Data Warehousing

Read chapter 8 of Atzeni et al. BD Architetture
Section 20.4, 20.5 of Garcia-Molina et al.

Slides derived from those by Hector Garcia-Molina

Outline

- What is a data warehouse?
- Why a warehouse?
- Models & operations
- Implementing a warehouse

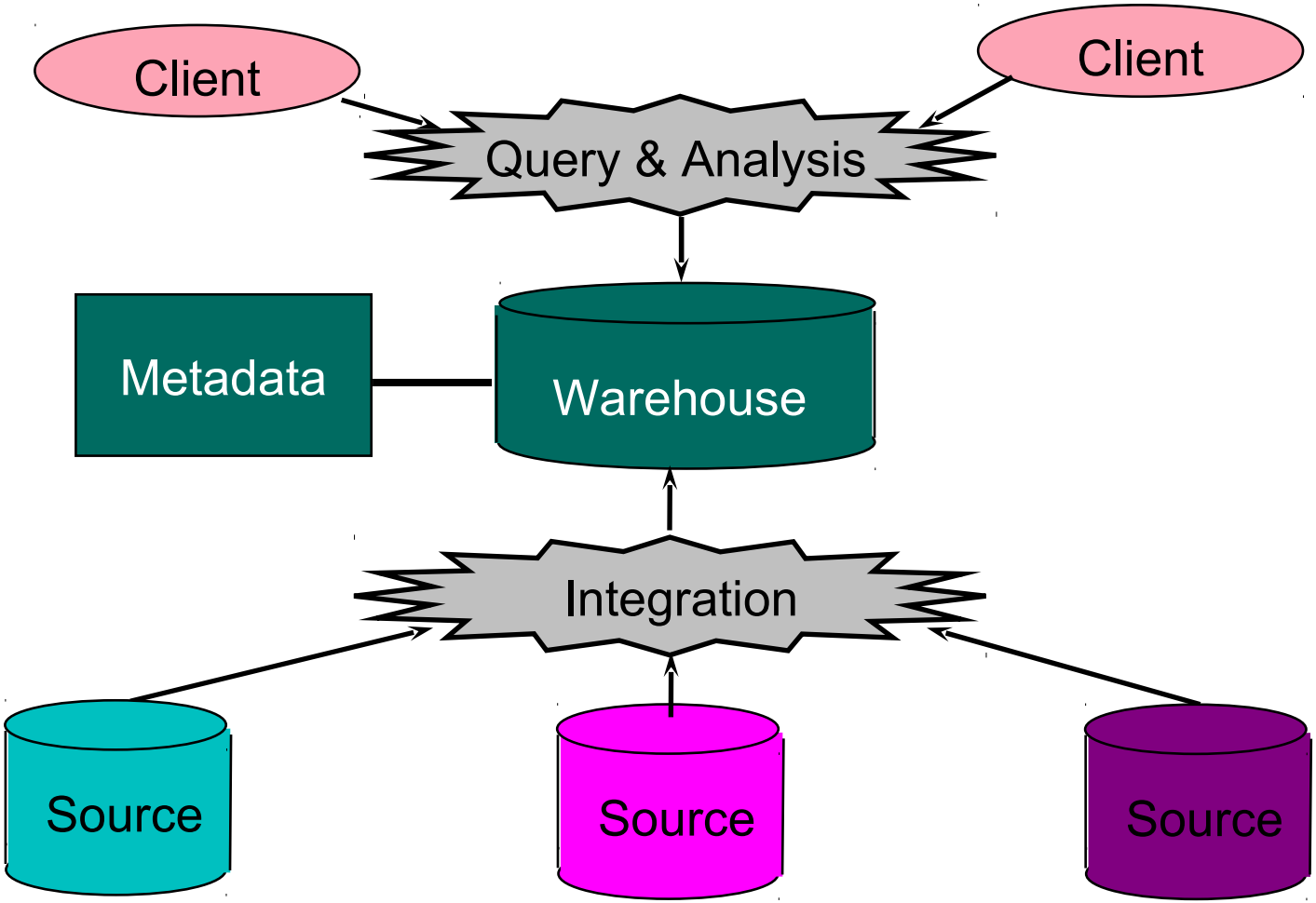
What is a Warehouse?

- 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

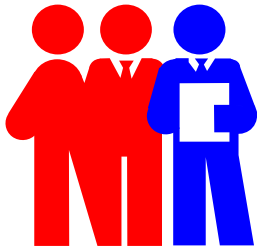


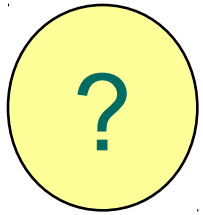
Motivating Examples

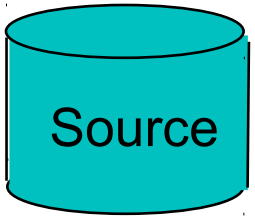
- Forecasting
- Comparing performance of units
- Monitoring, detecting fraud
- Visualization

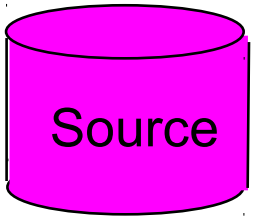
Why a Warehouse?

- Two Approaches:
  - Query-Driven (Lazy)
  - Warehouse (Eager)

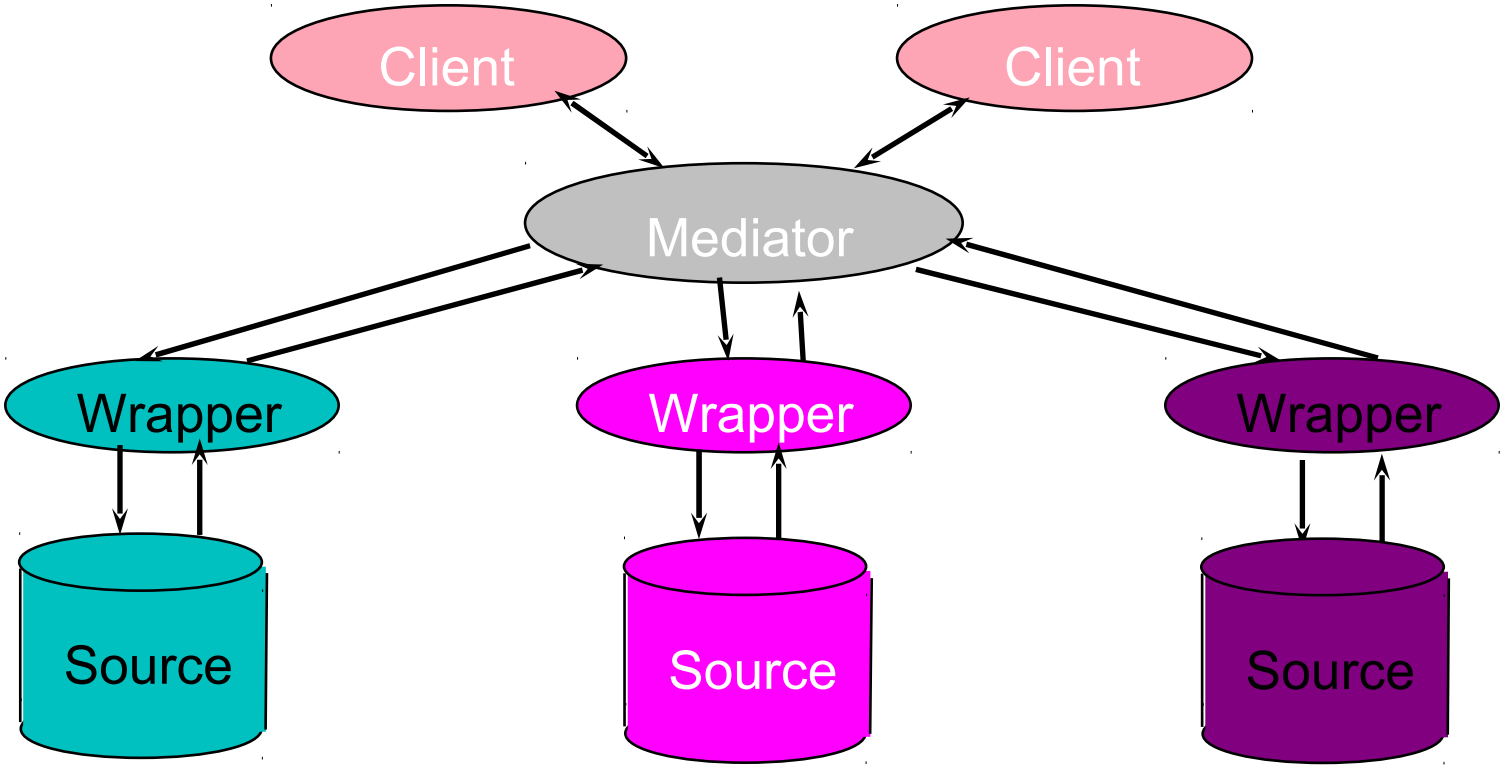








Query-Driven Approach



Advantages of Warehousing

- High query performance
- Queries not visible outside warehouse
- Local processing at sources unaffected
- Can operate when sources unavailable
- Can query data not stored in a DBMS
- Extra information at warehouse
  - Modify, summarize (store aggregates)
  - Add historical information

Advantages of Query-Driven

- No need to copy data
  - less storage
  - no need to purchase data
- More up-to-date data
- Query needs can be unknown
- Only query interface needed at sources
- May be less draining on sources

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-Tb 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

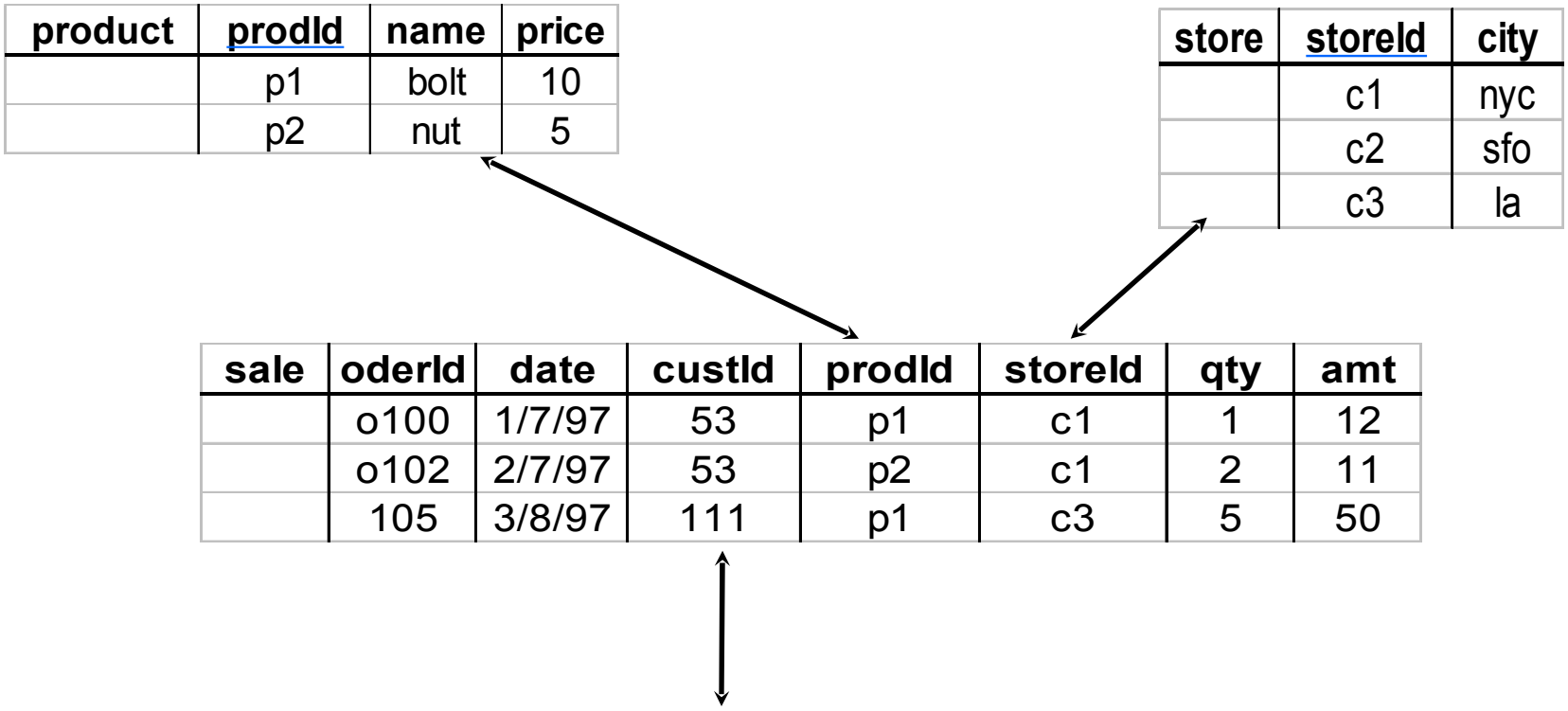
Data Marts

- Smaller warehouses
- Spans part of 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



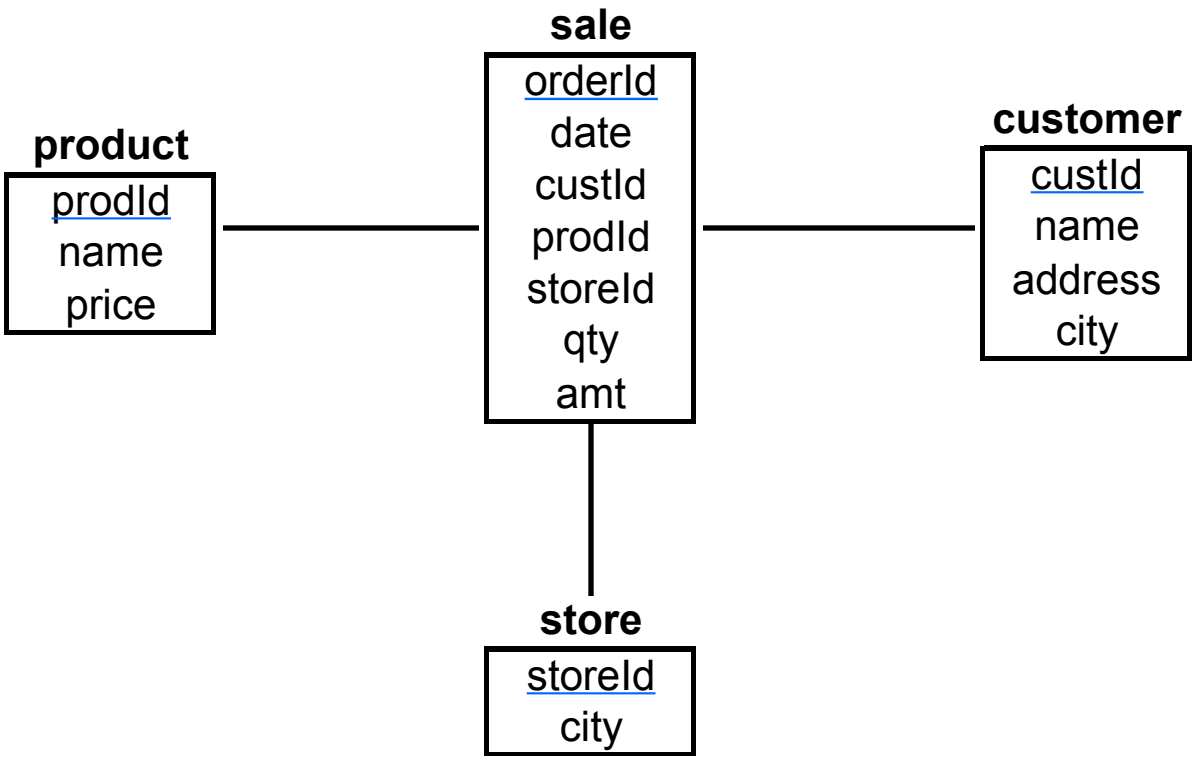
product	prodId	name	price
	p1	bolt	10
	p2	nut	5

store	storeId	city
	c1	nyc
	c2	sfo
	c3	la

sale	oderId	date	custId	prodId	storeId	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

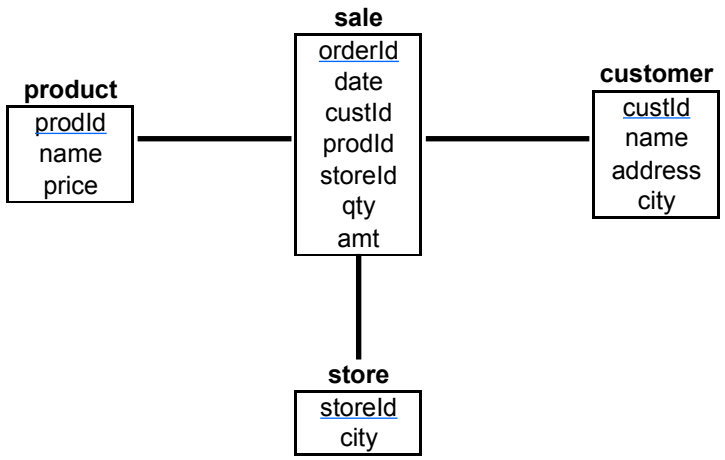
customer	custId	name	address	city
	53	joe	10 main	sfo
	81	fred	12 main	sfo
	111	sally	80 willow	la

Star Schema

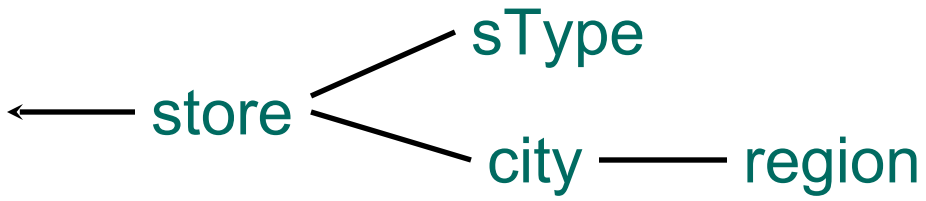

sale
orderId
date
custId
prodId
storeId
qty
amt
product
prodId
name
price
customer
custId
name
address
city
store
storeId
city

Terms

- Fact table
- Dimension tables
- Measures



Dimension Hierarchies



store	storeId	cityId	tId	mgr
	s5	sfo	t1	joe
	s7	sfo	t2	fred
	s9	la	t1	nancy

sType	tId	size	location
	t1	small	downtown
	t2	large	suburbs

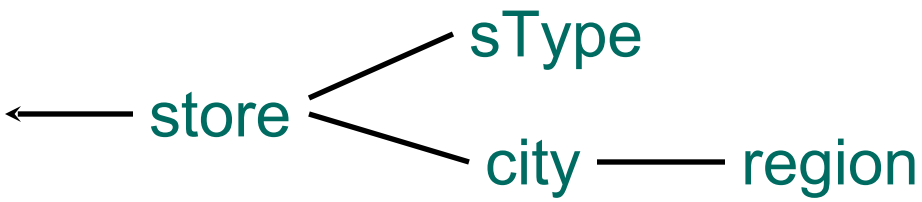
city	cityId	pop	regId
	sfo	1M	north
	la	5M	south

region	regId	name
	north	cold region
	south	warm region

➔ snowflake schema

Snowflake Schema

Sometimes not normalized: not in third normal form



store	storeId	cityId	tId	mgr
	s5	sfo	t1	joe
	s7	sfo	t2	fred
	s9	la	t1	nancy

sType	tId	size	location
	t1	small	downtown
	t2	large	suburbs

city	cityId	pop	regId	name
	sfo	1M	north	cold region
	la	5M	south	warm region

Cube

Fact table view:

sale	prodId	storeId	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	

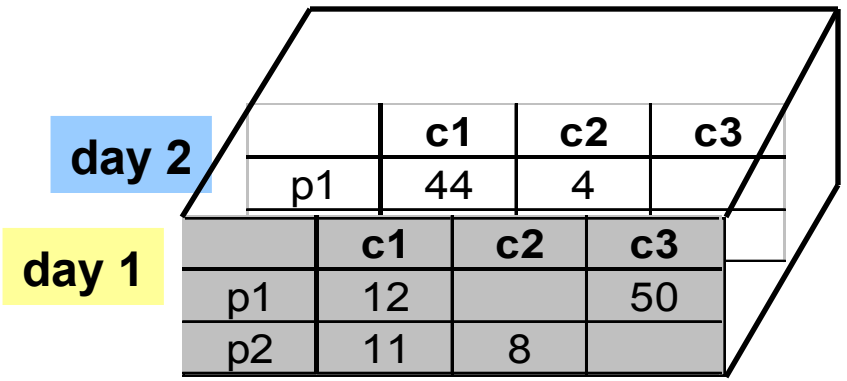
dimensions = 2

3-D Cube

Fact table view:

sale	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:



dimensions = 3

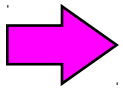
ROLAP vs. MOLAP

- ROLAP:
  Relational On-Line Analytical Processing
- MOLAP:
  Multi-Dimensional On-Line Analytical Processing

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

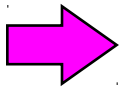


81

Aggregates

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

sale	prodId	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

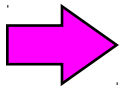


71

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

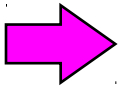
rollup

drill-down

Another Example

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

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



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

rollup →

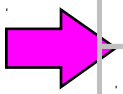
← drill-down

Another Example

- Add up amounts by month
- In SQL: SELECT month, prodId, storeId, sum(amt)

FROM SALE JOIN DATE GROUP BY month, prodId, storeId

sale	prodId	storeId	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

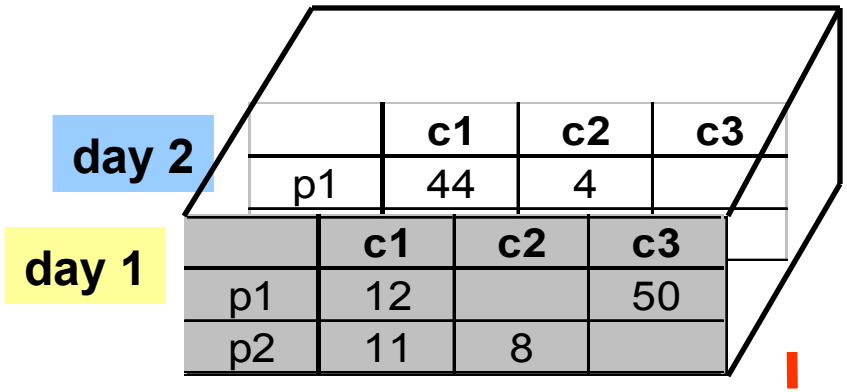


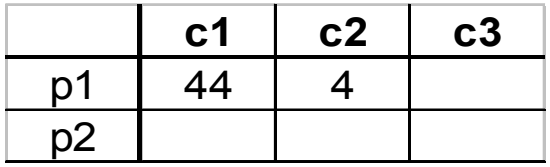
sale	prodId	storeId	month	amt
	p1	c1	sep	23
	p2	c2	sep	58
	p1	c3	oct	48

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



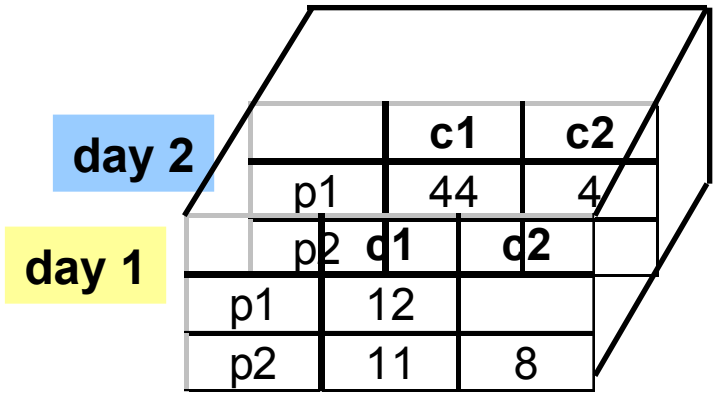


slicing

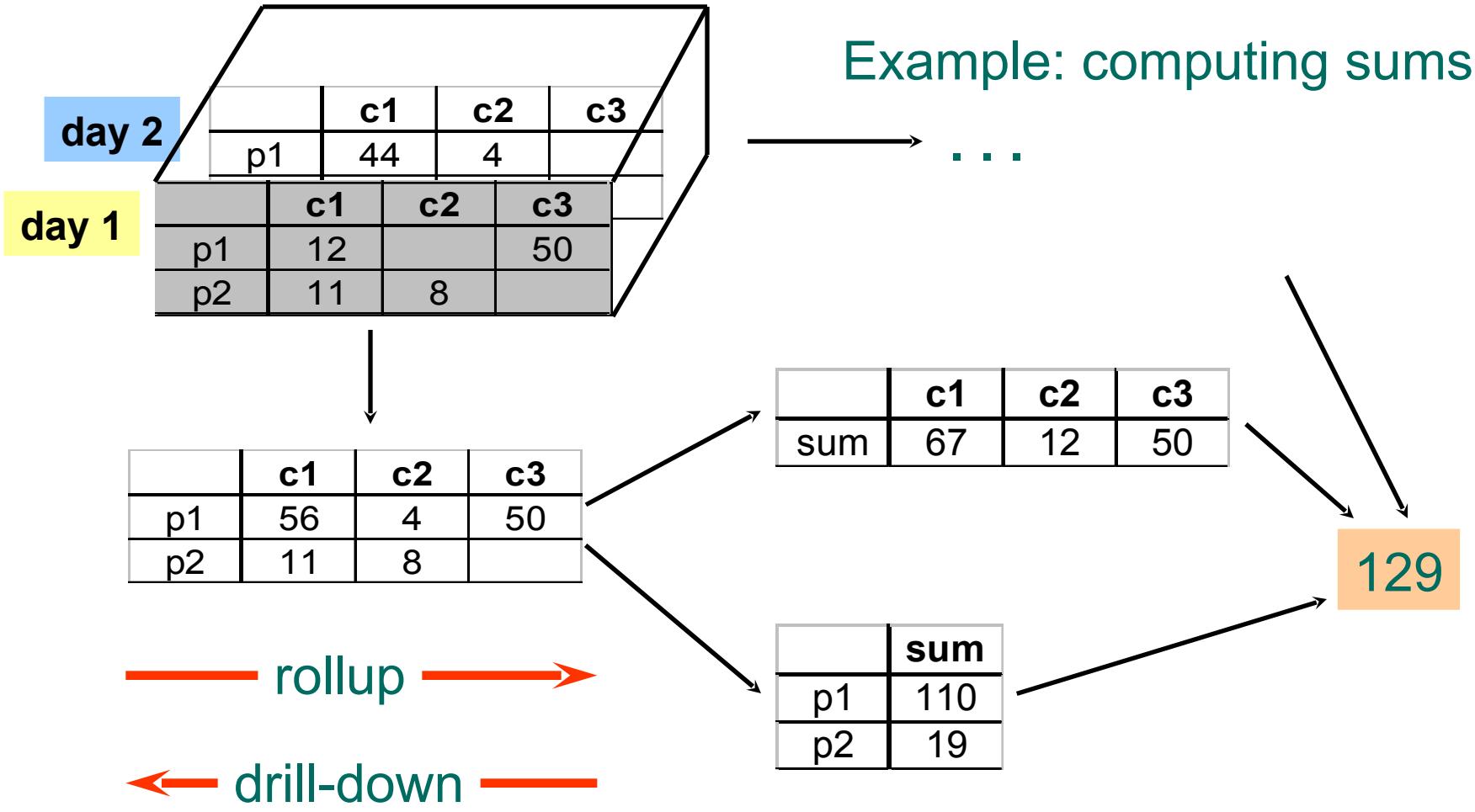
(equality selection)

dicing

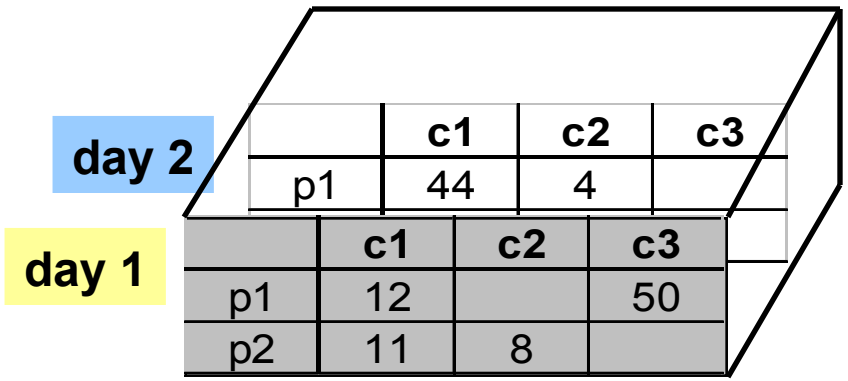
(range selection)



Cube Aggregation



Aggregation Using Hierarchies



rollup

drill-down

customer

region

country

	region A	region B
p1	56	54
p2	11	8

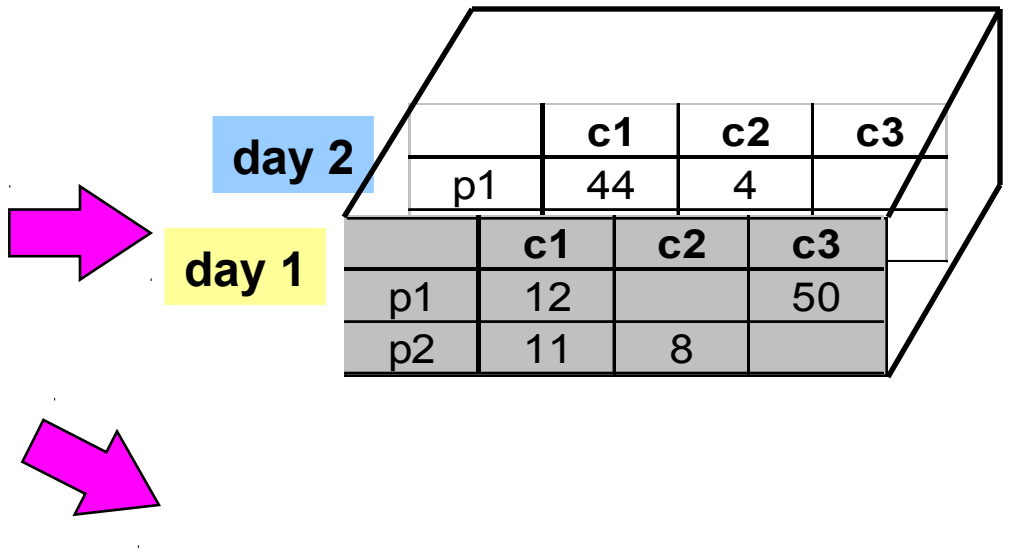
(customer c1 in Region A;
customers c2, c3 in Region B)

Pivoting

Fact table view:

sale	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:



	c1	c2	c3
p1	44	4	

	c1	c2	c3
p1	12		50
p2	11	8	

	c1	c2	c3
p1	56	4	50
p2	11	8	

Query & Analysis Tools

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

Other Operations

- Time functions
  - e.g., time average
- Computed Attributes
  - e.g., commission = sales * rate
- Text Queries
  - e.g., find documents with words X AND B
  - e.g., rank documents by frequency of words X, Y, Z

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	id	name	address	city
	53	joe	10 main	sfo
	81	fred	12 main	sfo
	111	**sally**	**80 willow**	**la**

Monitoring Techniques

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

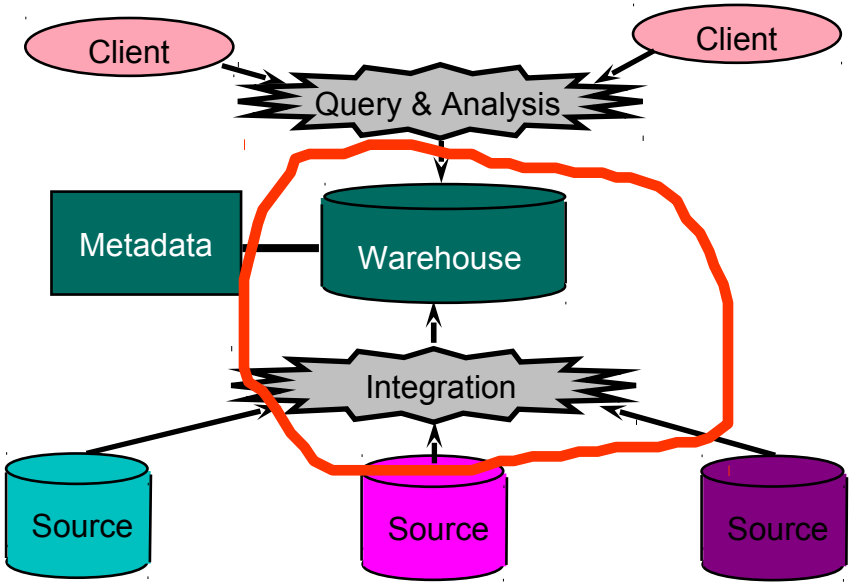
➚ Advantages & Disadvantages!!

Monitoring Issues

- Frequency
  - periodic: daily, weekly, …
  - triggered: on “big” change, lots of changes, ...
- Data transformation
  - convert data to uniform format
  - remove & add fields (e.g., add date to get history)
- Standards (e.g., ODBC)
- Gateways

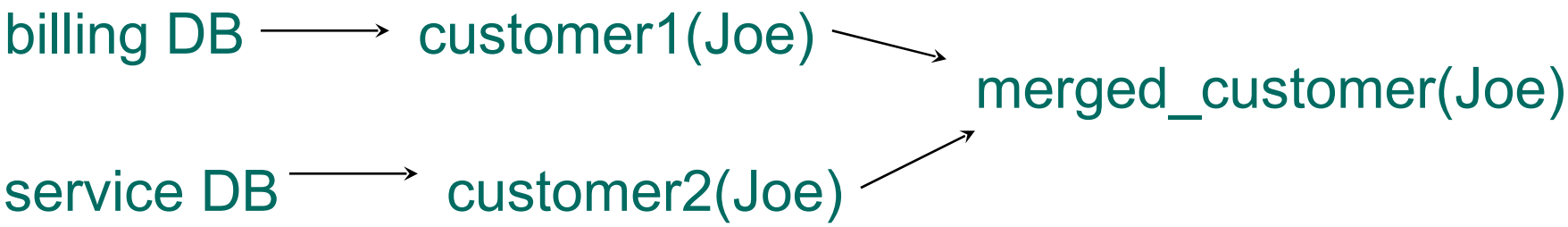
Integration

- Data Cleaning
- Data Loading
- Derived Data



Data Cleaning

- Migration (e.g., yen ⇨ 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	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

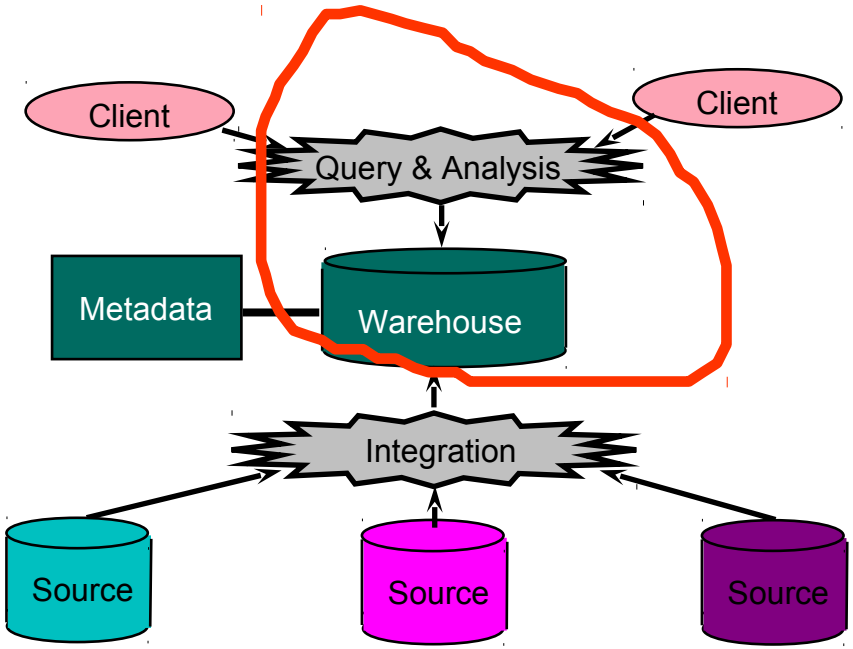
product	id	name	price
	p1	bolt	10
	p2	nut	5

joinTb	prodId	name	price	storeId	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

does not exist at any source

Processing

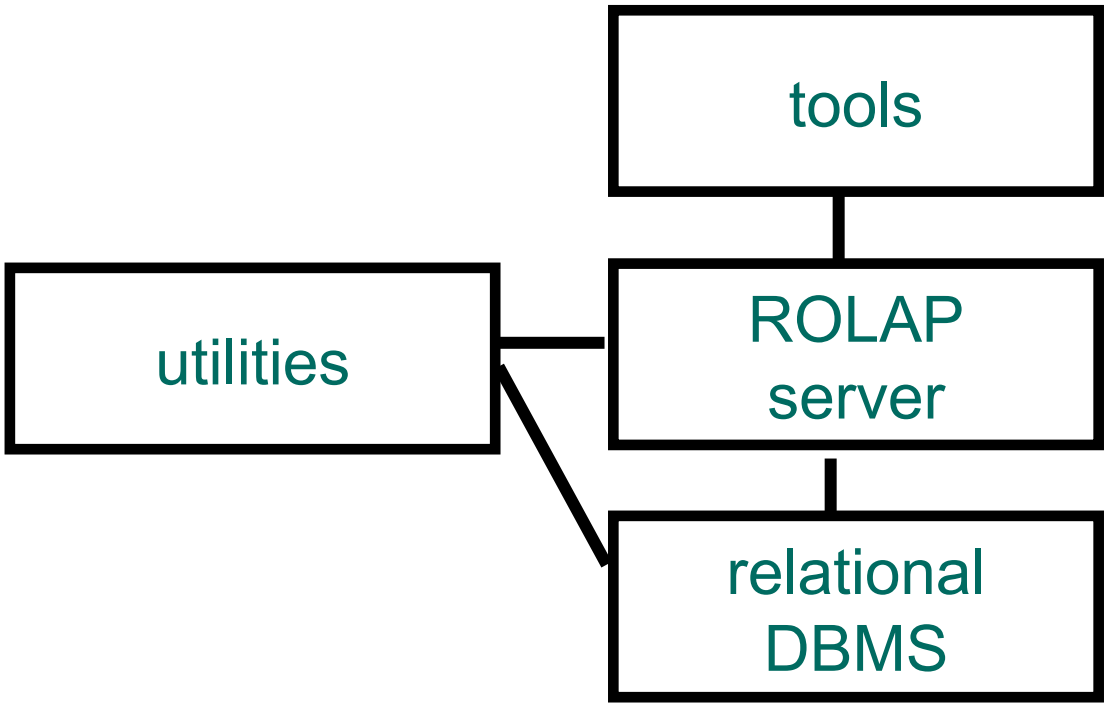
- ROLAP servers vs. MOLAP servers
- Index Structures
- What to Materialize?
- Algorithms



ROLAP Server

- Relational OLAP Server

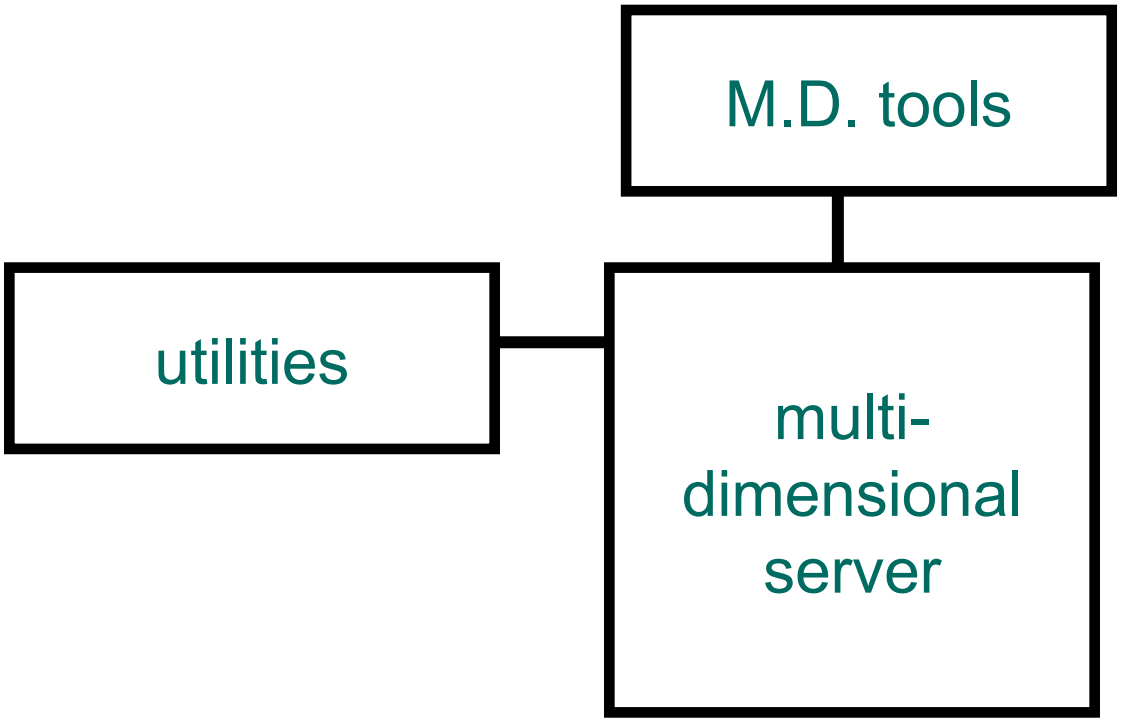
sale	prodId	date	sum
	p1	1	62
	p2	1	19
	p1	2	48

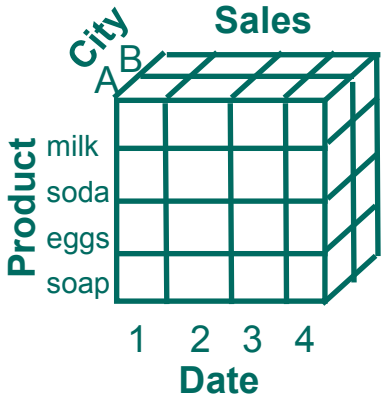


Special indices, tuning;
Schema is “denormalized”

MOLAP Server

- Multi-Dimensional OLAP Server



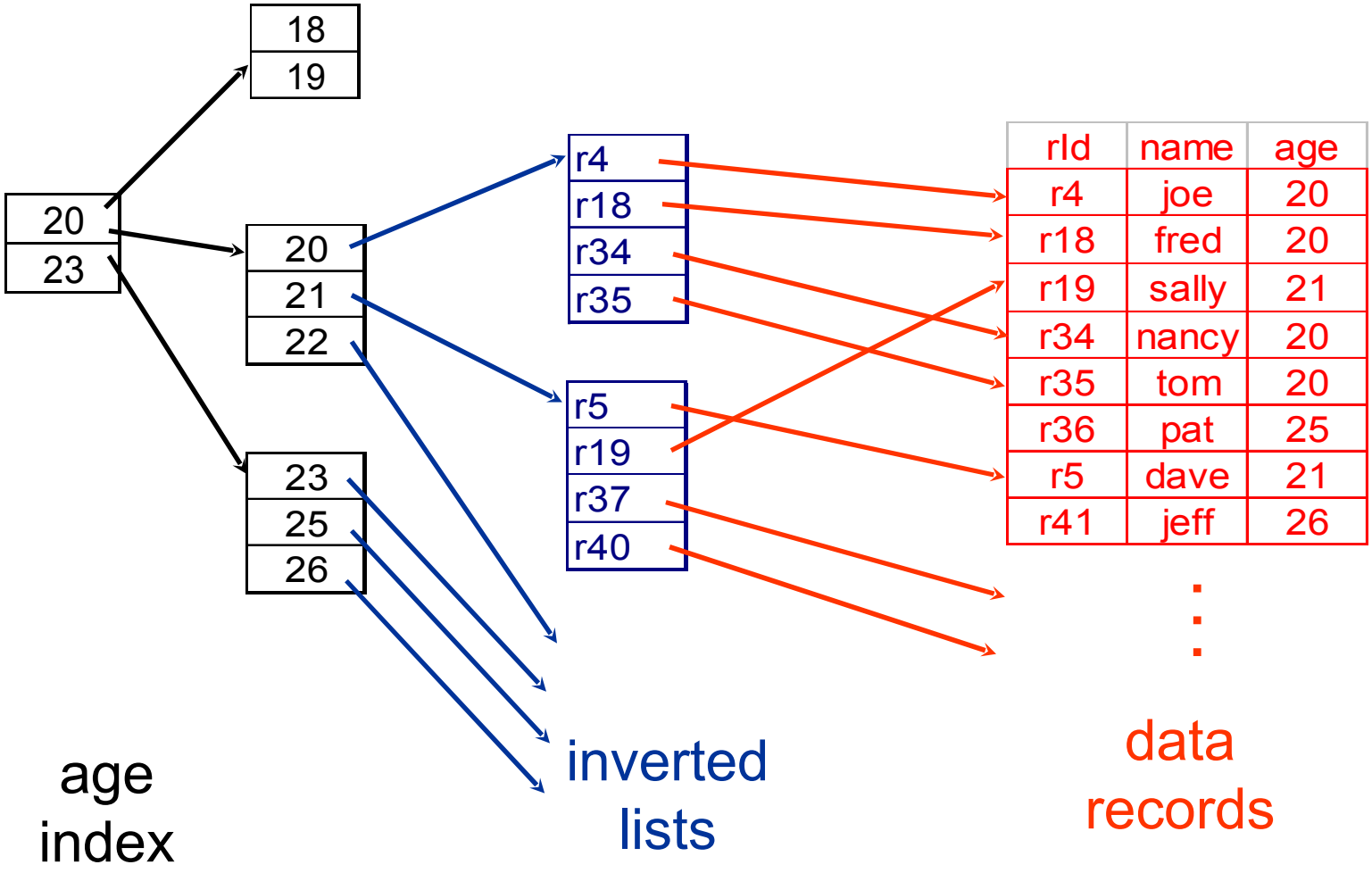


could also sit on relational DBMS

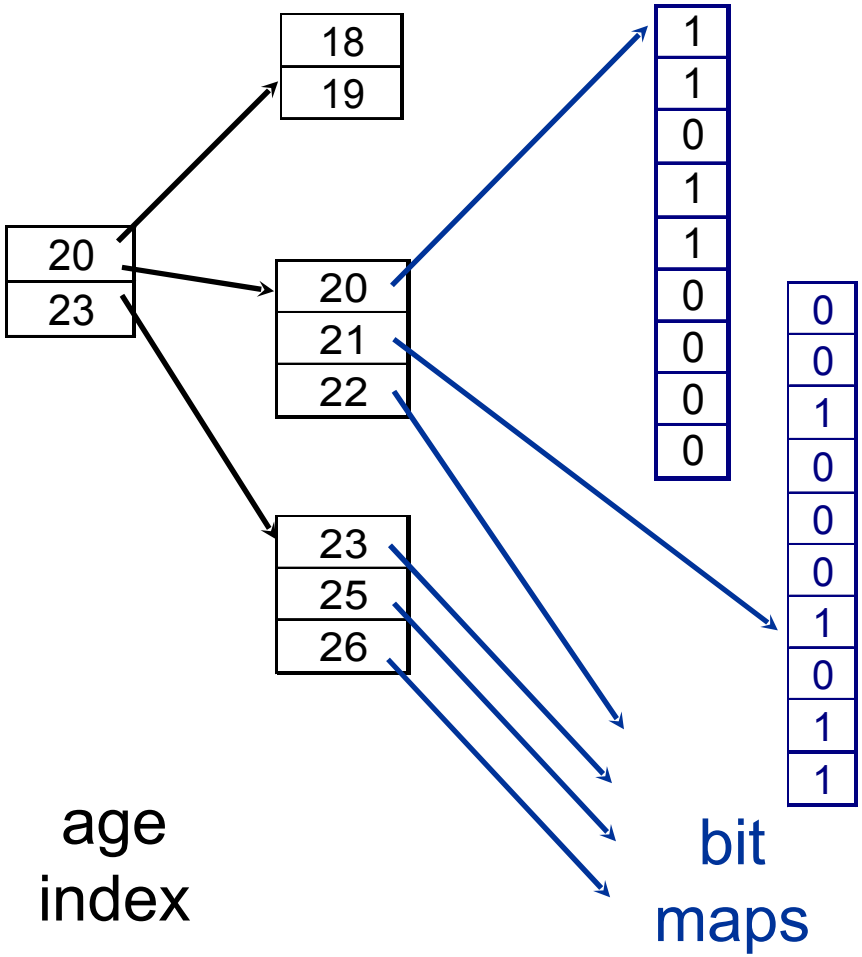
Index Structures

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

Join

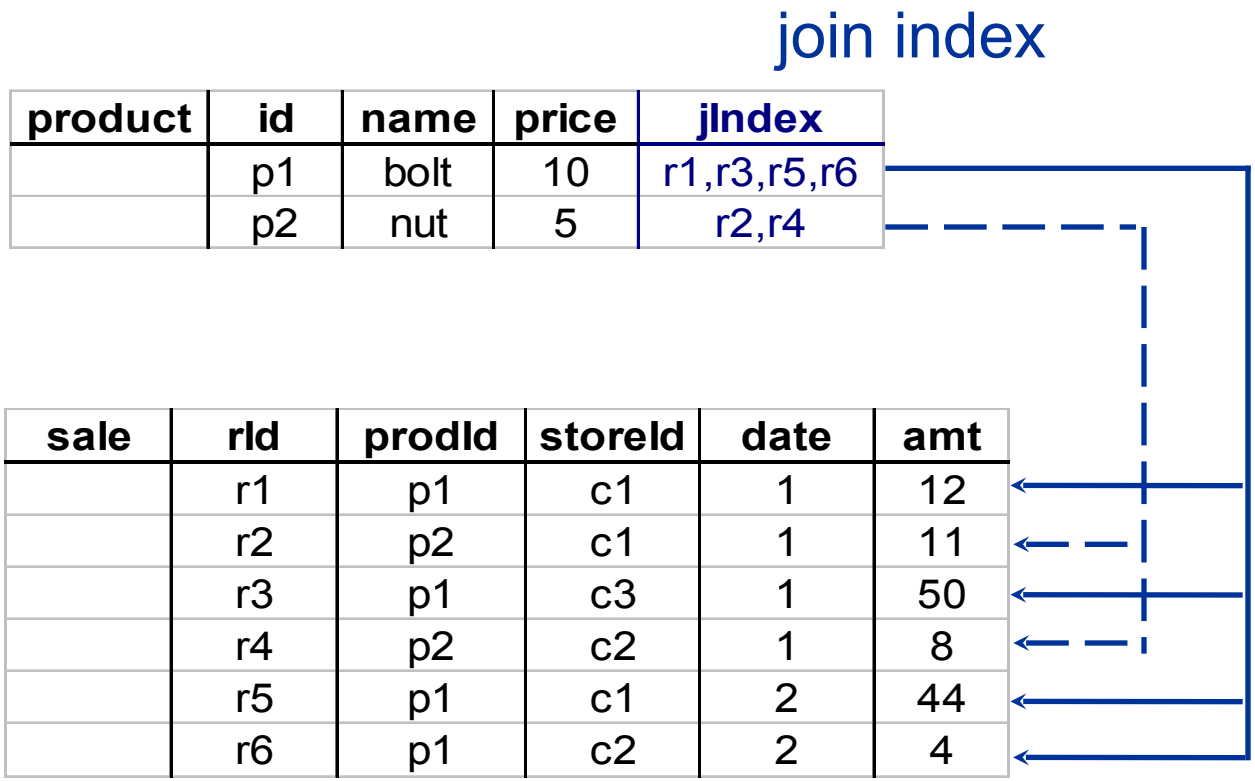
- "Combine" SALE, PRODUCT relations
- In SQL: SELECT * FROM SALE NATURAL JOIN PRODUCT

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

product	id	name	price
	p1	bolt	10
	p2	nut	5

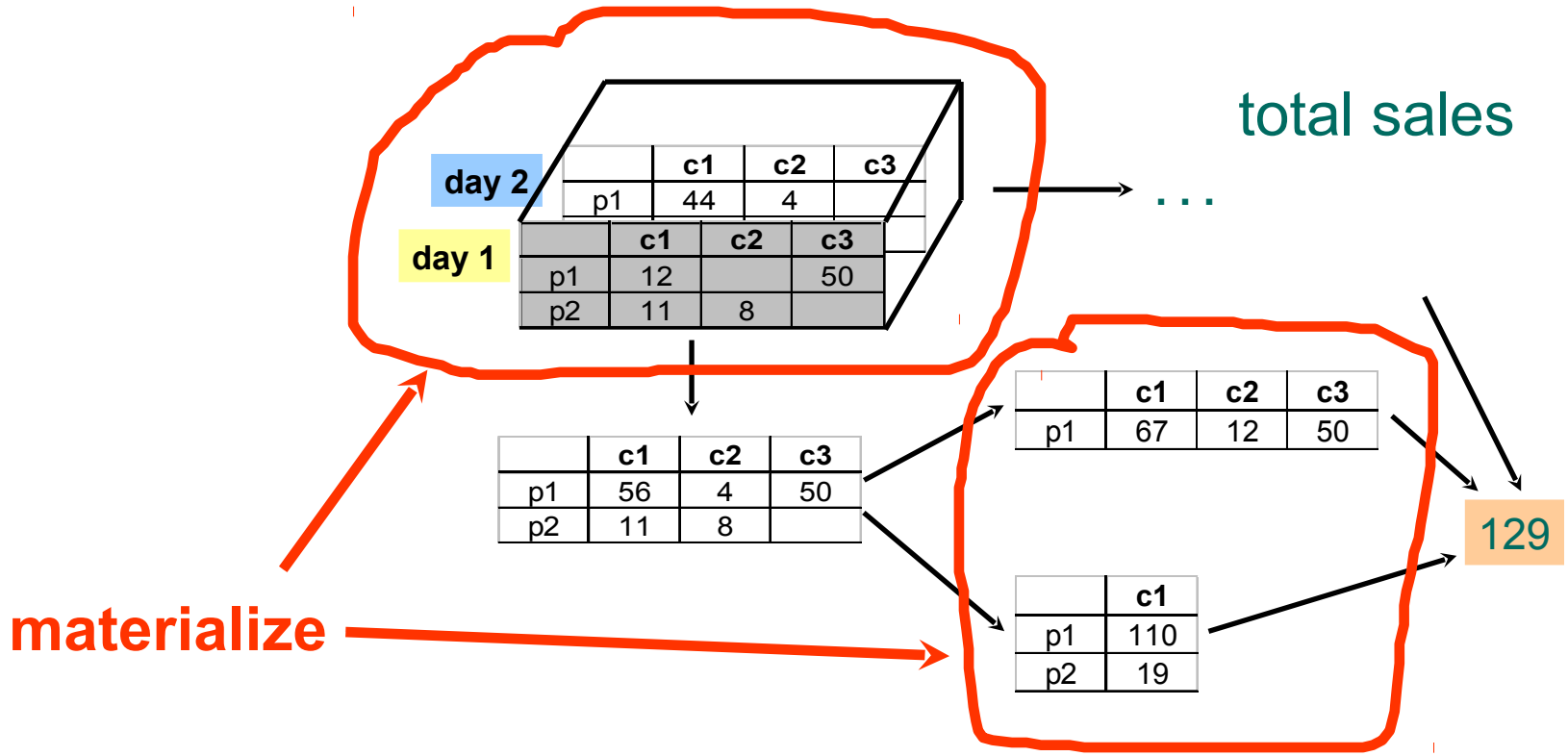
joinTb	prodId	name	price	storeId	date	amt
	p1	bolt	10	c1	1	12
	p2	nut	5	c1	1	11
	p1	bolt	10	c3	1	50
	p2	nut	5	c2	1	8
	p1	bolt	10	c1	2	44
	p1	bolt	10	c2	2	4

Join Indexes

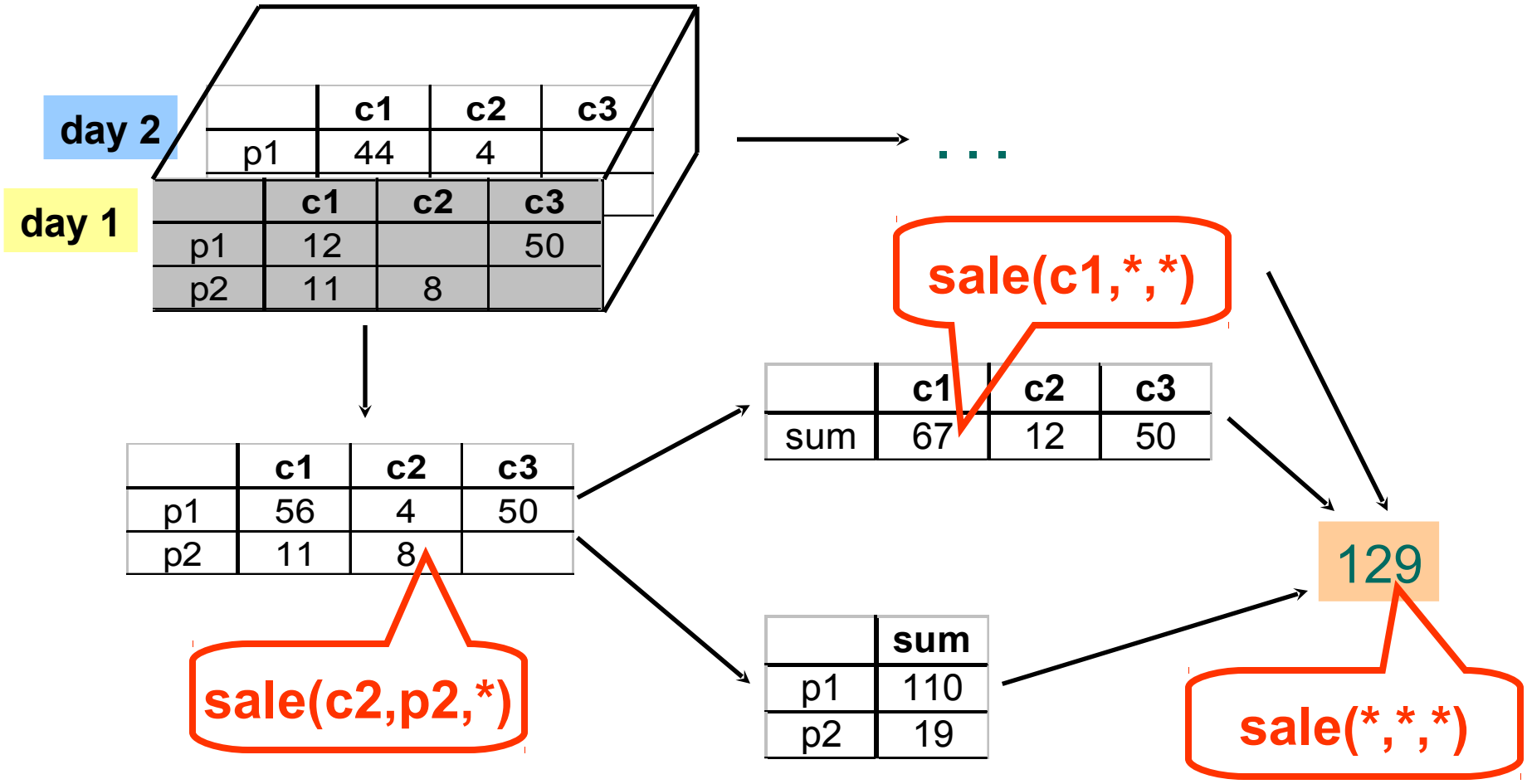


What to Materialize?

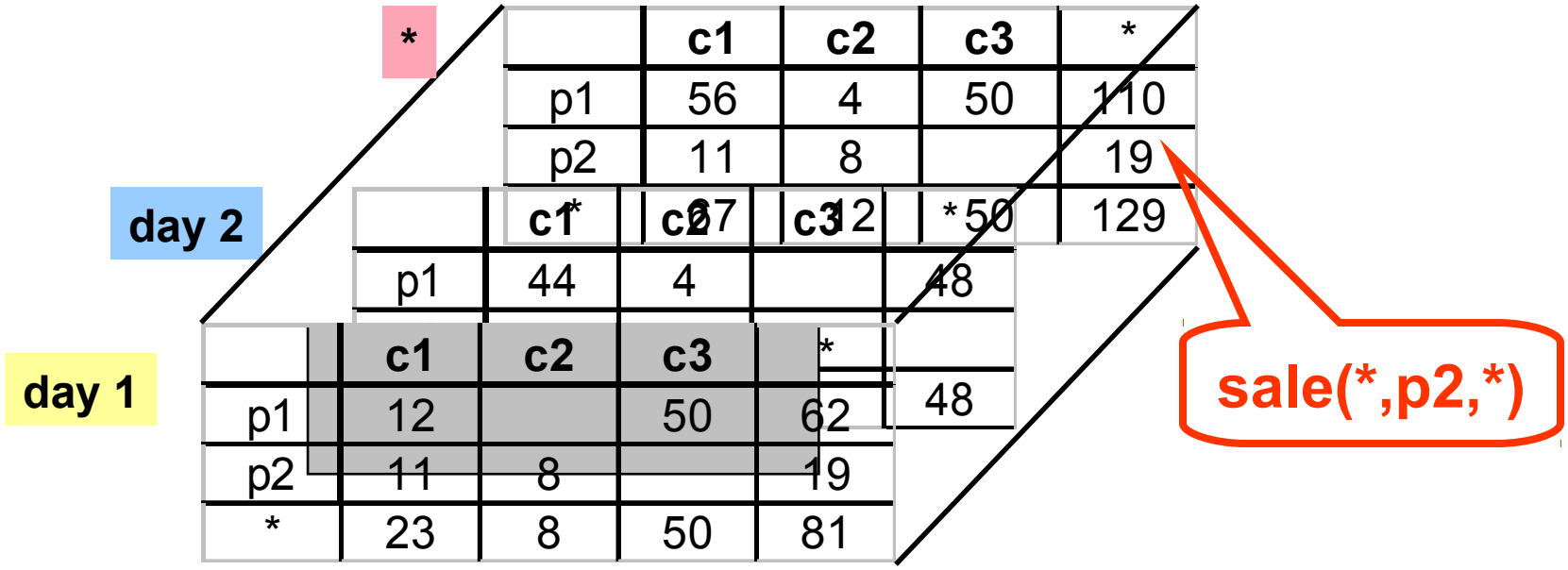
- Store in warehouse results useful for common queries
- Example:



Intermediate Results



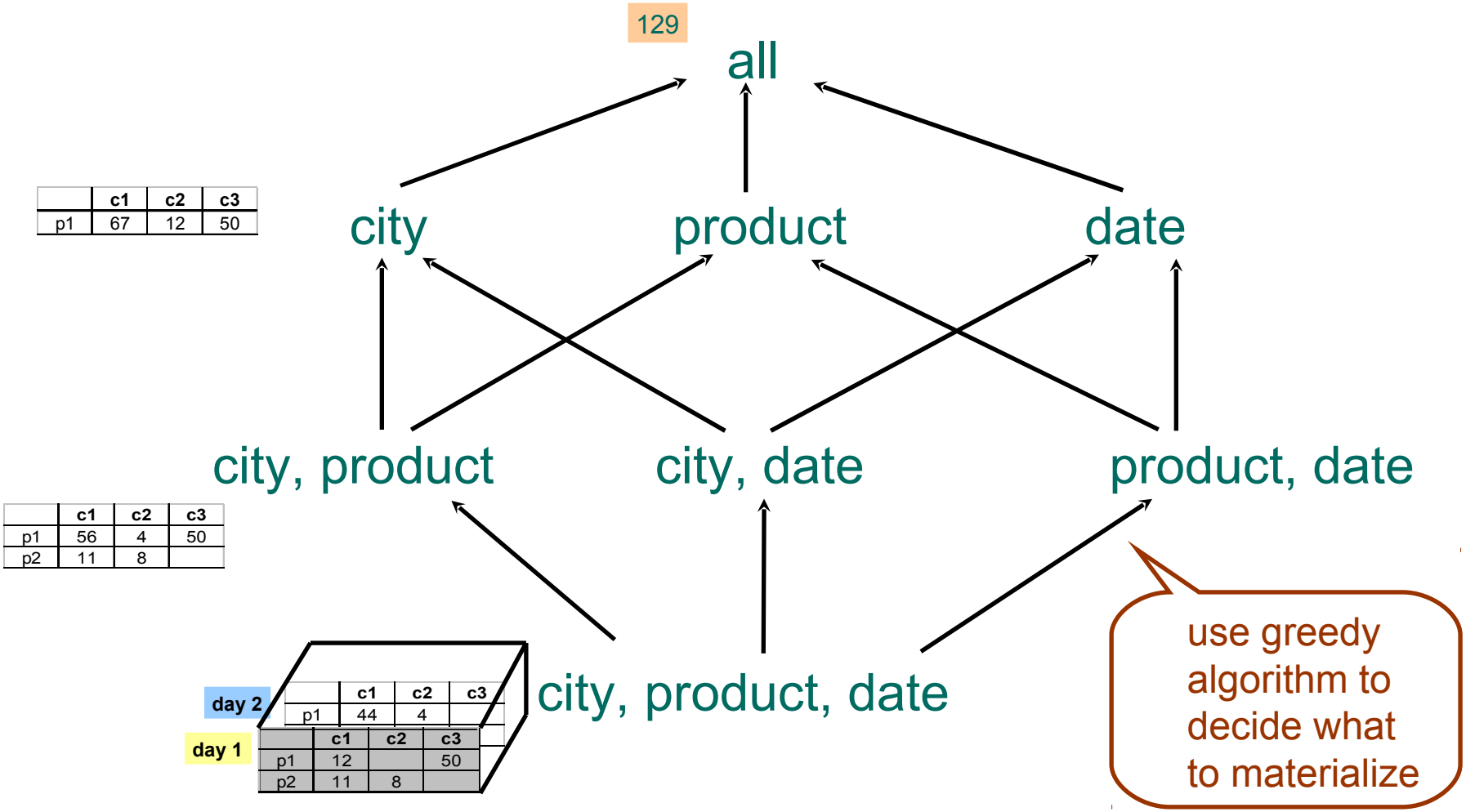
Extended Cube



Materialization Factors

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

Cube Aggregates Lattice



	c1	c2	c3
p1	67	12	50

	c1	c2	c3
p1	56	4	50
p2	11	8	

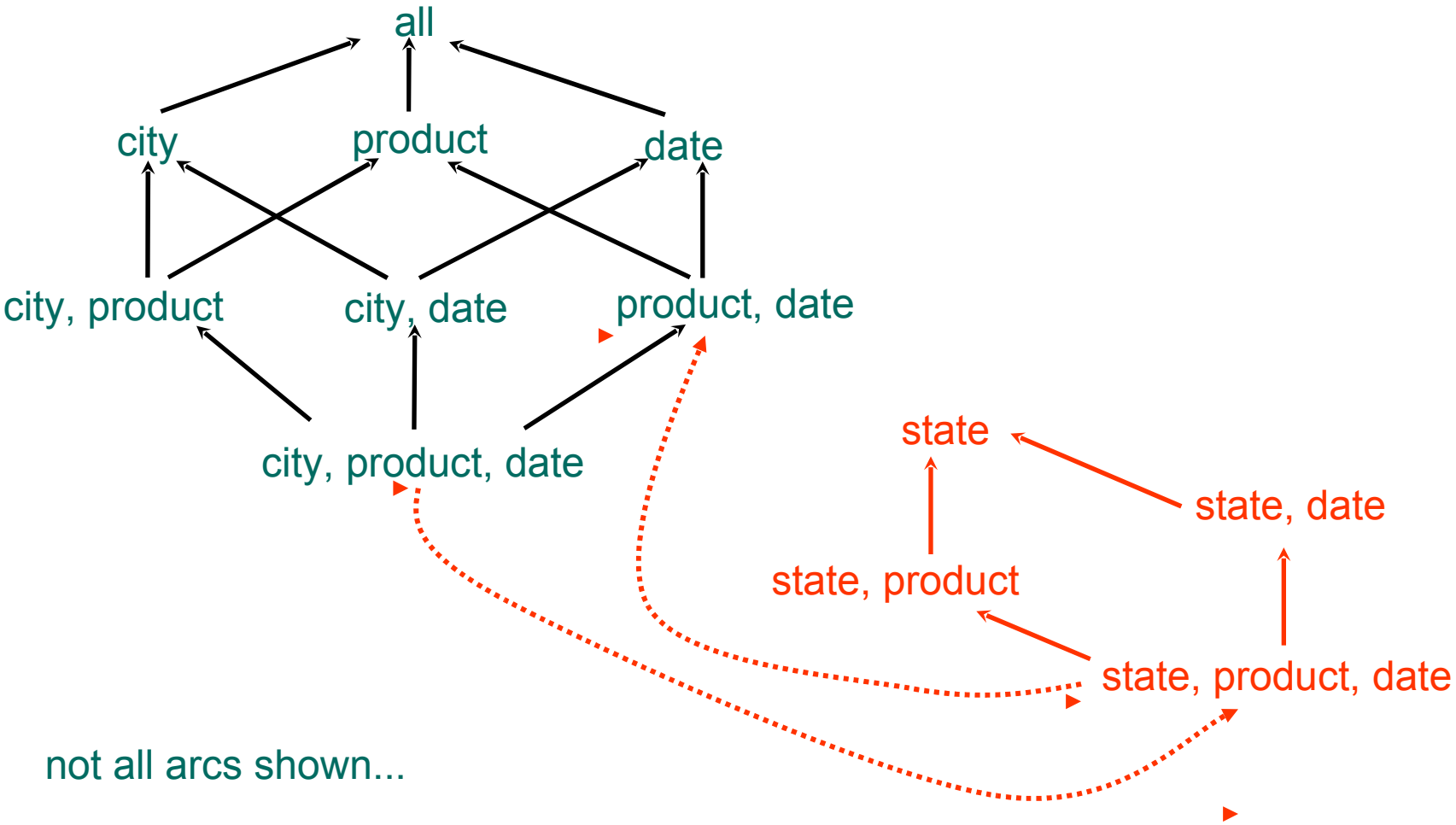
	c1	c2	c3
p1	44	4	

	c1	c2	c3
p1	12		50
p2	11	8	

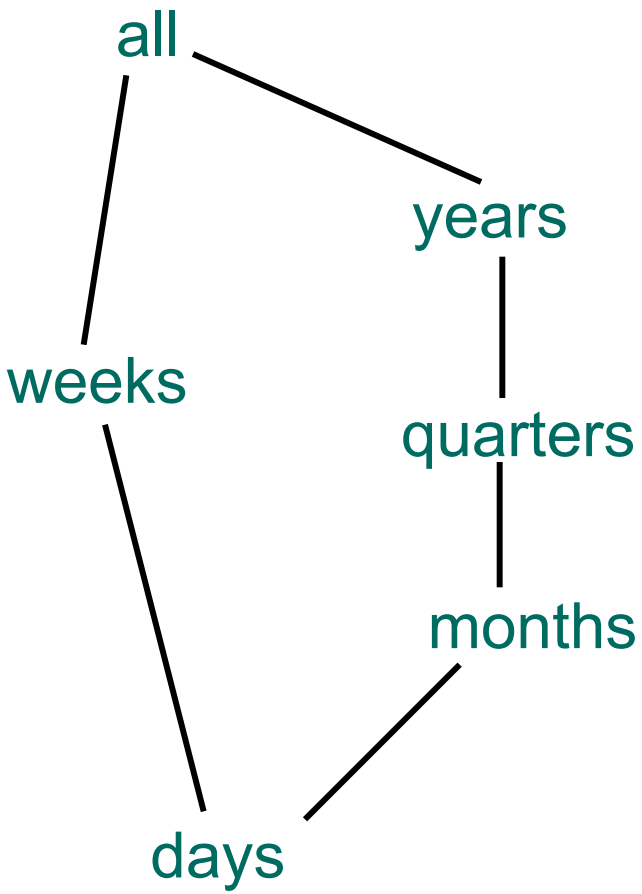
Dimension Hierarchies

cities	city	state
	c1	CA
	c2	NY

Dimension Hierarchies



Interesting Hierarchy

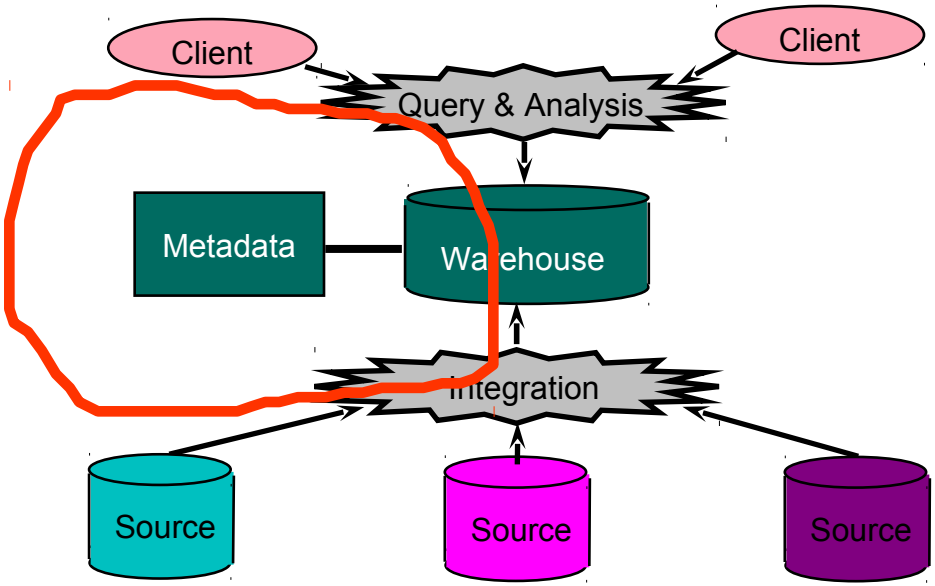


time	day	week	month	quarter	year
	1	1	1	1	2000
	2	1	1	1	2000
	3	1	1	1	2000
	4	1	1	1	2000
	5	1	1	1	2000
	6	1	1	1	2000
	7	1	1	1	2000
	8	2	1	1	2000

conceptual dimension table

Managing

- Metadata
- Tools



Metadata

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

Metadata

- Business
  - business terms & definition
  - data ownership, charging
- Operational
  - data lineage
  - data currency (e.g., active, archived, purged)
  - use stats, error reports, audit trails

Tools

- Development
  - design & edit: schemas, views, scripts, rules, queries, reports
- Planning & Analysis
  - what-if scenarios (schema changes, refresh rates), capacity planning
- Warehouse Management
  - performance monitoring, usage patterns, exception reporting
- System & Network Management
  - measure traffic (sources, warehouse, clients)
- Workflow Management
  - “reliable scripts” for cleaning & analyzing data

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