Customer Data Warehouse Design

Data modelling

Data warehouse development

- Requirements identification
- Logical design, data modelling
- Data extract, transform and load (ETL)
- Warehouse architecture, technology and tools
- Physical database design
- Delivery systems
- Operational policies

Designing a data warehouse - data design

- There are two main approaches to data modelling for data warehouse design
  - Entity relationship modelling and normalisation
  - Dimensional modelling
  - What works for CRM? (Todman)
The design of databases using a traditional E-R approach

- Entities and relationships
- Normalisation 3NF, 4NF

Entity relationship schema

Why do we normalise data?

- Normalisation is a process for converting complex data structures into simple, stable data structures
- Normalised data models are:
  - robust and stable
  - have minimum redundancy
Why do we normalise data?

- Normalised data (3NF):
  - No repeating groups
  - No partial dependences
  - No transitive dependences

- The goal is to protect database integrity by avoiding anomalies (update, delete, create)

Dimensional Modeling (star schema)

- It is a logical design technique that seeks to present data in a standard framework that is intuitive and allows for high-performance access

Dimensional Modelling vs E-R modelling

- The purpose of dimensional modelling is to structure data for easy and efficient retrieval and analysis

- E-R modelling creates a single model of data required to support organisation's processes, Whereas

DM creates individual models for different areas of business/decision interest eg.
  - model for sales info
  - model for Inventory info
Dimensional Model – Star Schema

• Components of dimensional model:
  – **Fact Tables**: contain measurements of business eg. Sales, purchase order, shipment
    • Most data warehouses have very large fact tables up to 50 billion records and approximately 1 to 5 terabytes
  – **Dimension Tables**: store the descriptions of the dimensions of the business eg. Product, customer, vendor, store
    • smaller than fact tables

Dimensional Modeling

• Each dimension table has a single primary key that corresponds exactly to one of the components of the multipart key in the fact table.

• A fact table always expresses a many to many relationship (the key is composed of foreign keys)

• The most useful facts in a fact table are numeric and additive (typically values are added up)
### A Sample Star Schema

- **Customer**
  - Customer key
  - Customer name
  - Customer type
  - Customer credit code
  - Salesperson number
  - Sales territory
  - Standard industry code

- **Salesperson**
  - Salesperson key
  - Salesperson name
  - Sales region
  - Sales branch

- **Product**
  - Product key
  - Customer key
  - Salesperson key
  - Time key
  - Sales units
  - Gross sales amount
  - Sales discount amount
  - Net sales amount
  - Sales commission amount

- **Time**
  - Time key
  - Day
  - Month
  - Quarter
  - Year

### Dimensions

- The salesperson dimension allows users to analyze sales by salesperson, region, branch.
- The product dimension allows users to analyze purchasing patterns by product and groupings of product.
- The customer dimension allows users to analyze purchasing customer purchasing patterns.
- Example query: How many white Toyota cars were sold in February by Melbourne stores to customers living in Ballarat.
- The time dimension?

### Dimension Tables and Normalization

- Dimension tables in star schema are denormalized resulting in:
  - Fewer tables
  - Simpler for users to navigate
  - Reduced number of complex multi-table joins

- Compare the following two models...
Entity relationship schema (3NF)

Corresponding Star schema

Snowflake schema

• Snowflake schema - all the tables are normalised
• Star schemas are preferable to snowflake – fewer joins for information retrieval
DM vs. E-R modeling debate (Kimball’s view)

- OLTP systems are volatile – high rates of update transactions
- In normalised models the goal is to reduce data redundancy and prevent update anomalies
- Data in a data warehouse does not need to be normalised because it is periodically refreshed – not updated by user transactions

Steps in the dimensional design process (Kimball’s approach)

- Business requirements
- Dimensional modelling steps:
  1. Choose a business process
  2. Choose the grain of the fact table
  3. Choose the dimensions
  4. Choose the measured facts

(Kimball, 2002)
Steps in the dimensional design process

- Example processes: orders, shipments, sales, inventory, marketing
- Example measured facts (business measurements):
  - Sales fact table: unit sales, sales dollar amount, cost...
  - Shipment fact table: qty shipped, invoice dollar amount, storage cost...
  - The most useful facts are numeric and additive

Fact table: choosing the Grain

- The grain specifies what an individual row in a fact table represents
  - daily item sales
  - customer transaction
  - line item on order
  - daily inventory level of a stock item

Granularity

- Granularity refers to the level of detail captured in the data warehouse
- Choosing the grain is a critical step in design of DW
- Example: each individual customer transaction vs daily sales
Example: retail trading (adapted from Kimball and Groth)

Requirement 1:
- Management wants to analyse what products are selling, in which stores, on what days, under what promotion conditions

Choosing the grain
Example: grain 1 - daily sales

Example: grain 2 - individual POS transaction

From Kimball (1996), p. 29
From Kimball (2002), p. 37
Requirement 2:

- Management wants to analyse what type of products and what brand are buying different types of customers

The Importance of Granularity

- Choosing the grain is a critical step in design of DW
- The grain determines the dimensionality of the database
- It affects the level of detail of a query
- Trade off - volume of data against the level of detail
### The effect of granularity (example)

- **High level of detail**
  - Individual phone call
  - Summary of phone calls
- **Low level of detail**
  - Not enough detail to answer

The questions that can be answered depend on the level of granularity.

---

### Example: Product dimension

<table>
<thead>
<tr>
<th>Product</th>
<th>Sales Fact</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SKU description</td>
<td>Unit sales</td>
<td>Time key</td>
</tr>
<tr>
<td>SKU number</td>
<td>Dollar costs</td>
<td>Store</td>
</tr>
<tr>
<td>Product type</td>
<td>Customer count</td>
<td>Promotion</td>
</tr>
<tr>
<td>subcategory</td>
<td>Dollar sales</td>
<td></td>
</tr>
<tr>
<td>category</td>
<td></td>
<td></td>
</tr>
<tr>
<td>department</td>
<td></td>
<td></td>
</tr>
<tr>
<td>brand</td>
<td></td>
<td></td>
</tr>
<tr>
<td>package type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unit of measure</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

### Example: time dimension

<table>
<thead>
<tr>
<th>Time</th>
<th>Sales Fact</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day of week</td>
<td>Unit sales</td>
<td>Product key</td>
</tr>
<tr>
<td>Day no in month</td>
<td>Dollar costs</td>
<td>Store key</td>
</tr>
<tr>
<td>Day no overall</td>
<td>Customer count</td>
<td>Promotion key</td>
</tr>
<tr>
<td>Week no overall</td>
<td>Dollar sales</td>
<td></td>
</tr>
<tr>
<td>Month</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Month no overall</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week no in year</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quarter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Holiday flag</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weekday flag</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Last day month flag</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fiscal period</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Season</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Adapted from Kimball.
Shared dimensions

- Dimension tables can be shared between subject areas.

If a dimension table connects to more than one fact table:
- Represent this dimension table in both schemas.
- The dimension tables are referred to as conformed between the two dimensional models.
- Conformed dimensions → consistent information.

Shared dimensions, example

- Customer
- Product
- Time
- Shipments
- Warehouse
- Orders
- Production facts

Slowly changing dimensions

- Many dimensions (like Product and Customer) evolve slowly over time:
  - People change names, addresses etc.
  - Sales forces change names of districts, regions etc.

- Three standard approaches are:
  - Overwrite old values
  - Create an additional dimension record
  - Create a current value field
Aggregates

- Most data warehouses have very large fact tables (up to 50 billion records and approximately 1 to 5 terabytes).
- Aggregates (pre-stored summaries) are the most effective way of improving data warehouse performance.
- An aggregate is a fact table record representing a summarisation of base level fact table records.

Aggregate fact tables

- Data model showing relationships between sales fact, category, and product dimensions.

Dimensional Modelling example

• Read DW examples from the Groth text
  - Retail
  - Telecommunications
  - Healthcare

References

• Groth R., Data Mining, Prentice Hall, 2000, chapter 9
• “Modern Data Base Management”, by Mc Fadden et al

What works for CRM?

• The challenges and issues in the development of customer-centric data warehouse