**Лекция 2**

Now that you understand fact and dimension tables, it’s time to bring the building blocks together in a dimensional model, as shown in Figure 1-5. Each business process is represented by a dimensional model that consists of a fact table containing the event’s numeric measurements surrounded by a halo of dimension tables that contain the textual context that was true at the moment the event occurred. This characteristic star-like structure is often called a *star join*, a term dating back to the earliest days of relational databases.



Here are illustrated the most granular or atomic data. These atomic data should be the foundation for every fact table design to withstand business users’ ad hoc attacks in which they pose unexpected queries. With dimensional models, you can add completely new dimensions to the schema as long as a single value of that dimension is defined for each existing fact row. Likewise, you can add new facts to the fact table, assuming that the level of detail is consistent with the existing fact table. You can supplement preexisting dimension tables with new, unanticipated attributes. In each case, existing tables can be changed in place either by simply adding new data rows in the table or by executing an SQL ALTER TABLE command. Data would not need to be reloaded, and existing BI applications would continue to run without yielding different results. Another way to think about the complementary nature of fact and dimension tables is to see them translated into a report. As illustrated in Figure 1-6, dimension attributes supply the report filters and labeling, whereas the fact tables supply the report’s numeric values.

You can easily envision the SQL that’s written (or more likely generated by a BI tool) to create this report:

|  |
| --- |
| SELECT  store.district\_name,  product.brand,  sum(sales\_facts.sales\_dollars) AS "Sales Dollars"  FROM  store,  product,  date,  sales\_facts  WHERE  date.month\_name="January" AND  date.year=2013 AND  store.store\_key = sales\_facts.store\_key AND  product.product\_key = sales\_facts.product\_key AND  date.date\_key = sales\_facts.date\_key  GROUP BY  store.district\_name,  product.brand |



**Kimball’s DW/BI Architecture**

Let’s build on your understanding of DW/BI systems and dimensional modeling fundamentals by investigating the components of a DW/BI environment based on the Kimball architecture. You need to learn the strategic significance of each component to avoid confusing their role and function.

As illustrated in Figure 1-7, there are four separate and distinct components to consider in the DW/BI environment: operational source systems, ETL system, data presentation area, and business intelligence applications.

**Operational Source Systems**

These are the operational systems of record that capture the business’s transactions. Think of the source systems as outside the data warehouse because presumably you have little or no control over the content and format of the data in these operational systems. The main priorities of the source systems are processing performance and availability.

Operational queries against source systems are narrow, one-record-at-a-time Data Warehousing, Business Intelligence, and Dimensional Modeling queries that are part of the normal transaction flow and severely restricted in their demands on the operational system. It is safe to assume that source systems are not queried in the broad and unexpected ways that DW/BI systems typically are queried.

Source systems maintain little historical data; a good data warehouse can relieve the source systems of much of the responsibility for representing the past. In many cases, the source systems are special purpose applications without any commitment to sharing common data such as product, customer, geography, or calendar with other operational systems in the organization. Of course, a broadly adopted cross-application enterprise resource planning (ERP) system or operational master data management system could help address these shortcomings.



**Extract, Transformation, and Load System**

The *extract, transformation, and load* (*ETL*) *system* of the DW/BI environment consists of a work area, instantiated data structures, and a set of processes. The ETL system is everything between the operational source systems and the DW/BI presentation area. We elaborate on the architecture of ETL systems and associated techniques: ETL Subsystems and Techniques, but we want to introduce this fundamental piece of the overall DW/BI system puzzle.

Extraction is the first step in the process of getting data into the data warehouse environment. *Extracting* means reading and understanding the source data and copying the data needed into the ETL system for further manipulation. At this point, the data belongs to the data warehouse.

After the data is extracted to the ETL system, there are numerous potential transformations, such as cleansing the data (correcting misspellings, resolving domain conflicts, dealing with missing elements, or parsing into standard formats), combining data from multiple sources, and de-duplicating data. The ETL system adds value to the data with these cleansing and conforming tasks by changing the data and enhancing it. In addition, these activities can be architected to create diagnostic metadata, eventually leading to business process reengineering to improve data quality in the source systems over time.

The final step of the ETL process is the physical structuring and loading of data into the presentation area’s target dimensional models. Because the primary mission of the ETL system is to hand off the dimension and fact tables in the delivery step, these subsystems are critical. Many of these defined subsystems focus on dimension table processing, such as surrogate key assignments, code lookups to provide appropriate descriptions, splitting, or combining columns to present the

appropriate data values, or joining underlying third normal form table structures into flattened denormalized dimensions. In contrast, fact tables are typically large and time consuming to load, but preparing them for the presentation area is typically straightforward. When the dimension and fact tables in a dimensional model have been updated, indexed, supplied with appropriate aggregates, and further quality assured, the business community is notified that the new data has been published.