sales of all "green colored" items that even cross product categories (washing machine, refrigerator, split-AC, etc.).

Looking at a combination of both.

ROLAP Issue: Convention

Conventions are NOT absolute

Example: What is calendar year? What is a week?

- Calendar:

  01 Jan. to 31 Dec (Calendar year for HR dept) or

  01 Jul. to 30 Jun. (Financial year for finance dept) or

  01 Sep to 30 Aug. (Sales year for sales dept)

- Week:

  Mon. to Sat. or Thu. to Wed.

ROLAP Issue: Storage space explosion

Summary tables required for non-standard grouping

Summary tables required along different definitions of year, week etc.

ROALP Issues: Aggregation pitfalls

- Throwing away the detail data after aggregation.

- Means higher the level of aggregation, lower the detail so to make aggregates at higher level we are removing the chance of getting back to previous data as with higher level we are also losing the data by throwing.

How to Reduce Summary tables?

Many ROLAP products have developed means to reduce the number of summary tables by:

- Building summaries on-the-fly as required by end-user applications.
- Providing smart tools to assist DBAs in selecting the "best” aggregations to build i.e. trade-off between speed and space.

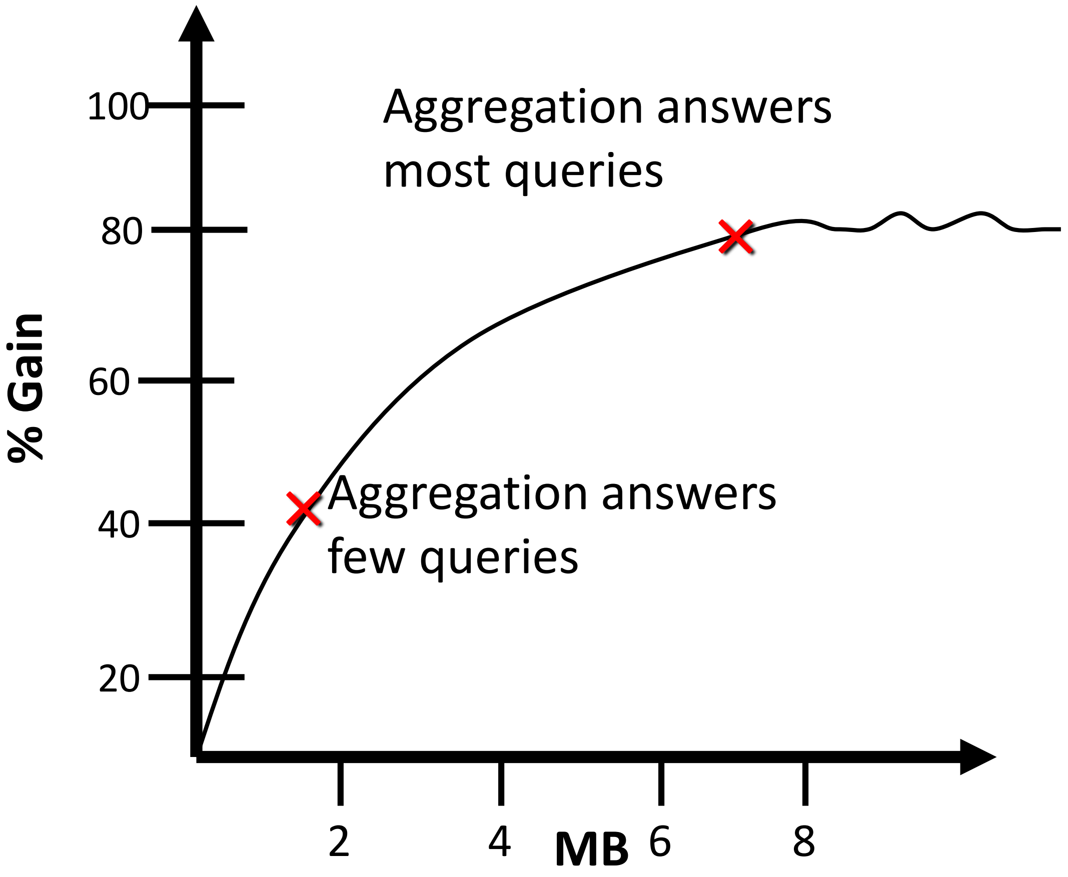
Performance vs. Space Trade-Off

- Maximum performance boost implies using lots of disk space for storing every pre-calculation.

- Minimum performance boost implies no disk space with zero pre-calculation.

- Using meta data to determine best level of pre-aggregation from which all other aggregates can be computed.

Performance vs. Space Trade-off using Wizard



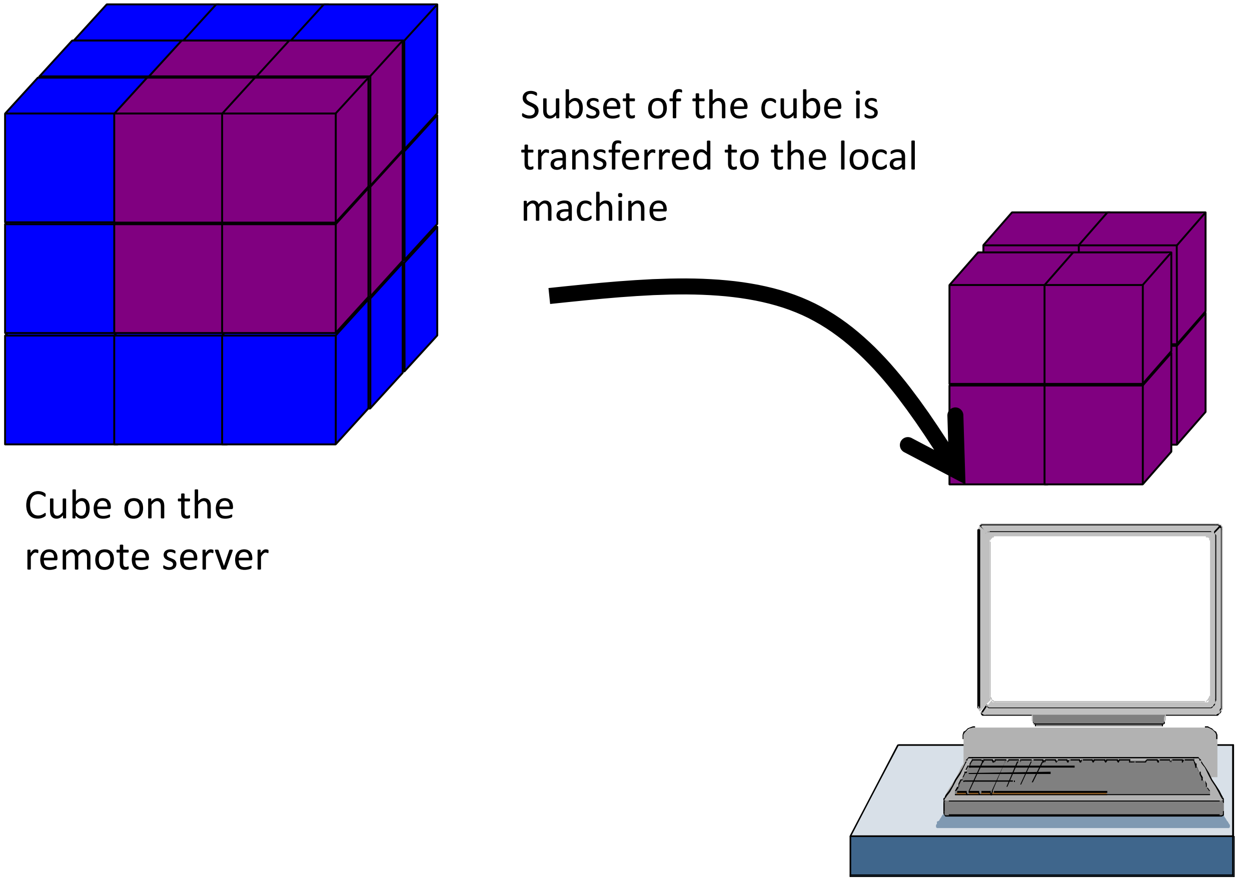
Aggregation answers most queries

Aggregation answers few queries

HOLAP

- Target is to get the best of both worlds.
- HOLAP (Hybrid OLAP) allow co-existence of pre-built MOLAP cubes alongside relational OLAP or ROLAP structures.

DOLAP



Subset of the cube is transferred to the local machine

Cube on the remote server

Local Machine/Server