Data Warehousing Overview
CS245 Notes 12

Hector Garcia-Molina
Stanford University

Warehousing

- Growing industry: $8 billion in 1998
- Range from desktop to huge:
  - Walmart: 900-CPU, 2,700 disk, 23TB Teradata system
- Lots of buzzwords, hype
  - slice & dice, rollup, MOLAP, pivot, ...

Outline

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

What is a Warehouse?

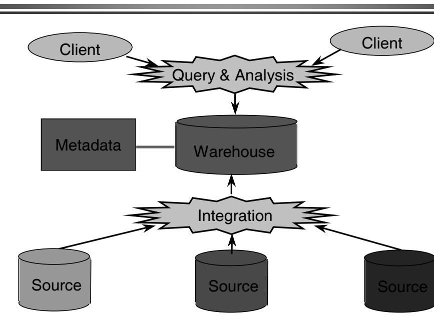
- Collection of diverse data
  - subject oriented
  - aimed at executive, decision maker
  - often a copy of operational data
  - with value-added data (e.g., summaries, history)
  - integrated
  - time-varying
  - non-volatile

more

What is a Warehouse?

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

Warehouse Architecture

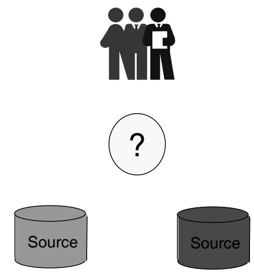


Motivating Examples

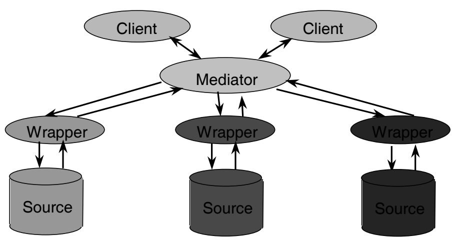
- Forecasting
- Comparing performance of units
- Monitoring, detecting fraud
- Visualization

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

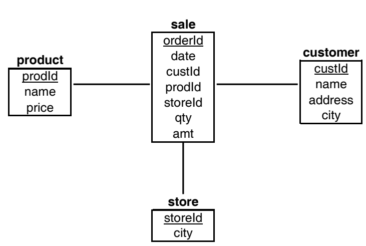
product	prodId	name	price
	p1	bolt	10
	p2	nut	5

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

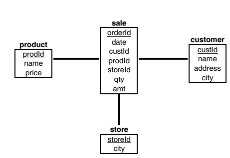
customer	custId	name	address	city
	53	joe	10 main	sfo
	81	fred	12 main	sfo
	111	sally	80 willow	la

Star Schema

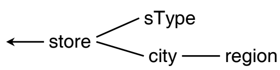


Terms

- 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
➔ constellations

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	

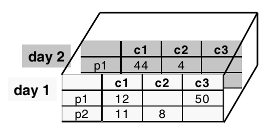
dimensions = 2

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 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

Another Example

- Add up amounts by day, product
- In SQL: SELECT 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 →

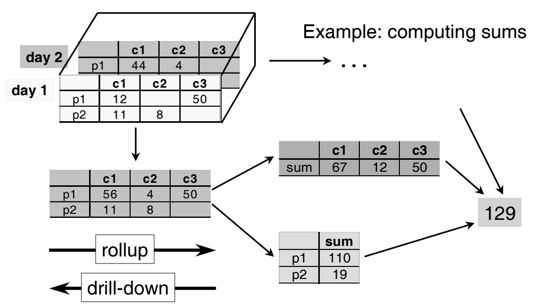
← drill-down

Aggregates

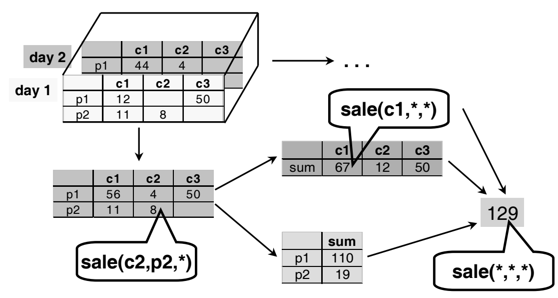
- Operators: sum, count, max, min, median, ave
- "Having" clause
- Using dimension hierarchy
  - average by region (within store)
  - maximum by month (within date)

Cube Aggregation

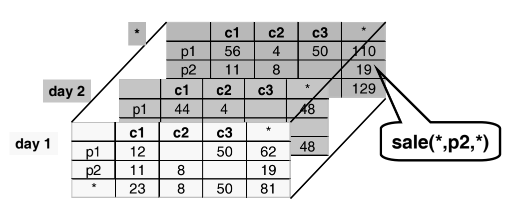
Example: computing sums



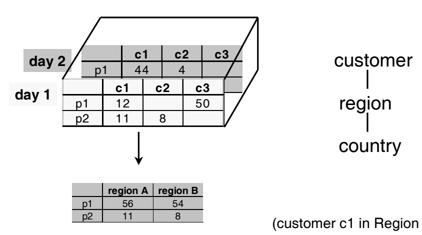
Cube Operators



Extended Cube



Aggregation Using Hierarchies



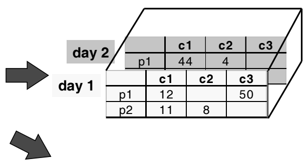
(customer c1 in Region A; customers c2, c3 in Region B)

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	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

Data Mining

- Decision Trees
- Clustering
- Association Rules

Decision Trees

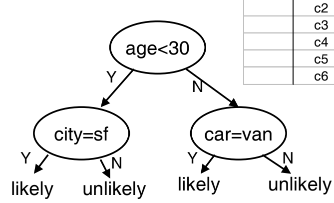
Example:
- Conducted survey to see what customers were interested in new model car
- Want to select customers for advertising campaign

sale	custId	car	age	city	newCar
	c1	taurus	27	sf	yes
	c2	van	35	la	yes
	c3	van	40	sf	yes
	c4	taurus	22	sf	yes
	c5	merc	50	la	no
	c6	taurus	25	la	no

training set

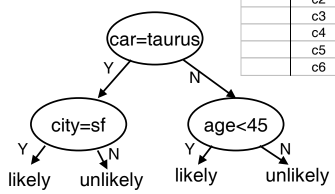
One Possibility

sale	custId	car	age	city	newCar
	c1	taurus	27	sf	yes
	c2	van	35	la	yes
	c3	van	40	sf	yes
	c4	taurus	22	sf	yes
	c5	merc	50	la	no
	c6	taurus	25	la	no



Another Possibility

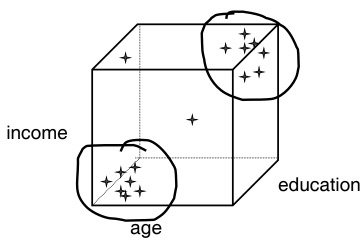
sale	custId	car	age	city	newCar
	c1	taurus	27	sf	yes
	c2	van	35	la	yes
	c3	van	40	sf	yes
	c4	taurus	22	sf	yes
	c5	merc	50	la	no
	c6	taurus	25	la	no



Issues

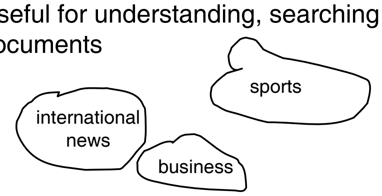
- Decision tree cannot be “too deep”
  - → would not have statistically significant amounts of data for lower decisions
- Need to select tree that most reliably predicts outcomes

Clustering



Another Example: Text

- Each document is a vector
  - e.g., <100110...> contains words 1,4,5,...
- Clusters contain “similar” documents
- Useful for understanding, searching documents



Issues

- Given desired number of clusters?
- Finding “best” clusters
- Are clusters semantically meaningful?
  - e.g., “yuppies’’ cluster?
- Using clusters for disk storage

Association Rule Mining

sales records:

transaction id	customer id	products bought
tran1	cust33	p2, p5, p8
tran2	cust45	p5, p8, p11
tran3	cust12	p1, p9
tran4	cust40	p5, p8, p11
tran5	cust12	p2, p9
tran6	cust12	p9

market-basket data

- Trend: Products p5, p8 often bough together
- Trend: Customer 12 likes product p9

Association Rule

- <u>Rule</u>: $\{p_1, p_3, p_8\}$
- <u>Support</u>: number of baskets where these products appear
- <u>High-support set</u>: support ≥ threshold *s*
- <u>Problem</u>: find all high support sets

Finding High-Support <u>Pairs</u>

- Baskets(basket, item)
- SELECT I.item, J.item, COUNT(I.basket)
  FROM Baskets I, Baskets J
  WHERE I.basket = J.basket AND
  I.item < J.item
  GROUP BY I.item, J.item
  HAVING COUNT(I.basket) >= s;

WHY?

Example

basket	item
t1	p2
t1	p5
t1	p8
t2	p5
t2	p8
t2	p11
...	...

→

basket	item1	item2
t1	p2	p5
t1	p2	p8
t1	p5	p8
t2	p5	p8
t2	p5	p11
t2	p8	p11
...	...	...

check if count ≥ s

Issues

- Performance for size 2 rules

big

basket	item
t1	p2
t1	p5
t1	p8
t2	p5
t2	p8
t2	p11
...	...

basket	item1	item2
t1	p2	p5
t1	p2	p8
t1	p5	p8
t2	p5	p8
t2	p5	p11
t2	p8	p11
...	...	...

even bigger!

- Performance for size *k* rules

Implementing a Warehouse

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

Monitoring

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

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

new

Monitoring Techniques

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

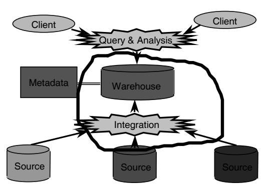
Advantages & Disadvantages!!

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., mail list, customer merging)

billing DB → customer1(Joe) → merged_customer(Joe)

service DB → customer2(Joe) → merged_customer(Joe)

- Auditing: discover rules & relationships (like data mining)

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

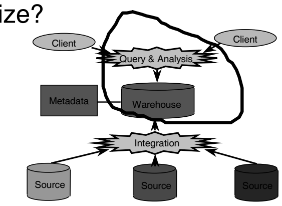
product	id	name	price
	p1	bolt	10
	p2	nut	5

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

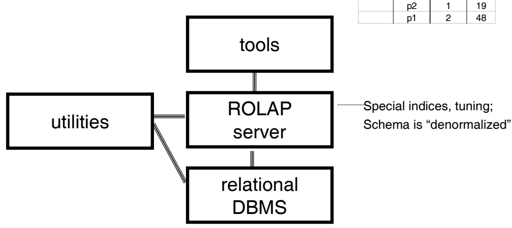
- ROLAP servers vs. MOLAP servers
- Index Structures
- What to Materialize?
- Algorithms



ROLAP Server

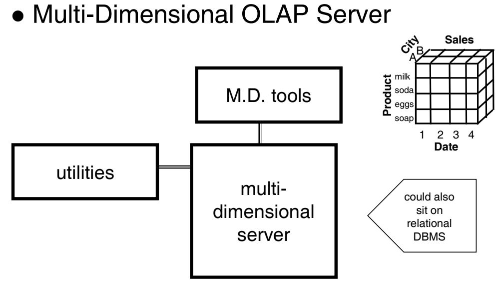
- Relational OLAP Server

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



MOLAP Server

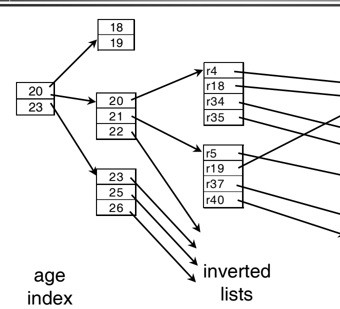
- Multi-Dimensional OLAP Server



Index Structures

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

Inverted Lists



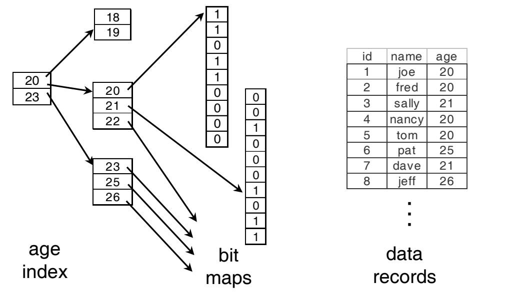
rId	name	age
r4	joe	20
r18	fred	20
r19	sally	21
r34	nancy	20
r35	tom	20
r36	pat	25
r5	dave	21
r41	jeff	26

Using Inverted Lists

- Query:
  - Get people with age = 20 and name = “fred”
- List for age = 20: r4, r18, r34, r35
- List for name = “fred”: r18, r52
- Answer is intersection: r18

CS 245 Notes12 61

Bit Maps



CS 245 Notes12 62

Using Bit Maps

- Query:
  - Get people with age = 20 and name = “fred”
- List for age = 20: 1101100000
- List for name = “fred”: 0100000001
- Answer is intersection: 010000000000

- Good if domain cardinality small
- Bit vectors can be compressed

CS 245 Notes12 63

Join

- “Combine” SALE, PRODUCT relations
- In SQL: SELECT * FROM SALE, 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

CS 245 Notes12 64

Join Indexes

join index

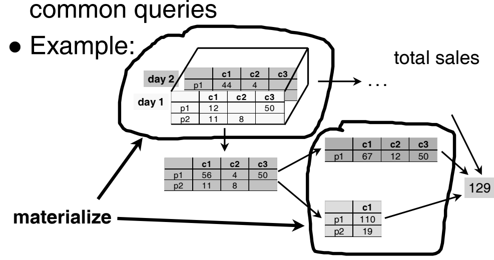
product	id	name	price	jIndex
	p1	bolt	10	r1,r3,r5,r6
	p2	nut	5	r2,r4

sale	rId	prodId	storeId	date	amt
	r1	p1	c1	1	12
	r2	p2	c1	1	11
	r3	p1	c3	1	50
	r4	p2	c2	1	8
	r5	p1	c1	2	44
	r6	p1	c2	2	4

CS 245 Notes12 65

What to Materialize?

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

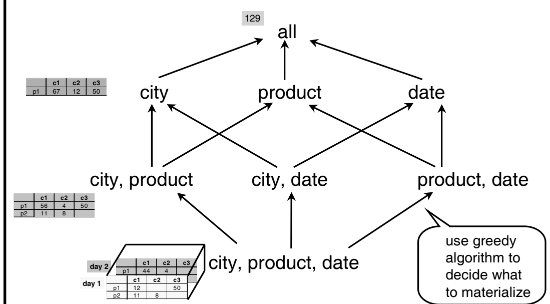


CS 245 Notes12 66

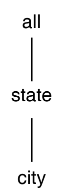
Materialization Factors

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

Cube Aggregates Lattice

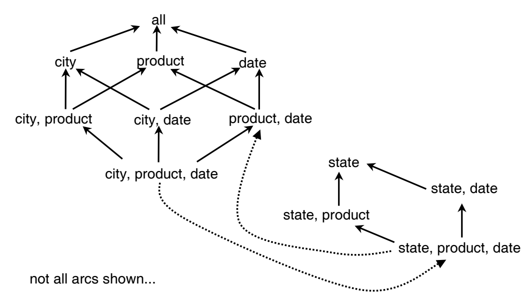


Dimension Hierarchies

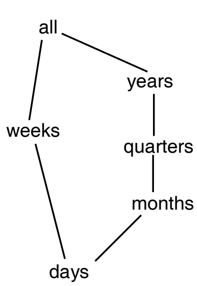


cities	city	state
	c1	CA
	c2	NY

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

Algorithms

- Query Optimization
- Parallel Processing
- Data Mining

Example: Association Rules

- How do we perform rule mining efficiently?
- Observation: If set *X* has support *t*, then each *X* subset must have at least support *t*
- For 2-sets:
  - if we need support *s* for {*i, j*}
  - then each *i, j* must appear in at least *s* baskets

Algorithm for 2-Sets

(1) Find OK products
- those appearing in *s* or more baskets

(2) Find high-support pairs
using only OK products

Algorithm for 2-Sets

- INSERT INTO okBaskets(basket, item)
  SELECT basket, item
  FROM Baskets
  GROUP BY item
  HAVING COUNT(basket) >= s;
- Perform mining on okBaskets

```
SELECT I.item, J.item, COUNT(I.basket)
FROM okBaskets I, okBaskets J
WHERE I.basket = J.basket  AND
        I.item < J.item
GROUP BY I.item, J.item
HAVING COUNT(I.basket) >= s;
```


Counting Efficiently

- One way:

threshold = 3

basket	I.item	J.item
t1	p5	p8
t2	p5	p8
t2	p8	p11
t3	p2	p3
t3	p5	p8
t3	p2	p8
...	...	...

sort →

basket	I.item	J.item
t3	p2	p3
t3	p2	p8
t1	p5	p8
t2	p5	p8
t3	p5	p8
t2	p8	p11
...	...	...

count & remove →

count	I.item	J.item
3	p5	p8
5	p12	p18
...	...	...

Counting Efficiently

- Another way:

threshold = 3

basket	I.item	J.item
t1	p5	p8
t2	p5	p8
t2	p8	p11
t3	p2	p3
t3	p5	p8
t3	p2	p8
...	...	...

scan & count →

count	I.item	J.item
1	p2	p3
2	p2	p8
3	p5	p8
5	p12	p18
1	p21	p22
2	p21	p23
...	...	...

remove →

count	I.item	J.item
3	p5	p8
5	p12	p18
...	...	...

~~keep counter array in memory~~

Yet Another Way

threshold = 3

basket	I.item	J.item
t1	p5	p8
t2	p5	p8
t2	p8	p11
t3	p2	p3
t3	p5	p8
t3	p2	p8
...	...	...

(1) scan & hash & count →

in-memory hash table

count	bucket
1	A
5	B
2	C
1	D
8	E
1	F
...	...

(2) scan & remove

basket	I.item	J.item
t1	p5	p8
t2	p5	p8
t2	p8	p11
t3	p5	p8
t5	p12	p18
t8	p12	p18
...	...	...

false positive

(3) scan& count

in-memory counters

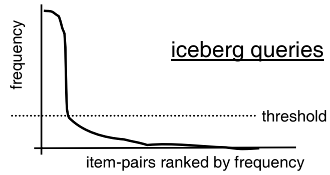
count	I.item	J.item
3	p5	p8
1	p8	p11
5	p12	p18
...	...	...

(4) remove

count	I.item	J.item
3	p5	p8
5	p12	p18
...	...	...

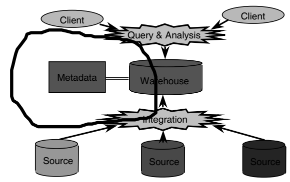
Discussion

- Hashing scheme: 2 (or 3) scans of data
- Sorting scheme: requires a sort!
- Hashing works well if few high-support pairs and many low-support ones



Managing

- Metadata
- Warehouse Design
- 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

Design

- What data is needed?
- Where does it come from?
- How to clean data?
- How to represent in warehouse (schema)?
- What to summarize?
- What to materialize?
- What to index?

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
- What are companies & research labs working on?

Research (1)

- Incremental Maintenance
- Data Consistency
- Data Expiration
- Recovery
- Data Quality
- Error Handling (Back Flush)

Research (2)

- Rapid Monitor Construction
- Temporal Warehouses
- Materialization & Index Selection
- Data Fusion
- Data Mining
- Integration of Text & Relational Data

Conclusions

- Massive amounts of data and complexity of queries will push limits of current warehouses
- Need better systems:
  - easier to use
  - provide quality information