A

Term	Definition
Accumulating Snapshot Fact Tables	A row in an *accumulating snapshot fact table* summarizes the measurement events occurring at predictable steps between the beginning and the end of a process. Pipeline or workflow processes, such as order fulfillment or claim processing, that have a defined start point, standard intermediate steps, and defined end point can be modeled with this type of fact table. There is a date foreign key in the fact table for each critical milestone in the process. An individual row in an accumulating snapshot fact table, corresponding for instance to a line on an order, is initially inserted when the order line is created. As pipeline progress occurs, the accumulating fact table row is revisited and updated. This consistent updating of accumulating snapshot fact rows is unique among the three types of fact tables. In addition to the date foreign keys associated with each critical process step, accumulating snapshot fact tables contain foreign keys for other dimensions and optionally contain degenerate dimensions. They often include numeric lag measurements consistent with the grain, along with milestone completion counters. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/accumulating-snapshot-fact-table/
Additive, Semi-Additive, and Non-Additive Facts	The numeric measures in a fact table fall into three categories. The most flexible and useful facts are fully *additive*; additive measures can be summed across any of the dimensions associated with the fact table. *Semi-additive* measures can be summed across some dimensions, but not all; balance amounts are common semi-additive facts because they are additive across all dimensions except time. Finally, some measures are completely *non-additive*, such as ratios. A good approach for non-additive facts is, where possible, to store the fully additive components of the non-additive measure and sum these components into the final answer set before calculating the final non-additive fact. This final calculation is often done in the BI layer or OLAP cube. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/additive-semi-additive-non-additive-fact/
Allocated Facts	It is quite common in header/line transaction data to encounter facts of differing granularity, such as a header freight charge. You should strive to *allocate* the header facts down to the line level based on rules provided by the business, so the allocated facts can be sliced and rolled up by all the dimensions. In many cases, you can avoid creating a header-level fact table, unless this aggregation delivers query performance advantages.

	http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/allocated-fact/

B

Term	Definition
Bridge Tables	In a classic dimensional schema, each dimension attached to a fact table has a single value consistent with the fact table's grain. But there are a number of situations in which a dimension is legitimately *multivalued*. For example, a patient receiving a healthcare treatment may have multiple simultaneous diagnoses. In these cases, the multivalued dimension must be attached to the fact table through a group dimension key to a bridge table with one row for each simultaneous diagnosis in a group. A *multivalued bridge table* may need to be based on a type 2 slowly changing dimension. For example, the bridge table that implements the many-to-many relationship between bank accounts and individual customers usually must be based on type 2 account and customer dimensions. In this case, to prevent incorrect linkages between accounts and customers, the bridge table must include effective and expiration date/time stamps, and the requesting application must constrain the bridge table to a specific moment in time to produce a consistent snapshot. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/multivalued-dimension-bridge-table/

C

Term	Definition
Calendar Date Dimensions	*Calendar date dimensions* are attached to virtually every fact table to allow navigation of the fact table through familiar dates, months, fiscal periods, and special days on the calendar. You would never want to compute Easter in SQL, but rather want to look it up in the calendar date dimension. The calendar date dimension typically has many attributes describing characteristics such as week number, month name, fiscal period, and national holiday indicator. To facilitate partitioning, the primary key of a date dimension can be more meaningful, such as an integer representing YYYYMMDD, instead of a sequentially-assigned surrogate key. However, the date dimension table needs a special row to represent unknown or to-be-determined dates. If a smart date key is used, filtering and grouping should be based on the dimension table's attributes, not the smart key.

	When further precision is needed, a separate date/time stamp can be added to the fact table. The date/time stamp is not a foreign key to a dimension table, but rather is a standalone column. If business users constrain or group on time-of-day attributes, such as day part grouping or shift number, then you would add a separate time-of-day dimension foreign key to the fact table. http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/calendar-date-dimension/
Cardinality	In SQL (Structured Query Language), the term **cardinality** refers to the uniqueness of data values contained in a particular column (attribute) of a **database** table. The lower the **cardinality**, the more duplicated elements in a column. Source: https://en.wikipedia.org/wiki/Cardinality_(SQL_statements)
Conformed Dimensions	Dimension tables *conform* when attributes in separate dimension tables have the same column names and domain contents. Information from separate fact tables can be combined in a single report by using conformed dimension attributes that are associated with each fact table. When a conformed attribute is used as the row header (that is, the grouping column in the SQL query), the results from the separate fact tables can be aligned on the same rows in a drill-across report. This is the essence of integration in an enterprise DW/ BI system. *Conformed dimensions*, defined once in collaboration with the business's data governance representatives, are reused across fact tables; they deliver both analytic consistency and reduced future development costs because the wheel is not repeatedly re-created. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/conformed-dimension/
Conformed Facts	If the same measurement appears in separate fact tables, care must be taken to make sure the technical definitions of the facts are identical if they are to be compared or computed together. If the separate fact definitions are consistent, the *conformed facts* should be identically named; but if they are incompatible, they should be differently named to alert the business users and BI applications. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/conformed-fact/

D

Term	Definition
Degenerate Dimensions	Sometimes a dimension is defined that has no content except for its

primary key. For example, when an invoice has multiple line items, the line item fact rows inherit all the descriptive dimension foreign keys of the invoice, and the invoice is left with no unique content. But the invoice number remains a valid dimension key for fact tables at the line item level. This *degenerate dimension* is placed in the fact table with the explicit acknowledgment that there is no associated dimension table. Degenerate dimensions are most common with transaction and accumulating snapshot fact tables.
Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/degenerate-dimension/

Dimension Hierarchies: Fixed Depth Positional Hierarchies

A *fixed depth hierarchy* is a series of many-to-one relationships, such as product to brand to category to department. When a fixed depth hierarchy is defined and the hierarchy levels have agreed upon names, the hierarchy levels should appear as separate positional attributes in a dimension table. A fixed depth hierarchy is by far the easiest to understand and navigate as long as the above criteria are met. It also delivers predictable and fast query performance. When the hierarchy is not a series of many-to-one relationships or the number of levels varies such that the levels do not have agreed upon names, a ragged hierarchy technique must be used.
Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/fixed-depth-hierarchy/

Dimension Hierarchies: Multiple Hierarchies in Dimensions

Many dimensions contain more than one natural hierarchy. For example, calendar date dimensions may have a day to week to fiscal period hierarchy, as well as a day to month to year hierarchy. Location intensive dimensions may have multiple geographic hierarchies. In all of these cases, the separate hierarchies can gracefully coexist in the same dimension table.
Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/multiple-hierarchies/

Dimension Hierarchies: Ragged/Variable Depth Hierarchies

Ragged hierarchies of indeterminate depth are difficult to model and query in a relational database. Although SQL extensions and OLAP access languages provide some support for recursive parent/child relationships, these approaches have limitations. With SQL extensions, alternative ragged hierarchies cannot be substituted at query time, shared ownership structures are not supported, and time varying ragged hierarchies are not supported. All these objections can be overcome in relational databases by modeling a ragged hierarchy with a specially constructed *bridge table*. This bridge table contains a row for every possible path in the ragged hierarchy and enables all forms of hierarchy

	traversal to be accomplished with standard SQL rather than using special language extensions. The use of a bridge table for ragged variable depth hierarchies can be avoided by implementing a *pathstring attribute* in the dimension. For each row in the dimension, the pathstring attribute contains a specially encoded text string containing the complete path description from the supreme node of a hierarchy down to the node described by the particular dimension row. Many of the standard hierarchy analysis requests can then be handled by standard SQL, without resorting to SQL language extensions. However, the pathstring approach does not enable rapid substitution of alternative hierarchies or shared ownership hierarchies. The pathstring approach may also be vulnerable to structure changes in the ragged hierarchy that could force the entire hierarchy to be relabeled. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/ragged-variable-depth-hierarchy/
Dimension Hierarchies: Slightly Ragged/Variable Depth Hierarchies	*Slightly ragged* hierarchies don't have a fixed number of levels, but the range in depth is small. Geographic hierarchies often range in depth from perhaps three levels to six levels. Rather than using the complex machinery for unpredictably variable hierarchies, you can force-fit slightly ragged hierarchies into a fixed depth positional design with separate dimension attributes for the maximum number of levels, and then populate the attribute value based on rules from the business. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/slightly-ragged-variable-depth-hierarchy/
Drilling Down	*Drilling down* is the most fundamental way data is analyzed by business users. Drilling down simply means adding a row header to an existing query; the new row header is a dimension attribute appended to the GROUP BY expression in an SQL query. The attribute can come from any dimension attached to the fact table in the query. Drilling down does not require the definition of predetermined hierarchies or drill-down paths. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/drilling-down/

E

Term	Definition
Enterprise Data Warehouse Bus Architecture	The *enterprise data warehouse bus architecture* provides an incremental approach to building the enterprise DW/BI system. This architecture decomposes the DW/ BI planning process into manageable pieces by focusing on business processes, while delivering integration

	via standardized conformed dimensions that are reused across processes. It provides an architectural framework, while also decomposing the program to encourage manageable agile implementations corresponding to the rows on the enterprise data warehouse bus matrix. The bus architecture is technology and database platform independent; both relational and OLAP dimensional structures can participate. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/enterprise-data-warehouse-bus-architecture/
Enterprise Data Warehouse Bus Matrix	The *enterprise data warehouse bus matrix* is the essential tool for designing and communicating the enterprise data warehouse bus architecture. The rows of the matrix are business processes and the columns are dimensions. The shaded cells of the matrix indicate whether a dimension is associated with a given business process. The design team scans each row to test whether a candidate dimension is well-defined for the business process and also scans each column to see where a dimension should be conformed across multiple business processes. Besides the technical design considerations, the bus matrix is used as input to prioritize DW/BI projects with business management as teams should implement one row of the matrix at a time. The *detailed implementation bus matrix* is a more granular bus matrix where each business process row has been expanded to show specific fact tables or OLAP cubes. At this level of detail, the precise grain statement and list of facts can be documented. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/enterprise-data-warehouse-bus-matrix/

↖**F**

Term	Definition
Factless Fact Tables	Although most measurement events capture numerical results, it is possible that the event merely records a set of dimensional entities coming together at a moment in time. For example, an event of a student attending a class on a given day may not have a recorded numeric fact, but a fact row with foreign keys for calendar day, student, teacher, location, and class is well-defined. Likewise, customer communications are events, but there may be no associated metrics. *Factless fact tables* can also be used to analyze what didn’t happen. These queries always have two parts: afactless coverage table that contains all the possibilities of events that might happen and an activity table that contains the events that did happen. When the activity is subtracted from the coverage, the result is the set of events that did not happen.

Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/factless-fact-table/

J

Term	Definition
Junk Dimensions	Transactional business processes typically produce a number of miscellaneous, low-cardinality flags and indicators. Rather than making separate dimensions for each flag and attribute, you can create a single *junk dimension* combining them together. This dimension, frequently labeled as a *transaction profile dimension* in a schema, does not need to be the Cartesian product of all the attributes' possible values, but should only contain the combination of values that actually occur in the source data. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/junk-dimension/

M

Term	Definition
Multivalued Dimensions	See "Bridge Tables"

N

Term	Definition
Natural, Durable, and Supernatural Keys	*Natural keys* created by operational source systems are subject to business rules outside the control of the DW/BI system. For instance, an employee number (natural key) may be changed if the employee resigns and then is rehired. When the data warehouse wants to have a single key for that employee, a new *durable key* must be created that is persistent and does not change in this situation. This key is sometimes referred to as a *durable supernatural key*. The best durable keys have a format that is independent of the original business process and thus should be simple integers assigned in sequence beginning with 1. While multiple surrogate keys may be associated with an employee over time as their profilechanges, the durable key never changes. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/natural-durable-supernatural-key/
Null Attributes in Dimensions	Null-valued dimension attributes result when a given dimension row has not been fully populated, or when there are attributes that are not applicable to all the dimension's rows. In both cases, we recommend substituting a descriptive string, such as Unknown or Not Applicable in place of the null value. Nulls in dimension attributes should be avoided because different databases handle grouping and constraining on nulls inconsistently. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-

resources/kimball-techniques/dimensional-modeling-techniques/null-dimension-attribute/

O

Term	Definition
Opportunity/Stakeholder Matrix	After the enterprise data warehouse bus matrix rows have been identified, you can draft a different matrix by replacing the dimension columns with business functions, such as marketing, sales, and finance, and then shading the matrix cells to indicate which business functions are interested in which business process rows. The *opportunity/stakeholder matrix* helps identify which business groups should be invited to the collaborative design sessions for each process-centric row. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/opportunity-stakeholder-matrix/
Outrigger Dimensions	A dimension can contain a reference to another dimension table. For instance, a bank account dimension can reference a separate dimension representing the date the account was opened. These secondary dimension references are called *outrigger dimensions*. Outrigger dimensions are permissible, but should be used sparingly. In most cases, the correlations between dimensions should be demoted to a fact table, where both dimensions are represented as separate foreign keys. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/outrigger-dimension/

P

Term	Definition
Periodic Snapshot Fact Tables	A row in a *periodic snapshot fact table* summarizes many measurement events occurring over a standard period, such as a day, a week, or a month. The grain is the period, not the individual transaction. Periodic snapshot fact tables often contain many facts because any measurement event consistent with the fact table grain is permissible. These fact tables are uniformly dense in their foreign keys because even if no activity takes place during the period, a row is typically inserted in the fact table containing a zero or null for each fact. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/periodic-snapshot-fact-table/

R

Term	Definition
Role-Playing Dimensions	A single physical dimension can be referenced multiple times in a fact table, with each reference linking to a logically distinct role for the

	dimension. For instance, a fact table can have several dates, each of which is represented by a foreign key to the date dimension. It is essential that each foreign key refers to a separate view of the date dimension so that the references are independent. These separate dimension views (with unique attribute column names) are called *roles*. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/role-playing-dimension/

↖**S**

Term	Definition
Shrunken Rollup Dimensions	*Shrunken dimensions* are conformed dimensions that are a *subset* of rows and /or columns of a base dimension. *Shrunken rollup* dimensions are required when constructing aggregate fact tables. They are also necessary for business processes that naturally capture data at a higher level of granularity, such as a forecast by month and brand (instead of the more atomic date and product associated with sales data). Another case of conformed dimension subsetting occurs when two dimensions are at the same level of detail, but one represents only a subset of rows. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/shrunken-rollup-dimension/
Surrogate Keys (Dimension Surrogate Keys)	A dimension table is designed with one column serving as a unique primary key. This primary key cannot be the operational system's natural key because there will be multiple dimension rows for that natural key when changes are tracked over time. In addition, natural keys for a dimension may be created by more than one source system, and these natural keys may be incompatible or poorly administered. The DW/BI system needs to claim control of the primary keys of all dimensions; rather than using explicit natural keys or natural keys with appended dates, you should create anonymous integer primary keys for every dimension. These *dimension surrogate keys* are simple integers, assigned in sequence, starting with the value 1, every time a new key is needed. The date dimension is exempt from the surrogate key rule; this highly predictable and stable dimension can use a more meaningful primary key. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/dimension-surrogate-key/

↖**T**

Term	Definition
Transaction Fact Tables	A row in a *transaction fact table* corresponds to a measurement event at a point in space and time. Atomic transaction grain fact tables are the most dimensional and expressive fact tables; this robust dimensionality

	enables the maximum slicing and dicing of transaction data. Transaction fact tables may be dense or sparse because rows exist only if measurements take place. These fact tables always contain a foreign key for each associated dimension, and optionally contain precise time stamps and degenerate dimension keys. The measured numeric facts must be consistent with the transaction grain. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/transaction-fact-table/
Type 4: Add Mini-Dimension	Slowly changing dimension *type 4* is used when a group of attributes in a dimension rapidly changes and is split off to a*mini–dimension*. This situation is sometimes called a *rapidly changing monster dimension*. Frequently used attributes in multimillion-row dimension tables are mini-dimension design candidates, even if they don’t frequently change. The type 4 mini-dimension requires its own unique primary key; the primary keys of both the base dimension and mini-dimension are captured in the associated fact tables. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/type-4-mini-dimension/

V

Term	Definition
Value Chain	A *value chain* identifies the natural flow of an organization’s primary business processes. For example, a retailer’s value chain may consist of purchasing to ware- housing to retail sales. A general ledger value chain may consist of budgeting to commitments to payments. Operational source systems typically produce transactions or snapshots at each step of the value chain. Because each process produces unique metrics at unique time intervals with unique granularity and dimensionality, each process typically spawns at least one atomic fact table. Source: http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/value-chain/