Relational Data Modeling

Saturday, April 6, 2019 6:45 AM

What is a Database Management System?

All information in a relational data base is represented explicitly at the logical level and in exactly one way – by values in tables.

Edgar Codd's 12 Rules - https://en.wikipedia.org/wiki/Codd%27s_12_rules

Computer software that interacts with a database and provides access to the data inside of it

OLAP – Online Analytic Processing – Database System Optimized for complex queries. Optimized for reads
OLTP – Online Transaction Processing - Database System Optimized for Transactions (Real time applications)

Normalization

- Normalization reduces data redundancy, and increases data integrity'
- Denormalization will improve performance for read heavy queries (more focused on the queries that will be run on them)
- We want to have the ability to update our data in once place and have it be the single source of truth

Objectives of normalization

- Update data in one place, thus decreasing unnecessary work
- Reduce the amount of refactoring needed when new items are added to the database
- To make the model more informative to users
- To be agnostic to query statistics (ie perform all queries equally well)

Normal Forms

First Normal Form (1NF):

- Atomic values: each cell contains unique and single values
- Be able to add data without altering tables
- Separate different relations into different tables
- Keep relationships between tables together with foreign keys

Second Normal Form (2NF):

- Have reached 1NF
- All columns in the table must rely on the Primary Key

Third Normal Form (3NF):

- Must be in 2nd Normal Form
- No transitive dependencies
- Remember, transitive dependencies you are trying to maintain is that to get from A-> C, you want to avoid going through B.

Denormalization

Increasing read performance (by joining data in advance) at the expense of losing write performance (due to duplication of data)

https://en.wikipedia.org/wiki/Denormalization

Denormalization must be done after normalization (and requires more space)

- Not a limiting factor as storage has gotten less expensive over time

Fact and Dimension Tables

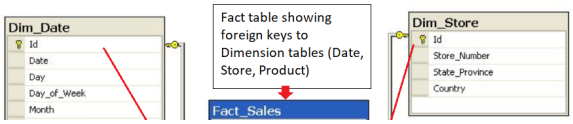
- Work together to create an organized model
- Conceptually important for organization

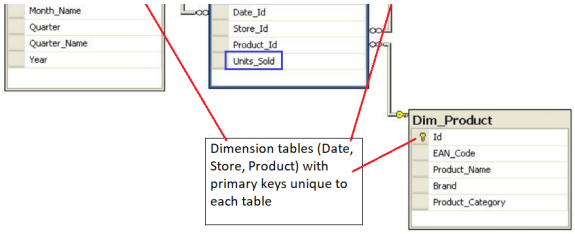
Fact Tables – Made up of facts (events that have actually happened) - measurements, metrics etc. Normally are numeric in nature (say the amount a customer spent in a particular month)

- Not meant to be updated in place as they are reporting a fact which is immutable

Dimension Table – Categorical Data relating to the fact table (people, products, places, time)

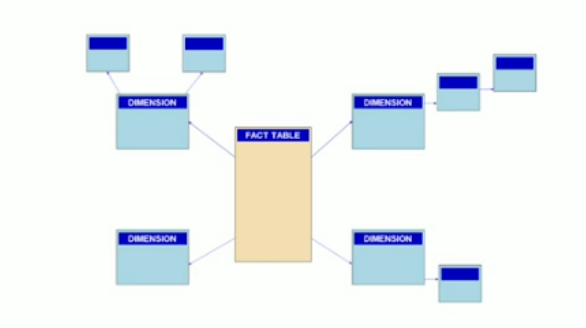
Example:





Schemas for data marts based on fact and dimension tables

- Star Schema
  - Simplest Option
  - 1 or more fact tables with any number of dimension tables (see example above)
  - Fact table has many FK's to all the dimension tables (aka the points of the star)
  - **https://en.wikipedia.org/wiki/Star_schema**
  - Benefits of a star schema
    - Deformalized for speed and simplified queries
    - Fast aggregations
  - Drawbacks
    - Issues around denormalization
    - Data Integrity (due to duplication)
    - Decreased Query Flexibility (due to modeling the schema to a query)
    - Many to Many aren't supported.
- Snowflake Schema



  - Centralized fact table that relates to multiple dimension schemas (star schema is a simplified snowflake schema due to only one level of dimension tables
  - Generally more normalized than a star schema

Summary

- What makes a database a relational database and Cobb's 12 rules of relations database design
- The difference between different types of workloads for databases OLAP and OLTP
- The process of database normalization and the normal forms.
- Denormalization and when it should be used.
- Fact vs dimension tables as a concept and how to apply that to our data modeling
- How the star and snowflake schemas use the concepts of fact and dimension tables to make getting value out of the data easier.