Week 13: Data Warehousing

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

Warehouse Architecture

![Diagram of Warehouse Architecture]

- Client
  - Query & Analysis
    - Metadata
    - Warehouse
    - Integration
      - Source
      - Source
      - Source

Client
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

<table>
<thead>
<tr>
<th>OLTP</th>
<th>OLAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Mostly updates</td>
<td>- Mostly reads</td>
</tr>
<tr>
<td>- Many small transactions</td>
<td>- Queries long, complex</td>
</tr>
<tr>
<td>- Mb-Tb of data</td>
<td>- Gb-Tb of data</td>
</tr>
<tr>
<td>- Raw data</td>
<td>- Summarized, consolidated data</td>
</tr>
<tr>
<td>- Clerical users</td>
<td>- Decision-makers, analysts as users</td>
</tr>
<tr>
<td>- Up-to-date data</td>
<td></td>
</tr>
<tr>
<td>- Consistency, recoverability critical</td>
<td></td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>product</th>
<th>prodId</th>
<th>name</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>bolt</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>p2</td>
<td>nut</td>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>store</th>
<th>storeId</th>
<th>city</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>nyc</td>
<td></td>
</tr>
<tr>
<td>c2</td>
<td>sfo</td>
<td></td>
</tr>
<tr>
<td>c3</td>
<td>la</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sale</th>
<th>orderId</th>
<th>date</th>
<th>custId</th>
<th>prodId</th>
<th>storeId</th>
<th>qty</th>
<th>amt</th>
</tr>
</thead>
<tbody>
<tr>
<td>o100</td>
<td>1/7/97</td>
<td>53</td>
<td>p1</td>
<td>c1</td>
<td>1</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>o102</td>
<td>2/7/97</td>
<td>53</td>
<td>p2</td>
<td>c1</td>
<td>2</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3/8/97</td>
<td>111</td>
<td>p1</td>
<td>c3</td>
<td>5</td>
<td>50</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>customer</th>
<th>custId</th>
<th>name</th>
<th>address</th>
<th>city</th>
</tr>
</thead>
<tbody>
<tr>
<td>53</td>
<td>joe</td>
<td>10 main</td>
<td>sfo</td>
<td></td>
</tr>
<tr>
<td>81</td>
<td>fred</td>
<td>12 main</td>
<td>sfo</td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>sally</td>
<td>80 willow</td>
<td>la</td>
<td></td>
</tr>
</tbody>
</table>
Star Schema

Terms

- Fact table
- Dimension tables
- Measures
Dimension Hierarchies

store | sType | city | region
------|-------|-----|-----
s5   | sfo   | t1  | joe
s7   | sfo   | t2  | fred
s9   | la    | t1  | nancy

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

sType | tld | size  | location
------|-----|-------|-------
t1    | small | downtown
      | large | suburbs

region | regId | name
-------|-------|-----
north  |       | cold region
       |       | south warm region

Cube

Fact table view:

<table>
<thead>
<tr>
<th>sale</th>
<th>prodId</th>
<th>storeId</th>
<th>amt</th>
</tr>
</thead>
</table>
p1   | c1     | 12      |     |
p2   | c1     | 11      |     |
p1   | c3     | 50      |     |
p2   | c2     | 8       |     |

Multi-dimensional cube:

<table>
<thead>
<tr>
<th>c1</th>
<th>c2</th>
<th>c3</th>
</tr>
</thead>
</table>
p1  | 12 | 50 |
p2  | 11 | 8  |

dimensions = 2
3-D Cube

Fact table view:

<table>
<thead>
<tr>
<th>sale</th>
<th>prodId</th>
<th>storeId</th>
<th>date</th>
<th>amt</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>c1</td>
<td>1</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td>c1</td>
<td>1</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>p1</td>
<td>c3</td>
<td>1</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td>c2</td>
<td>1</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>p1</td>
<td>c1</td>
<td>2</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>p1</td>
<td>c2</td>
<td>2</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>sale</th>
<th>proddId</th>
<th>storeId</th>
<th>date</th>
<th>amt</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>c1</td>
<td></td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>p2</td>
<td>c1</td>
<td></td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>p1</td>
<td>c3</td>
<td></td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>p2</td>
<td>c2</td>
<td></td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>p1</td>
<td>c1</td>
<td></td>
<td>2</td>
<td>44</td>
</tr>
<tr>
<td>p1</td>
<td>c2</td>
<td></td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

81

Aggregates

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

<table>
<thead>
<tr>
<th>ans</th>
<th>date</th>
<th>sum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>48</td>
</tr>
</tbody>
</table>
Another Example

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

<table>
<thead>
<tr>
<th>sale</th>
<th>prodId</th>
<th>storeId</th>
<th>date</th>
<th>amt</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>c1</td>
<td>1</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>p2</td>
<td>c1</td>
<td>1</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>p1</td>
<td>c3</td>
<td>1</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td>c2</td>
<td>1</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>p1</td>
<td>c1</td>
<td>2</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>p1</td>
<td>c2</td>
<td>2</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

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

sale(c1,*,*)
sale(c2,p2,*)
sale(*,*,*)
Extended Cube

Aggregation Using Hierarchies

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

Fact table view:

<table>
<thead>
<tr>
<th>sale</th>
<th>prodId</th>
<th>storeId</th>
<th>date</th>
<th>amt</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>c1</td>
<td>1</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>p2</td>
<td>c1</td>
<td>1</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>p1</td>
<td>c3</td>
<td>1</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>p2</td>
<td>c2</td>
<td>1</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>p1</td>
<td>c1</td>
<td>2</td>
<td>2</td>
<td>44</td>
</tr>
<tr>
<td>p1</td>
<td>c2</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

Multi-dimensional cube:

day 1

<table>
<thead>
<tr>
<th></th>
<th>c1</th>
<th>c2</th>
<th>c3</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>12</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td>11</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

day 2

<table>
<thead>
<tr>
<th></th>
<th>c1</th>
<th>c2</th>
<th>c3</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>44</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td>11</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

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

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., mail list, customer merging)

![Diagram showing data flow]

- 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

<table>
<thead>
<tr>
<th>sale</th>
<th>prodId</th>
<th>storeId</th>
<th>date</th>
<th>amt</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>c1</td>
<td>1</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td>c1</td>
<td>1</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>p1</td>
<td>c3</td>
<td>1</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td>c2</td>
<td>1</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>p1</td>
<td>c1</td>
<td>2</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>p1</td>
<td>c2</td>
<td>2</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>product</th>
<th>id</th>
<th>name</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>bolt</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td>nut</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>joinTb</th>
<th>prodId</th>
<th>name</th>
<th>price</th>
<th>storeId</th>
<th>date</th>
<th>amt</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>bolt</td>
<td>10</td>
<td>c1</td>
<td>1</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td>nut</td>
<td>5</td>
<td>c1</td>
<td>1</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>p1</td>
<td>bolt</td>
<td>10</td>
<td>c3</td>
<td>1</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td>nut</td>
<td>5</td>
<td>c2</td>
<td>1</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>p1</td>
<td>bolt</td>
<td>10</td>
<td>c1</td>
<td>2</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>p1</td>
<td>bolt</td>
<td>10</td>
<td>c2</td>
<td>2</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

Join Tb does not exist at any source.
Processing

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

ROLAP Server

- Relational OLAP Server

<table>
<thead>
<tr>
<th>sale</th>
<th>prodId</th>
<th>date</th>
<th>sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>1</td>
<td>62</td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td>1</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>p1</td>
<td>2</td>
<td>48</td>
<td></td>
</tr>
</tbody>
</table>

Special indices, tuning; Schema is "denormalized"
MOLAP Server

- Multi-Dimensional OLAP Server

![Diagram of M.D. tools and multi-dimensional server](image)

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
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
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