Introduction to Data Warehousing

Introduction

- Organizations need a single source of the truth from which to analyze in order to understand the different aspects of their business and gain a business advantage. Typical organizations have their data:
    - Scattered throughout the organization in different DBMS from same or different vendors, and in files of different formats, e.g. MS Word, Excel, plain ASCII files,…
    - Shared data on servers across the organization or localized to a department or specific user, e.g. An MS Access database, a spreadsheet, etc.
    - Stored at the organization's data center or at the vendor's data center when using a cloud based application. For example, the inventory system could be hosted at the corporate data center but the ERP system is hosted by SAP.
    - Duplicated across departments, e.g. the culinary department and the supply systems department will have different copies of the restaurant's menu recipes for different purposes.
    - Stored by similar name but have different definition, e.g. Sales could be gross sales in one system but net sales in another system.
- In addition to the data problems above, the business analysis is also scattered throughout the organization, therefore analysts may reach different conclusions about certain business metrics, e.g. the operations, the marketing, and the finance departments may state different %sales increase for the week over last year's performance.
- Whether small, large, or very large, the DW has become a necessity for today's businesses and not a "nice to have" because it's a centralized single source of the truth.
- DW is not a transactional data storage (OLTP) where you have users entering, editing and deleting data on regular basis (e.g. an inventory system) but instead a dedicated optimized storage for reporting, multi-dimensional analysis, and predictive analysis. OLTP systems are the primary source of data for the DW system.
    - OLTP supports operational processing; DW supports analytical processing
    - OLTP has predictable data processing patterns: insert, edit, delete, some queries; DW has a less predictable pattern of data queries, low to medium transaction throughput.
    - OLTP contains typically very granular data; DW contains lightly and highly summarized data plus some detailed data.
    - The age of data in an OLTP is current; DW's data is typically historic but trending towards new and current.
    - Reporting in an OLTP is relatively static and fixed; DW reporting is unpredictable, multi-dimensional, and dynamic.
- A DW is a "*subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision making process*"
    - Subject oriented. Warehouse is organized around the data and not the application that manages the data in its corresponding OLTP.

    - Integrated. The data will be centralized and transformed into the same format, same definition.
    - Time-variant. The warehouse contains the data at a various points in time unlike an OLTP that just updates the current data with newer data, e.g. a customer address.
    - Nonvolatile. The data is a snapshot and will not be updated but instead supplemented with new snapshots at pre-determined time intervals.
        - Exceptions exist in the case of corrupted or incomplete data loads.
- Also referred to as Information Warehousse or Enterprise Data Warehouse (EDW).
- What used to be known as *Decision Support* has grown to become *Business Analytics* and *Business Intelligence*[1]. BI is something you create by asking the right questions of the right data using the right analytical tools[1.]

Data Warehousing Challenges

- Underestimation of resources required or data ETL – varies based on sophistication of the ETL tools and the extent of knowledge of the Analyst of the source data system (schema or API)
- Hidden problems with source systems – unclean data and incomplete data issues, e.g. the data field is text in the source system therefore allowed badly formatted dates are in the system; or, the start of employment date is Null for some employee data.
- Required data not captured – the requirements of the DW calls for data that the organization doesn't own, doesn't currently capture, or doesn't capture to the required granularity. The missing data could either be in the form of an entire data set, e.g. Weather data that is not captured or payroll data that is captured at the weekly and not daily or hourly level.
- Data homogenization. To produce a consolidated and integrated view of the organization's data.
- High demand for resources. Data warehouses accumulate a lot of data, require a lot of aggregations, a lot of indexes, and many more database objects that require space.
- Data ownership. A DW creates awareness of the importance of data quality and sensitivity due to the visibility of data across all departments.
- High maintenance. Data Warehouses have to be updated as the source systems are updated and as business processes change.
- Long-duration projects. DW projects typically take over a year or two to complete.
- Complexity of integration. A DW project requires various tools to be integrated together in order to work with uniformity on the DW platform.
- Management challenges. Many stakeholders: Executives, Directors, Managers, Analysts, end-users. Who is the most important to satisfy? Implementation decisions cater to different crowds.

Data Warehouse DBMS

- Similar to typical DBMS requirements of scalability, reliability, availability with a focus on performance, i.e. speed of access to the information.
- Performance
    - Load performance. DW will typically need to load gigabytes of data per hour
    - Load processing. Ability to perform data conversion, formatting, integrity checks, indexing, metadata update,…

    - Query performance. Ah-hoc analysis for complex Analyst queries must perform very quickly. Also, the underlying queries used by the report and cube generator need to perform at very high speed.
- Warehouse administration.
- Support of Parallelism: Symmetric multiprocessing (SMP) that share memory and disk; Massively Parallel Processing (MPP) shared nothing architecture. Must be capable of parallel queries.

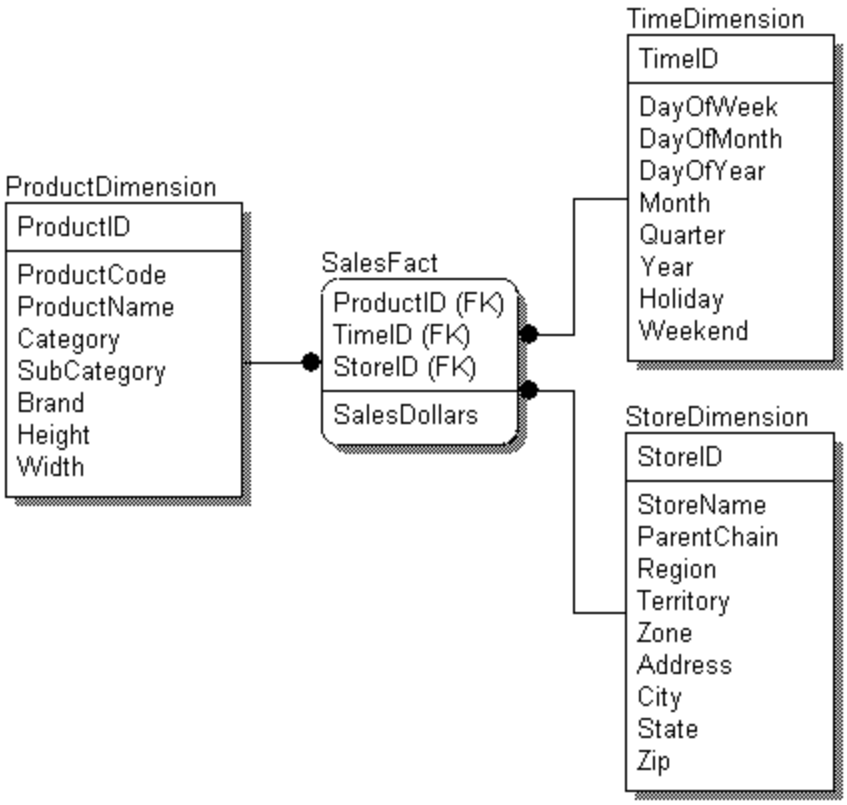
Data Warehouse Architecture Concepts

- Operational data, source data, input data is typically the transactional data in the data format needed for the business applications.
- An Operational Data Store (ODS) is a DW like storage containing a centralized cleaned operational data. It may be used as a "staging area" (SA) for populating the DW.
- A Staging Area (SA) is where the data is cleaned, transformed, combined, and duplicated to prepare the data for a bulk load into the data warehouse.
- ETL – Extract/Translate/Load
    - Extraction of data from the source (databases, files, systems)
    - Translation of the data from its original form to the form needed inside the DW which may involve anything from a simple name change to aggregation of data.
    - Loading of data into the DW. Frequency of the loads, impact on DW availability.
    - Extraction could occur into an ODS/SA (Operational Data Source or Staging Area)
    - Loading could occur during the transformation process or afterwards.
- Data profiling and data quality controls – To produce a good ETL process, source data can be analyzed to assess missing data, null data, incomplete entries, and the distribution of values in each column (e.g. how many sales checks are ToGo orders vs dine in vs Catering).
- Metadata: ETL metadata, warehouse metadata.
- End-User Access Tools – Users interact with the DW using tools. Critical to pre-plan requirements for joins, aggregations, and reporting for high performance.
    - Reporting Tools – Report writing and report generation.
    - Query tools.
    - Application Development tools – to build custom applications
    - Dashboarding tools.
    - OLAP tools.
    - Data mining tools.
- Data Marts
    - A database that contains a subset of the Warehouse data to support the analytical requirements of a particular business unit (such as the Marketing department) or to support users who share the same requirement to analyze a particular business process (such as property sales).
    - It could be a star schema modeled on a particular business process, e.g. Restaurant Server Tips or a database containing all data related to a particular department, e.g. Sales Department.

Data Warehousing Designs

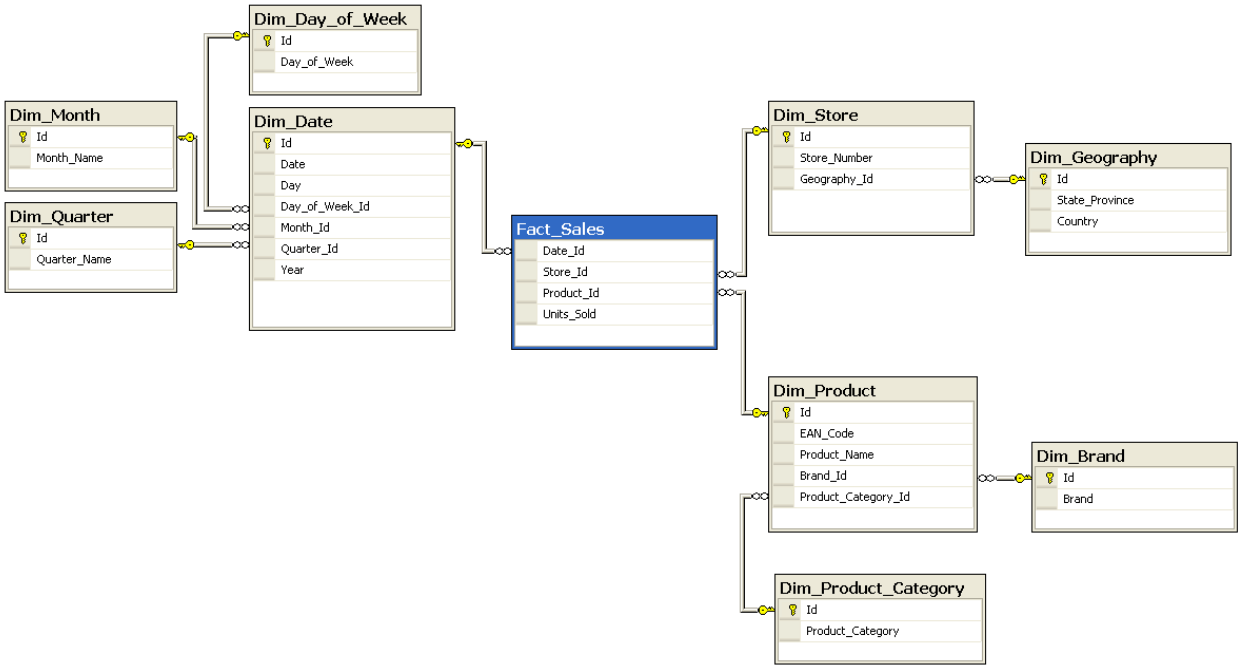
- Conducting interviews with the business users to gather requirements.

- Understanding the budgetary restrictions in terms of time, people, and funds. A data mart may be more appropriate.
- Assessing the appetite of the organization for long projects.
- Inmon's Corporate Information Factory approach
    - Start by creating a data model of all the Enterprise's data.
    - Build the EDW using that model and create Data Marts from there to the different groups.
    - Uses traditional db techniques
    - Is favored when the goal is to achieve EDW as the very first deployment; project will take a long time to produce any deliverables.
- Kimball's Business Dimensional Lifecycle approach
    - Identify the information requirements and the business processes
    - Create a Data Mart for each business unit
    - Integrate all the data marts into an EDW, a difficult task once they spread throughout the Enterprise and are heavily used for critical functions.
    - Is favored when individual departments have to be taken care of quickly and have something tangible for particular departments in a shorter amount of time, i.e. is quicker to demonstrate value even if it's not at the Enterprise level.
- Dimensional Modeling
    - Unlike ER modeling, it is not focused on normalizing tables to reduce redundancy. ER Modeling results in a good OLTP schema but not efficient for ad hoc queries due to all the needed joins. Dimensionality modeling is a logical design technique that aims to present the data in a standard, intuitive form that allows for high-performance access.
    - Primary components are the Fact table and the Dimension tables which form the Star Schema.



- The fact table contains a number of foreign keys to the Dimension tables plus the facts, also called measures.
- The Dimension tables have simply integer surrogate keys instead of natural keys.
- The primary key of the fact table consists of a composite key of foreign keys.
- Star Schema is a dimensional data model that has a fact table in the center, surrounded by denormalized dimension tables.
- The facts/measures are the data in the fact table other than the foreign keys, e.g. SalesDollars is the only fact in the above schema.
    - Facts are a read only fact that will not change.
    - Facts are numeric. A salesman name or an address is not a valid fact.
    - Facts are continuously valued. A sales for the hour fact has a different value whenever it is measured every hour.
    - Facts are additive so they can be rolled up. Profit margin % as a fact for every menu item on a restaurant menu is not additive.
    - Facts have to be of the same granularity, e.g. if the grain is hourly, a sales table will have the sales for the hour on a row but cannot have a Monthly Rent fact.
- Dimension table contain descriptive information about the facts. It is how the data/facts/measures will be viewed for analysis by the business. In the example above, the Dimension table describe what was sold, where, when, …they also act as constraints for the queries, i.e. limited set of valid values.
- Dimension table are denormalized to reduce the number of joins needed and hence query performance.

- Snowflake Schema is a dimensional data model that has a fact table in the center, surrounded by normalized dimension tables.



- A hybrid of normalized and denormalized dimensions is call Starflake schema
- Advantages:
    - Efficiency. The consistency of the schema allows for more efficient access to the data by various tools such as report writers and query tools, i.e. all joins to dimensions occur through the fact table and that will not change as we add dimensions and facts.
    - Extensibility. It's easy to add facts (as long as same granularity as rest of facts), easy to add dimensions, easy to add dimensional attributes or breaking existing records into lower levels of granularity from a certain point in time forward, e.g. we can change the time dimension from year to month, put a default value for historical data and capture more accurate facts moving forward.

Suggested Reading

Data Warehouse Gotchas: http://www.dwinfocenter.org/gotchas.html

http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/

References

1. Business Intelligence and the Cloud, Michael S. Gendron.