Data Warehousing Design

1

Logical Design in Data Warehouses

- Logical Versus Physical Design in Data Warehouses
- Creating a Logical Design
- Data Warehousing Schemas
- Data Warehousing Objects

Logical Versus Physical Design in Data Warehouses

- The logical design is more conceptual and abstract than the physical design.
- In the logical design, you look at the logical relationships among the objects.
- In the physical design, you look at the most effective way of storing and retrieving the objects

Logical Versus Physical Design in Data Warehouses

- as well as handling them from a transportation and backup/recovery perspective.
- Orient your design toward the needs of the end users.
- End users typically want to perform analysis and look at aggregated data,
- rather than at individual transactions.

Logical Versus Physical Design in Data Warehouses

- However, end users might not know what they need until they see it.
- In addition, a well-planned design allows for growth and changes as the needs of users change and evolve.
- By beginning with the logical design, you focus on the information requirements and save the implementation details for later.

Creating a Logical Design

- A logical design is conceptual and abstract.
- You do not deal with the physical implementation details yet.
- You deal only with defining the types of information that you need.
- One technique you can use to model your organization's logical information requirements is entity-relationship modeling.

Creating a Logical Design

- Entity-relationship modeling involves identifying the things of importance (entities),
- the properties of these things (attributes),
- and how they are related to one another (relationships).
- The process of logical design involves arranging data into a series of logical relationships called entities and attributes.

Creating a Logical Design

- An entity represents a chunk of information.
- In relational databases, an entity often maps to a table.
- An attribute is a component of an entity that helps define the uniqueness of the entity.
- In relational databases, an attribute maps to a column.

Creating a Logical Design

- To be sure that your data is consistent, you need to use unique identifiers.
- A unique identifier is something you add to tables so that you can differentiate between the same item when it appears in different places.
- In a physical design, this is usually a primary key.

Creating a Logical Design

- While entity-relationship diagramming has traditionally been associated with highly normalized models such as OLTP applications,
- the technique is still useful for data warehouse design in the form of dimensional modeling.
- In dimensional modeling, instead of seeking to discover atomic units of information (such as entities and attributes) and

Creating a Logical Design

- all of the relationships between them,
- you identify which information belongs to a central fact table and which information belongs to its associated dimension tables.
- You identify business subjects or fields of data, define relationships between business subjects, and name the attributes for each subject.

Creating a Logical Design

- Your logical design should result in
- (1) a set of entities and attributes corresponding to fact tables and dimension tables and
- (2) a model of operational data from your source into subject-oriented information in your target data warehouse schema.

Creating a Logical Design

- You can create the logical design using a pen and paper,
- or you can use a design tool such as Oracle Warehouse Builder (specifically designed to support modeling the ETL process)
- or Oracle Designer (a general purpose modeling tool).

Data Warehousing Schemas

- A schema is a collection of database objects, including tables, views, indexes, and synonyms.
- You can arrange schema objects in the schema models designed for data warehousing in a variety of ways.
- Most data warehouses use a dimensional model.

Data Warehousing Schemas

- The model of your source data and the requirements of your users help you design the data warehouse schema.
- You can sometimes get the source model from your company's enterprise data model and reverse-engineer the logical data model for the data warehouse from this.

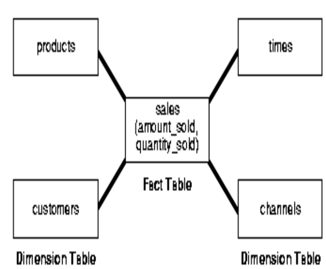
Data Warehousing Schemas

- The physical implementation of the logical data warehouse model may require some changes to adapt it to your system parameters-
- size of machine, number of users, storage capacity, type of network, and software

Data Warehousing Schemas

- The star schema is the simplest data warehouse schema.
- It is called a star schema because the diagram resembles a star,
- with points radiating from a center.
- The center of the star consists of one or more fact tables and the points of the star are the dimension tables, as shown in Figure below.

The star schema



The star schema

- The most natural way to model a data warehouse is as a star schema,
- only one join establishes the relationship between the fact table and any one of the dimension tables.
- A star schema optimizes performance by keeping queries simple and providing fast response time.
- All the information about each level is stored in one row

Data Warehousing Objects

- Fact tables and dimension tables are the two types of objects commonly used in dimensional data warehouse schemas.
- Fact tables are the large tables in your warehouse schema that store business measurements.
- Fact tables typically contain facts and foreign keys to the dimension tables.

Data Warehousing Objects

- Fact tables represent data, usually numeric and additive, that can be analyzed and examined.
- Examples include sales, cost, and profit.
- Dimension tables, also known as lookup or reference tables, contain the relatively static data in the warehouse.

Data Warehousing Objects

- Dimension tables store the information you normally use to contain queries.
- Dimension tables are usually textual and descriptive and you can use them as the row headers of the result set.
- Examples are customers or products.

Fact Tables

- A fact table typically has two types of columns:
- those that contain numeric facts (often called measurements), and those that are foreign keys to dimension tables.
- A fact table contains either detail-level facts or facts that have been aggregated.

Fact Tables

- Fact tables that contain aggregated facts are often called summary tables.
- A fact table usually contains facts with the same level of aggregation.
- Though most facts are additive, they can also be semi-additive or non-additive.

Fact Tables

- Additive facts can be aggregated by simple arithmetical addition.
- A common example of this is sales.
- Non-additive facts cannot be added at all.

Fact Tables

- An example of this is averages.
- Semi-additive facts can be aggregated along some of the dimensions and not along others.
- An example of this is inventory levels, where you cannot tell what a level means simply by looking at it.

Creating a New Fact Table

- You must define a fact table for each star schema.
- From a modeling standpoint, the primary key of the fact table is usually a composite key that is made up of all of its foreign keys.

Dimension Tables

- A dimension is a structure, often composed of one or more hierarchies, that categorizes data.
- Dimensional attributes help to describe the dimensional value.
- They are normally descriptive, textual values.

Dimension Tables

- Several distinct dimensions, combined with facts, enable you to answer business questions.
- Commonly used dimensions are customers, products, and time.

Dimension Tables

- Dimension data is typically collected at the lowest level of detail and then aggregated into higher level totals that are more useful for analysis.
- These natural rollups or aggregations within a dimension table are called hierarchies.

Hierarchies

- Hierarchies are logical structures that use ordered levels as a means of organizing data.
- A hierarchy can be used to define data aggregation.
- For example, in a time dimension, a hierarchy might aggregate data from the month level to the quarter level to the year level.

Hierarchies

- A hierarchy can also be used to define a navigational drill path and to establish a family structure.
- Within a hierarchy, each level is logically connected to the levels above and below it.
- Data values at lower levels aggregate into the data values at higher levels.

Hierarchies

- A dimension can be composed of more than one hierarchy.
- For example, in the product dimension, there might be two hierarchies--one for product categories and one for product suppliers.

Hierarchies

- Dimension hierarchies also group levels from general to granular.
- Query tools use hierarchies to enable you to drill down into your data to view different levels of granularity.
- This is one of the key benefits of a data warehouse.

Hierarchies

- When designing hierarchies, you must consider the relationships in business structures.
- For example, a divisional multilevel sales organization.
- Hierarchies impose a family structure on dimension values.

Hierarchies

- For a particular level value, a value at the next higher level is its parent, and values at the next lower level are its children.
- These familial relationships enable analysts to access data quickly.

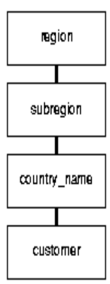
Level Relationships

- Level relationships specify top-to-bottom ordering of levels from most general (the root) to most specific information.
- They define the parent-child relationship between the levels in a hierarchy.

Level Relationships

- Hierarchies are also essential components in enabling more complex rewrites.
- For example, the database can aggregate an existing sales revenue on a quarterly base to a yearly aggregation when the dimensional dependencies between quarter and year are known.

Typical Levels in a Dimension Hierarchy



Unique Identifiers

- Unique identifiers are specified for one distinct record in a dimension table.
- Artificial unique identifiers are often used to avoid the potential problem of unique identifiers changing.
- Unique identifiers are represented with the # character.
- For example, #customer_id.

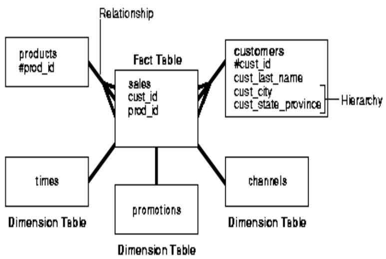
Relationships

- Relationships guarantee business integrity.
- An example is that if a business sells something,
- there is obviously a customer and a product.
- Designing a relationship between the sales information in the fact table and
- the dimension tables products and customers enforces the business rules in databases.

Example of Data Warehousing Objects and Their Relationships

- Figure below illustrates a common example of a sales fact table and dimension tables customers, products, promotions, times, and channels.

Typical Data Warehousing Objects



Questions

44