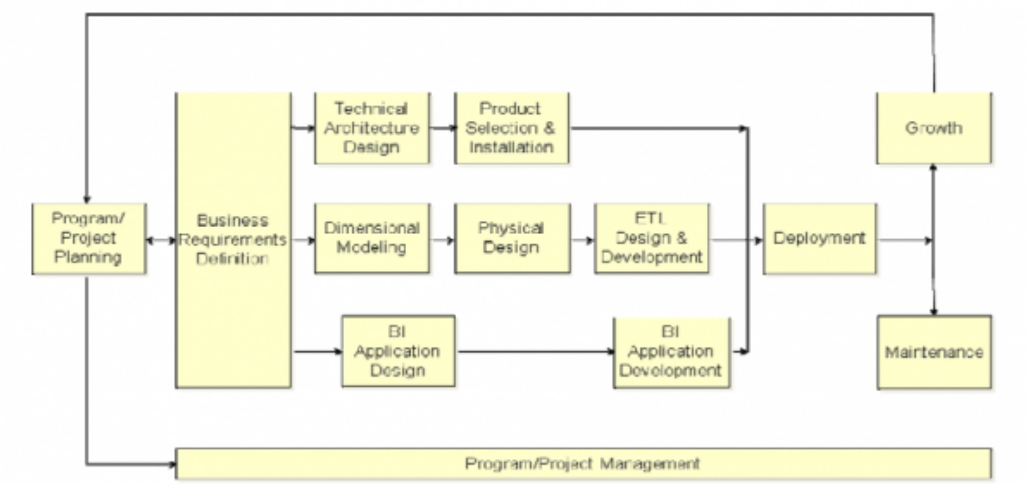
Chapter 6 – DIMENSIONAL MODELING

What is a Dimensional Modelling

I. Dimensional Modelling
   a. Suited for Data Warehouse and Business Intelligence applications
   b. Denormalized structure, optimized for faster retrieval of data
   c. Easier to understand and to use by business users
   d. Groups the data according to business categories
   e. Dimensional models stored are often referred to as star schemas
   f. Both the dimensional models and the 3NF models are logical models of data that can be physically stored in relational databases
   g. Part of the data warehouse lifecycle

II. Kimball Data Warehouse Lifecycle



III. Program/Project Planning

   a. Program
      a. Long term strategic approach, no fixed deadline
      b. Contain multiple projects, can be linked
      c. Governed by senior stakeholders
      d. Planning
         i. Why is data warehouse project is required
         ii. Access Readiness
            1. Sponsor support, investment, culture, resources, and business motivation

        iii. Benefits/ROI new data warehouse project will bring in
- b. Project
    - a. Aim is to deliver one or more business products
    - b. Is time-bound, has an end date
    - c. Contains project team
    - d. Planning
        - i. Develop project plan (Cost, Resourcing, Start Date, Completion Date)
        - ii. Project team assemble

IV. Requirement Gathering

- a. Interview with Business Users & Executives
    - a. Understand the needs of the business
    - b. Understand KPI, process, issues, all the information to support their analytical needs
- b. Data source analysis or data profiling
- c. Documentation write up
    - a. Findings from interviews
    - b. Success criteria
    - c. Data sources list
    - d. Business Processes identified
        - i. Operational activities performed by your organization
        - ii. Generate or capture performance metrics that translate into facts in a fact table
        - iii. Fact tables focus on the results of a single business process
        - iv. Defines a specific design target
        - v. Allows to define the grain, dimensions, and facts
        - vi. Corresponds to a row in the bus matrix
    - e. Users stories
    - f. Business Processes laid out in Enterprise Bus Matrix
        - i. Essential tool to implement the Dimensional Data Warehouse
        - ii. Defines high level entities
        - iii. Rows are business processes
        - iv. Columns are dimensions
        - v. Helps prioritize the project direction and workload

V. Concept and Steps of Dimensional Modeling

a. Dimension
    a. A dimension is essentially a descriptive information about facts
    b. Example: Customer, Date, Products
    c. Attributes defines characteristic of a dim (customer_name, product_description)
b. Fact
    a. A "fact" indicates business measurements or business activity (Transaction, sales)
    b. Example Business Activity: Product was sold for $50
    c. Measurement in fact defines quantitative values (count, sum, avg)
    d. Example measure: price $50
c. 4 Steps of Dimensional Modeling
    a. Select the business process
        i. Operational activities performed by your organization
        ii. Capture performance metrics that will get translated into fact tables
    b. Declare the grain
        i. A critical process that establishes what a single row of fact table represents
        ii. Must be declared before anything else as it dictates the design of dim and fact tables
    c. Identify the Dimensions
        i. Who, what, where, when, why, how context of a business process
        ii. Whenever possible a dim should be a single value when associated with a given fact
    d. Identify the Facts
        i. Measurements in numeric values that result from a business process
        ii. Only facts consistent with the declared grain are allowed

VI. Declare the Grain

a. Declare the grain means defining the level of detail for your star schema
b. Indicates the lowest level at which data is captured (Daily, hourly, monthly, etc.)
c. Both Dim and Fact table contains some level of details
d. In a fact table, individual transactions, line item of order contains level of detail

e. Data should be stored as granular as possible
f. Granularity should be defined before identifying dim and fact

VII. Dimensions

a. Dimension tables contain descriptive fields
b. Usually flat denormalized table
c. Have a single primary key column
d. Attributes are the primary target for declaring constraints
e. Dimension characteristics should be verbose (full words), descriptive, and complete (no missing values)
f. Usually represent many-to-one hierarchical relationships

VIII. Types of Dimensions

a. Conformed Dimensions
    a. Common dimensions that are joined to multiple facts
    b. Provides same structure, attributes, and same meaning in every fact table
    c. Improved data consistency
    d. Every row is unique and is at atomic level
    e. Enables cross process analysis by allowing various fact in same query
    f. Easy to update as all business rules are in one place
    g. Contains primary and surrogate keys
    h. Date dimension is common conformed dimension (year, month, week, days, etc.)
    i. Conformed Dimensions are needed to build Dimensional Data Warehouse

b. Junk Dimensions
    a. Dim with simple attributes (flags, yes/no, true/false, id/description)
    b. Values that do not change frequently
    c. Eliminate small dimensions for performance and better management
    d. Group highly correlated attributes into a single dimension
    e. Reduce the number of dimensions (<= 26 dimensions per fact table)
    f. Reduce the number of columns in the fact table
    g. Reduce joins between facts and dimensions
c. Degenerate Dimensions
    a. Dimension without attributes
    b. Receipt/Invoice number, tracking number, order number, etc.
    c. It is stored inside a fact table to reduce duplications (Fact dimensions)
    d. Degenerate dimensions are added due to grain of fact tables
d. Role-Playing Dimensions
    a. Same dimension used for multiple purpose
    b. Used multiple times within the same fact giving different business context
    c. Best example is date dimension
    d. fact_order contains (order_date, due_date, cancelled_date etc.)
    e. Don't create multiple dim, instead create one dim_date
e. Slowly Changing Dimensions
    a. Dimensions that change over time
    b. Manages current and historical version
    c. Build to track changes
    d. There are different types of SCD such as:
        i. Type 0 – Retain original
            1. There is essentially no change
            2. The attribute will never change
            3. Facts always grouped by the original value
        ii. Type 1 – Overwriting the Old Value
            1. Overwrite Old Value
            2. Attributes always reflect the most recent assignment, disregarding historical changes
            3. This is easy to implement but not the most optimum solution as we lose track of historical data
        iii. Type 2 – New Additional Record
            1. Create new additional record
            2. New primary key (SK), flag column, and date columns added to track

iv. Type 3 – Adding a New Attribute Column
    1. Add a new column to the dimension to preserve historical information
    2. Allows to query in 2 different realities
v. Type 4 – Using Historical Table
    1. Historical table used to track any changes separate to the dimension table
    2. The main dimension only keeps current data based on the present time period
vi. Store as Snapshots
    1. Use table partitions and store as snapshot for dimensions
    2. All data is added to snapshot daily or weekly

e. Bridge Tables
    i. Used to resolve many-to-many relationships
    ii. Sits between Fact and Dimension
    iii. Only contain key columns for the various tables
    iv. Access requirement before implementing
    v. Loading of table can be complex

IX. Facts

a. Contains the measurement created by an operational system
b. At the lowest granularity captured by the business process
c. Design entirely based on a physical activity, not influenced by the report
d. Contain foreign keys for each dimension associated
e. Measures are used for queries and aggregations
f. Primary key is usually a composite key

IX. Types of Facts

a. Measures
    a. Additive
        i. Measures that can be summed across any of the dimension within the fact table
        ii. Results we get from aggregations is useful and gives us business meanings
        iii. SUM, GROUP BY

b. Semi-Additive
    i. Measures that can be summed across some of the dimensions within the fact table
    ii. Some values do not provide business value or is misleading
c. Non-Additive Facts
    i. Measures that cannot be summed across any of the dimensions within the fact table
    ii. Percentage and unit price are examples

b. Tables
    a. Transaction Fact Tables
        i. Most common in dimensional modeling
        ii. Grain one row per transaction
        iii. Lowel level of granularity and date dimension
        iv. Additive measures
        v. Can grow very large quick
        vi. No update happens in these tables
    b. Periodic Fact Tables
        i. Snapshot of data for specific time (Day, Week, Month, Hours, etc.)
        ii. Grain is one row per time period
        iii. Semi-additive
        iv. Usually built from Transaction Fact Table
        v. Smaller table size compared to Transaction Fact Table
        vi. Useful to get overview of KPI's
    c. Accumulating Fact Tables
        i. One row per entire lifetime of an event or product
        ii. Has beginning and an end date
        iii. Contains multiple date columns
        iv. Update happens when each milestone is completed
        v. Example: Processing of an order, insurance processing, material processing
        vi. Aggregation can be difficult to perform
        vii. Smallest in table size

X. Star Schema

a. Pros
    a. Simpler queries compared to a normalized model
    b. Simplified business reporting logic
    c. Better performing queries

b. Cons
    a. Data integrity not enforced
    b. Does not inform many to many relationships
    c. Dependent on business process

XI. Snowflake Schema

a. Pros
    a. Improved data quality as data is more structured
    b. Uses less storage space than a denormalized schema
    c. Suitable for data with deep hierarchies
    d. Easier to design and develop
b. Cons
    a. Requires more complex queries
    b. Increase number of joins potentially impacting on performance
    c. Level of integrity still lower than a highly normalized schema
    d. Difficult for Business Users to understand data