

---

## Fact Tables

By Ralph  
Kimball

Fact tables are the foundation of the data warehouse. They contain the fundamental measurements of the enterprise, and they are the ultimate target of most data warehouse queries. There is no point in hoisting fact tables up the flagpole unless they have been chosen to reflect urgent business priorities, have been carefully quality assured and are surrounded by dimensions that provide a wealth of entry points for constraining and grouping. Now that we have paved the way for fact tables, let's see how to build them and use them.

### Stay True to the Grain

The first and most important design step is declaring the fact table grain. The grain is the business definition of what a single fact table record represents. The grain declaration is not a list of dimensional foreign keys that implement a primary key for the fact table. Rather, the grain is the description of the measurement event in the physical world that gives rise to a measurement. When the grocery store scanner measures the quantity and the charged price of a product being purchased, the grain is literally the beep of the scanner. That is a great grain definition!

Immediately after declaring the grain, it is possible to list the dimensional foreign keys that exist at that grain. By declaring the grain first, the discussion of foreign keys remains grounded and precise.

The real purpose of the fact table is to be the repository of the numeric facts that are observed during the measurement event. It is critically important for these facts to be true to the grain. The grocery store "beep" measures the quantity and extended price of the product being scanned. We never include other numeric measurements that violate the grain, such as the overall category sales or the sales of this product last month. Even though these other measurements might be narrowly helpful for selected calculations, they cannot be combined across fact records and they introduce weird asymmetries in the design of applications. We let our business intelligence (BI) tools compute these off-topic values at query time rather than hard coding them into our fact tables.

We always strive to make the facts additive across the dimensions and exactly consistent with the grain. Notice that we don't store the price of the product being scanned because the price is nonadditive. Rather, we store the extended price, which can be added freely across products, stores, times and all the other dimensions.

### Build Up from the Lowest Possible Grain

The data warehouse should always be built on fact tables expressed at the lowest possible grain. In the example, the beep of the grocery store cash register is the lowest possible grain because it cannot be divided any further. Fact tables at the lowest grain are the most expressive because they have the most complete set of possible dimensions for that business process. The beep grain fact table could have Date, Store, Product, Cashier, Manager, Customer, Promotion, Competition, Basket and even Weather if all these data sources can be marshaled when the fact records are created. Higher grain aggregated tables such as category sales by district cannot support all these dimensions and therefore are much less expressive. It is a fundamental mistake to publish only aggregated tables to the end users without making the lowest grain fact tables smoothly accessible by drilling down. Most of the false notions that dimensional tables presuppose the business question come from making this fundamental mistake.

### Three Kinds of Fact Tables

If you stay true to the grain, then all of your fact tables can be grouped into just three types: transaction grain, periodic snapshot grain and accumulating snapshot grain (the three types are shown in Figure 1). In Figure 1, the dimensions are designated by FK (foreign key) and the numeric facts are italicized.

The transaction grain corresponds to a measurement taken at a single instant. The grocery store beep is a transaction grain. The measured facts are valid only for that instant and for that event. The next measurement event could happen one millisecond later or next month or never. Thus, transaction grain fact tables are unpredictably sparse or dense. We have no guarantee that all the possible foreign keys will be represented. Transaction grain fact tables can be enormous, with the largest containing many billions of records.

The periodic snapshot grain corresponds to a predefined span of time, often a financial reporting period. Figure 1 illustrates a monthly account periodic snapshot. The measured facts summarize activity during or at the end of the time span. The periodic snapshot grain carries a powerful guarantee that all of the reporting entities (such as the bank account in Figure 1) will appear in each snapshot, even if there is no activity. The periodic snapshot is predictably dense, and applications can rely on combinations of keys always being present. Periodic snapshot fact tables can also get large. A bank with 20 million accounts and a 10-year history would have 2.4 billion records in the monthly account periodic snapshot!

The accumulating snapshot fact table corresponds to a predictable process that has a well-defined beginning and end. Order processing, claims processing, service call resolution and college admissions are typical candidates. The grain of an accumulating snapshot for order processing, for example, is usually the line item on the order. Notice in Figure 1 that there are multiple dates representing the standard scenario that an order undergoes. Accumulating snapshot records are revisited and overwritten as the process progresses through its steps from beginning to end. Accumulating snapshot fact tables generally are much smaller than the other two types because of this overwriting strategy.

Figure 1. The Three Fact Table Types		
Transaction Grain	Periodic Snapshot Grain	Accumulating Snapshot Grain
Date (FK)	Month (FK)	Order Date (FK)
Product (FK)	Account (FK)	Ship Date (FK)
Store (FK)	Branch (FK)	Delivery Date (FK)
Customer (FK)	Household (FK)	Payment Date (FK)
Cashier (FK)	<i>balance</i>	Return Date (FK)
Manager (FK)	<i>fees paid</i>	Warehouse (FK)
Promotion (FK)	<i>interest earned</i>	Customer (FK)
Weather (FK)	<i>transaction count</i>	Promotion (FK)
Basket (FK)		Order Status (FK)
<i>time of day</i>		<i>quantity</i>
<i>quantity</i>		<i>extended list price</i>
<i>extended price</i>		<i>discounts</i>
		<i>extended net price</i>

