The data warehouse toolkit Chapter 1

💡 A DW/BI system must always first start with needs of a business in mind. With the needs firmly in mind we can work backwards through the logical and then physical designs, along with decisions about technology and tools.

Data capture and analysis

💡 Today data is one of the most important assets and organization has. It is almost always used for two purposes: operational record keeping and analytical decision making. The operational systems are where you put the data in, and the DW/BI system is where you get the data out.

Operational systems

💡 Operational systems do not maintain history but rather update data to reflect the most current state of a business.

DW

💡 Data warehouse users however, capture these states over time and use this data to evaluate performance. When this data is captured over a period of time, and used on mass it can provide tremendous value.

Goals of DW/BI

The DW/BI system must make information easily accessible

The contents of these systems must be easily understandable. The data must be intuitive and obvious to the business user, not merely the developer. The data structure and labels must mirror the thought process and vocabulary of the business user. **Simple to use and fast to access are key.**

The DW/BI system must present information consistently

The data in the system must be credible, it must be carefully assembled from a variety of sources, cleansed, quality assured and released only when fit for consumption. Consistency is crucial, if two measures have the same name, they must mean the same thing.

The DW/BI system must adapt to change

User’s needs, business conditions, data and technology are all subject to change. The system must be designed to handle changes gracefully. Existing data and applications should not be changed when new questions are asked or data is added.

The DW/BI system must present information in a timely way

When operational decisions are made raw data needs to be transformed into actionable information within hours.

The DW/BI system must be a secure bastion that protects the information assets

💡 The system must control access to the organizations confidential information, which is it’s crown jewel.

The DW/BI system must serve as the authoritative and trustworthy foundation for improved decision making

💡 The system must have the right data to support decision making. The most important output from this system are the decisions that are made atop the data that it delivers.

The business community must accept the DW/BI system to deem it successful

💡 The business community needs to embrace the system for it to be successful. It will be embraced if it is a **simple and fast source** for actionable information.

⚠️ The final two points are the most important rules. A successful DW requires you to have one foot in the IT door and one in Business.

DW/BI managers

💡 The main responsibility of a DW is to serve the users. There is a critical need to focus outwards on customers as opposed to inwards on products and processes. The technology used in DW is a means to an end.

Key responsibilities as a DW manager

Understand the business users

- Understand their job responsibilities, goals and objectives.
- Determine the decisions that the business users want to make with the help of the DW system
- Identify the best users who make effective, high-impact decisions
- Find potential new users and make them aware of the DW system's capabilities

Deliver high-quality, relevant, and accessible information and analytics to the business users

- Choose the most robust, actionable data to present in the DW system, carefully selected from the vast universe of possible data sources in your organization
- Make the UI and applications simple and template-driven, explicitly matched to the users' cognitive processing profiles
- Make sure the data is accurate and can be trusted, labeling it consistently across the enterprises
- Continuously monitor the accuracy of the data and analyses
- Adapt to changing user profiles, requirements, and business priorities, along with the availability of new data sources

Sustain the DW/BI environment

- Take a portion of the credit for the business decisions made using the DW/BI system, and use these successes to justify staffing and ongoing expenditures
- Update the system regularly
- Maintain the business user's trust
- Keep the users, exec sponsors and IT management happy

Dimensional modeling

💡 Dimensional modeling is accepted as the best modeling technique for presenting analytical data because it addresses two requirements:

- Deliver data that is understandable to the business users
- Deliver fast query performance

💡 The ability to visualize something as abstract as a set of data in a concrete and tangible way is the secret of understandability. A data model that starts simple has a good chance of remaining simple.

Dimensional modeling vs 3NF

💡 The core different between both models is the degree of normalization.

3NF

💡 3NF is incredibly useful in operational processing because an update or insert transaction touches the database in only one place. Normalized models however, are too complicated for analytical queries. Additionally most query engines can efficiently query a normalized model.

Dimensional model

💡 A dimensional model contains the same information as a normalized model, but packages the data in a format that delivers user understandability, query performance and resilience to change.

Star schema vs OLAP cubes

💡 Dimensional models implemented in RDBMS are known as star schemas. Dimensional models implemented in multidimensional database environments are known as OLAP cubes.

- [ ] Look more into OLAP cubes

Fact tables for measurements

✏️ The fact table in a dimensional model stores the performance measurements resulting from an organization’s business process events.

💡 Measurements data is overwhelming the largest set of data. It should not be replicated, and should serve as a single source of truth to ensure consistency for business users throughout the enterprise.

💡 **Each row in a fact table corresponds to a measurement event.** The data on each row is at a specific level of detail, known as the grain. A core tenant of dimensional modelling is that all the measurement rows in a fact table must be at the same grain. This ensures that measurements aren’t double counted.

💡 A measurement event in the physical world has a one-to-one relationship to a single row in the corresponding fact table is a bedrock principle. Everything else builds from this foundation.

Facts

✏️ A fact represents a business measure. Such as the price of a product, or the quantity when sold. The most useful facts are numeric and additive.

Additivity

💡 Additivity is crucial because BI applications rely on many fact rows at a time to extract value and the most useful thing to do is add them up.

Non or semi additive facts

💡 Sometimes facts can be semi-additive or non-additive. These can’t be added but can be used with averages or counts to have value.

Textual data as facts

💡 It is possible to have text as a fact, however this is rare and in most cases a textual measurement is a description of something and is drawn from a discrete list of values.

💡 Fact tables tend to be quite sparse, and deep in terms of the numbers of rows but usually contain less columns.

Fact grain

💡 Fact grains fall into one of three categories:

- Transaction
- Periodic snapshot
- Accumulating snapshot

Foreign keys

💡 All fact tables have two or more foreign keys that to connect to the dimension tables' primary keys. When all keys in the fact table correctly match their respective primary keys in the corresponding dimension tables, this satisfies **referential integrity. You can access the fact table via the dimension tables joined to it.**

Composite key

💡 The fact table generally has its own primary key composed of a subset of the foreign keys. This key is often called a **composite key.** Every table that has a composite key is a fact table. Fact tables expression many-to-many relationships. All others are dimension tables.

Dimension tables for descriptive context

💡 Dimension tables are integral companions to a fact table. The dimension table contains the textual context associated with a business process measurement event. They describe the "who, what, where, when, how, and why" associated the event. Dimension tables often have many columns and attributes. Each dimension is defined by a single primary key, which serves as the basis for referential integrity with any given fact table for which it is joined.

Dimension attributes

💡 Dimension attributes serve as the primary source of query constraints, groupings and report labels. Attributes should consist of real words as opposed to cryptic abbreviations.

Value of attributes

💡 A data warehouse is only as good as the dimension attributes; the analytical power of the DW/BI environment is directly proportional to the quality and depth of the dimension attributes.

Fact vs attributes

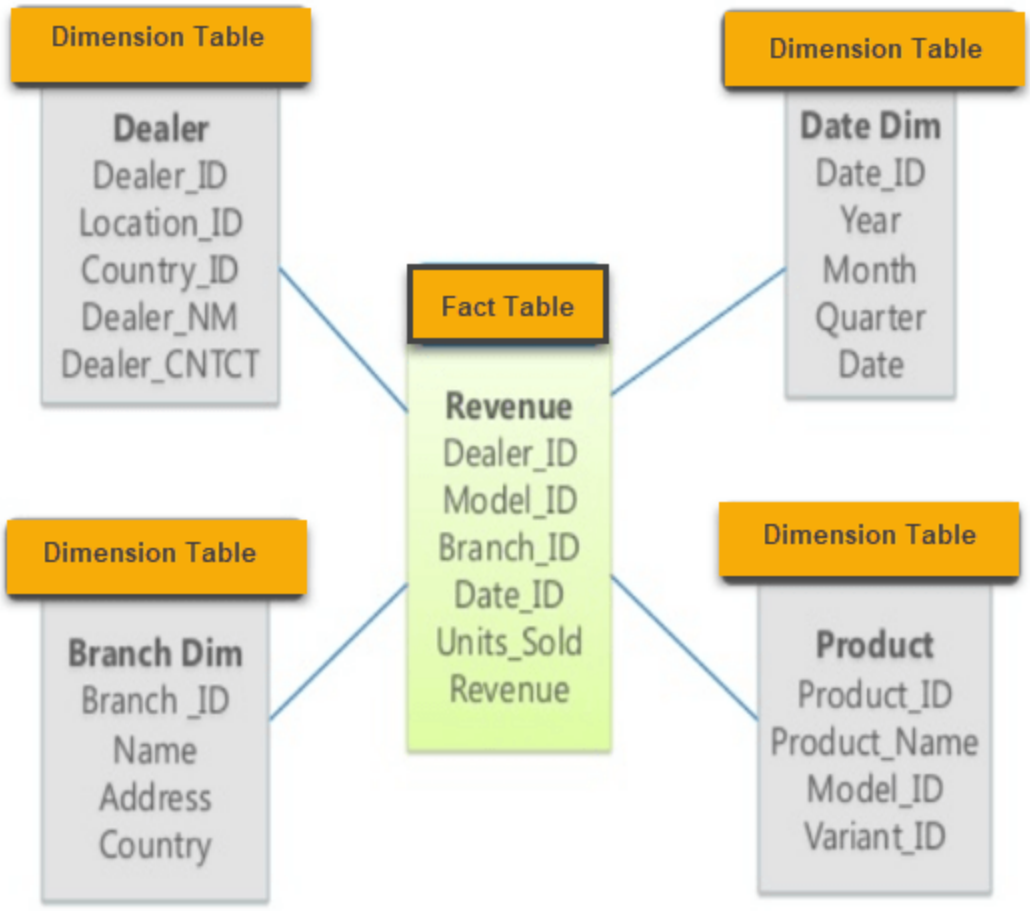
💡 To decide between a fact or attribute from source data, you should ask whether the column is a measurement of some event that takes on lots of values and participates in calculations or is a discretely valued description that is more or less constant and participates in constraints and row labels.

Snowflaking

💡 One should resist the urge to normalize data by storing only a code in the product dimension and creating a separate lookup table. Instead of a highly normalized form, dimension tables typically are highly denormalized with flattened many-to-one relationships within a single dimension table. The increase in storage capacity makes normalizing dimension tables a useless activity.

Facts and dimensions joined in a star schema

💡 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 is called a **star join.**



Data granularity

✏️ Grain is the specificity of the data. Granularity refers to the amount of detail in the data. For example, in sales data a more granular data point would be an individual transaction whereas a the sales per day would be less granular.

Dimensionality based on grain

💡 The most granular data has the most dimensionality, meaning that it has more attributes or dimensions than less granular data. A transaction might have several dimensions based on location, time, and product. Whereas a daily sum of a store has the dimension of location and time.

Adding dimensions

💡 In 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.

Operational source systems

💡 You should think of the operational source systems that capture a business's transactions as outside of the warehouse as you have no control over the content or format of the data.

Extract, Transformation, and Load system

💡 The ETL system is the DW/BI environment consists of a work area, instantiated data structures, and a set of processes. Extraction is the first step in the process of getting data into the DW. Once the data is extracted and copied in it belongs to the DW. You will then transform the data such as cleansing it or dealing with missing elements. This adds value to the data. It can also create metadata for data quality. The load part is the physical structuring of the data into the presentation area's target dimensional models. 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.

3NF format

💡 Sometimes the data arrives in a 3NF relational format. ETL system developers may be more comfortable performing the cleansing and transformation tasks using normalized structures. However, this is not the end goal. We should denormalize the data before it arrives to the warehouse.

Presentation area to support BI

💡 The DW presentation area is where data is organized, stored, and made available for direct querying by users, report writers, and other analytical BI apps. This area is all the business users are concerned with.

💡 The data in a presentation area should be presented, stored and accessed in dimensional schemas, either relational start schemas or OLAP cubes.

Detailed data

💡 The presentation area must also contain detailed data that is atomic. This is required to withstand unpredictable ad hoc queries. We should never only store summary data in dimensional models while the atomic data is locked up in normalized models. Users may still be interested in seeing the most fine grain data.

Structure

💡 The presentation area should be structured around business process measurements events. This naturally alligns with the operational source data capture systems. They should not be desined to deliver the report of the day. **You should construct a single fact table for atomic sales metrics rather than populating several similar databases.**

Warehouse bus architecture

💡 All the dimensional structures must be built using common, conformed dimensions. The bus architecture is crucial in the presentation area as it prevents standalone tables that cannot be tied together. This is the bane of DW as it will lead to incompatible views of the enterprise. The bus architecture is a framework that enables agile, decentralized, realistically scoped, and interatively made DW.

💡 The presentation area ultimately consists of dozens of dimensional models with many of the associated dimension tables shared across fact tables.

⚠️ Data should not be structured in accordance to individual departments' interpretation of the data.

BI applications

💡 A BI application refers to the range of capabilities provided to business users to leverage the presentation are for analytic decision making.

Restaurant metaphor for the Kimball architecture

ETL in the back room kitchen

💡 The data warehouse ETL system the restaurants kitchen. Source data is magically transformed into meaningful, presentable information. The back room ETL system must be laid out and architected long before any data is extracted from the source. The kitchen is designed to ensure throughput. Data quality comes in like raw ingredients that must be checked , conditions are continually monitored to ensure high integrity and the kitchen is cut off from the presentation area where the users eat.

Data presentation and BI in the front dining room

💡 Disconnected from the kitchen this where people eat the exploits provided by the kitchen. The kitchen is off limits to the restaurant patrons. Restaurants are usually judged on four metrics:

- Food (quality, taste and presentation)
- Decor (appealing)
- Service (prompt food delivery, attentive staff)
- Cost

Alternative DW architecture

Independent data mart architecture

💡 Analytic data is deployed on a departemental basis without concern to sharing and integrating information across the enterprise. This is an issue as many departments might be interested in the same source data. With independent sources of truth the metrics across departments do not add up.

Hub and Spoke corporate information factory inmon architecture

This is similar to the kimball method as CIF advocates enterprise data coordination and integration. But CIF says the normalize Enterprise Data Warehouse fills this roles, whereas KImball stresses the important of an enterprise bus with conformed dimensions.

Hybrid hub and spoke and kimball architecture

This may be useful but it will require a higher budget and organizational patience to fully normalize the data and instantiate it before it into dimensional structures.

Dimensional modeling myths

Dimensional models are only for summary data

This is wrong as you cannot anticipate the business uses and queries on the data, and therefore serving granular data will allow your business users to find uses for it and roll it up based on their business questions.

Dimensional models are departmental, Not enterprise

Rather than drawing boundaries based on organizational departments, dimensional models should be organized around business processes, such as orders, functions and invoices.

Dimensional models are not scalable

💡 Fact tables can scale extremely easily and new dimensional tables can be added very easily.

Dimensional models are only for predictable usage

💡 Dimensional models should not be desined based on predefined reports or analyses, the design should center on the measurement processes. This is because meansurement events are typically stable as opposed to analyses. The secret to query flexibility is designing fact tables at the most granular level. The correct starting is to present data at the lowest detail possible for maximum flexibility.

Dimensional models can’t be integrated

💡 The data warehouse bus architecture allows for the integration of dimensional models. Dimensions are built and maintained as centralized persistent master data in the ETL system and then reused across dimensional models to enable data integration and ensure semantic consistency.

More reasons to think dimensionally

💡 You should continuously ask yourself about the business process measurement events producing the report or dashboard metrics. You should rank each business process on highest value and feasibility and then tackle them based on highest business impact.

- [ ] Look more into the enterprise data warehouse bus matrix in chapter 4

Agile considerations

- Focus on delivering business value.
- Value collaboration between the development team and business stakeholders.
- Stress ongoing face-to-face communication, feedback, and prioritization with the business stakeholders
- Adapt quickly to inevitably evolving requirements
- Tackle deployment in an interative, incremental manner

> ⚠️ A common criticism of the agile approaches is the lack of planning and architecture, coupled with ongoing governance challenges. Use the bus matrix

- [ ] Look more into conformed dimensions