

CSE 562 Database Systems

Data Warehousing Overview

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



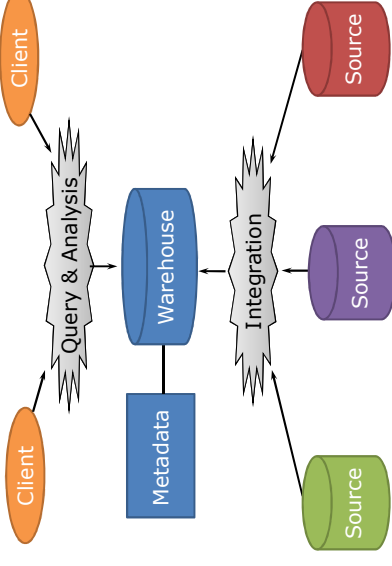
What is a Warehouse?

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

UB CSE 562 Spring 2009

5

Warehouse Architecture



UB CSE 562 Spring 2009

6

Motivating Examples

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

UB CSE 562 Spring 2009

7

Why a Warehouse?

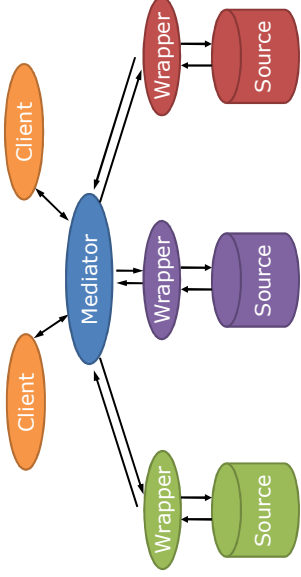
- Two Approaches:
 - Query-Driven (Lazy)
 - Warehouse (Eager)



UB CSE 562 Spring 2009

8

Query-Driven Approach



UB CSE 562 Spring 2009

9

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

UB CSE 562 Spring 2009

10

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

UB CSE 562 Spring 2009

11

OLTP vs. OLAP

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

UB CSE 562 Spring 2009

12

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

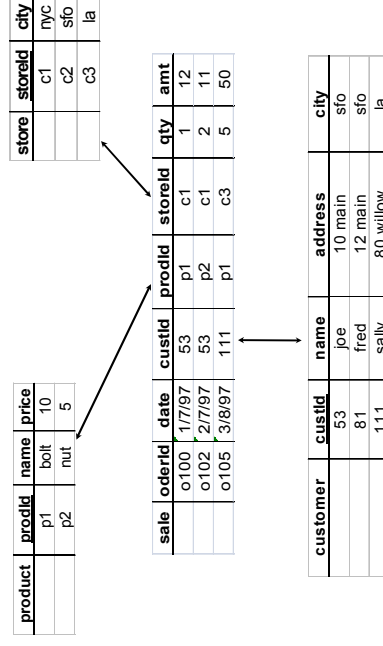
Data Marts

- 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

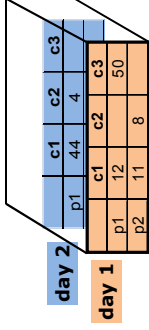


3-D Cube

Fact table view:

sale	prodlid	storelid	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	prodlid	storelid	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	prodlid	storelid	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 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

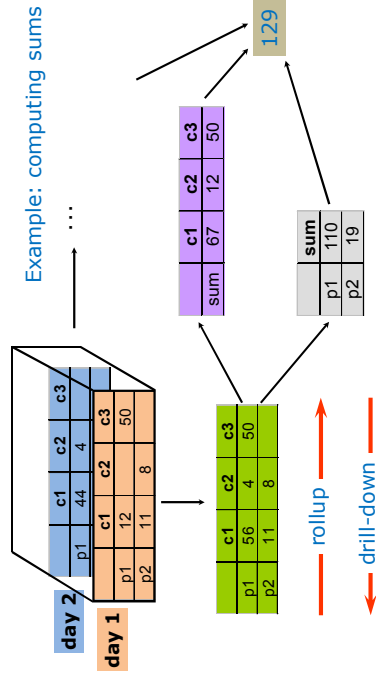
— rollup —→
— drill-down ←—

Aggregates

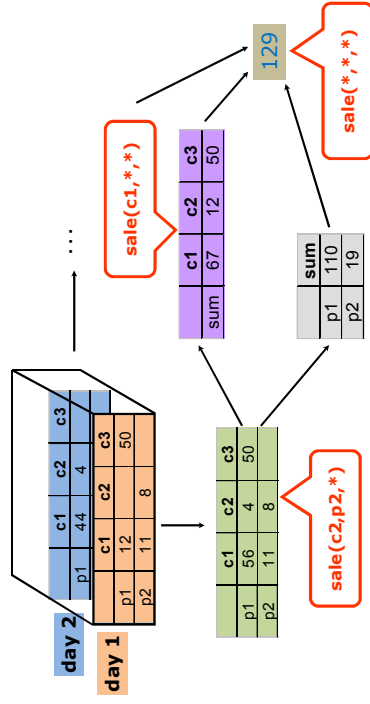
- Operators: sum, count, max, min, median, avg
- "Having" clause
- Using dimension hierarchy
 - average by region (within store)
 - maximum by month (within date)

Cube Aggregation

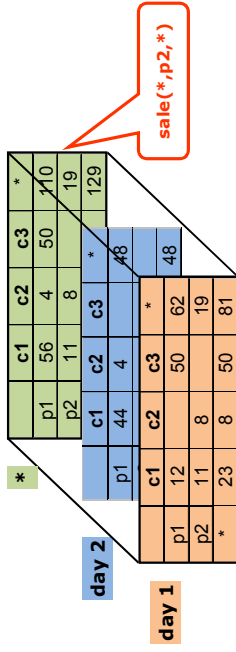
Example: computing sums



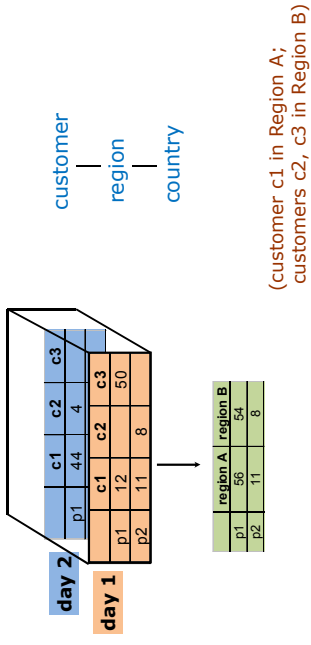
Cube Operators



Extended Cube



Aggregation Using Hierarchies

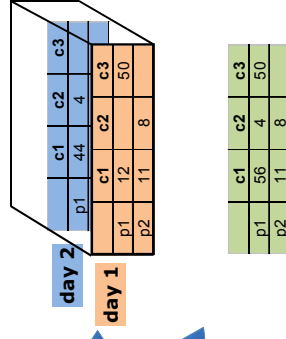


Pivoting

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:



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

UB CSE 562 Spring 2009

33

Implementing a Warehouse

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

UB CSE 562 Spring 2009

34

Monitoring

- Source Types: relational, flat file, IMS, VSAM, IDMS, 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



UB CSE 562 Spring 2009

35

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!!



UB CSE 562 Spring 2009

36

Monitoring Issues

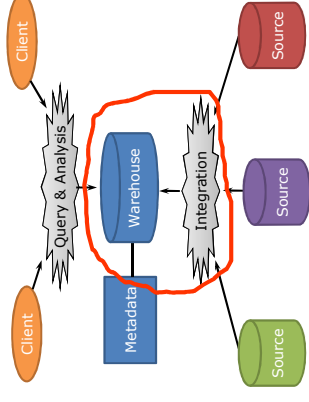
- Frequency
 - periodic: daily, weekly, ...
 - triggered: on “big” change, lots of changes, ...
- Data transformation
 - convert data to uniform format
 - remove & add fields (e.g., add date to get history)
- Standards (e.g., ODBC)

UB CSE 562 Spring 2009

37

Integration

- Data Cleaning
- Data Loading
- Derived Data



UB CSE 562 Spring 2009

38

Data Cleaning

- Migration (e.g., yen ⇔ dollars)
- Scrubbing: use domain-specific knowledge (e.g., social security numbers)
- Fusion (e.g., mail list, customer merging)



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

UB CSE 562 Spring 2009

39

Loading Data

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

UB CSE 562 Spring 2009

40

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

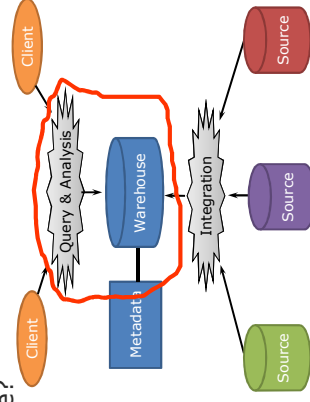
sale	prodid	storeid	date	amt	product	id	name	price
	p1	c1	1	12		p1	bolt	10
	p2	c1	1	11		p2	nut	5
	p1	c3	1	50				
	p2	c2	1	8				
	p1	c1	2	44				
	p1	c2	2	4				

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

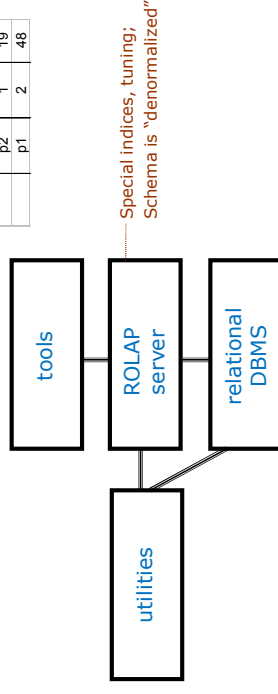
- ROLAP servers vs. MOLAP servers
- Index Structures
- What to Materialize?
- Algorithms



ROLAP Server

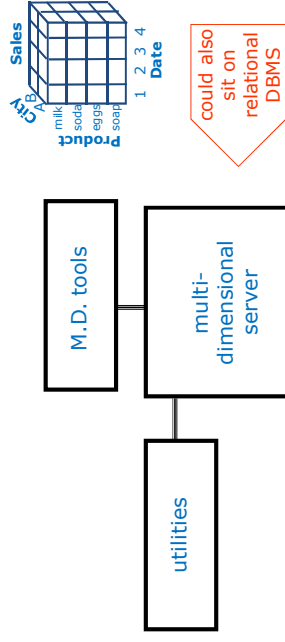
- Relational OLAP Server

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



MOLAP Server

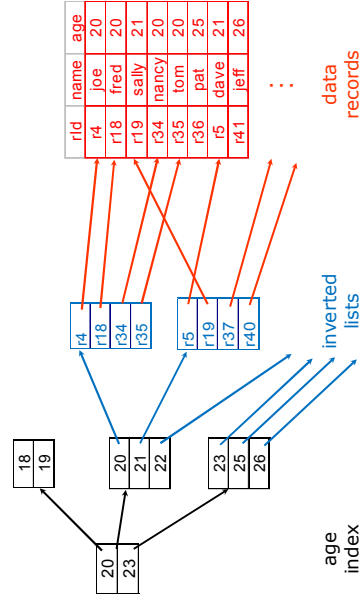
- Multi-Dimensional OLAP Server



Index Structures

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

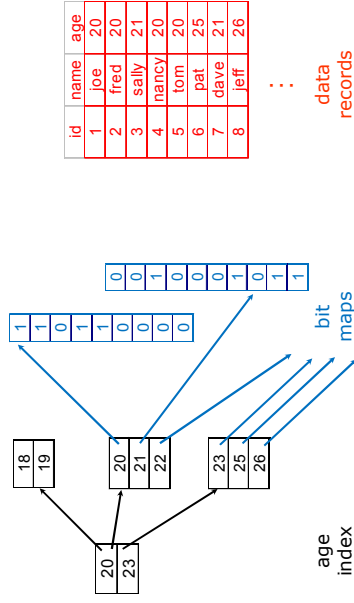
Inverted Lists



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

Bit Maps

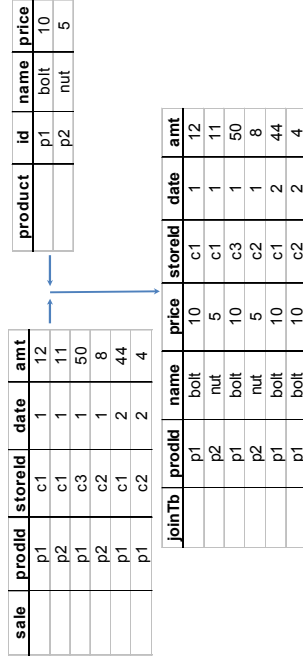


Using Bit Maps

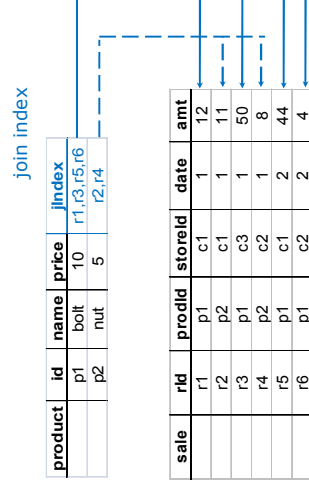
- Query:
 - Get people with age = 20 and name = "fred"
- List for age = 20: **1101100000**
- List for name = "fred": **0100000001**
- Answer is intersection: **0100000000**
- Good if domain cardinality small
- Bit vectors can be compressed

Join

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

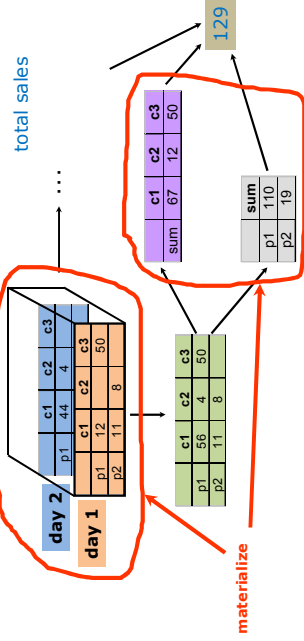


Join Indexes



What to Materialize?

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



UB CSE 562 Spring 2009

53

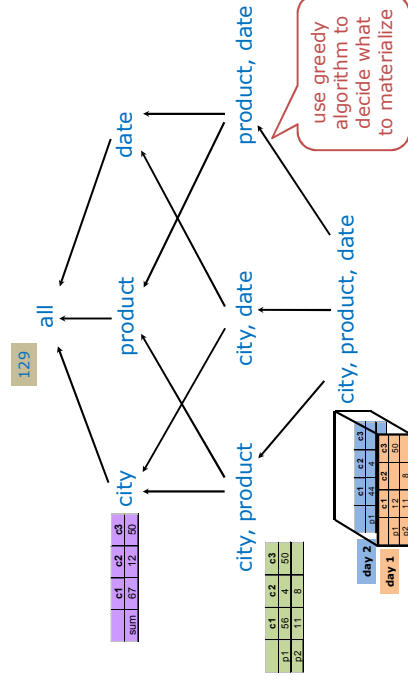
Materialization Factors

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

UB CSE 562 Spring 2009

54

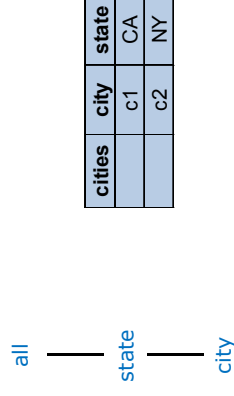
Cube Aggregates Lattice



UB CSE 562 Spring 2009

55

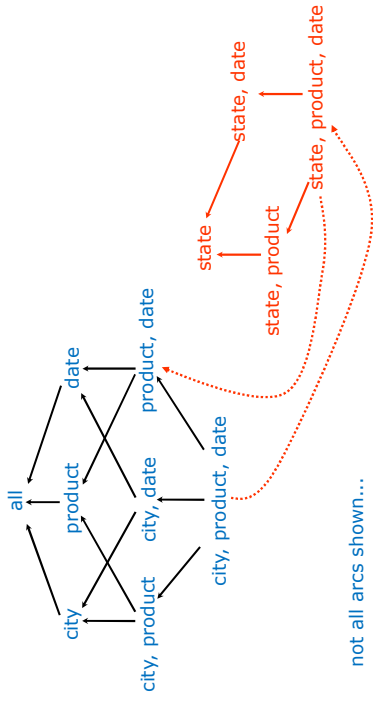
Dimension Hierarchies



UB CSE 562 Spring 2009

56

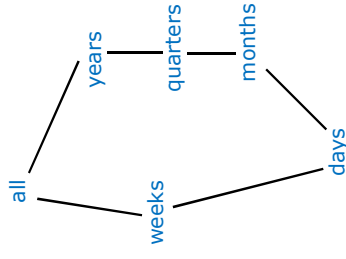
Dimension Hierarchies



UB CSE 562 Spring 2009

57

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

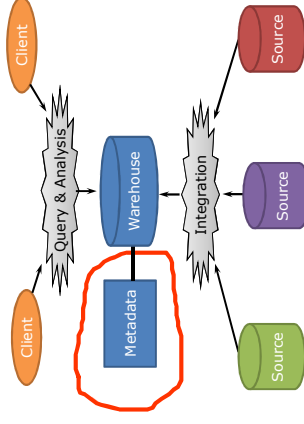
conceptual dimension table

UB CSE 562 Spring 2009

58

Managing

- Metadata
- Warehouse Design
- Tools



UB CSE 562 Spring 2009

59

Metadata

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

UB CSE 562 Spring 2009

60

Metadata

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

UB CSE 562 Spring 2009

61

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?

UB CSE 562 Spring 2009

62

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

UB CSE 562 Spring 2009

63

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

UB CSE 562 Spring 2009

64

Future Directions

- Better performance
- Larger warehouses
- Easier to use
- What are companies & research labs working on?

UB CSE 562 Spring 2009

65

Research (1)

- Incremental Maintenance
- Data Consistency
- Data Expiration
- Recovery
- Data Quality
- Error Handling

UB CSE 562 Spring 2009

66

Research (2)

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

UB CSE 562 Spring 2009

67

Conclusions

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

UB CSE 562 Spring 2009

68