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

Motivating Examples

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

Warehouse Architecture

Why a Warehouse?

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

![Diagram](https://via.placeholder.com/150)

**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

**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

**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

Warehouse Models & Operators

- Data Models
  - relations
  - stars & snowflakes
  - cubes

- Operators
  - slice & dice
  - roll-up, drill down
  - pivoting
  - other

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?
Star Schema

- Product
  - prodId
  - name
  - price

- Order
  - orderId
  - date
  - custId
  - prodId
  - storeId
  - qty
  - amt

- Customer
  - custId
  - name
  - address
  - city

- Store
  - storeId
  - city

Dimension Hierarchies

- Store
  - storeId
  - cityId
  - tid
  - mgr
  - sType
  - city
  - region

- Customer
  - custId
  - name
  - address
  - city

- Product
  - prodId
  - name
  - price

- Store
  - storeId
  - cityId
  - pop
  - regId

- Region
  - name
  - regId

- City
  - name
  - pop
  - regId

- Snowflake schema
  - Constellations

Terms

- Fact table
- Dimension tables
- Measures

Cube

- Fact table view:
  - sale
  - prodId
  - storeId
  - amt

- Multi-dimensional cube:
  - dimensions = 2

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

<table>
<thead>
<tr>
<th>c1</th>
<th>c2</th>
<th>c3</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>11</td>
<td>8</td>
</tr>
</tbody>
</table>
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>1</td>
<td>12</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>80</td>
<td>50</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>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>

- **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>1</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>48</td>
<td></td>
</tr>
</tbody>
</table>

<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>
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>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>
</tbody>
</table>

Cube Aggregation

Example: computing sums

Cube Operators

Aggregates

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

Pivoting

Aggregation Using Hierarchies

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)

\[ \text{billing DB} \rightarrow \text{customer1}(\text{Joe}) \rightarrow \text{merged\_customer}(\text{Joe}) \]

\[ \text{service DB} \rightarrow \text{customer2}(\text{Joe}) \]

- 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

<table>
<thead>
<tr>
<th>sale</th>
<th>prodid</th>
<th>storeId</th>
<th>date</th>
<th>amount</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>amount</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>

ROLAP Server

- Relational OLAP Server
  - Special indices, tuning; Schema is "denormalized"
MOLAP Server

- Multi-Dimensional OLAP Server

```
  M.D. tools
  |      |
  |      |
  | utilities

multi-dimensional 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

- 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

Using Inverted Lists

```
+---+-----+-----+-----+
|    |    |    | 20 |
+---+-----+-----+-----+
| 23 | 20  | 20  |
+---+-----+-----+-----+
| 23 | 21  | 22  |
+---+-----+-----+-----+
| 23 | 25  | 26  |
+---+-----+-----+-----+

rid | name | age
---+-----+-----
 4  | joe  | 20
 4  | fred | 20
 34 | sally| 21
 35 | nancy| 20
 35 | tom  | 20
 36 | pat  | 25
 5  | dave | 21
 41 | jeff | 26

```

age index

inverted lists

data records
Managing

- Metadata
- Warehouse Design
- Tools

Metadata

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

Metadata

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

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

Future Directions

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

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

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