

# Data Warehousing

*Read chapter 13 of Riguzzi et al Sistemi Informativi*

*Slides derived from those by Hector Garcia-Molina*

# What is a Warehouse?

---

- Collection of diverse data
  - subject oriented
  - aimed at executive, decision maker
  - often a copy of operational data
  - with value-added data (e.g., summaries, history)
  - integrated
  - time-varying
  - non-volatile



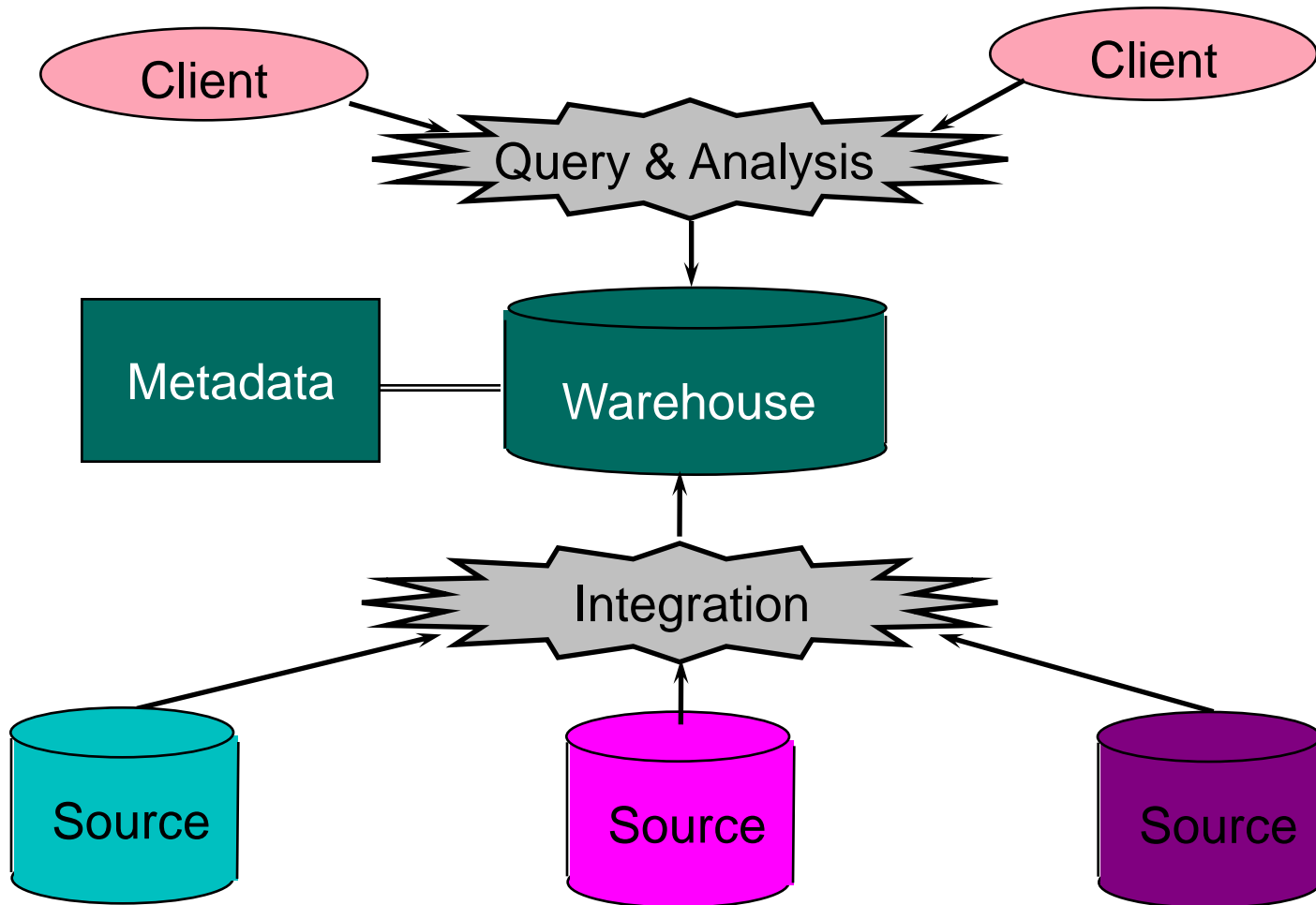
# What is a Warehouse?

---

- Collection of tools
  - gathering data
  - cleansing, integrating, ...
  - querying, reporting, analysis
  - data mining
  - monitoring, administering warehouse

# Warehouse Architecture

---



# Motivating Examples

---

- Forecasting
- Comparing performance of units
- Monitoring, detecting fraud
- Visualization

# OLTP vs. OLAP

---

- OLTP: On Line Transaction Processing
  - Describes processing at operational sites
- OLAP: On Line Analytical Processing
  - Describes processing at warehouse

# OLTP vs. OLAP

---

## OLTP

- Mostly updates
- Many small transactions
- Mb-Gb of data
- Raw data
- Clerical users
- Up-to-date data
- Consistency, recoverability critical

## OLAP

- Mostly reads
- Queries long, complex
- Gb-Tb of data
- Summarized, consolidated data
- Decision-makers, analysts as users

# Data Marts

---

- Smaller warehouses
- Spans a part of an organization
  - e.g., marketing (customers, products, sales)
- Do not require enterprise-wide consensus
  - but long term integration problems?



# Warehouse Models & Operators

---

- Data Models
  - relations
  - stars & snowflakes
  - cubes
- Operators
  - slice & dice
  - roll-up, drill down
  - pivoting
  - other

# Star

product	<u>prold</u>	name	price
	p1	bolt	10
	p2	nut	5

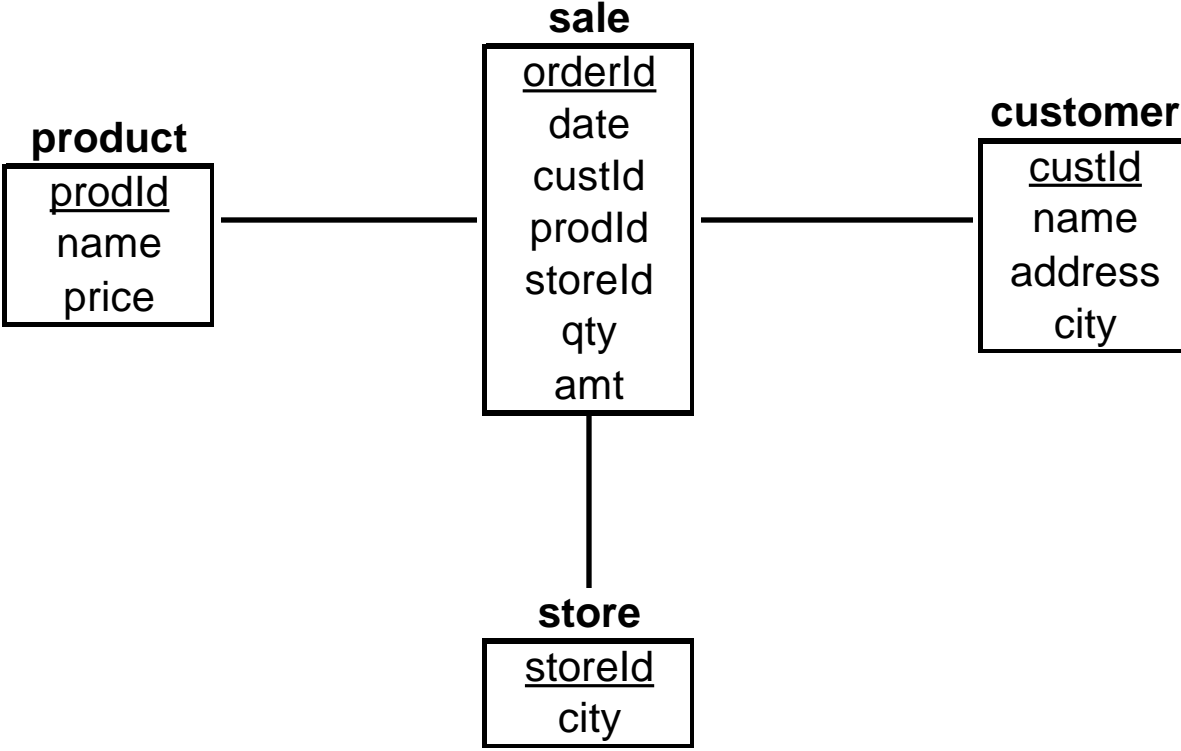
store	<u>storeld</u>	city
	c1	nyc
	c2	sfo
	c3	la

sale	oderld	date	custld	prold	storeld	qty	amt
	o100	1/7/97	53	p1	c1	1	12
	o102	2/7/97	53	p2	c1	2	11
	105	3/8/97	111	p1	c3	5	50

customer	<u>custld</u>	name	address	city
	53	joe	10 main	sfo
	81	fred	12 main	sfo
	111	sally	80 willow	la

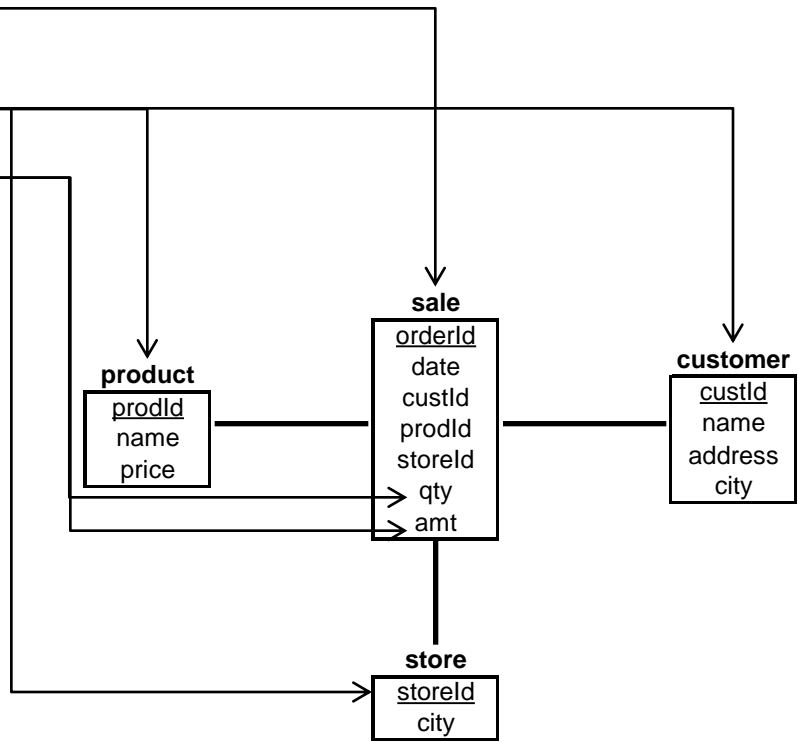
# Star Schema

---

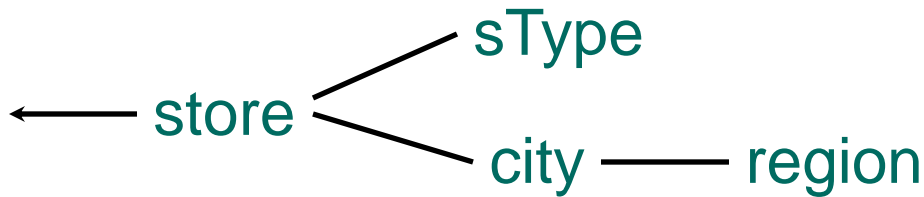


# Terms

- Fact table
- Dimension tables
- Measures



# Dimension Hierarchies



store	<u>storeld</u>	cityld	tld	mgr
	s5	sfo	t1	joe
	s7	sfo	t2	fred
	s9	la	t1	nancy

sType	<u>tld</u>	size	location
	t1	small	downtown
	t2	large	suburbs

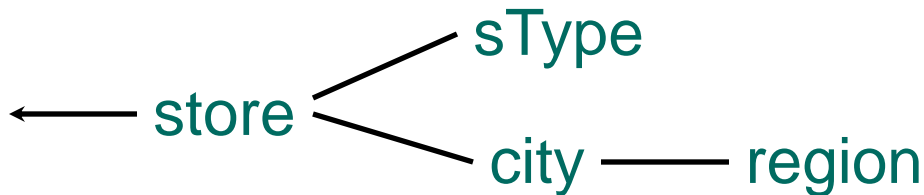
city	<u>cityld</u>	pop	regld
	sfo	1M	north
	la	5M	south

→ snowflake schema

region	<u>regld</u>	name
	north	cold region
	south	warm region

# Snowflake Schema

Sometimes not normalized: not in third normal form



store	<u>storeld</u>	cityld	tld	mgr
	s5	sfo	t1	joe
	s7	sfo	t2	fred
	s9	la	t1	nancy

sType	<u>tld</u>	size	location
	t1	small	downtown
	t2	large	suburbs

city	<u>cityld</u>	pop	regld	name
	sfo	1M	north	cold region
	la	5M	south	warm region

# Cube

---

Fact table view:

sale	prold	storeld	amt
	p1	c1	12
	p2	c1	11
	p1	c3	50
	p2	c2	8



Multi-dimensional cube:

	c1	c2	c3
p1	12		50
p2	11	8	

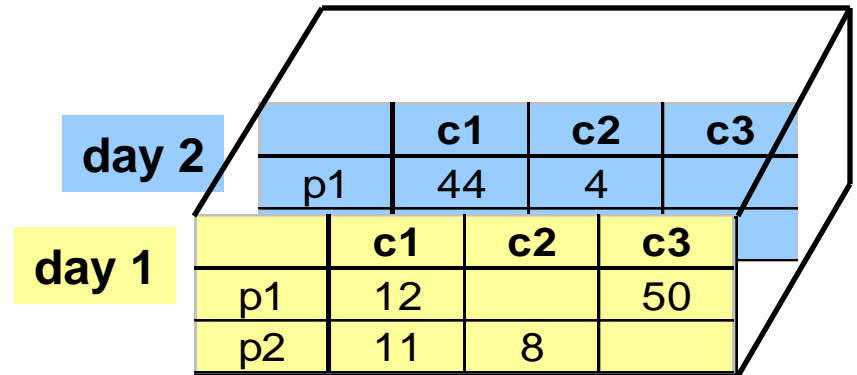
dimensions = 2

# 3-D Cube

Fact table view:

sale	prold	storeld	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:



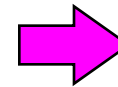
dimensions = 3



# Aggregates

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

sale	prodlid	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

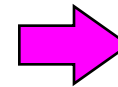
— rollup —→

← drill-down —

# Aggregates

- Add up amounts for days 1 and 2
- In SQL: `SELECT sum(amt) FROM SALE WHERE date >= 1 AND date <=2`

sale	prodlid	storeid	date	amt
	p1	c1	1	12
	p2	c1	1	11
	p1	c3	2	40
	p2	c2	2	8
	p1	c1	3	44
	p1	c2	3	4



71

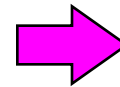
— rollup —→

← drill-down —

# Aggregates

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

sale	prodl	storeld	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

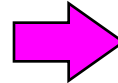
— rollup —→

← drill-down —

# Another Example

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

sale	prodl	storeld	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



sale	prodl	date	amt
	p1	1	62
	p2	1	19
	p1	2	48

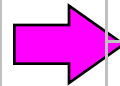
— rollup —→

← drill-down —

# Another Example

- Add up amounts by month
- In SQL: `SELECT month, prodlid, storeld, sum(amt)`  
`FROM SALE JOIN DATE GROUP BY month, prodlid, storeld`

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



sale	prodlid	storeld	month	amt
	p1	c1	sep	23
	p2	c2	sep	58
	p1	c3	oct	48

rollup →

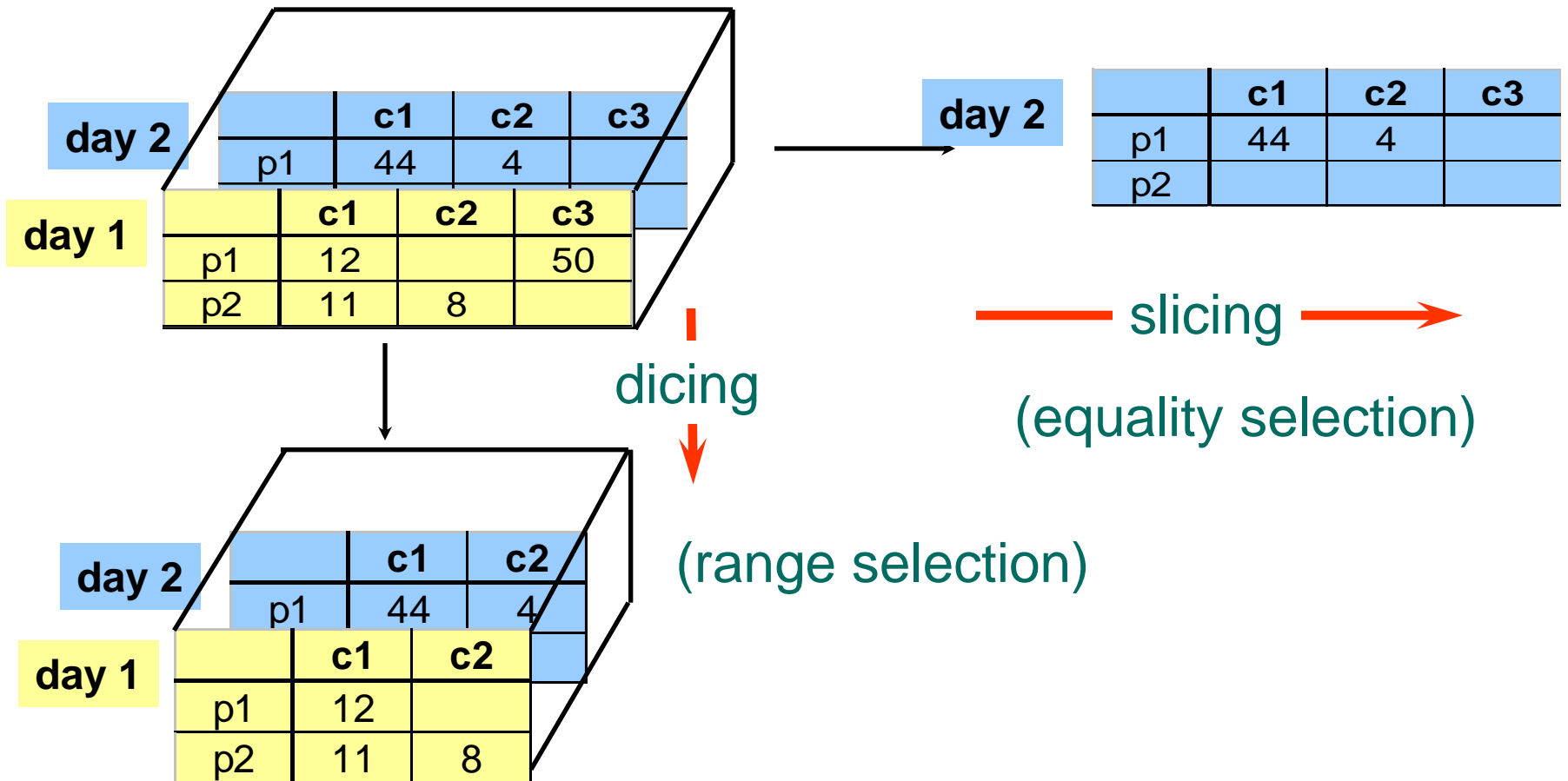
← drill-down

# Aggregates

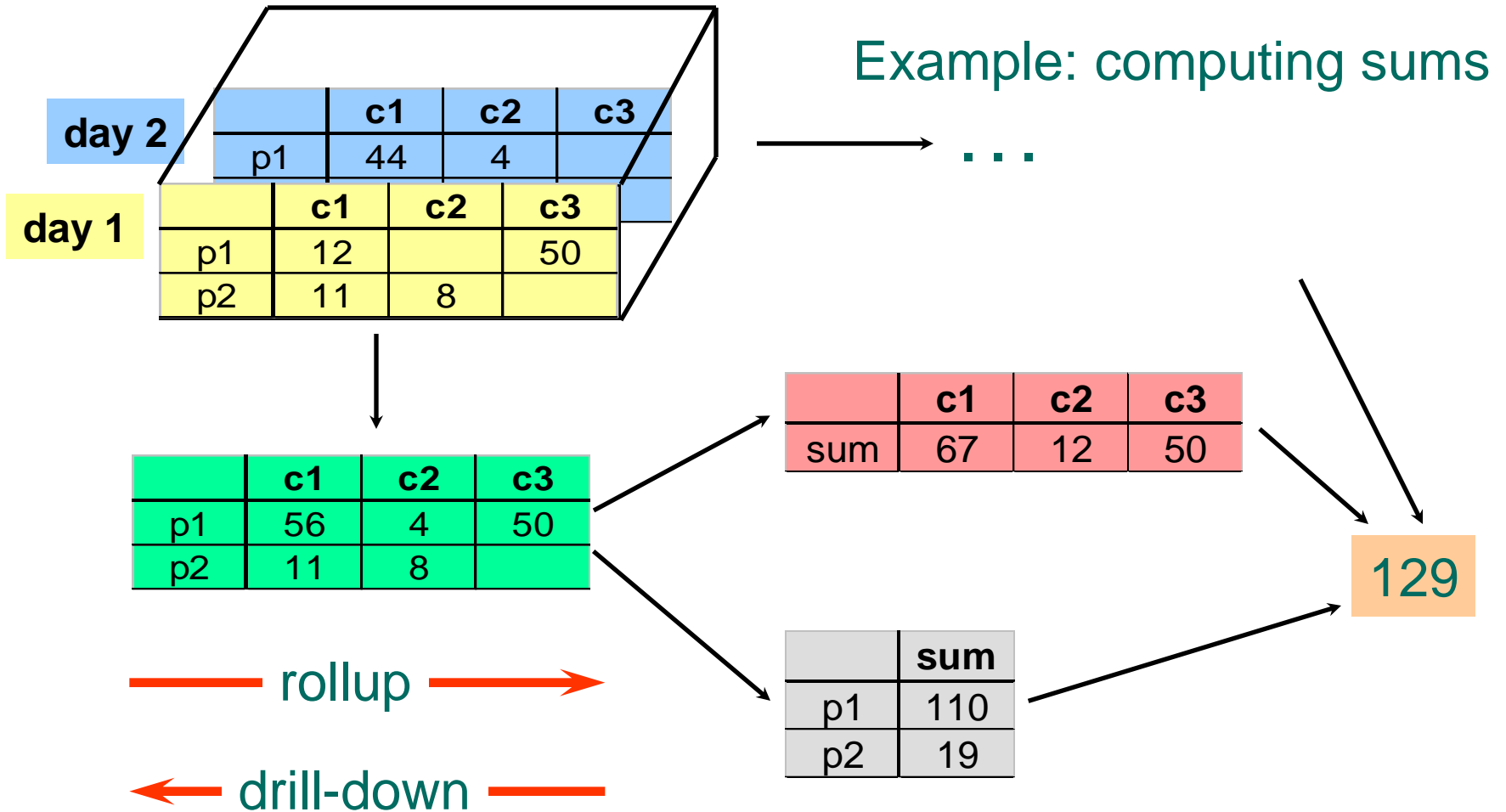
---

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

# Operations on the Cube



# Cube Aggregation





# Aggregation Using Hierarchies

		day 2		
		c1	c2	c3
day 1	p1	44	4	
		c1	c2	c3
	p1	12		50
	p2	11	8	

rollup



drill-down



	region A	region B
p1	56	54
p2	11	8



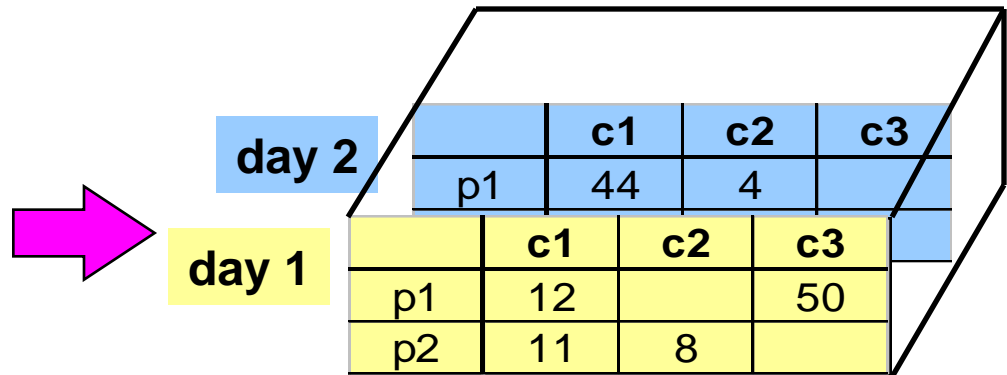
(customer c1 in Region A; customers c2, c3 in Region B)

# Pivoting

Fact table view:

sale	prodlid	storeld	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:



# Query & Analysis Tools

---

- Query Building
- Report Writers (comparisons, growth, graphs,...)
- Spreadsheet Systems
- Web Interfaces
- Data Mining

# Implementing a Warehouse

---

- *Monitoring*: Sending data from sources
- *Integrating*: Loading, cleansing,...
- *Processing*: Query processing, indexing, ...
- *Managing*: Metadata, tools

# Monitoring

---

- Source Types: relational, flat files, IMS, VSAM, WWW, news-wire, ...
- Incremental vs. Refresh

customer	<u>id</u>	name	address	city
	53	joe	10 main	sfo
	81	fred	12 main	sfo
	<b>111</b>	<b>sally</b>	<b>80 willow</b>	<b>la</b>



# Monitoring Techniques

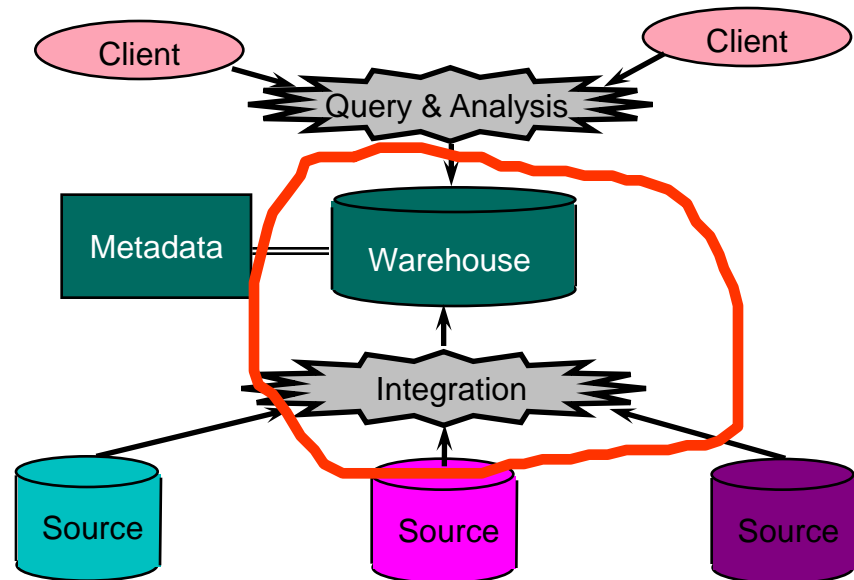
---

- Periodic snapshots
- Polling (queries to source)
- Database triggers
- Log shipping
- Data shipping (replication service)
- Transaction shipping
- Application level monitoring

➔ Advantages & Disadvantages!!

# Integration

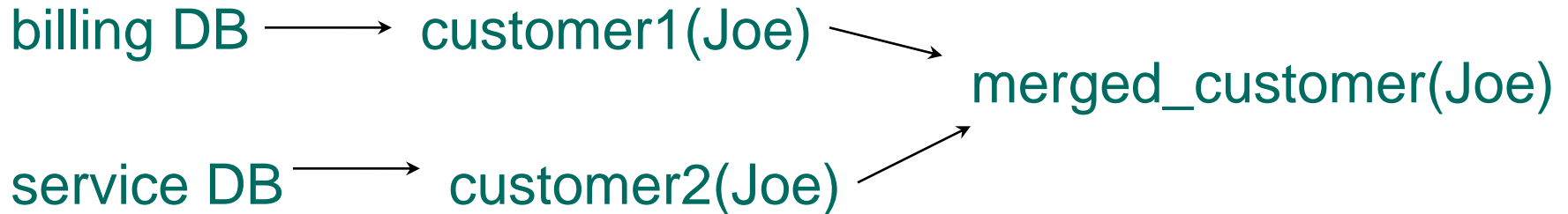
- Data Cleaning
- Data Loading
- Derived Data



# Data Cleaning

---

- Migration (e.g., yen  $\Rightarrow$  dollars)
- Scrubbing: use domain-specific knowledge (e.g., social security numbers)
- Fusion (e.g., customer merging)





# Loading Data

---

- Incremental vs. refresh
- Off-line vs. on-line
- Frequency of loading
  - At night, 1x a week/month, continuously
- Parallel/Partitioned load

# Derived Data

---

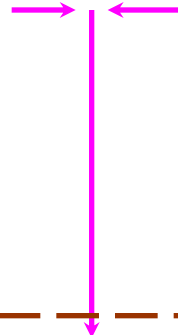
- Derived Warehouse Data
  - indexes
  - aggregates
  - materialized views (next slide)
- When to update derived data?
- Incremental vs. refresh

# Materialized Views

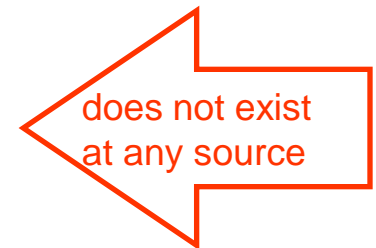
- Define new warehouse relations using SQL expressions

sale	prodlid	storeld	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

product	id	name	price
	p1	bolt	10
	p2	nut	5

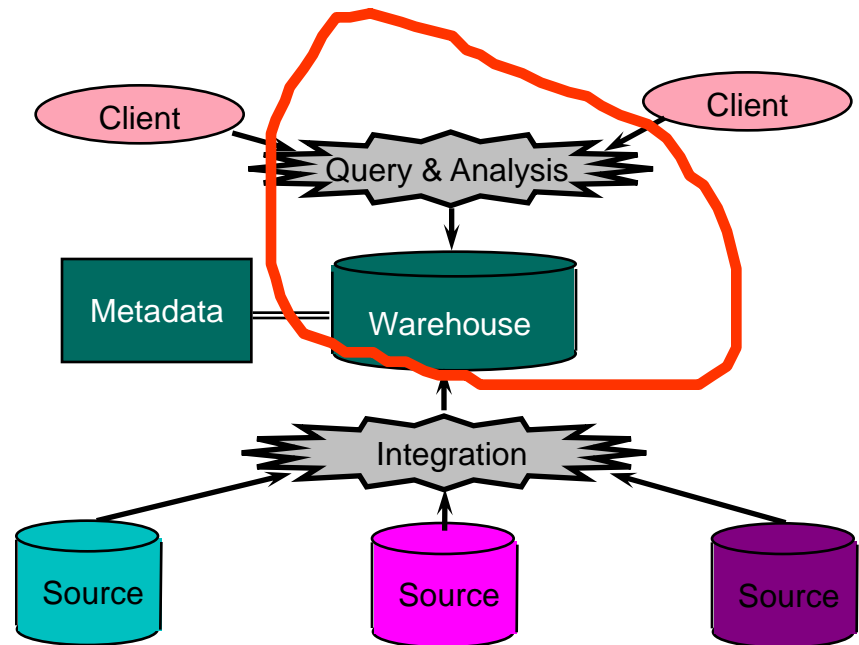


joinTb	prodlid	name	price	storeld	date	amt
	p1	bolt	10	c1	1	12
	p2	nut	5	c1	1	11
	p1	bolt	10	c3	1	50
	p2	nut	5	c2	1	8
	p1	bolt	10	c1	2	44
	p1	bolt	10	c2	2	4



# Processing

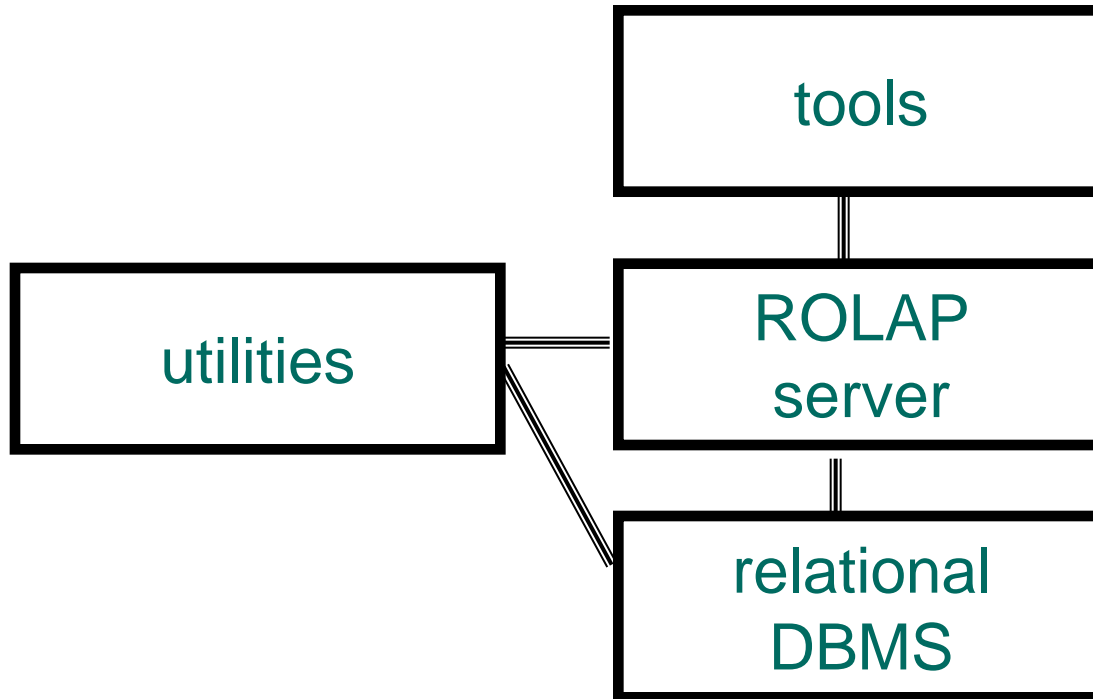
- ROLAP: Relational On-Line Analytical Processing
- MOLAP: Multi-Dimensional On-Line Analytical Processing
- Index Structures
- What to Materialize?
- Algorithms



# ROLAP Server

- Relational OLAP Server

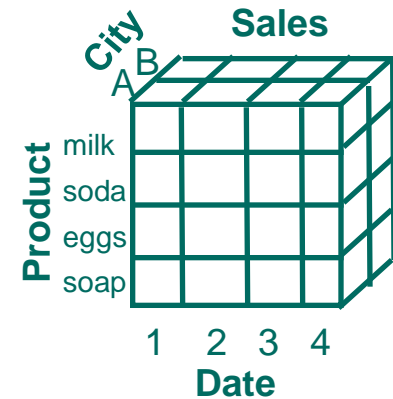
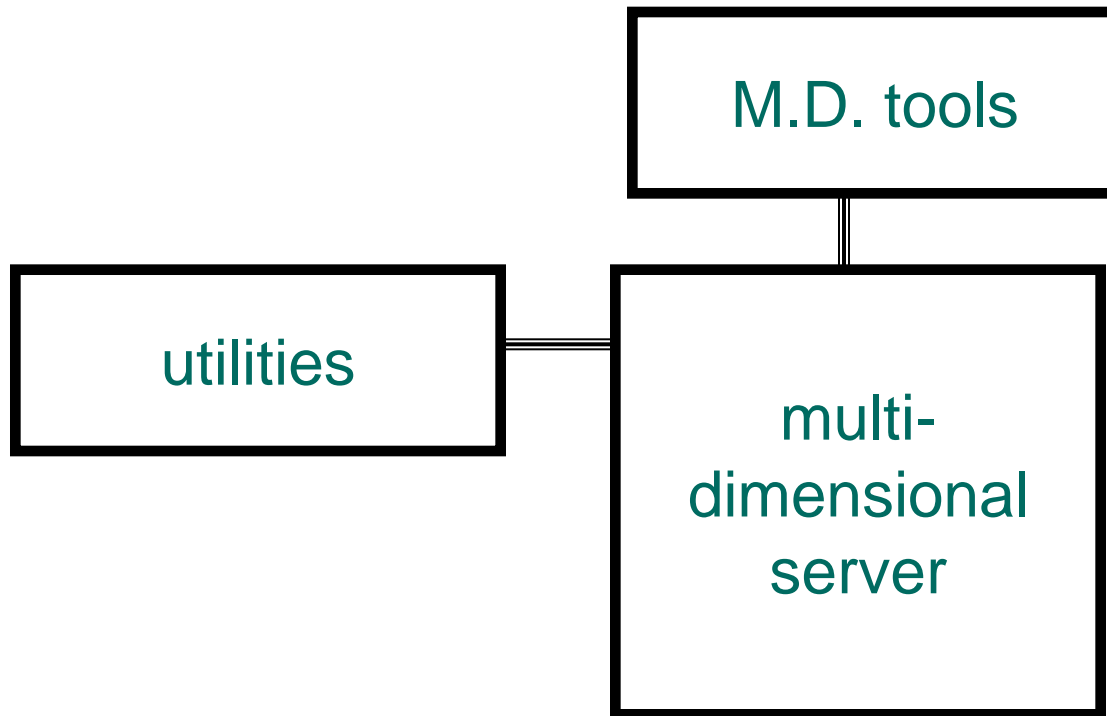
sale	prodlid	date	sum
	p1	1	62
	p2	1	19
	p1	2	48



.....Special indices, tuning;  
Schema is “denormalized”

# MOLAP Server

- Multi-Dimensional OLAP Server

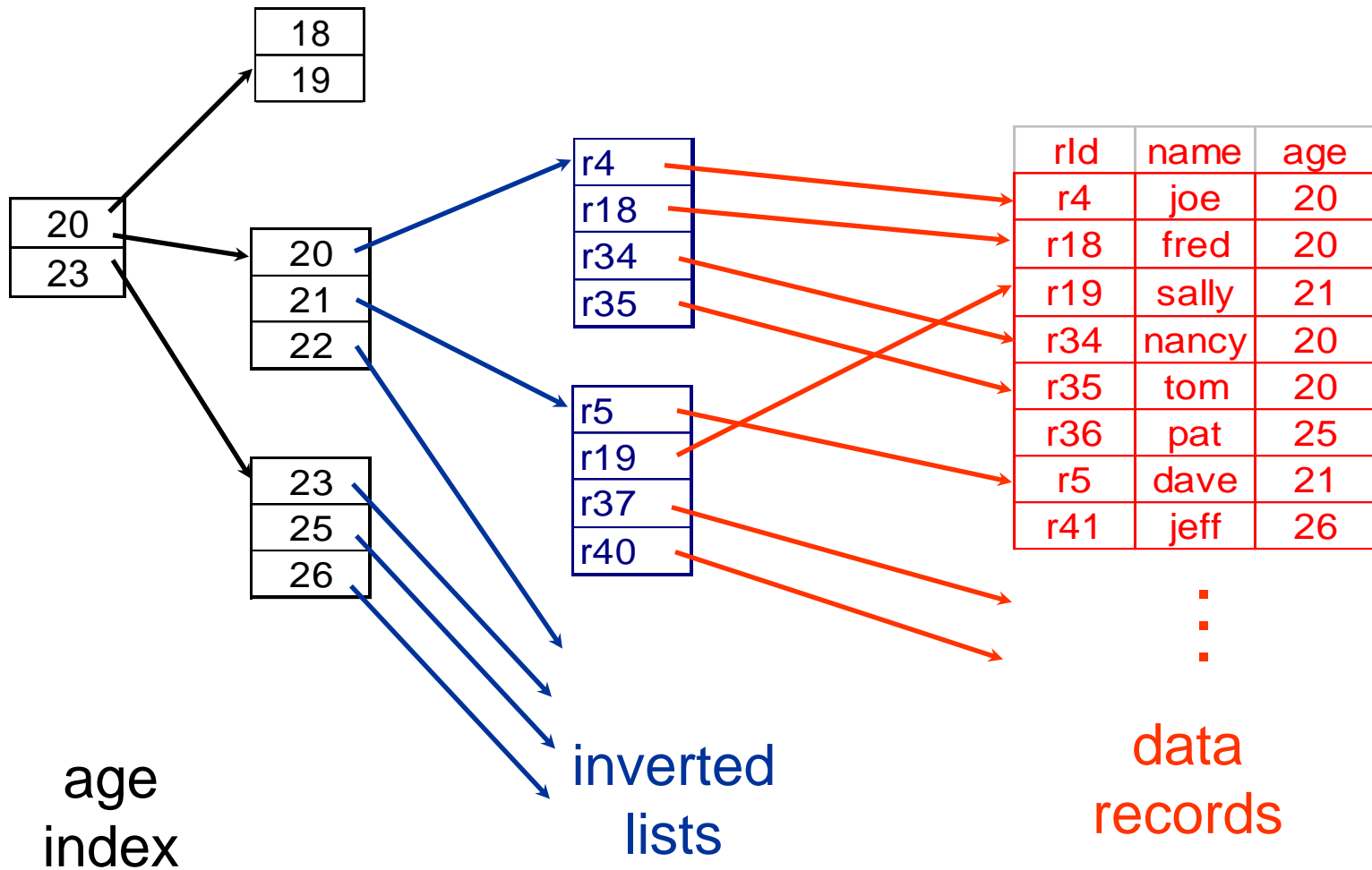


# Index Structures

---

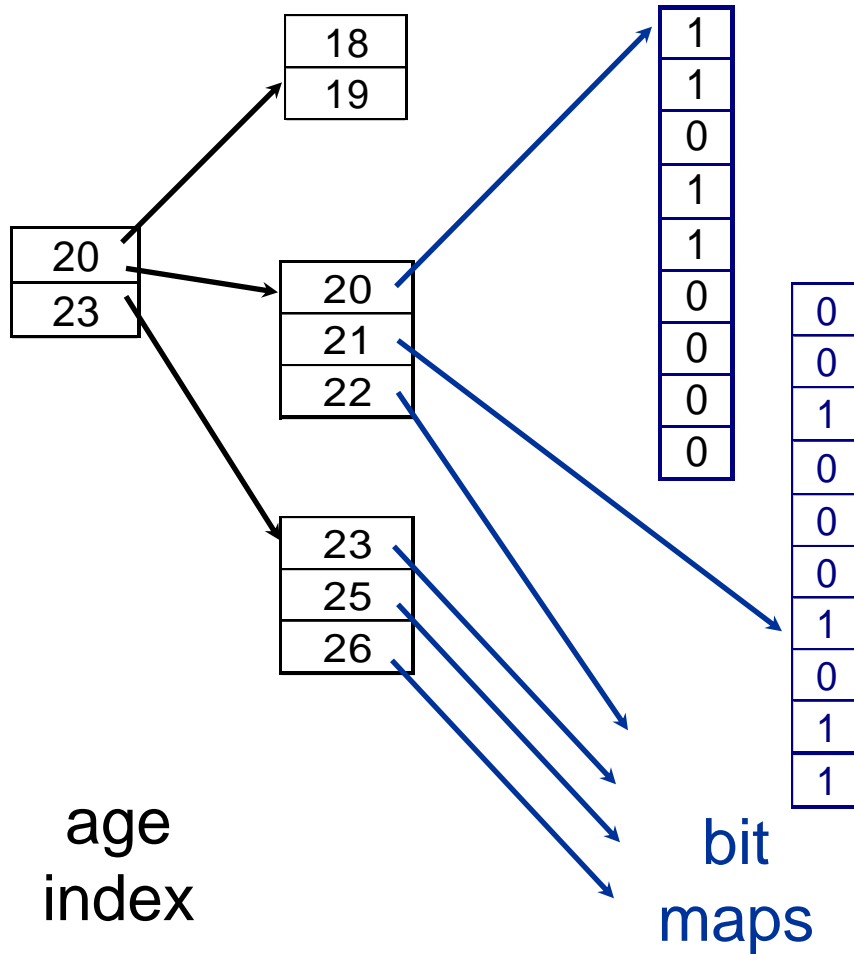
- Traditional Access Methods
  - B-trees, hash tables, grids, ...
- Popular in Warehouses
  - inverted lists
  - bit map indexes

# Inverted Lists





# Bit Maps



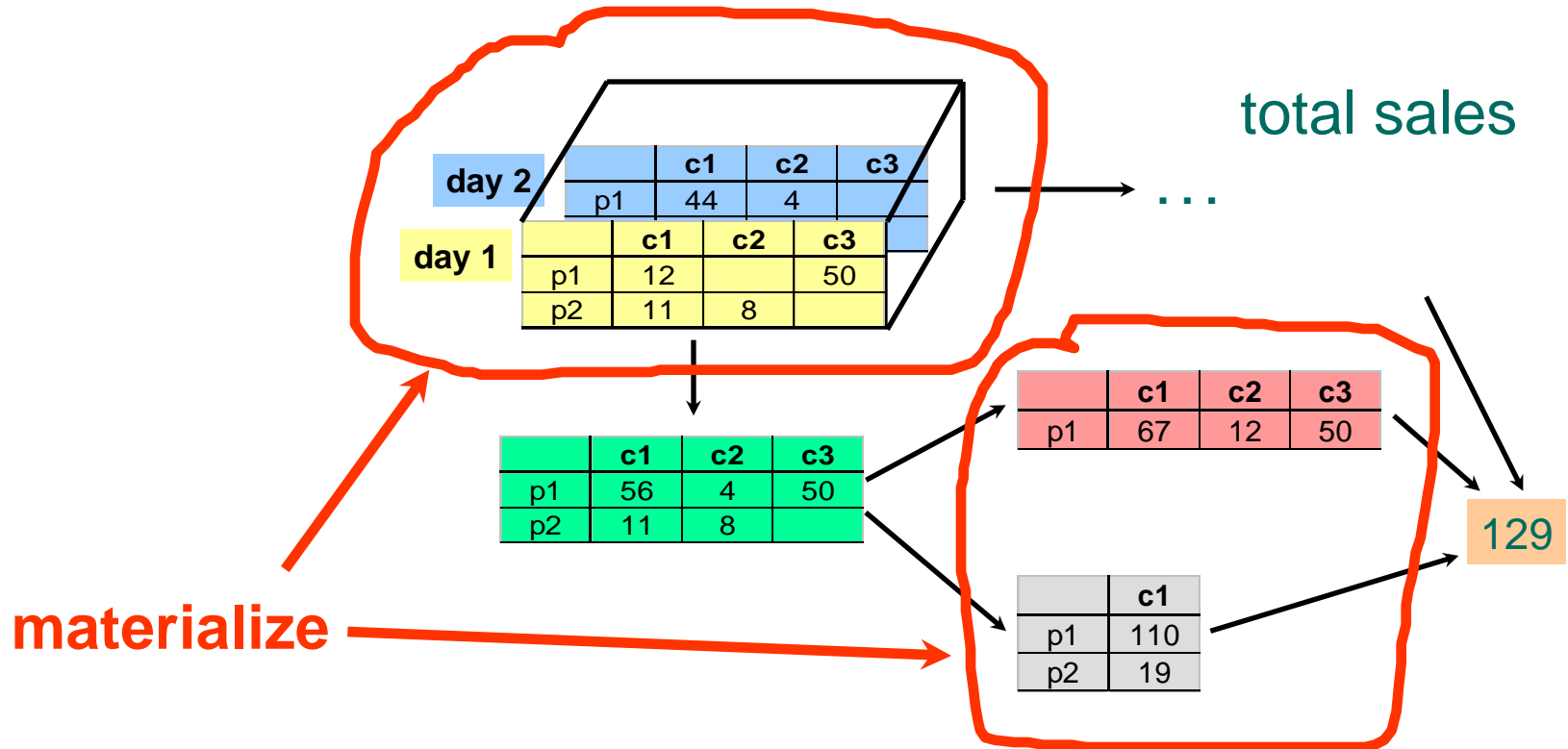
id	name	age
1	joe	20
2	fred	20
3	sally	21
4	nancy	20
5	tom	20
6	pat	25
7	dave	21
8	jeff	26

⋮

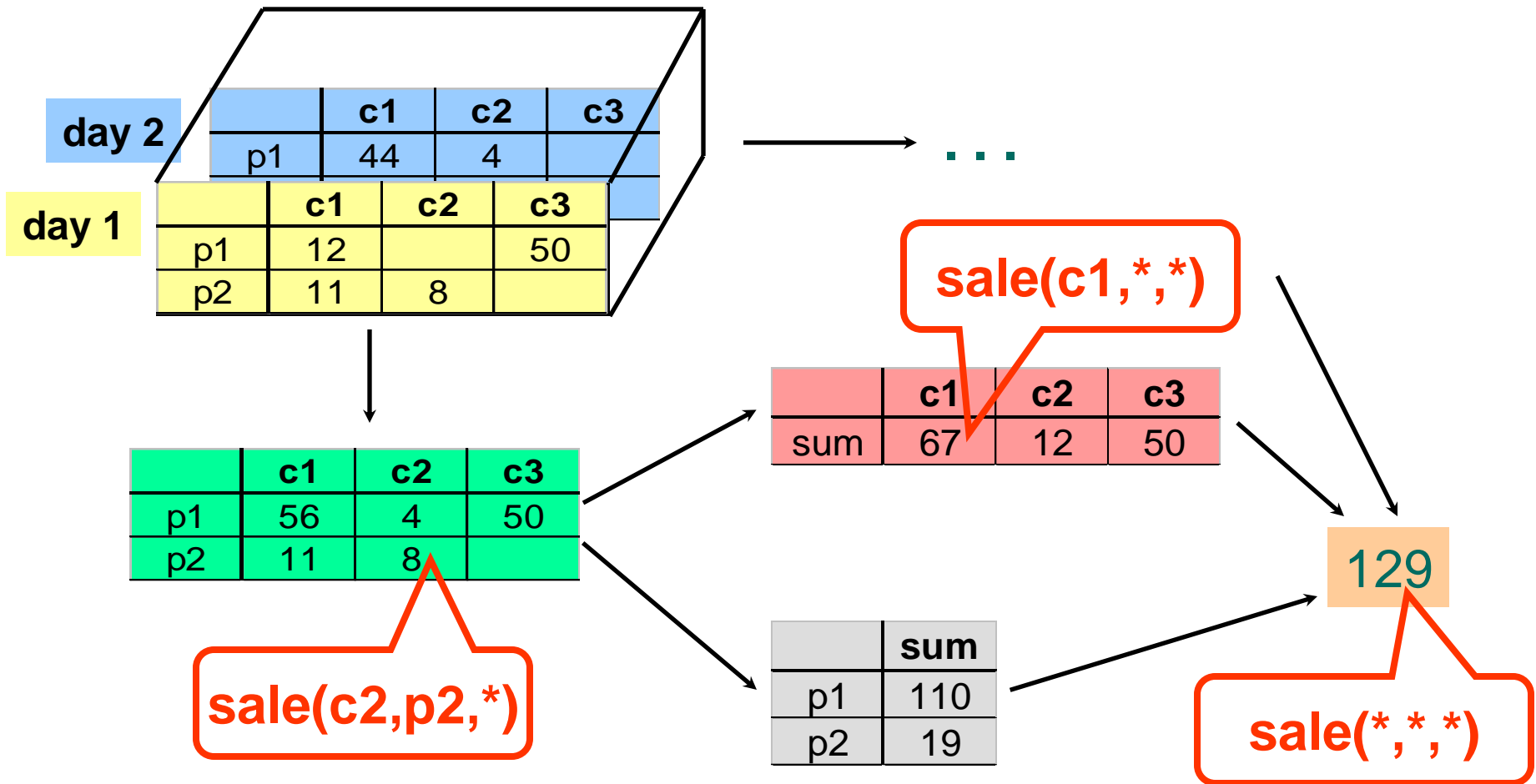
data records

# What to Materialize?

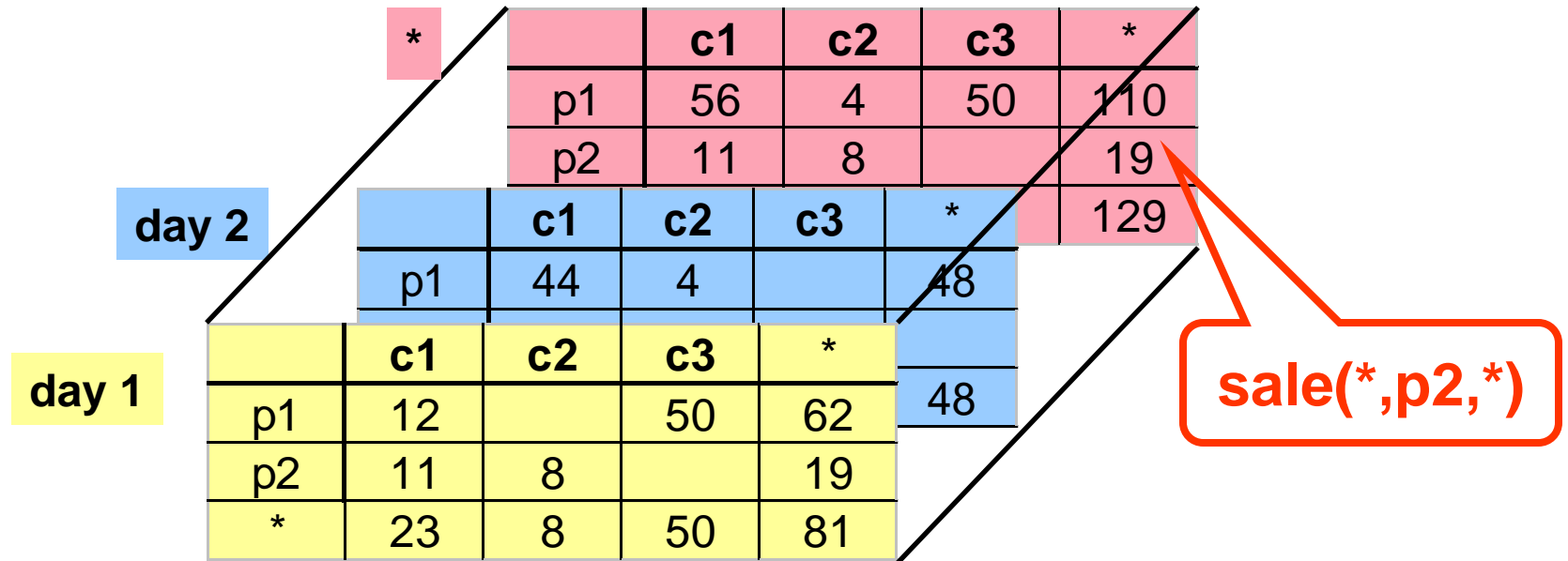
- Store in warehouse results useful for common queries
- Example:



# Intermediate Results



# Extended Cube

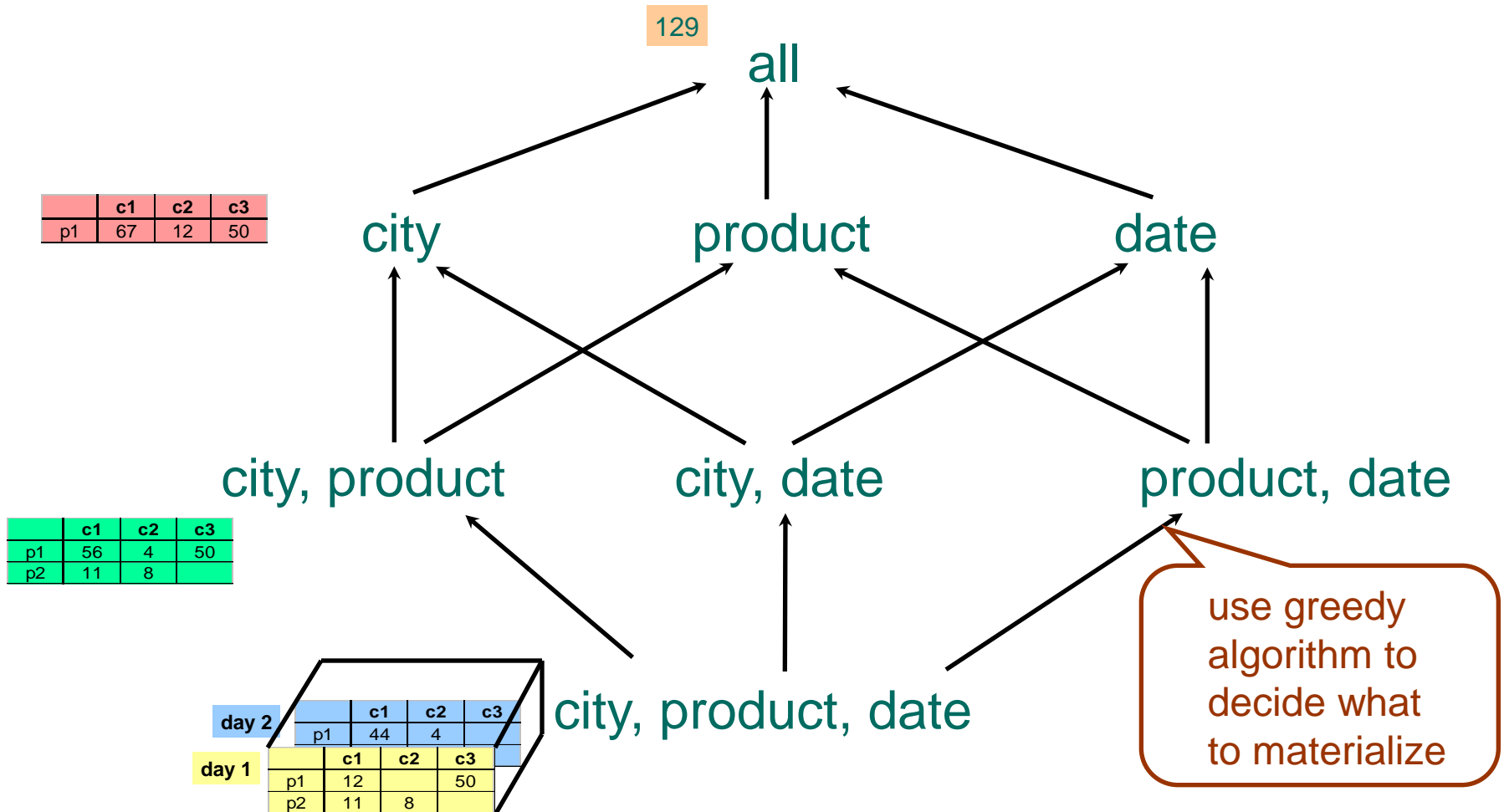


# Materialization Factors

---

- Type/frequency of queries
- Query response time
- Storage cost
- Update cost

# Cube Aggregates Lattice



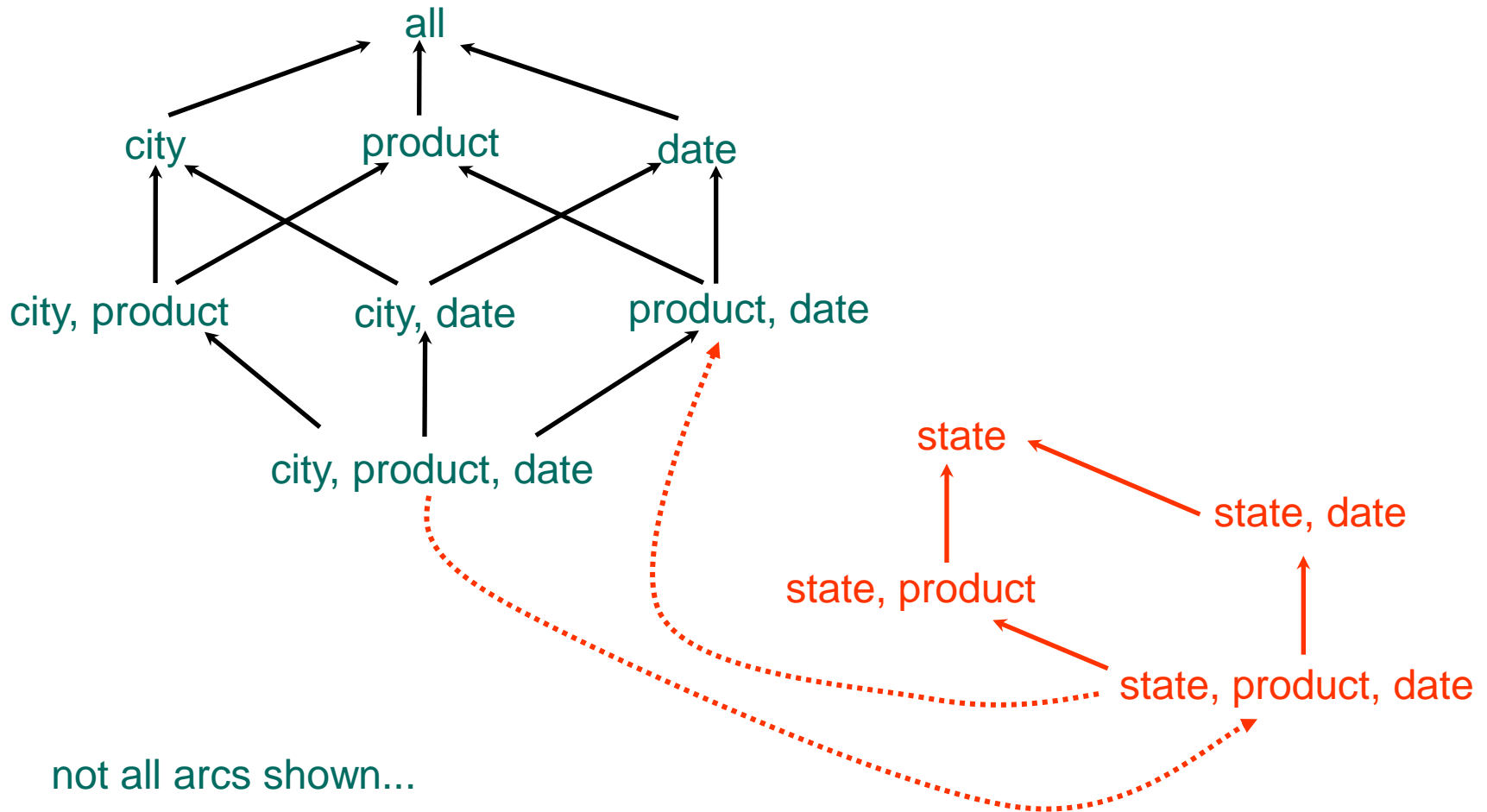
# Dimension Hierarchies

---



<b>cities</b>	<b>city</b>	<b>state</b>
	c1	CA
	c2	NY

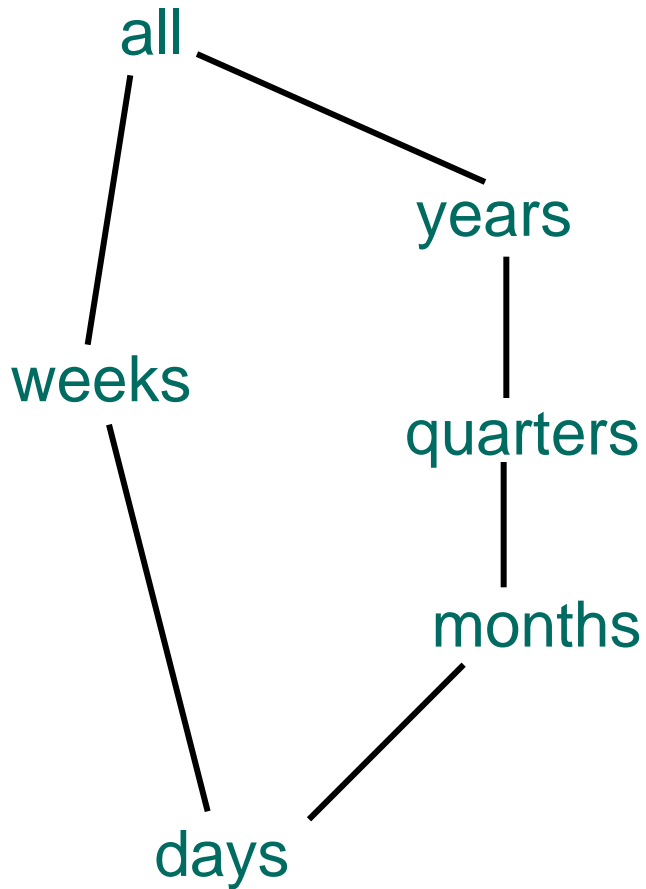
# Dimension Hierarchies



not all arcs shown...



# Interesting Hierarchy

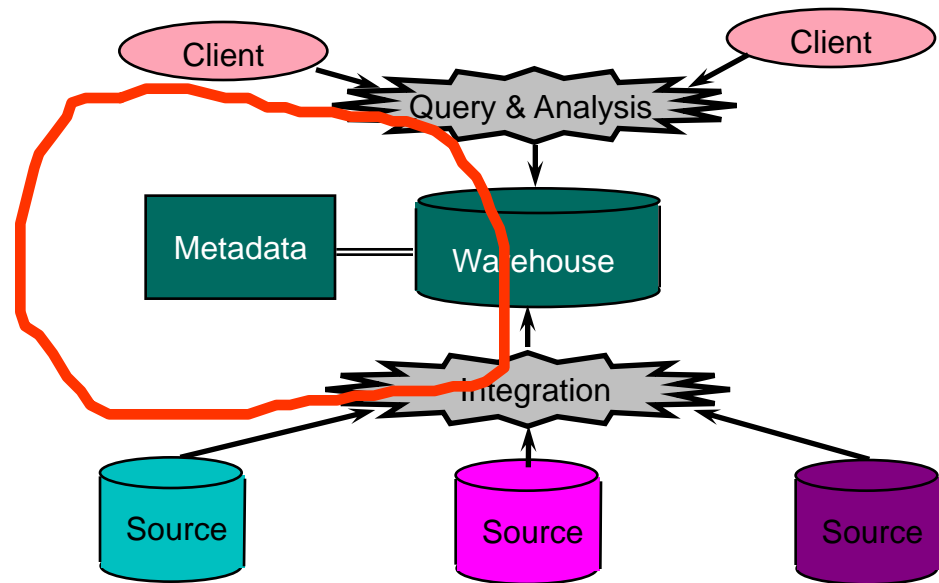


time	day	week	month	quarter	year
	1	1	1	1	2000
	2	1	1	1	2000
	3	1	1	1	2000
	4	1	1	1	2000
	5	1	1	1	2000
	6	1	1	1	2000
	7	1	1	1	2000
	8	2	1	1	2000

conceptual  
dimension table

# Managing

- Metadata



# Metadata

---

- Administrative
  - definition of sources, tools,
  - schemas, dimension hierarchies,
  - rules for extraction, cleaning,
  - refresh, purging policies
  - user profiles, access control

# Current State of Industry

---

- Extraction and integration done off-line
  - Usually in large, time-consuming, batches
- Everything copied at warehouse
  - Not selective about what is stored
  - Query benefit vs storage & update cost
- Query optimization aimed at OLTP
  - High throughput instead of fast response
  - Process whole query before displaying anything

# Future Directions

---

- Better performance
- Larger warehouses
- Easier to use