Introduction to Multidimensional data modelling

Types of data model

- Conceptual data model
- Logical data model
- Physical data model

Conceptual data model

- In conceptual data model we identify the various entities and the highest level relationship between them as per given requirements.

Conceptual data model

- It identifies most important entities
- It identifies relationship between different entities.
- It does not support the specification of attributes.
- It does not support the specification of primary key.

Logical data model

- The logical data model is used to describe data in as much detail as possible. While describing data, practically no consideration is given to the physical implementation aspect.

Logical data model

- It identifies all entities and relationship among them.
- It identifies all the attribute for the each entity.
- It specifies primary key for each entity.
- It specifies foreign keys
- Normalization of entities is performed at this stage.

Physical Model

- A physical model is a representation of how the model will be built in the database.

Physical Model

- Specification of all tables and columns.
- Foreign keys are used to identify relationships between tables.
- While logical data model is about normalization, physical data model may support de-normalization based on requirements.
- Physical considerations may cause the physical data model to be quite different from logical data model.
- Physical data model will be different for different RDBMS

Overview

- 1st Normal Form
- 2nd Normal Form
- 3rd Normal Form
- Conclusion

Database Normalization

- The main goal of Database Normalization is to restructure the logical data model of a database to:
- Eliminate redundancy
- Organize data efficiently
- Reduce the potential for data anomalies.

Data Anomalies

- Data anomalies are inconsistencies in the data stored in a database as a result of an operation such as update, insertion, and/or deletion.
- Such inconsistencies may arise when have a particular record stored in multiple locations and not all of the copies are updated.

- We can prevent such anomalies by implementing 7 different level of normalization called Normal Forms (NF)
- We'll only look at the first three. ☺

Brief History/Overview

- Database Normalization was first proposed by Edgar F. Codd.
- Codd defined the first three Normal Forms, which we'll look into, of the 7 known Normal Forms.

- In order to do normalization we must know what the requirements are for each of the three Normal Forms that we'll go over.
- One of the key requirements to remember is that Normal Forms are progressive. That is, in order to have 3rd NF we must have 2nd NF and in order to have 2nd NF we must have 1st NF.

1st Normal Form
The Requirements

- The requirements to satisfy the 1st NF:
  - Each table has a primary key: minimal set of attributes which can uniquely identify a record
  - The values in each column of a table are atomic (No multi-value attributes allowed).
  - There are no repeating groups: two columns do not store similar information in the same table.

1st Normal Form Example

Un-normalized Students table:

Student#	AdvID	AdvName	AdvRoom	Class1	Class2
123	123A	James	555	102-8	104-9
124	123B	Smith	467	209-0	102-8

Normalized Students table: 1st NF

Student#	AdvID	AdvName	AdvRoom	Class#
123	123A	James	555	102-8
123	123A	James	555	104-9
124	123B	Smith	467	209-0
124	123B	Smith	467	102-8

2nd Normal Form
The Requirements

- The requirements to satisfy the 2nd NF:
  - All requirements for 1st NF must be met.
  - Redundant data across multiple rows of a table must be moved to a separate table.
    - The resulting tables must be related to each other by use of foreign key.

2nd Normal Form Example

Students table

Student#	AdvID	AdvName	AdvRoom
123	123A	James	555
124	123B	Smith	467

Registration table

Student#	Class#
123	102-8
123	104-9
124	209-0
124	102-8

3rd Normal Form
The Requirements

- The requirements to satisfy the 3rd NF:
  - All requirements for 2nd NF must be met.
  - Eliminate fields that do not depend on the primary key;
    - That is, any field that is dependent not only on the primary key but also on another field must be moved to another table.

3rd Normal Form Example

Students table:

Student#	AdvID
123	123A
124	123B

Registration table:

Student#	Class#
123	102-8
123	104-9
124	209-0
124	102-8

Advisor table:

AdvID	AdvName	AdvRoom
123A	James	555
123B	Smith	467

Conclusion

- We have seen how Database Normalization can decrease redundancy, increase efficiency and reduce anomalies by implementing three of seven different levels of normalization called Normal Forms. The first three NF's are usually sufficient for most small to medium size applications.

FACT TABLE

- Dimensional modeling divides the database into two parts
  - Measurement
  - Context

  Measurement are captured by the various business processes. These measurements are usually numeric values called facts.

  Facts are enclosed by various contexts which are divided into individual logical clumps called dimensions.

FACT TABLE

- A fact table consists of various measurements. The measures are factual or quantitative in representation and are generally numeric in nature.

Types of Fact

- Additive facts
- Semi-Additive facts
- Non additive facts

Additive facts

- These are the facts that can be summed up/ aggregated across all dimension in a fact table.
  - Quantity sold
  - Dollars sold
  - Sales amount
  - revenue

Semi-Additive facts

- These are the facts that can be summed up for some dimensions in the fact table, but not all.
  - Account balances
  - Inventory level

Non additive facts

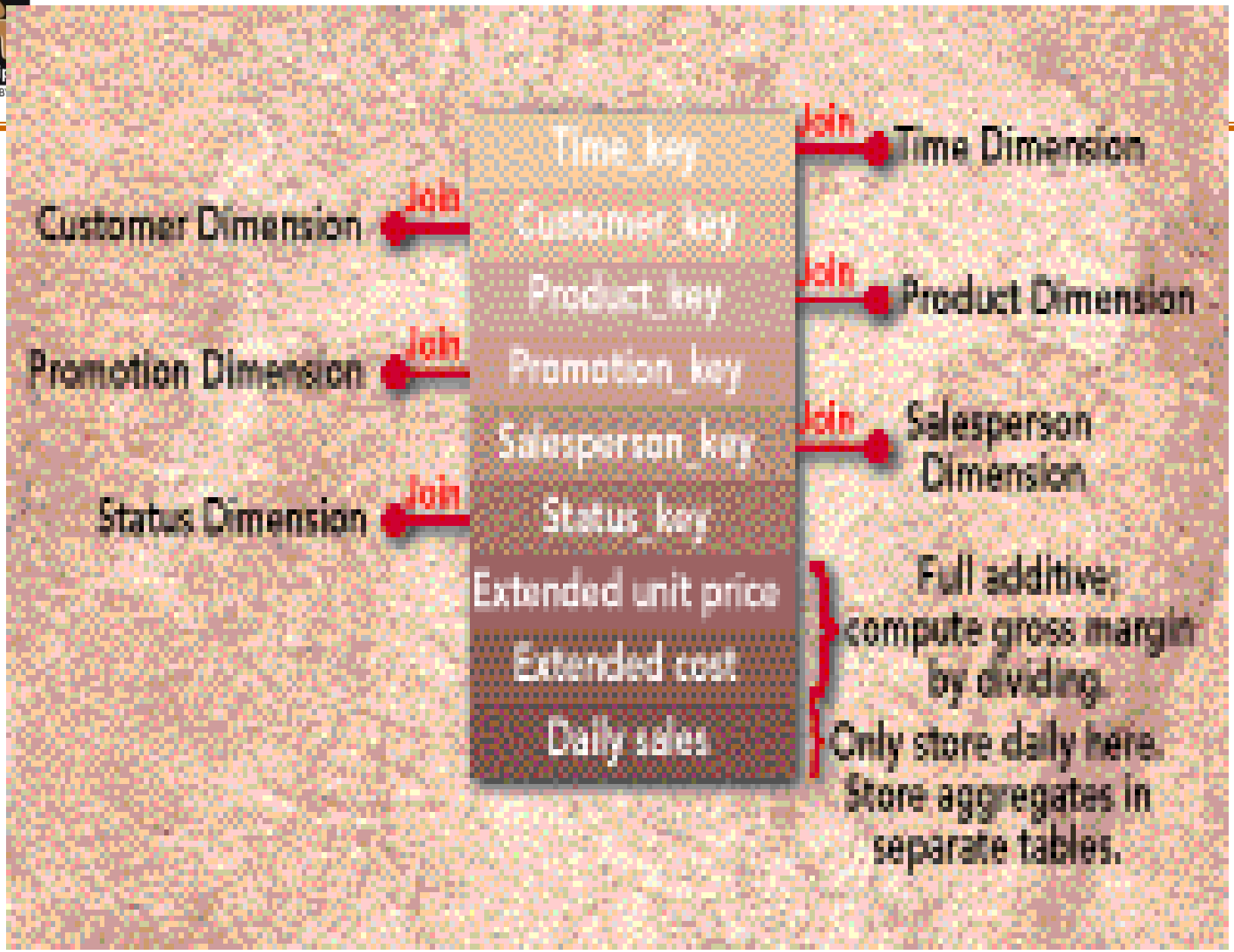
- These are the dimensions that cannot be summed up for any of the dimensions present in the fact table.
  - Room temperature
  - Percentages
  - Ratios
  - Fact less facts

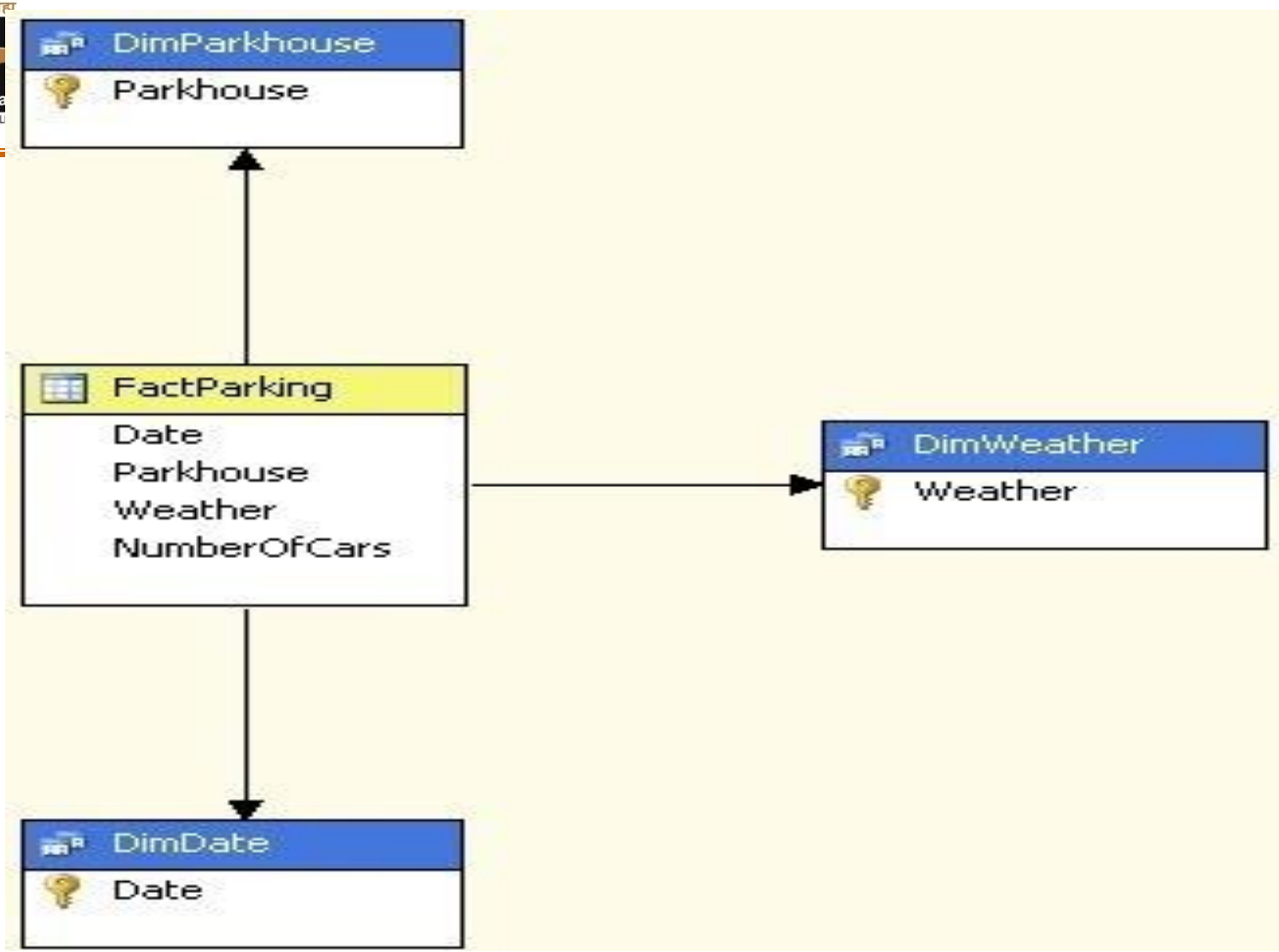
Non additive facts

- Textual facts: you cannot add but you can count.
- Averages:

Fact less facts

- These are also called as event based fact tables. These are the tables which record events.
  - Web page clicks
  - Employee attendence


Customer Dimension
Join
Promotion Dimension
Join
Status Dimension
Join
Time_key
Customer_key
Product_key
Promotion_key
Salesperson_key
Status_key
Extended unit price
Extended cost
Daily sales
Join
Time Dimension
Join
Product Dimension
Join
Salesperson Dimension
Full additive; compute gross margin by dividing.
Only store daily here. Store aggregates in separate tables.


DimParkhouse
Parkhouse
FactParking
Date
Parkhouse
Weather
NumberOfCars
DimWeather
Weather
DimDate
Date

Dimension Table

- Dimension tables consist of dimension attributes which describe the dimension elements to enhance comprehension.

Dimension Table

- Dimension Table attributes must be
  - **Verbose**: Labels must consist of full words
  - **Descriptive** : convey the meaning effectively
  - **Complete**: must not contain missing values
  - **Discrete values**: must contain only one value per row in a dimension table.
  - **Quality assured**: dimension attributes must not contain misspelt or impossible values.

Dimension Table

- Dimension hierarchy: Dimension hierarchy is a cascaded series of many to one relationships and consists of different levels. Each level in a hierarchy corresponds to a dimension attribute.
- Example: daily sales rollup to weekly sales etc.

Types of Dimension tables

- Degenerate dimension
- Slowly changing dimension
- Rapidly changing dimension
- Role playing dimension
- Junk dimension

Degenerate dimension

- It is a dimension without any attributes. Usually it is a transaction based number. There can be more than one degenerate dimension in a fact table.

Degenerate dimension

- Degenerate dimension often cause confusion as they do not feel or look like normal dimensions. They act as dimension keys in fact tables; however they are not joined to corresponding dimensions in other dimension tables as all their attributes are already present in other dimension tables.
  - Transaction number

Slowly changing dimension (SCD)

- In a dimensional model, dimension attributes are not fixed as their value can change slowly over a period of time.
- A SCD is a dimension whose attributes for a record(i.e. row) change slowly over time rather than change on a regular timely basis.

SCD

Sales Rep ID	Sales Rep Name	Sales Territory
1001	Bret Watson	Chicago

SCD

- Type-I (Over Writing the history)

Sales Rep ID	Sales Rep Name	Sales Territory
1001	Bret Watson	Los Angeles

Type-1 Advantages

- Easiest and simplest method to implement
- Effective in those situations requiring the correction of bad data.
- No change is needed to the structure of the dimension table.

Type-1 Disadvantages

- All history may be lost if used inappropriately. It is typically not possible to trace history.
- All previously made aggregated table need to be rebuilt.

SCD

- Type-2 (preserving the history)

Sales Rep ID	Sales Rep Name	Sales Territory
1001	Bret Watson	Chicago
1006	Bret Watson	Los Angeles

Type-2 Advantages

- This approach enables us to accurately keep track of all historical information.

Type-2 Disadvantages

- This approach will cause the size of the table to grow fast.
- Storage and performance can become a serious concern, especially in cases where the number of rows for the table is very high to start with.
- It complicates the ETL process too.

Type-3(Preserving one or more version of history)

SalesRepID	SalesRepName	OriginalSales Territory	Current Sales Territory	Effective from
1001	Bret Watson	Chicago	Los Angeles	01-May-2011

Type-3(Preserving one or more version of history)

- This approach is used when it is compulsory for the data warehouse to track historical changes.

Type-3 Advantages

- Since only old information is updated with new information, this does not increase the size of the table.
- It allows us to keep some part of history.

Type-3 Disadvantages

- Type-3 SCDs will not be able to keep all history where an attribute is changed more than once.

Rapidly Changing Dimension (RCD)

- RCD dimension change frequently in several rows.
  - Age
  - Income
  - Test score
  - Rating
  - Credit history score
  - Customer account status
  - weight

RCD

- One way to handle RCD is to breakoff RCD into one or more separate dimensions known as mini dimensions.
- The fact table would then have two separate foreign keys- one for the primary dimension table and other for the rapidly changing attributes.

Role Playing dimension

- A single dimension that is expressed differently in a fact table with the usage of views is called a role playing dimension.
  - Order date and delivery date
  - FromCity to toCity

Junk Garbage dimension

- The garbage dimension is a dimension that contains low cardinality columns/attributes such as indicators, codes and status flag.
- The attributes in a garbage dimension are not associated with any hierarchy.
- Go for garbage dimension if cardinality of each attribute is relatively low.

Typical Dimensional models

- Star Schema
- Snowflake schema
- Galaxy or fact constellation schema

Star Schema

- It consists of large central table called fact table with no redundancy. The fact table is in 3NF or higher form of normalization. All the dimensions are usually in a de-normalized manner and are present in 2NF or lower. The dimension tables are also known as lookup or reference tables.

Snow flake Schema

- In the Snow flake schema dimensions are present in a normalized form in a multiple related tables.
  - Easy to maintain and save storage space
  - More joins are needed to execute the query thus reducing the performance.

Galaxy Schema

- It consists of multiple fact tables (multiple stars). This allows dimension tables to be shared among various fact tables.
  - Enterprise wide data warehouse
  - Not used in case of data marts.

References

- David Loshin, “Business Intelligence”, Morgan Kaufmann Publishers, 2003
- Mike Biere, “Business Intelligence for the Enterprise”, 2nd edition, IBM Press,2003.
- R N Prasad, Seema Acharya, “Fundamentals of Business Analytics”, Wiley India, 2011