ON-LINE ANALYTICAL PROCESSING

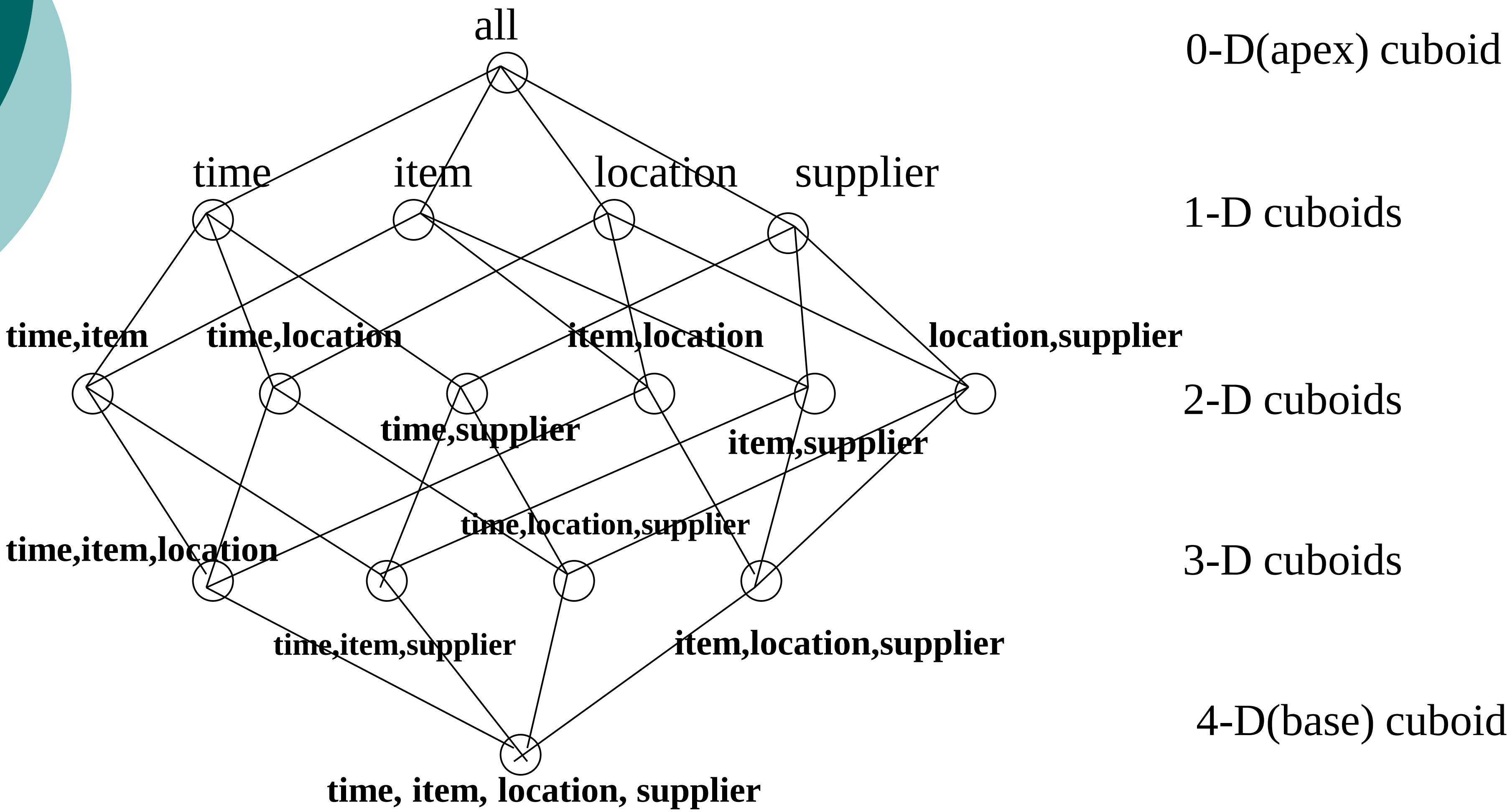
Dr. Yashvardhan Sharma

Department of Computer Science & Information Systems

BITS, Pilani

L12

Lattice of Cuboids



all

0-D(apex) cuboid

time item location supplier

1-D cuboids

time,item time,location item,location location,supplier

time,supplier item,supplier

2-D cuboids

time,location,supplier

time,item,location

3-D cuboids

time,item,supplier item,location,supplier

4-D(base) cuboid

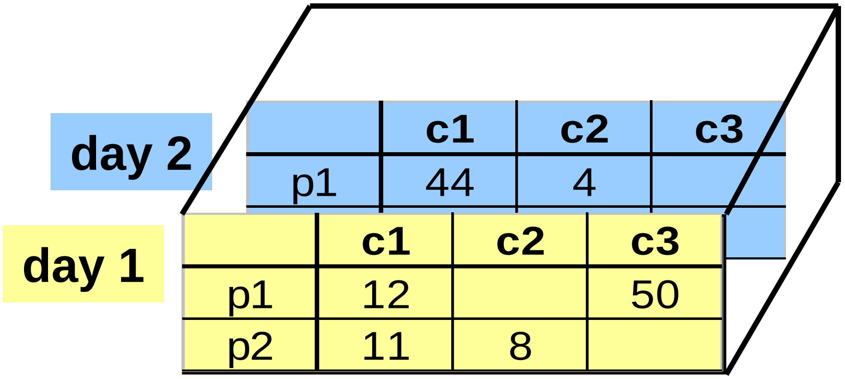
time, item, location, supplier

CUBE

Fact table view:

sale	prodId	storeId	date	amt
	p1	c1	1	12
	p2	c1	1	11
	p1	c3	1	50
	p2	c2	1	8
	p1	c1	2	44
	p1	c2	2	4

Multi-dimensional cube:



day 2

	c1	c2	c3
p1	44	4	

day 1

	c1	c2	c3
p1	12		50
p2	11	8	

dimensions = 3

Aggregates

- Add up amounts for day 1
- In SQL: SELECT sum(amt) FROM SALE
  WHERE date = 1

sale	prodId	storeId	date	amt
	p1	c1	1	12
	p2	c1	1	11
	p1	c3	1	50
	p2	c2	1	8
	p1	c1	2	44
	p1	c2	2	4

→ 81

Aggregates

- Add up amounts by day
- In SQL: SELECT date, sum(amt) FROM SALE
  GROUP BY date

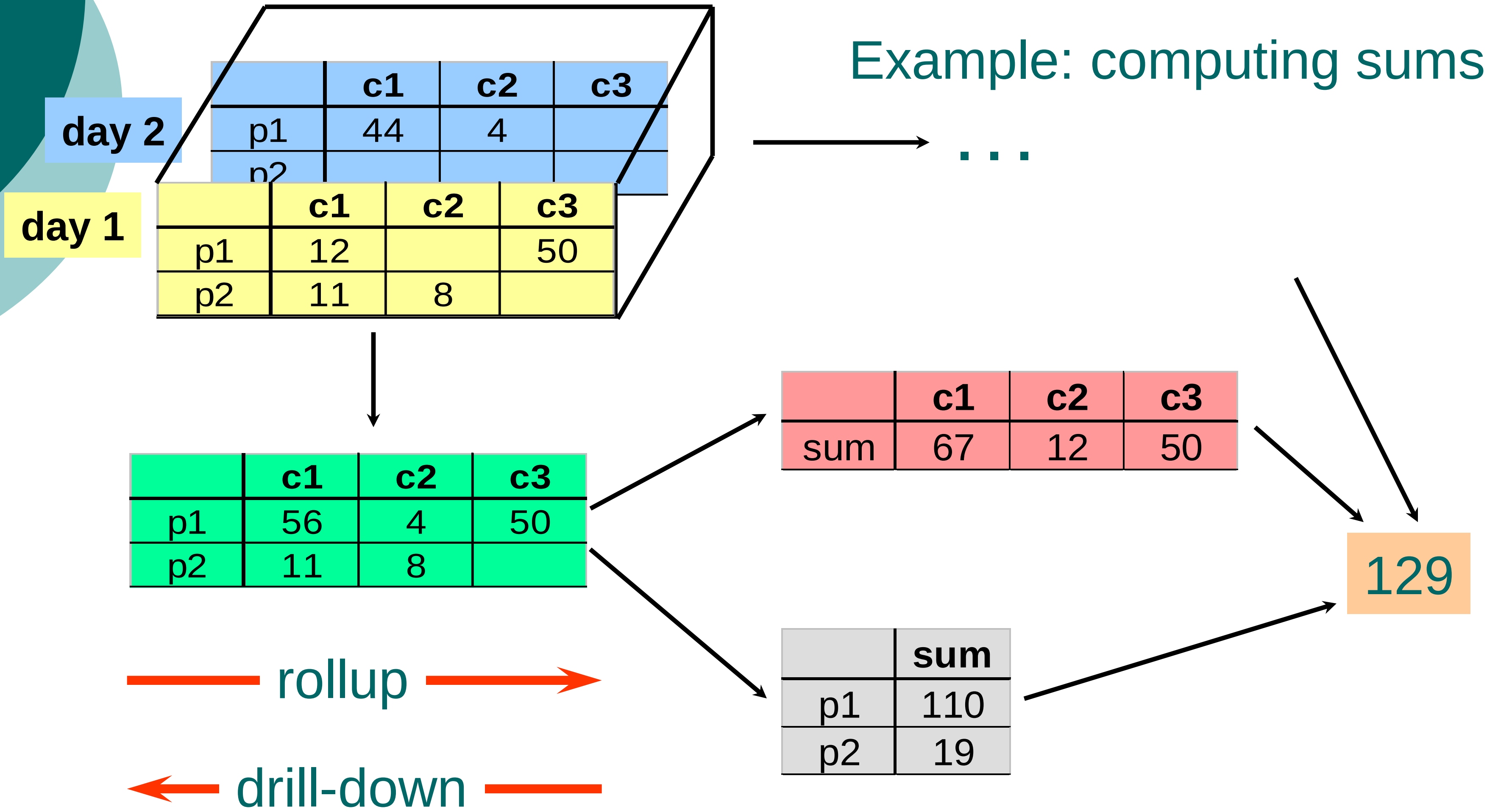
sale	prodId	storeId	date	amt
	p1	c1	1	12
	p2	c1	1	11
	p1	c3	1	50
	p2	c2	1	8
	p1	c1	2	44
	p1	c2	2	4

ans	date	sum
	1	81
	2	48

Aggregates

- Operators: sum, count, max, min, median, ave
- "Having" clause
- Using dimension hierarchy
  - average by region (within store)
  - maximum by month (within date)

Cube Aggregation



Example: computing sums

. . .

day 2

	c1	c2	c3
p1	44	4	
p2			

day 1

	c1	c2	c3
p1	12		50
p2	11	8	

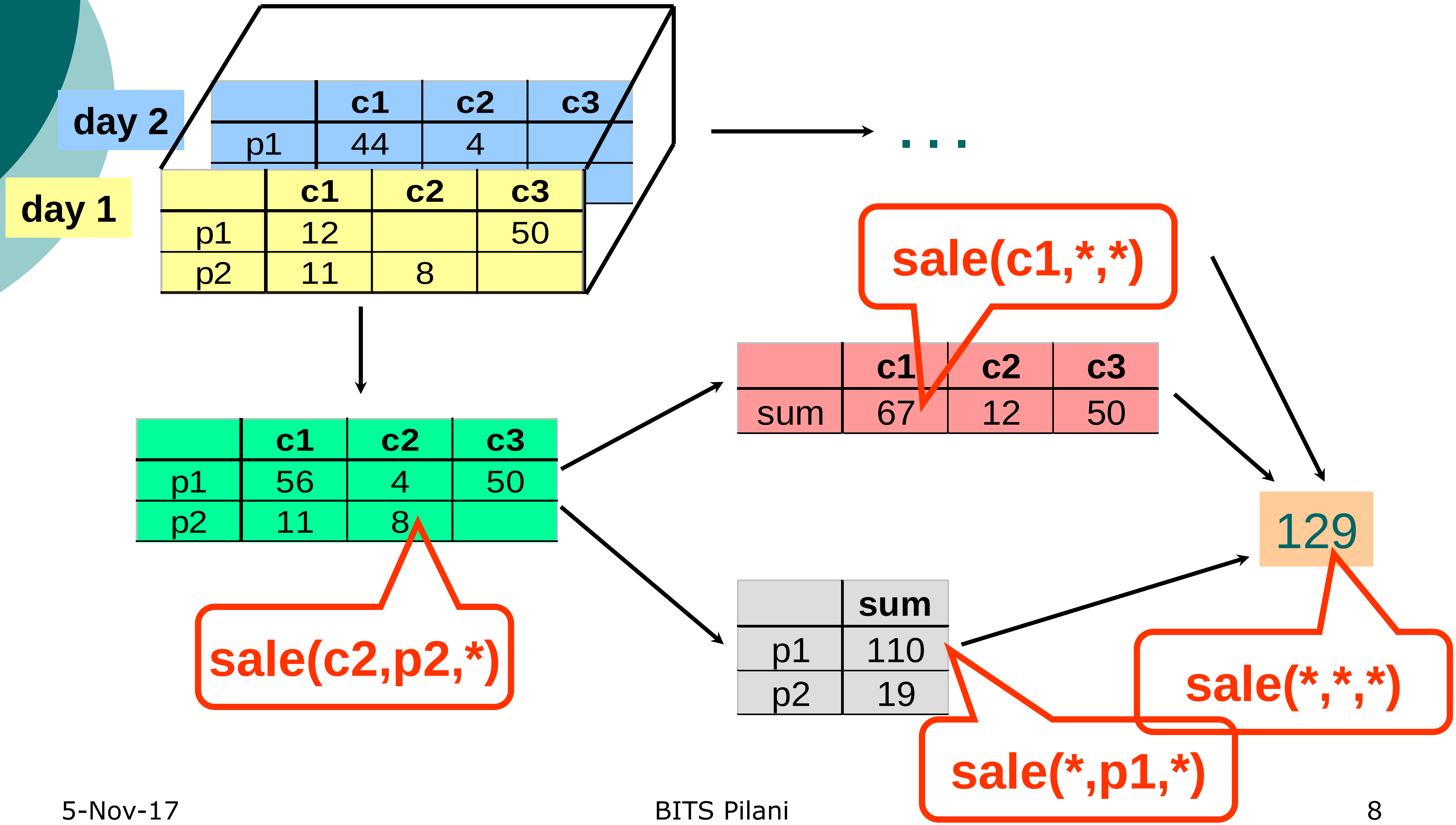
	c1	c2	c3
p1	56	4	50
p2	11	8	

	c1	c2	c3
sum	67	12	50

	sum
p1	110
p2	19

129

Cube Operators



day 2

	c1	c2	c3
p1	44	4	

day 1

	c1	c2	c3
p1	12		50
p2	11	8	

	c1	c2	c3
p1	56	4	50
p2	11	8	

	c1	c2	c3
sum	67	12	50

	sum
p1	110
p2	19

129

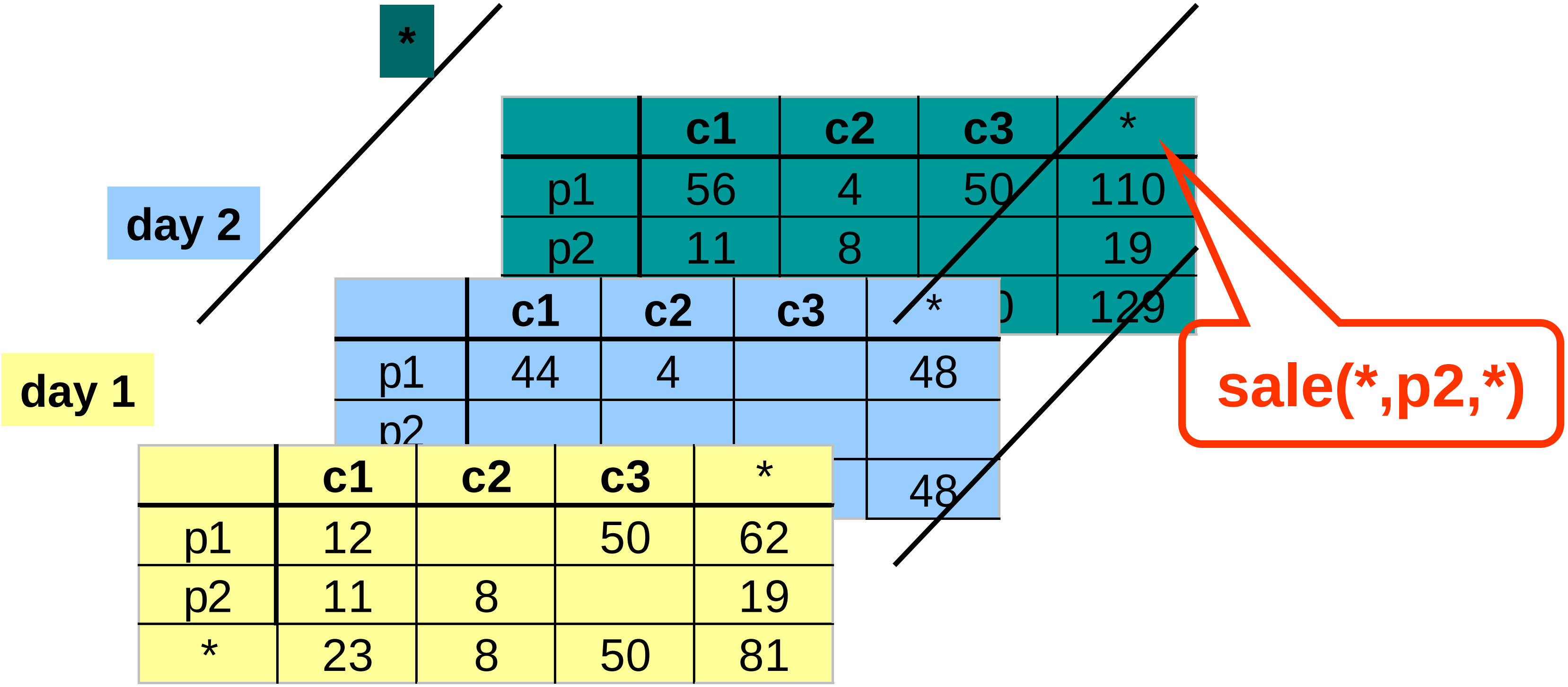
sale(c1,*,*)

sale(c2,p2,*)

sale(*,p1,*)

sale(*,*,*)

Extended Cube



*

day 2

day 1

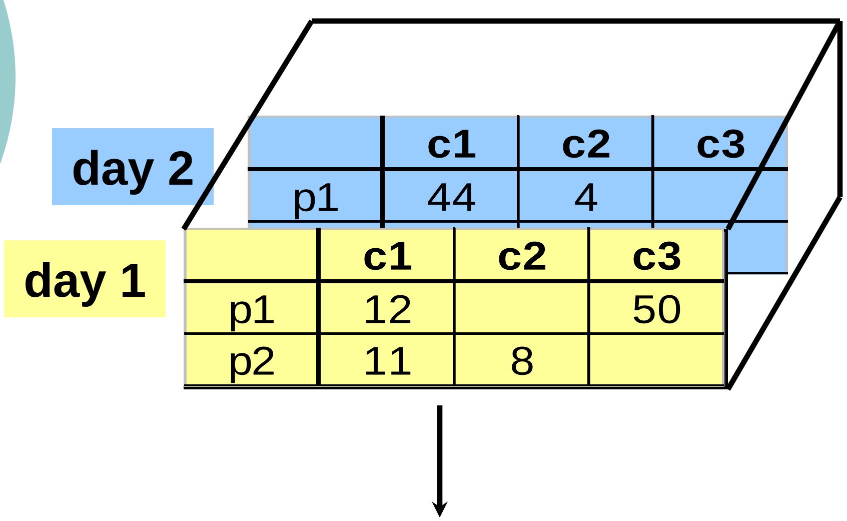
	c1	c2	c3	*
p1	56	4	50	110
p2	11	8		19
			0	129

	c1	c2	c3	*
p1	44	4		48
p2				
				48

	c1	c2	c3	*
p1	12		50	62
p2	11	8		19
*	23	8	50	81

sale(*,p2,*)

Aggregation Using Hierarchies



day 2

	c1	c2	c3
p1	44	4	

day 1

	c1	c2	c3
p1	12		50
p2	11	8	

	region A	region B
p1	56	54
p2	11	8

customer
|
region
|
country

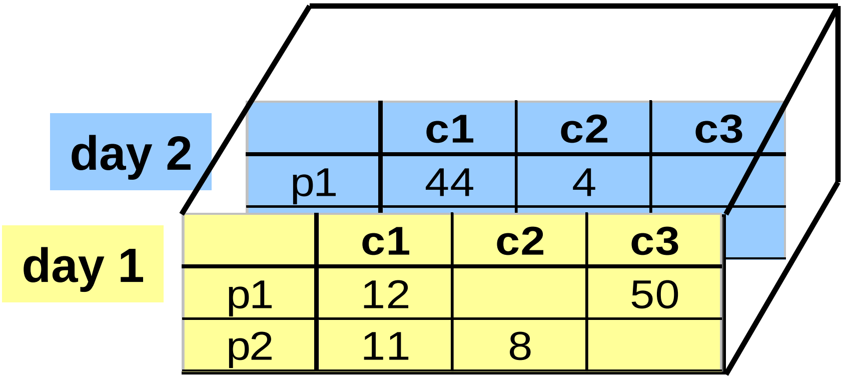
(customer c1 in Region A;
customers c2, c3 in Region B)

Pivoting

Fact table view:

sale	prodId	storeId	date	amt
	p1	c1	1	12
	p2	c1	1	11
	p1	c3	1	50
	p2	c2	1	8
	p1	c1	2	44
	p1	c2	2	4

Multi-dimensional cube:



day 2

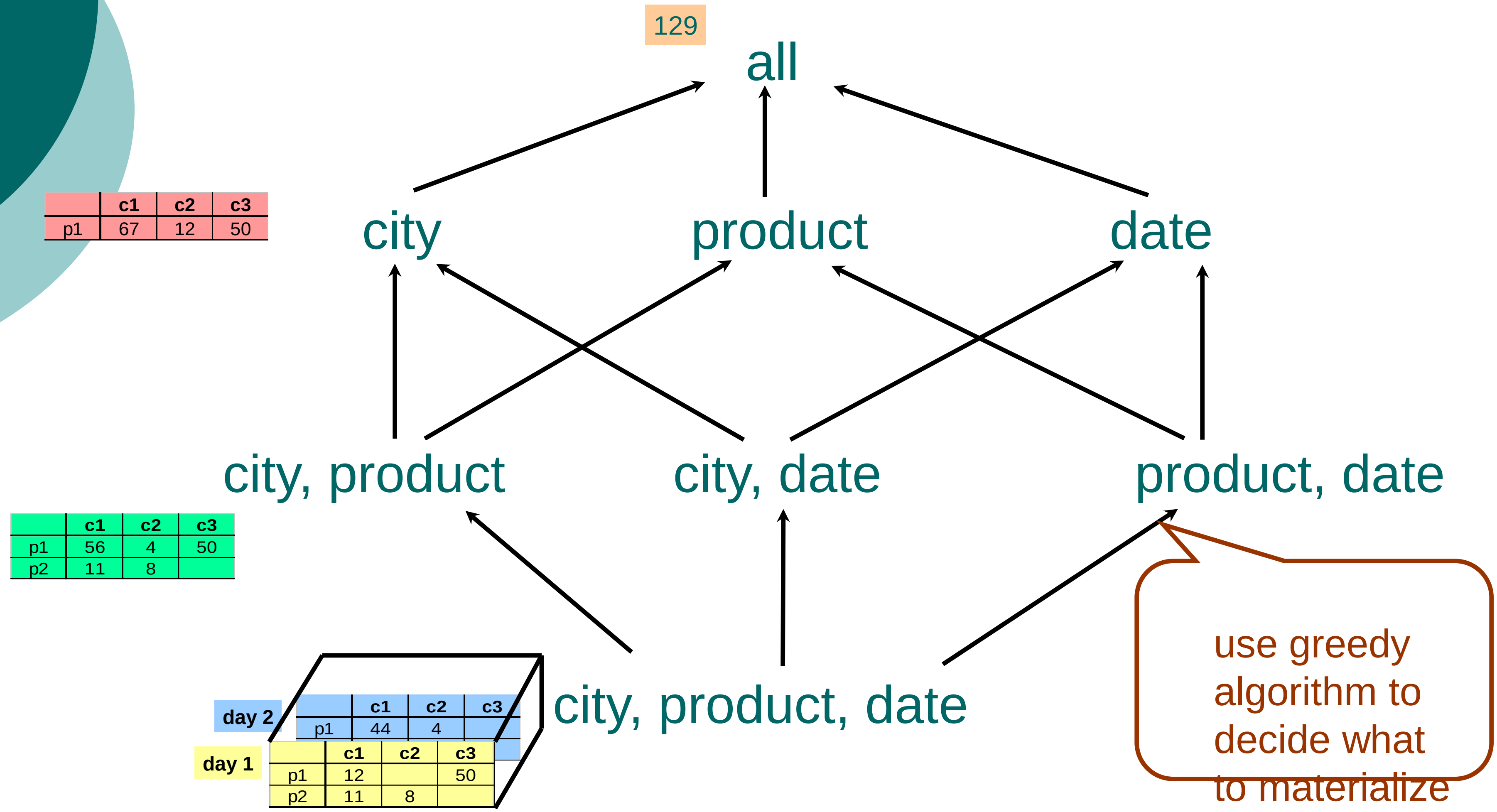
	c1	c2	c3
p1	44	4	

day 1

	c1	c2	c3
p1	12		50
p2	11	8	

	c1	c2	c3
p1	56	4	50
p2	11	8	

Cube Aggregates Lattice



129

all

city

product

date

city, product

city, date

product, date

city, product, date

	c1	c2	c3
p1	67	12	50

	c1	c2	c3
p1	56	4	50
p2	11	8	

day 2

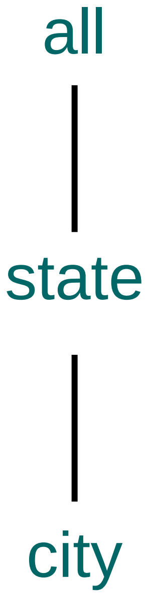
	c1	c2	c3
p1	44	4	

day 1

	c1	c2	c3
p1	12		50
p2	11	8	

use greedy algorithm to decide what to materialize

Dimension Hierarchies



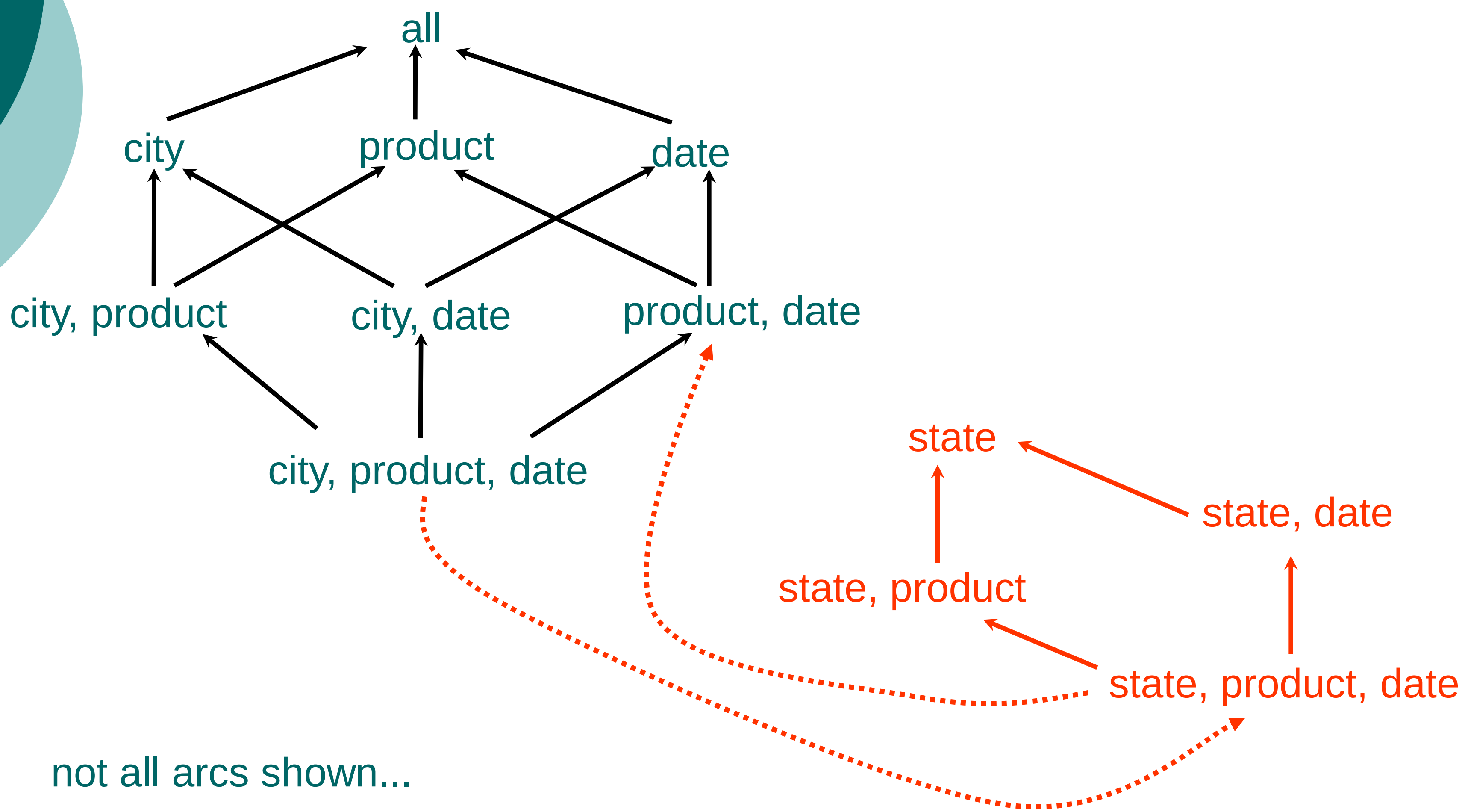
all

state

city

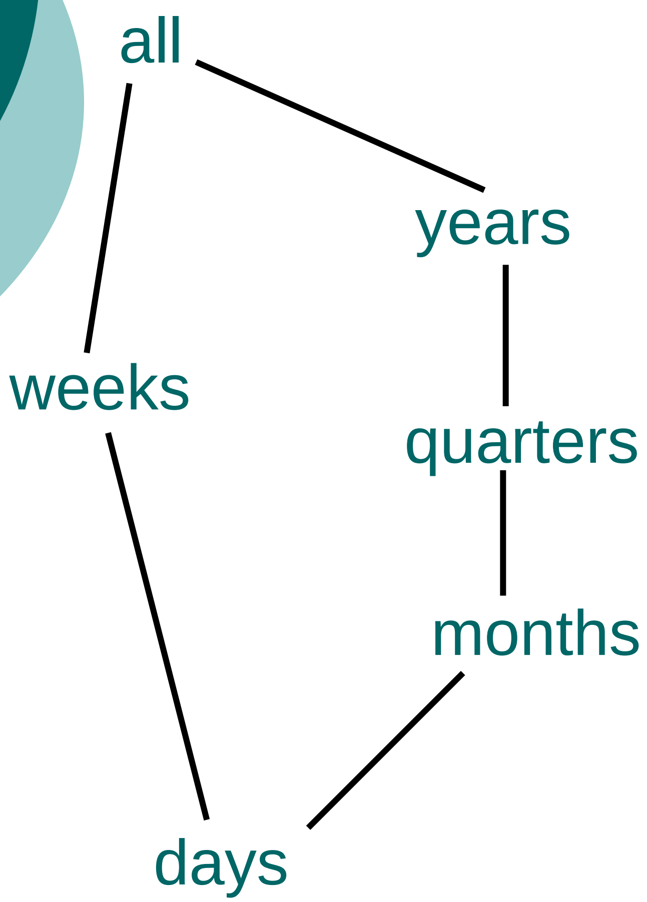
cities	city	state
	c1	CA
	c2	NY

Dimension Hierarchies



not all arcs shown...

Interesting Hierarchy



all

years

weeks

quarters

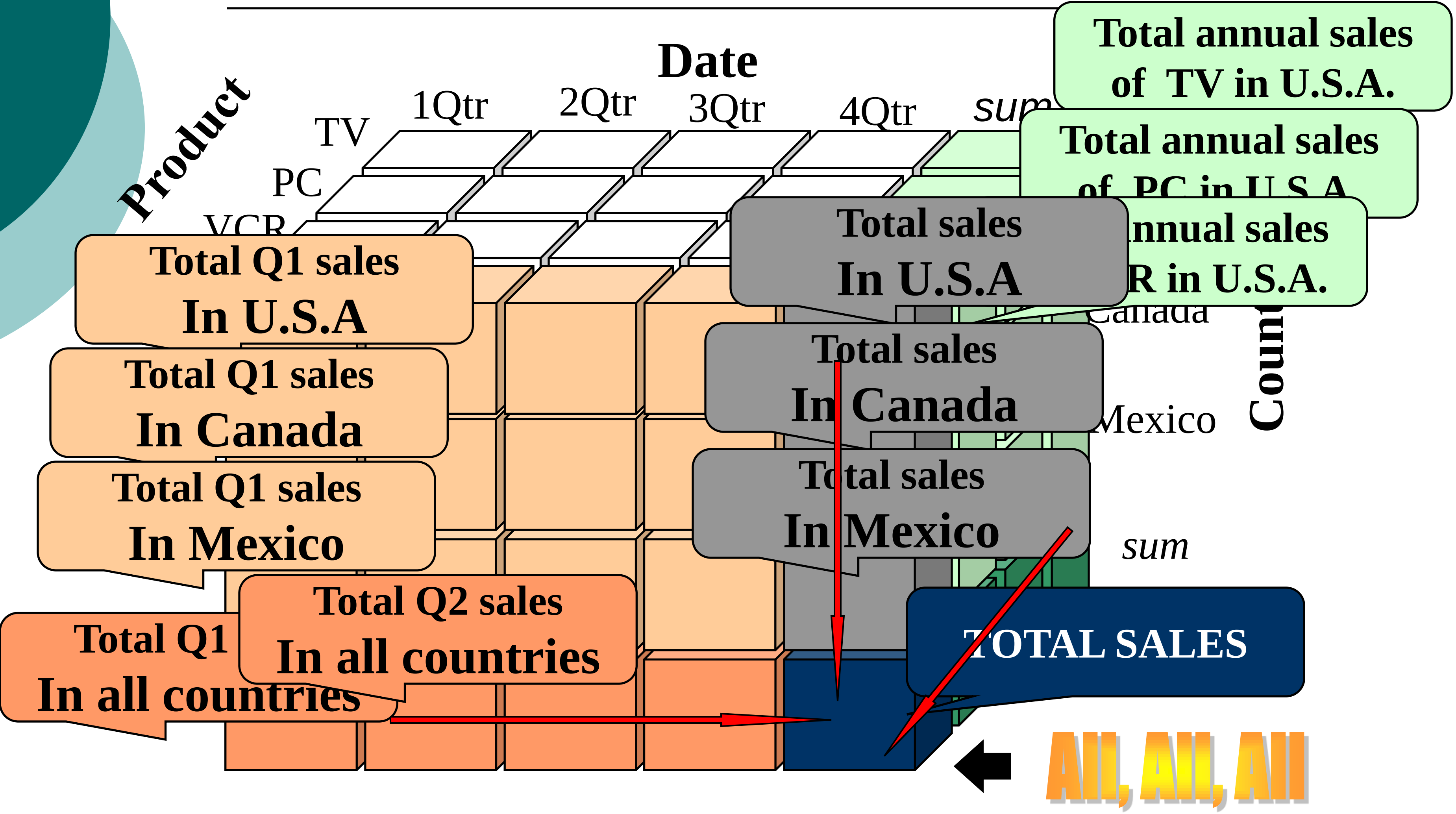
months

days

time	day	week	month	quarter	year
	1	1	1	1	2000
	2	1	1	1	2000
	3	1	1	1	2000
	4	1	1	1	2000
	5	1	1	1	2000
	6	1	1	1	2000
	7	1	1	1	2000
	8	2	1	1	2000

conceptual dimension table

SAMPLE CUBE



Total annual sales of TV in U.S.A.

Total annual sales of PC in U.S.A

nnual sales R in U.S.A.

Total sales In U.S.A

Total sales In Canada

Total sales In Mexico

Total Q1 sales In U.S.A

Total Q1 sales In Canada

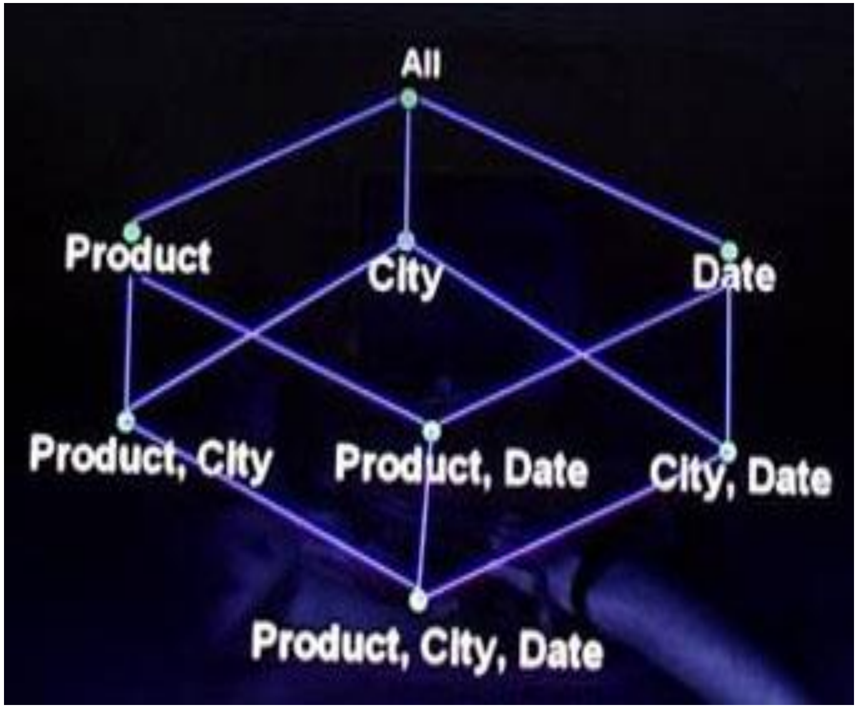
Total Q1 sales In Mexico

Total Q1 In all countries

Total Q2 sales In all countries

TOTAL SALES

Suppose if we have 3 dimensions product, City and Date. We consider these as our 3 elements which are nothing but the perspectives or dimensions. If we take these, then we have 3 sets of 2 element combinations {Product, City}, {Product, Date} and also {Date, City} and also contains the 3 element set {Product, City, Date} which forms a lattice of cuboids. Once we have 2 dimensional computations we can as well compute total sales for each product over all cities and over all dates. Using one dimensional cuboid we can compute grand sales over all products, over all cities and over all dates. If you observe the 'all' the grand sales and the base cuboid, the base cuboid is the greatest lower bound and apex cuboid is the least upper bound.



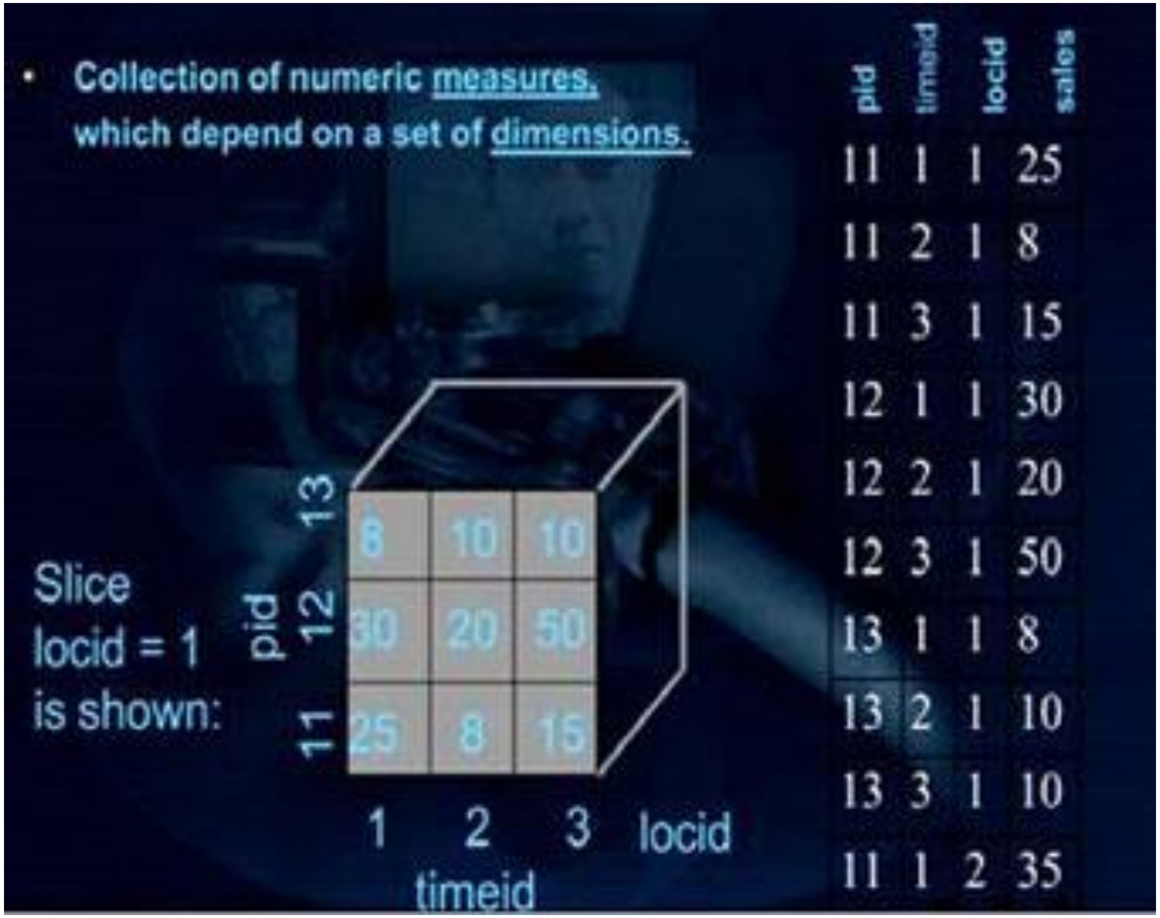
That is why a cube is nothing but a collection of cuboids, and each cuboid is nothing but a aggregation. If we integrate one dimensional, two dimensional and three dimensional aggregations then that is a data cube.

Conceptual vs. Actual

- The "cube" is a logical way of visualizing the data in an OLAP setting
- Not how the data is actually represented on disk
- Two ways of storing data:
  - ROLAP: Relational OLAP
  - MOLAP: Multidimensional OLAP

OLAP & CUBE

- Construction of the data cube is key to the operation of OLAP
- The computation process creates a set of aggregates on the various dimensions of the data
- The CUBE operator



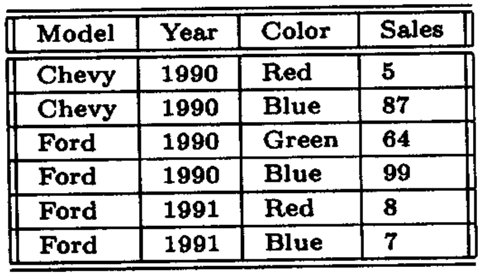
- Collection of numeric measures, which depend on a set of dimensions.

Slice locid = 1 is shown:

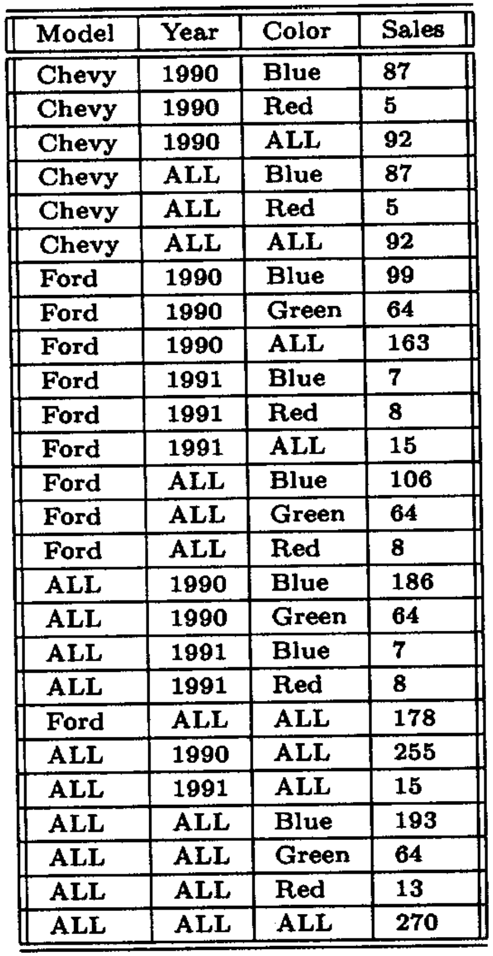
pid			
13	8	10	10
12	30	20	50
11	25	8	15
	1	2	3

pid	timeid	locid	sales
11	1	1	25
11	2	1	8
11	3	1	15
12	1	1	30
12	2	1	20
12	3	1	50
13	1	1	8
13	2	1	10
13	3	1	10
11	1	2	35

An example of the CUBE Operator



Model	Year	Color	Sales
Chevy	1990	Red	5
Chevy	1990	Blue	87
Ford	1990	Green	64
Ford	1990	Blue	99
Ford	1991	Red	8
Ford	1991	Blue	7



Model	Year	Color	Sales
Chevy	1990	Blue	87
Chevy	1990	Red	5
Chevy	1990	ALL	92
Chevy	ALL	Blue	87
Chevy	ALL	Red	5
Chevy	ALL	ALL	92
Ford	1990	Blue	99
Ford	1990	Green	64
Ford	1990	ALL	163
Ford	1991	Blue	7
Ford	1991	Red	8
Ford	1991	ALL	15
Ford	ALL	Blue	106
Ford	ALL	Green	64
Ford	ALL	Red	8
ALL	1990	Blue	186
ALL	1990	Green	64
ALL	1991	Blue	7
ALL	1991	Red	8
Ford	ALL	ALL	178
ALL	1990	ALL	255
ALL	1991	ALL	15
ALL	ALL	Blue	193
ALL	ALL	Green	64
ALL	ALL	Red	13
ALL	ALL	ALL	270

Cube Operator

Locid	City	State	Country
1	Madison	WI	USA
2	Fresno	CA	USA
5	Chennai	TN	India

Pid	Pname	category	Price
11	Lee Jeans	Apparel	25
12	Zord	Toys	18
13	Biro Pen	Stationery	2

Timeid	Date	Month	Year	Holiday
1	10/11/95	Nov	1995	N
2	11/11/96	Nov	1996	N
3	12/11/97	Nov	1997	N

pid	timeid	Locid	sales
11	1	1	25
11	2	1	8
11	3	1	15
12	1	1	30
12	2	1	20
12	3	1	50
13	1	1	8
13	2	1	10
13	3	1	10
11	1	2	35
11	2	2	22
11	3	2	10
12	1	2	26
12	2	2	45
12	3	2	20
13	1	2	20
13	2	2	40
13	3	2	5

Cube Operator

```
Select    T.year, L.state, SUM (sales)
from      Sales S, Times T, Locations L
Where     S.timeid=T.timeid &
     S.locid=L.locid
Group By T.year, L.state
```

```
Select    T.year, SUM (sales)
from      Sales S, Times T
Where     S.timeid=T.timeid
Group By T.year
```

```
Select    L.state, SUM (sales)
from      Sales S, Locations L
Where     S.locid=L.locid
Group By L.state
```

	WI	CA	Total
1995	63	81	144
1996	38	107	145
1997	75	35	110
Total	176	223	399

```
Select     SUM (sales)
from       Sales S, Locations L
Where      S.locid=L.locid
          OR
Select     SUM (sales)
from       Sales S, Time T
Where      S.timeid=T.timeid
```

How many such SQL queries to build cross-tab?

Cube Operator

```
Select    T.year, L.state, SUM (sales)
from      Sales S, Times T, Locations L
Where     S.timeid=T.timeid & S.locid=L.locid
Group By CUBE (T.year, L.state)
```

	WI	CA	Total
1995	63	81	144
1996	38	107	145
1997	75	35	110
Total	176	223	399

T.Year	L.State	SUM(sales)
1995	WI	63
1995	CA	81
1995	All	144
1996	WI	38
1996	CA	107
1996	All	145
1997	WI	75
1997	CA	35
1997	All	110
All	WI	176
All	CA	223
All	All	399

Rollup Operator

```
Select    T.year, L.state, SUM (sales)
from      Sales S, Times T, Locations L
Where     S.timeid=T.timeid & S.locid=L.locid
Group By ROLLUP (T.year, L.state)
```

	WI	CA	Total
1995	63	81	144
1996	38	107	145
1997	75	35	110
Total	176	223	399

T.Year	L.State	SUM(sales)
1995	WI	63
1995	CA	81
1995	All	144
1996	WI	38
1996	CA	107
1996	All	145
1997	WI	75
1997	CA	35
1997	All	110
All	All	399

Find out what the following SQL will generate?

```
Select    T.year, L.state, SUM (sales)
from      Sales S, Times T, Locations L
Where     S.timeid=T.timeid & S.locid=L.locid
Group By ROLLUP (L.state, T.year)
```

The CUBE Operator

- Proposed by Gray et al*
- Effectively involves a series of GROUP-BY operations to aggregate data
- Creates power set on all attributes according to:
  - A measure
  - An aggregator function

* *J. Gray, S. Chaudhuri, A. Bosworth, A. Layman,D. Reichart, M. Venkatrao, F. Pellow and H. Pirahesh.*
Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-totals.
Data Mining and Knowledge Discovery, 1:29-54, 1997.

CUBING Problem

- Problem: this generates a lot of data and work (2^n sets in total, where n is the number of dimensions)
- Solution: optimized algorithms to run faster, consume less memory, and perform fewer I/Os.

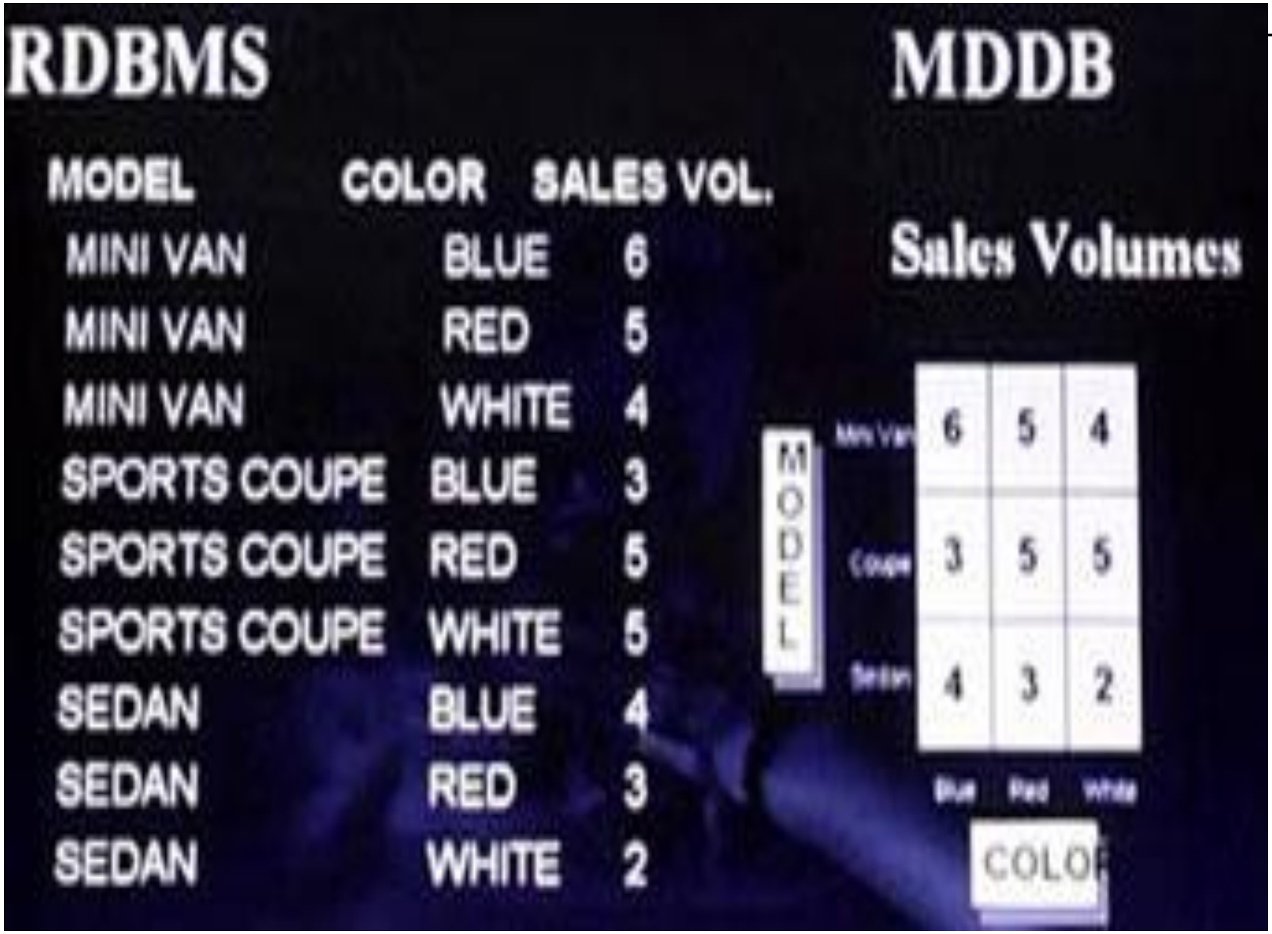
Efficient Computation of Data Cubes

- ROLAP-based cubing algorithms (Agarwal et al'96)
- Array-based cubing algorithm (Zhao et al'97)

S. Agarwal, R. Agrawal, P. M. Deshpande, A.Gupta, J. F. Naughton, R. Ramakrishnan and S.Sarawagi.
On the computation of multidimensional aggregates. In VLDB'96.
Y. Zhao, P. M. Deshpande, and J. F. Naughton.
An array-based algorithm for simultaneous multidimensional aggregates. In SIGMOD'97.

Approaches to OLAP Servers

- It is all about which DBMS you choose to store your data warehouse data
- RDBMS – ROLAP
- MDDB – MOLAP
- BOTH - HOLAP



RDBMS

MODEL	COLOR	SALES VOL.
MINI VAN	BLUE	6
MINI VAN	RED	5
MINI VAN	WHITE	4
SPORTS COUPE	BLUE	3
SPORTS COUPE	RED	5
SPORTS COUPE	WHITE	5
SEDAN	BLUE	4
SEDAN	RED	3
SEDAN	WHITE	2

MDDB

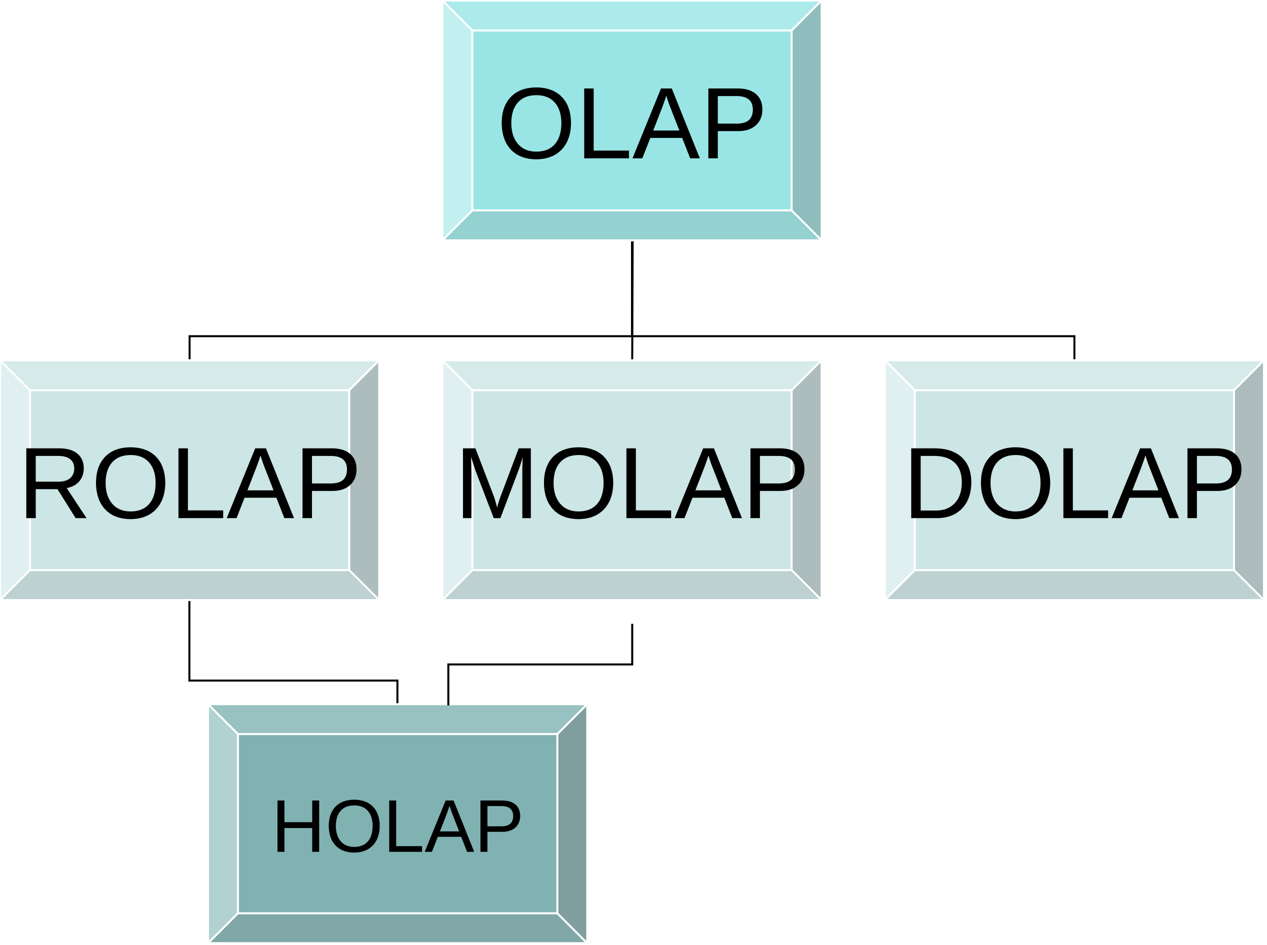
Sales Volumes

MODEL

Mini Van	6	5	4
Coupe	3	5	5
Sedan	4	3	2
	Blue	Red	White

COLOR

OLAP Flavours



OLAP

ROLAP

MOLAP

DOLAP

HOLAP

Approaches to OLAP Servers

Three possibilities for OLAP servers

(1) Relational OLAP (ROLAP)

- Relational and specialized relational DBMS to store and manage warehouse data
- OLAP middleware to support missing pieces

(2) Multidimensional OLAP (MOLAP)

- Array-based storage structures
- Direct access to array data structures

(3) Hybrid OLAP (HOLAP)

- Storing detailed data in RDBMS
- Storing aggregated data in MDBMS
- User access via MOLAP tools

ROLAP

- Special schema design: *star, snowflake*
- Special indexes: bitmap, multi-table join
- Proven technology (relational model, DBMS), tend to outperform specialized MDDB especially on large data sets
- Products
  - IBM DB2, Oracle, Sybase IQ, RedBrick, Informix

ROLAP

- Defines complex, multi-dimensional data with simple model
- Reduces the number of joins a query has to process
- Allows the data warehouse to evolve with relatively low maintenance
- Can contain both detailed and summarized data.
- ROLAP is based on familiar, proven, and already selected technologies.

BUT!!!

- SQL for multi-dimensional manipulation of calculations.

MOLAP

- MDDB: a special-purpose data model
- Facts stored in multi-dimensional arrays
- Dimensions used to index array
- Sometimes on top of relational DB
- Products
  - Pilot, Arbor Essbase, Gentia

MOLAP

- Pre-calculating or pre-consolidating transactional data improves speed.

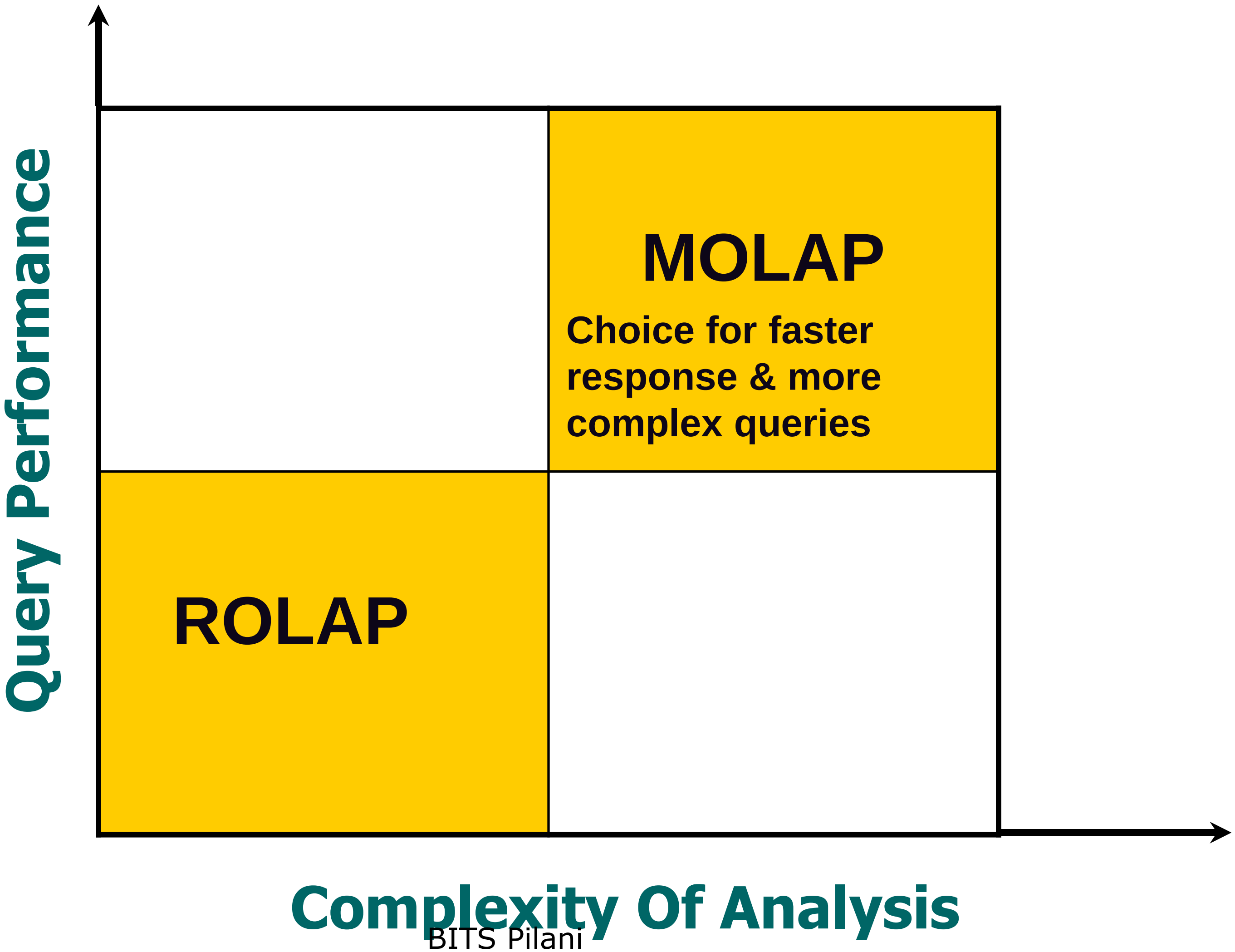
BUT

Fully pre-consolidating incoming data, MDDs require an enormous amount of overhead both in processing time and in storage. An input file of 200MB can easily expand to 5GB

MDDBs are great candidates for the **<** 100GB department data marts.

- With MDDs, application design is essentially the definition of dimensions and calculation rules, while the RDBMS requires that the database schema be a star or snowflake.

MOLAP vs. ROLAP



Query Performance

MOLAP

Choice for faster response & more complex queries

ROLAP

Complexity Of Analysis

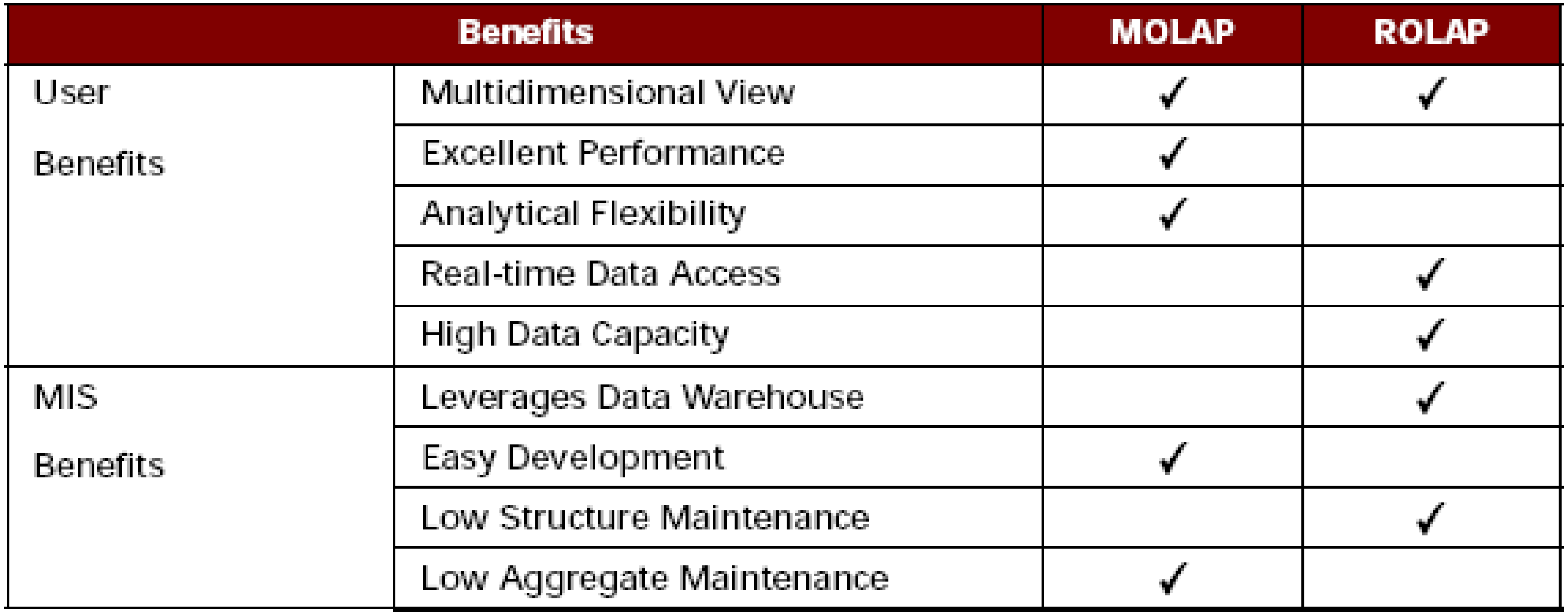
MOLAP vs. ROLAP

	Data Storage	Underlying Technologies	Functions & Features
ROLAP	Data Stored as Relational Tables in DW Detailed & light summary data available Very large data volumes	Use of complex SQL ROLAP engine in analytical server creates data cubes on the fly Multidimensional views by presentation layer	Known environment and availability of many tools Limitations on complex analysis functions Drill-through easy but not drill-across
MOLAP	Data srored as relational tables in DW Summary data kept in MDDBs Moderate data volumes Summary data access from MDDB and detailed data from DW	creation of pre-fabricated cubes by MLAP engine. Proprietary technology to store multidimensional views in arrays. High speed data matrix retrieval Sparse matrix handling techniques to manage data sparsity in summaries	Faster access Large library of functions for complex calculations Easy analysis irrespective of the number of dimensions Excessive drill-down and slice and dice capabilities

HOLAP

HOLAP technologies attempt to combine the advantages of MOLAP and ROLAP. For summary-type information, HOLAP leverages cube technology for faster performance. When detail information is needed, HOLAP can "drill through" from the cube into the underlying relational data.

ROLAP vs. MOLAP



Benefits		MOLAP	ROLAP
User Benefits	Multidimensional View	✓	✓
	Excellent Performance	✓	
	Analytical Flexibility	✓	
	Real-time Data Access		✓
	High Data Capacity		✓
MIS Benefits	Leverages Data Warehouse		✓
	Easy Development	✓	
	Low Structure Maintenance		✓
	Low Aggregate Maintenance	✓	

ROLAP vs. MOLAP

1) Performance:

- How fast will the system appear to the end-user?
- MDD server vendors believe this is a key point in their favor.

2) Data volume and scalability:

- While MDD servers can handle up to 100GB of storage, RDBMS servers can handle hundreds of gigabytes and terabytes.

MOLAP **ROLAP**

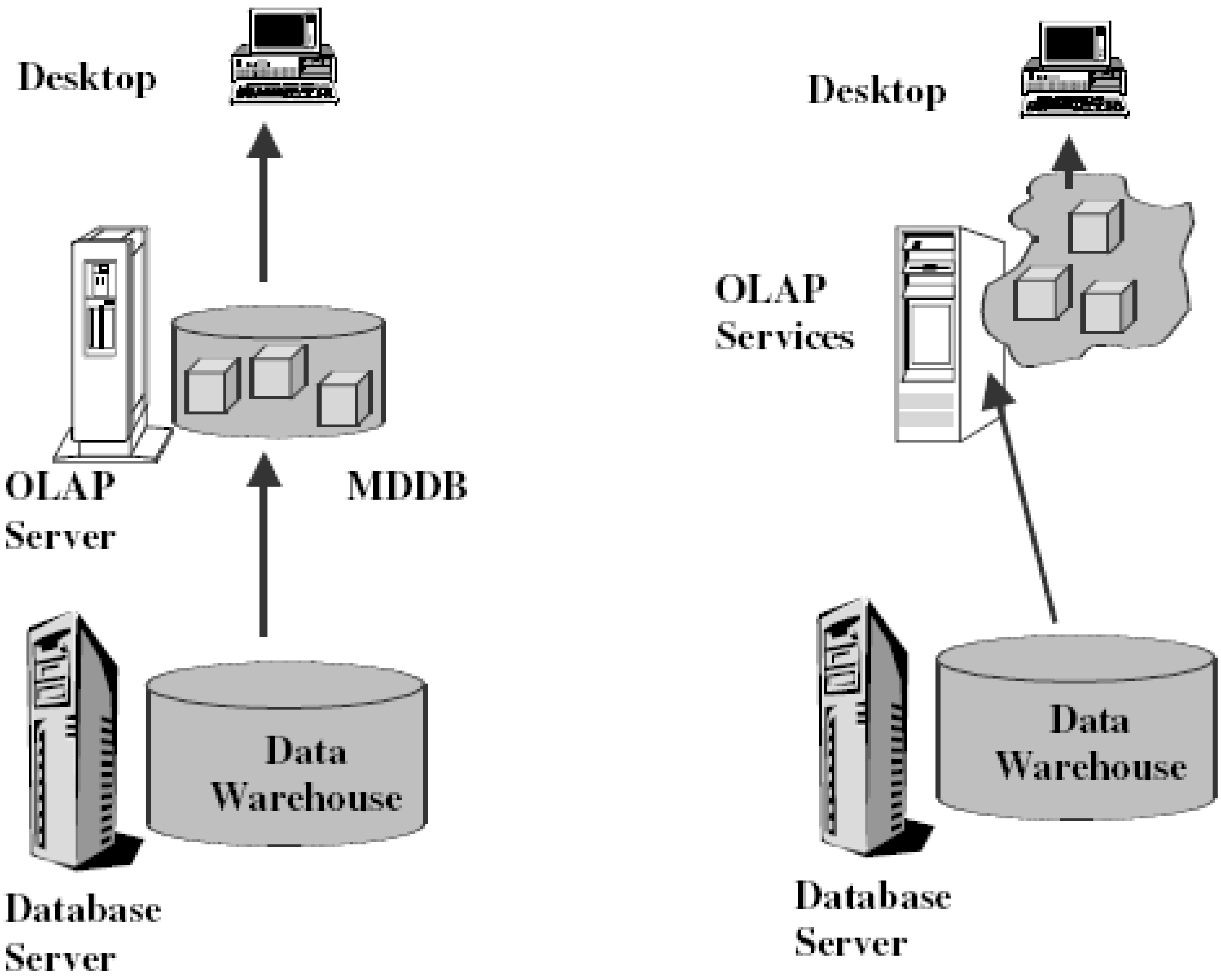
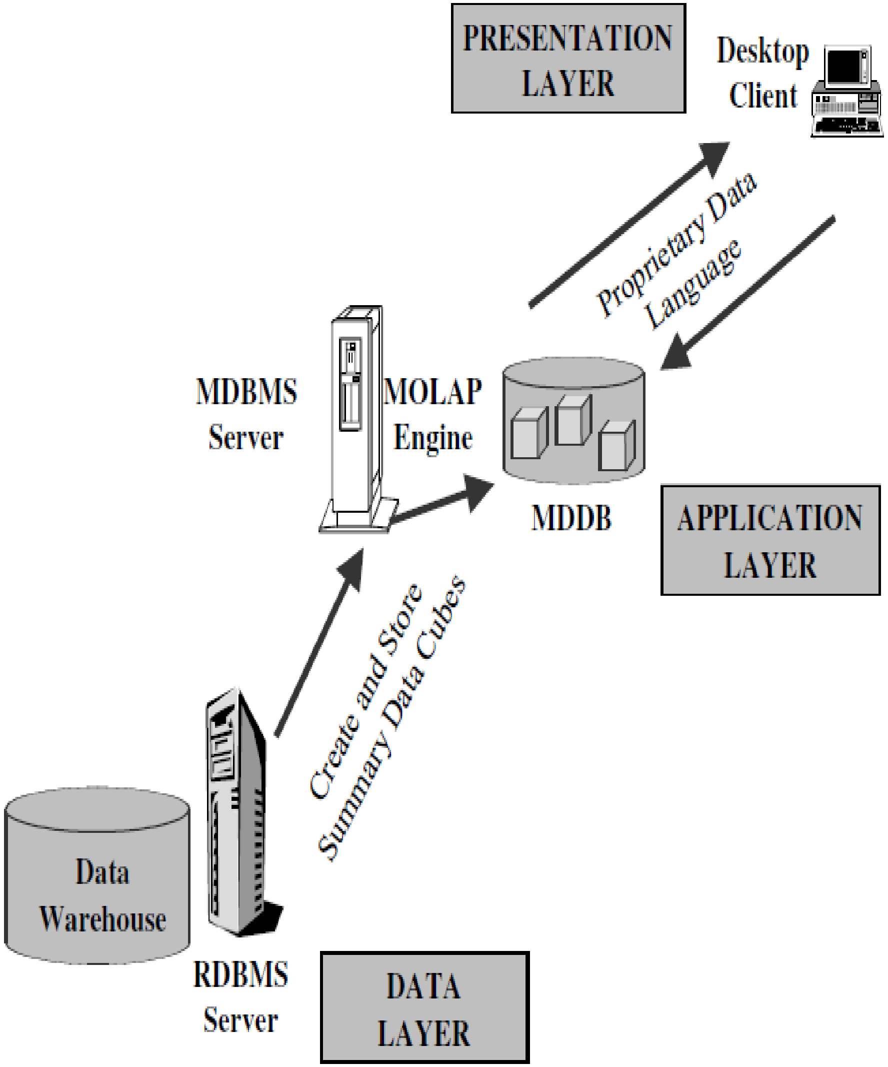


Figure 15-15 OLAP models.

OLAP Models

MOLAP :

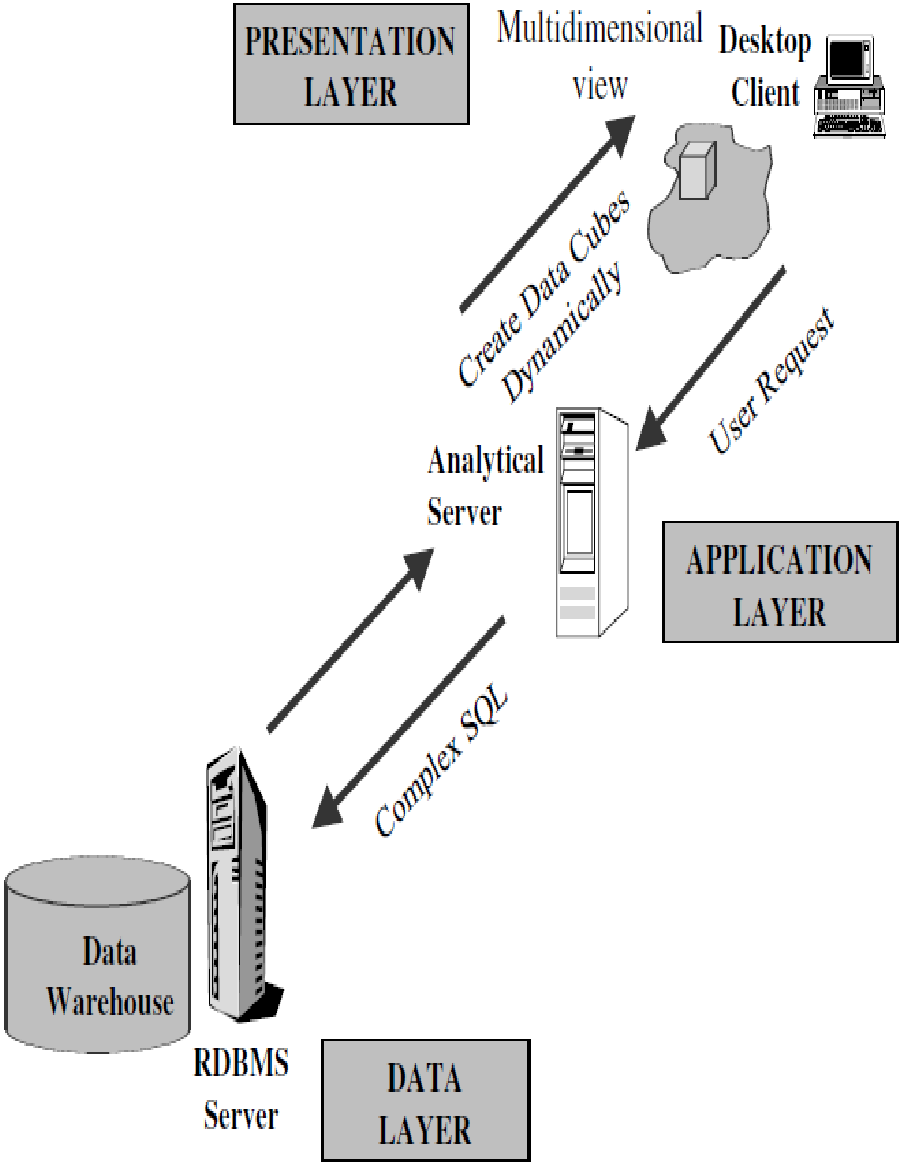
- Multidimensional data storage
- Pre-calculated and prefabricated multi-dimensional data cubes are stored in MDDBs
- MOLAP engine is used to push multidimensional view of the data from MDDBs to user



OLAP Models

ROLAP :

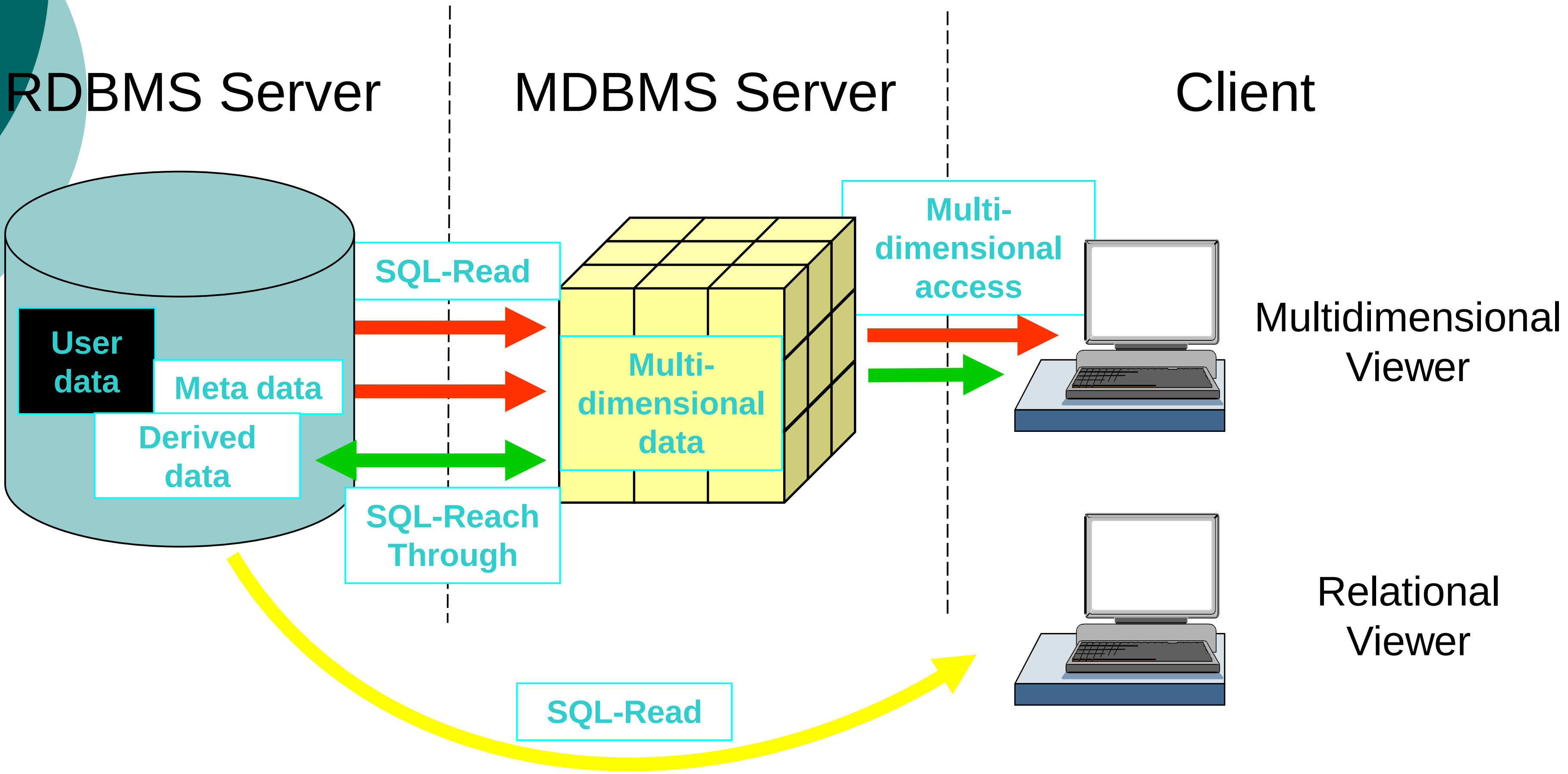
- Use of Relational database
- Complex multi-dimensional view based user queries are transformed into complex SQL queries directed to RDBMS
- Multidimensional view of the data created on the fly by analytical server



Hybrid OLAP - HOLAP

- Best of both worlds
- Storing detailed data in RDBMS
- Storing aggregated data in MDBMS
- User access via MOLAP tools

HOLAP


RDBMS Server
MDBMS Server
Client
User data
Meta data
Derived data
SQL-Read
SQL-Reach Through
Multi-dimensional data
Multi-dimensional access
Multidimensional Viewer
Relational Viewer
SQL-Read

ROLAP, MOPAL, or HOLAP

IF

A. You require write access
B. Your data is under 50 GB
C. Your timetable to implement is 60-90 days
D. Lowest level already aggregated
E. Data access on aggregated level
F. You’re developing a general-purpose application for inventory movement or assets management

THEN

Consider an MDD /MOLAP solution for your data mart

IF

A. Your data is over 100 GB
B. You have a "read-only" requirement
C. Historical data at the lowest level of granularity
D. Detailed access, long-running queries
E. Data assigned to lowest level elements

THEN

Consider an RDBMS/ROLAP solution for your data mart.

IF

A. OLAP on aggregated and detailed data
B. Different user groups
C. Ease of use and detailed data

THEN

Consider an HOLAP for your data mart

Conclusions

- ROLAP: RDBMS -> star/snowflake schema
- MOLAP: MDDB -> Cube structures
- ROLAP or MOLAP: Data models used play major role in performance differences
- MOLAP: for summarized and relatively lesser volumes of data (100GB)
- ROLAP: for detailed and larger volumes of data
- Both storage methods have strengths and weaknesses
- The choice is requirement specific, though currently data warehouses are predominantly built using RDBMSs/ROLAP.
- ***HOLAP is emerging as the OLPA server of choice***