Data Warehousing
Logical Database Design

Esteban Zimányi
ezimanyi@ulb.ac.be
Slides by Toon Calders
Outline

• Logical Database Design
  – Star schema
  – Snowflake schema

Chapter 8 of Golfarelli & Rizzi
ROLAP

• Relational OLAP
  – OLAP layer built on top of relational database
    • Facts, dimensions, hierarchies encoded as relations
  – Multi-dimensional data warehouse data structure

• Also:
  – MOLAP: Multi-dimensional OLAP = native support for datacubes
  – HOLAP: Hybrid form
Example of a Star Schema
Star Schema

• Dimension tables are not normalized
  – No consistency problems (given your ETL is fine)
  – Avoids the need for joins
  – Use surrogate key

• Dimensions such as Date are materialized

• Key for the fact table consists of the foreign keys to the dimension tables
  – Although some discussion here; case dependent
Why Surrogate Key

<table>
<thead>
<tr>
<th>CID</th>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>John</td>
<td>Dallas</td>
</tr>
<tr>
<td>002</td>
<td>Mary</td>
<td>Dallas</td>
</tr>
<tr>
<td>003</td>
<td>Pete</td>
<td>New York</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CID</th>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>John</td>
<td>New York</td>
</tr>
<tr>
<td>002</td>
<td>Mary</td>
<td>New York</td>
</tr>
<tr>
<td>004</td>
<td>Mark</td>
<td>Dallas</td>
</tr>
</tbody>
</table>

Customer

Sales

<table>
<thead>
<tr>
<th>CID</th>
<th>Product</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>Gun</td>
<td>5$</td>
</tr>
<tr>
<td>002</td>
<td>Beef</td>
<td>20$</td>
</tr>
<tr>
<td>003</td>
<td>Lava lamp</td>
<td>150$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CID</th>
<th>Product</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>Gun</td>
<td>15$</td>
</tr>
<tr>
<td>004</td>
<td>Pork</td>
<td>15$</td>
</tr>
<tr>
<td>004</td>
<td>Lava lamp</td>
<td>160$</td>
</tr>
</tbody>
</table>
Why Surrogate Key

• Star schema without surrogate key

Customer

<table>
<thead>
<tr>
<th>CID</th>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>John</td>
<td>New York</td>
</tr>
<tr>
<td>002</td>
<td>Mary</td>
<td>New York</td>
</tr>
<tr>
<td>003</td>
<td>Pete</td>
<td>New York</td>
</tr>
<tr>
<td>004</td>
<td>Mark</td>
<td>Dallas</td>
</tr>
</tbody>
</table>

Sales

<table>
<thead>
<tr>
<th>CID</th>
<th>PID</th>
<th>Date</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>G</td>
<td>2000</td>
<td>5$</td>
</tr>
<tr>
<td>001</td>
<td>G</td>
<td>2010</td>
<td>15$</td>
</tr>
<tr>
<td>002</td>
<td>B</td>
<td>2000</td>
<td>20$</td>
</tr>
<tr>
<td>003</td>
<td>L</td>
<td>2000</td>
<td>150$</td>
</tr>
<tr>
<td>004</td>
<td>P</td>
<td>2010</td>
<td>15$</td>
</tr>
<tr>
<td>004</td>
<td>L</td>
<td>2010</td>
<td>160$</td>
</tr>
</tbody>
</table>
Why Surrogate Key

- Star schema with surrogate key

<table>
<thead>
<tr>
<th>Key</th>
<th>OLTP key</th>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>001</td>
<td>John</td>
<td>Dallas</td>
</tr>
<tr>
<td>S2</td>
<td>002</td>
<td>Mary</td>
<td>Dallas</td>
</tr>
<tr>
<td>S3</td>
<td>003</td>
<td>Pete</td>
<td>New York</td>
</tr>
<tr>
<td>S4</td>
<td>004</td>
<td>Mark</td>
<td>Dallas</td>
</tr>
<tr>
<td>S5</td>
<td>001</td>
<td>John</td>
<td>New York</td>
</tr>
<tr>
<td>S6</td>
<td>002</td>
<td>Mary</td>
<td>New York</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key</th>
<th>PID</th>
<th>Date</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>G</td>
<td>2000</td>
<td>5$</td>
</tr>
<tr>
<td>S5</td>
<td>G</td>
<td>2010</td>
<td>15$</td>
</tr>
<tr>
<td>S2</td>
<td>B</td>
<td>2000</td>
<td>20$</td>
</tr>
<tr>
<td>S3</td>
<td>L</td>
<td>2000</td>
<td>150$</td>
</tr>
<tr>
<td>S4</td>
<td>P</td>
<td>2010</td>
<td>15$</td>
</tr>
<tr>
<td>S4</td>
<td>L</td>
<td>2010</td>
<td>160$</td>
</tr>
</tbody>
</table>
Why Surrogate Key

• Recall that data warehouse is non-volatile
  – Data can be updated
    • same customer, different address
    • discount gets assigned same code as earlier one
  – In operational database: overwrite
  – Not desirable in data warehouse
    • Incorrect data aggregation; when in 2010 querying sales in Dallas for 2000, John’s purchases should count for Dallas, not New York
Snowflake Schema

Order
- Order No
- Order Date

Customer
- Customer No
- Customer Name
- Customer Address
- City

Salesperson
- SalespersonID
- SalespersonName
- City
- Quota

Fact Table
- OrderNO
- CustomerNO
- SalespersonID
- ProdNo
- DateKey
- CityName
- Quantity
- Total Price

Product
- ProductNO
- ProdName
- ProdDescr
- CategoryName
- UnitPrice

Date
- DateKey
- Date
- Month

City
- CityName
- StateName

Month
- Year

State
- StateName

Country

Category
- CategoryName
- CategoryDescr
Snowflake Scheme

• More normalized dimensions
  – Avoid redundancy
  – Reduce size of large dimension
    • Customer → City → Country
  – More efficient data insertion

• Disadvantages: need for joins
  – However, join on foreign key; tables index on primary key
Fact Constellation

- Multiple fact tables share the same dimensions
  - E.g., (part, customer) shares Customer with (supplier, customer)
Fact Constellation

• Reuse as much as possible existing dimensions
  – Populating a dimension requires quite some effort
    → Reuse
  – Makes it easier to combine different cubes

• Notion of a “conformed dimension”
  – Definition of customer, date, product, ... agreed upon by all departments
  – Makes it easier to integrate different data marts
Examples
Corresponding Star-Schema

**Customer**
- CustKey
- OLTP_Custkey
- City
- Country
- Age-Cat

**Sale**
- CustKey
- DateKey
- ProdKey
- Quantity
- UnitPrice
- VATrate

**Product**
- ProdKey
- OLTP_Prodkey
- Brand
- Type

**Date**
- DateKey
- Date
- DayOfWeek
- Month
- Year
Example

<table>
<thead>
<tr>
<th>Sale</th>
</tr>
</thead>
<tbody>
<tr>
<td>quantity</td>
</tr>
<tr>
<td>unitPrice</td>
</tr>
<tr>
<td>VAT-rate</td>
</tr>
</tbody>
</table>

Optional dimension

DiscountName
Optional Dimension

• Avoid the use of “null”
  – Confusing and non-descriptive
  – Special case for queries (null≠null)

<table>
<thead>
<tr>
<th>Key</th>
<th>Name</th>
<th>Start</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>No discount</td>
<td>n/a</td>
</tr>
<tr>
<td>0002</td>
<td>Loyalty</td>
<td>Jan 2011</td>
</tr>
<tr>
<td>0003</td>
<td>Christmas</td>
<td>Dec 2012</td>
</tr>
</tbody>
</table>
Example

Cross-dimensional attribute

Tariff-plan

Unit-cost

called

number

Call

duration

cost

hour
Example

Tariff-plan

Cross-dimensional attribute

Unit-cost

Number
  - NID
  - Number
  - ... Plan

UnitCost
  - Hour Plan cost

Call
  - Time
    - TimeID
    - Hour Minute Second

Call
duration cost

called

number
Example
Recursive Hierarchy

• Special type of hierarchy: user hierarchy
  – Encode using attribute “parent” (unique)
  – Do not use “child” (not unique)
Multiple Arc – Option 1

- Use bridge table
Multiple Arc – Option 2

• Create artificial groups

Note: this is not a snowflake!
Summary

• Different types of aggregation operators
  – Distributive, algebraic, holistic

• Logical schemes for data warehouses
  – Star schema
  – Snowflake

• Translation of conceptual model to tables