D1B - ER Modelling Vs. Dimensional Modelling

Relational Database

Collection of tables and relationship between them.

- DBMS (a database management system): a software application that allows you to create, update, organize and query data from one or many databases (eg.MySQL)

Data Modelling

Technique to determine what data and relationship should be store in a database. We use data model to conceptually design our data.

- ER diagram (for database)
- Dimensional modelling (for data warehouse)

3 Level of Data Modelling

	Conceptual	Logical	Physical
Entity name	Y	Y	
Entity relationship	Y	Y	
Attribute		Y	
Primary key		Y	Y
Foreign key		Y	Y
Table name			Y
Column data type			Y

ER Modeling Vs. Dimensional Modeling

ER Modeling	Dimensional Modeling
Normalized	Denormalized
Many tables, many joins	Few tables, few joins
No redundancy	Allow redundancy
Slow response?	Fast response
Capture detail of operational information	Capture strategic information
Query is usually simple	Query may be complex
Not design for business users	East-to-use for business users
IT-driven	Business-driven
For centralized data warehouse (top-down)	For data marts (bottom-up)

ER Diagram (Used for Database)

ER diagram is a semantic, graphical data model

- It visually express business rules
- A communication toll between business people and IT people.
- Used to design **centralized data warehouse**

ER Diagram Building Blocks

Entity

- Nouns like student, customer, event
- We wish to collect data about entity
- Entity (type) is thing like city while entity instance is like Melbourne

Attribute

- Property or characteristic of an entity

Relationship

- Association between entities
- Verbs
- Has direction
- one-to-one / one-to-many / many-to-many
- optional / mandatory sign at both end

Key in ER Diagram

Candidate Key

An attribute (or a group of attributes) that an uniquely identify an entity instance is called candidate key

Primary Key

We choose one from candidate key to be primary key (underlined in diagram)

Foreign Key

A non-PK attribute in one entity that is PK in another entity.

Composite Key

A key that consists of more than one attribute.

Integrity Rules in ER Diagram

Entity Integrity

Every table must has a column which contain unique values (must has PK)

Referential Integrity

A value in a foreign key attribute of a table must be existing in the PK attribute of corresponding referenced table

Normalization (ER Diagram)

Normalization is the process for converting complex data structure to simple, stable structure.

- It is **good for operational database**, but not good for decisional database.
- Normalized relations (tables) contain minimum redundancy, and allow user to insert, modify and delete rows in table without error.
- It break one large table into several small tales.
- Normalised model is:

- robust and stable
- Minimally redundant
- flexible
- Storage efficiency

Anomaly in Unnormalized Data

Unnormalized table example: studentID, courseID, fee

- **Insertion Anomaly:**

Information of a course cannot be inserted until the first student enrolled.

- **Deletion Anomaly:**

If all students withdraw a course, we loose the information (courseID and fee) of the course.

- **Update Anomaly:**

If the fee of a course changed, we have to update every row about the course, otherwise inconsistence happens.

D2B - Dimensional Modeling

Dimensional Modeling (Used for Data Warehouse)

- Kimball's technique to design **data model** (Kimball 提出了bottom-up approach 和 dimensional modelling)
- Based on the **multi-dimensional model** of data, designed for **read-only database (data warehouse)**
- In dimensional modelling, data is **Denormalized** to make the DW easy-to-use for end-users, and support faster response time to complex query.
- Very simple, intuitive, easy-to-understand structure

- The resulting model reflects the business questions, but not the operational function
- Descriptive data

Denormalization

- Fewer tables (fewer join)
- Faster response to query
- Allow redundancy
- Aim to make the DW easy-to-use for end-users

Steps

1. Choose a business process (eg.Sale, EmployeeSale)
2. Choose the grain (eg. per order+line per row, per day per row)
3. Identify dimensions (eg. Customer, Product, Store, Date)
4. Identify facts (eg. UnitSales, DollarSales)

Granularity

Granularity is the level of detail.

- Finer/lower granularity, more detailed data, finer/lower grain.
- Larger/higher granularity, more summarized data, larger/higher grain.
- Higher storage space for finer grain
- More flexible querying and reporting for finer grain

Dimension Table

- Dimensions can be used to describes business events
- The attribute of dimension table includes what users would to sort, group, filter (customerNumber, StoreNumber)
- Each dimension should be conformed in the enterprise scope
- Generate surrogate key for each dimension table as PK, do not use the PK in source database

Embedded Hierarchy in Dimension Table

eg. Country > State > City in Employee dimension table
eg. Department > Team > Group in Employee dimension table
eg. Year > Quarter > Month > Day in Date dimension table

Surrogate Key

PK which is automatically generated. Used in DW to identify a row in dimension tables. Typically an integer.

Fact Table

- A Business event
- Contain DD (degenerate dimension)
- Contain FK (dimensional attribute) to connect dimension table
- Contain facts

Fact

Fact is also called measure, is a measurable metric which can be described by dimensions (UnitSales, OrderQuantity)

Grain

Level of detail a fact-table row represents

Star Schema

- Fact table 和 fact table 不可以相连
- 可以有多个fact table
- The dimension table is denormalized
- The dimension table attribute determine the granularity (grain of the fact, eg. How detailed the facts are)
- All dimension table must be conformed

Conformed Dimension

Conformed dimension is a list of dimensions that provide same meaning to all fact tables. Each conformed dimension table linked to at least 2 fact tables.

Snowflake Schema

Normalize dimension tables in star schema, each new table represents a level of hierarchy in dimension tables, then the diagram becomes snowflake schema

Advantage

- Save space to store
- Easy to maintain

Disadvantage

- Hard to understand by end-users

- Hard to browse
- Slow response time for query, lower the query performance

An Enterprise Data Warehouse

Must have

- Conformed dimensions
- Facts
- A bus architecture

Identify Dimensions From a User Story

- A user story can be represented by a row of fact table
- Answer 7 w questions from a user story:
  - When?
  - Where?
  - Who?
  - Why?
  - How?
  - What?
  - How many?
- Identify dimensions from the 7 answers.

Dimension Types

Slowly Changing Dimension (SCD)

A SCD is dimension with attributes that change it value slowly over time.

Eg. Address in Customer dimension change slowly

SCD Type 1

Overwrite and discard the old value

SCD Type 2

Keep the old value by adding a new row for the change, keep track the effective date of new value.

SCD Type 3

Keep the old value by putting it in a new historical-value column, keep track the effective date of new value.

- But we cannot keep the entire history, only can keep recent several changes.

Rapidly Changing Dimension

Rapidly changing dimension (also called monster dimension) is a dimension with attributes that change it value frequently.

- Cannot use method like SCD type2 to handle it, because that will produce a super-large table

Mini-Dimension

Technique for handling rapidly changing dimension.

Firstly split the rapidly changing attributes to a mini-dimension table and generate surrogate key for the mini-dimension. Then join the mini-dimension table to the fact table by using the mini-dimension surrogate key as FK. The fact table is still joined with the original dimension table using the original FK.

Degenerate Dimension

A dimensional attribute in a fact table, but is not linked to any dimension table. Fact table 里面, 既不是dimension table的FK, 也不是fact的那个attribute.

eg. SaleKey, EmployeeSaleKey

Fact Table Types

Transaction Fact Table

Eg. Sale

- Most common
- Usually has additive facts

Snapshot

Eg. EmployeeSale

- Periodical or accumulating view of business measures
- Usually semi-additive facts

Factless

- Describe event occurrence. Eg. Enrollment fact table
- No facts, only DD and FK to dimension tables

Fact Types

Additive

Facts that can be summed by all dimensions that is linked with its fact table.
Eg. DollarSales in Sale fact table can be summed by a Customer, by a Product, by a Store, by a Date

Semi-Additive

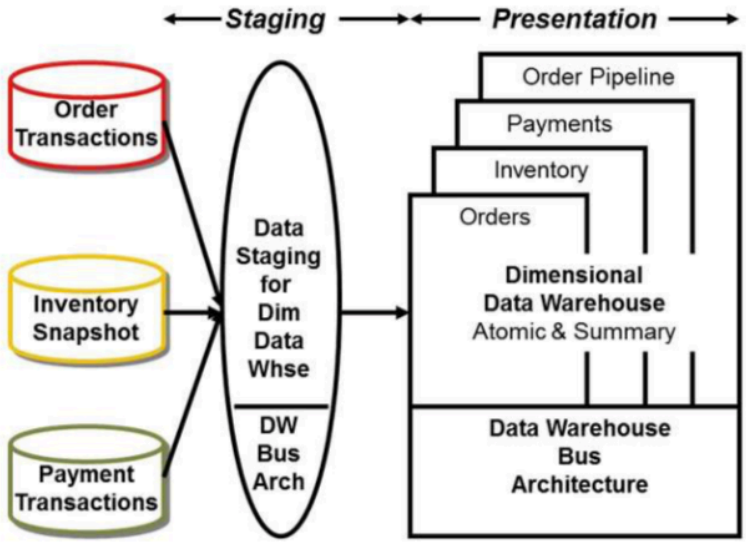
Facts that can only be summed by some of dimensions that is linked with its fact table.
Eg. DailyDollarSales in EmployeeSale table can be summed by a Store, by a Salesperson, but it is meaningless to sum it by a Date.

Non-Additive

Facts that cannot be summed by any dimensions that is linked to its fact table.
Eg. It is meaningless to sum any ratio value.

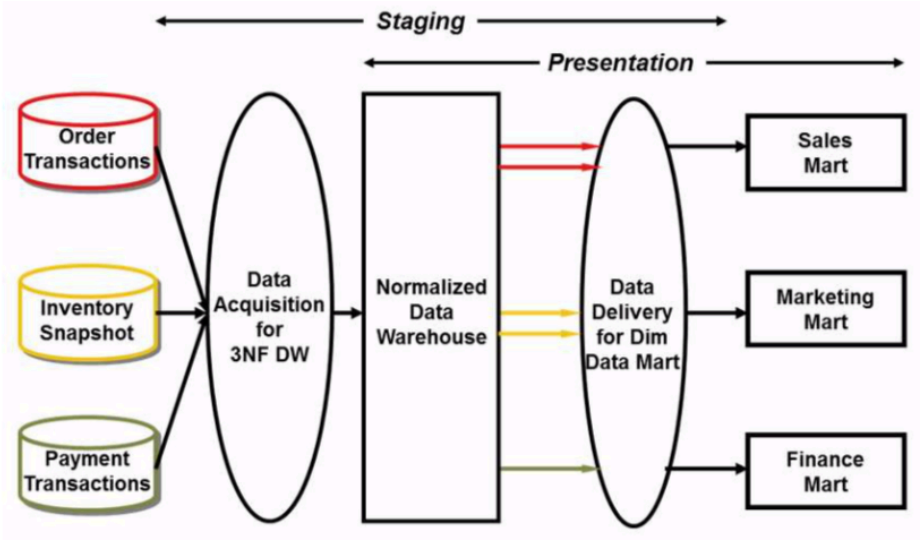
Kimball's Model

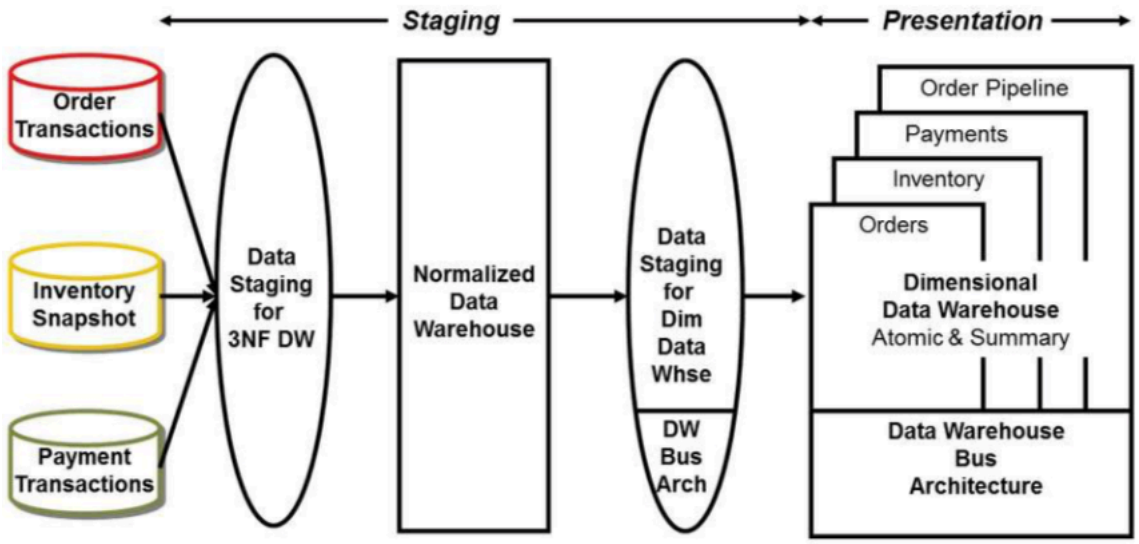
- Bottom-up
- Dimensional modelling (for data marts)
- build denormalized data mart
- Business-driven
- Decentralized bus architecture



Inmon's Model

- Top-down
- ER modelling (for centralized DW)
- First build normalized centralized DW (hub), then denormalized data marts (spoke)
- IT-driven
- Centralized hub-and-spoke architecture





Hybrid Model