Welcome! Module 4

- Data Warehouse – Business Intelligence Concepts

"A collection of integrated, subject-oriented databases designed to support the DSS function where each unit of data is relevant to some moment in time..."

Inmon, Imhoff and Sousa, The Corporate Information Factory

"A copy of transaction data specifically structured for query and analysis."

Ralph Kimball, The Data Warehouse Toolkit

Introduction

About Me

- **Parwaz Dalvi**
  Sr. Architect / Consultant DW-BI & Database

  TOGAF 8 Certified (The Open Group Architecture Framework)

My Session For you

- **Data Warehouse Concepts**

Session's Objective

- Understand what Data Warehousing means
- Realize the Need, Advantages & Challenges in implementation of a DW Solution
- Understand Data Warehouse Architecture and its components
- Understand IBM Reference DW-BI Architecture
- Understand IBM's IOD initiative and realize how DW-BI helps in achieving this objective
- Know the DW-BI Tools and Products, the trends in DW-BI
- Know your Growth Prospects in the DW-BI Arena within IBM

Course Content

Module	Content	Duration
1	Data Warehouse Evolution	
2	Data Warehouse Concepts	
3	Data Warehouse Architecture – Part 1 – GENERIC	
4	***Data Modeling in DW-BI***	
5	Data Warehouse Architecture – Part 2 – SPECIFIC	
6	DW-BI - IBM Reference Architecture & IOD	
7	DW-BI Tools and Products	
8	Trends in DW-BI	
9	Growth Path of DW-BI Professionals	

Course Title:

Module 4 : Data Modeling In DW-BI

Module Objectives

- At the completion of this chapter you should be able to understand:
  - Data Model – Definition
  - Data Modeling - Definition
  - ER – Relational Model – Normalization Process
  - Dimensional Model – Logical to Dimensional Process
  - Star Schema
  - Snowflake Schema
  - Multidimensional Schema
  - Slowly Changing Dimension – Type 1, Type 2, & Type 3
  - Type of Facts – Additive, Semi Additive & Non Additive

Module 4: Data Modeling In DW-BI : Agenda

- Topic 1. Types of Data Model in DW-BI
- Topic 2. ER – Relational Model
- Topic 3. Dimensional Model

Module 4: > Topic 1: Types of Data Model in DW-BI

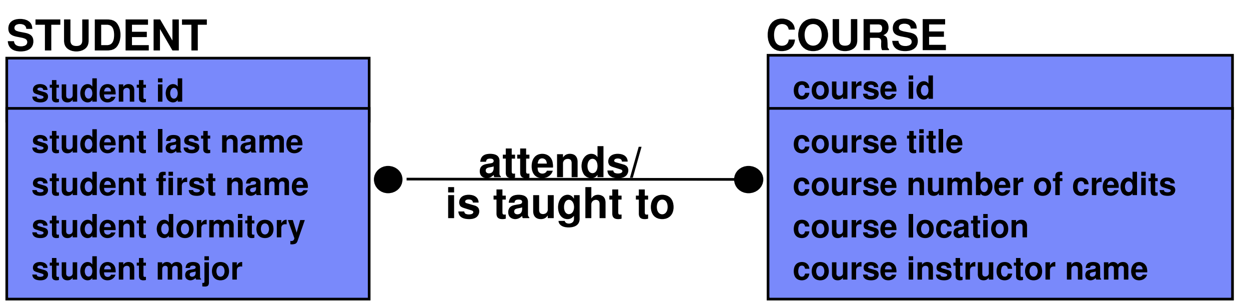
- Data Model – Definition
- Data Modeling - Definition
- Conceptual Data Model
- Logical Data Model
- Physical Data Model

Module 4: > Topic 1: Types of Data Model in DW-BI

Data Model - Definition

DATA MODEL

*The specification of **data structures** and **business rules** to represent business requirements*

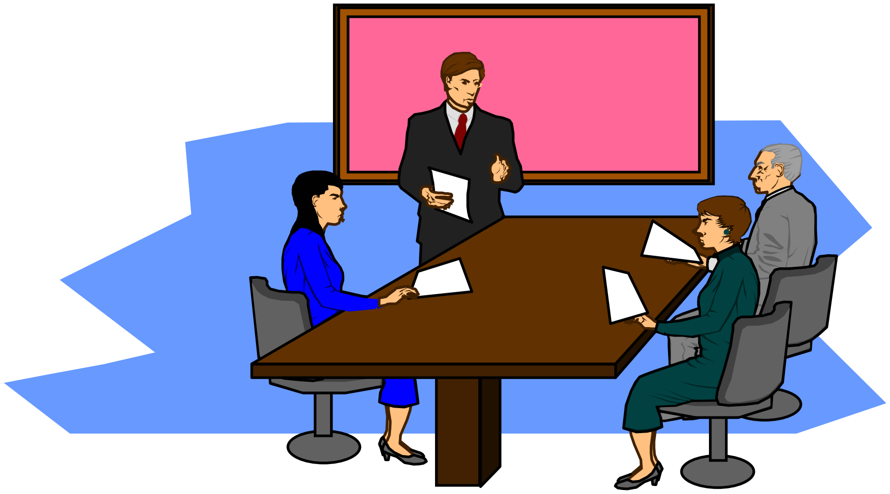


Module 4: > Topic 1: Types of Data Model in DW-BI

Data Modeling - Definition

DATA MODELING

A structured approach used to identify major components of an information system's specifications.



Module 4: > Topic 1: Types of Data Model in DW-BI

- ***Data Modeling** is the process used to:*
  - *Analyze the data*
  - *Identify the relationships*
  - *Create the data model*

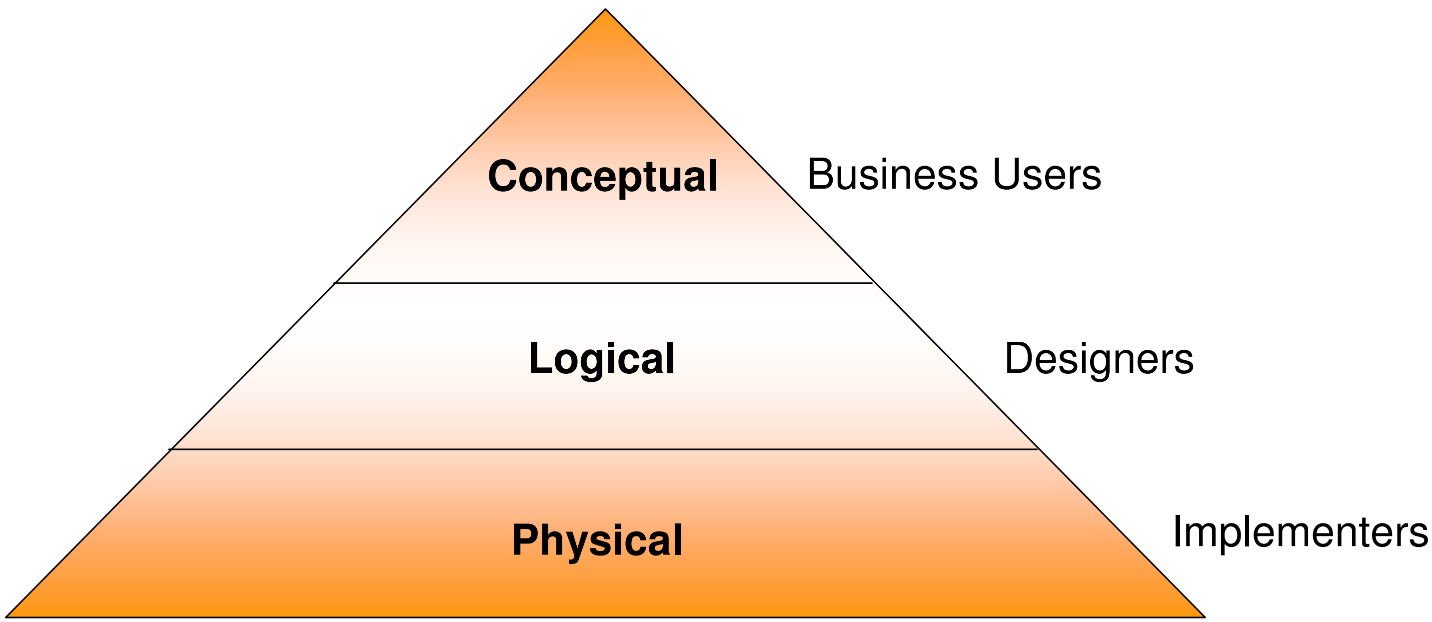
- *We use it to visualize informational needs in the form of an **E**ntity **R**elationship **D**iagram (ERD)*

- *Business models often coexist with Data Models to enable software development.*

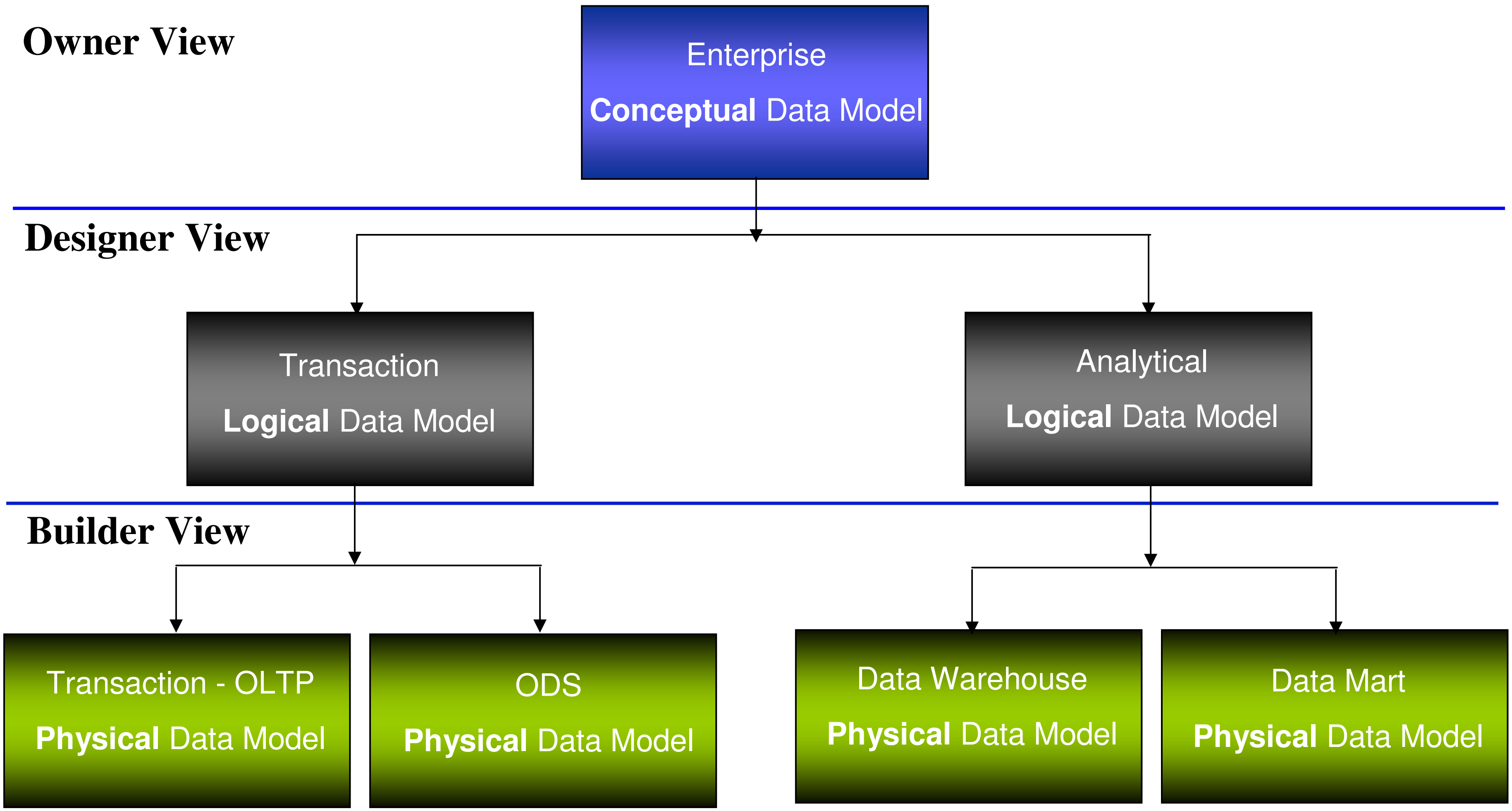
- *Other types of modeling:*
  - *Business Process Model*
  - *Functional Model*
  - *Object Model*

Module 4: > Topic 1: Types of Data Model in DW-BI

Viewers of Data Model



Module 4: > Topic 1: Types of Data Model in DW-BI



Module 4: > Topic 1: Types of Data Model in DW-BI

Conceptual Data Model

- *A **Conceptual Data Model** is a structured business view of the data required to support current business processes, business events, and related performance measures*

- *It is a **single integrated data structure** which reflects the structure of business functions rather than the processing flow or the physical arrangement of data*

- *Characteristics*
  - *Represents overall logical structure of data*
  - *Independent of software or data storage structure*
  - *Often contains objects not implemented in physical databases*
  - *Represents data needed to run an enterprise or a business activity*

Module 4: > Topic 1: Types of Data Model in DW-BI

Logical Data Model

- *A Logical Data Model builds upon the business requirements and includes a further level of detail that supports both the business and system requirements*

- *Business rules are incorporated into the LDM and it loses some of the 'generalities' from the Enterprise CDM*

- *Characteristics*
  - *Independent of specific software and data storage structure*
  - *Includes more specific entities and attributes*
  - *Includes business rules and relationships*
  - *Includes foreign keys, alternate keys, and inversion entries*

Module 4: > Topic 1: Types of Data Model in DW-BI

Physical Data Model

- *A Physical Data Model is specific to the software and performance constraints of the specific database management system to be used in the implementation*

- *Both software and data storage structures are considered and the model is often modified to meet performance or physical constraints*

- *Characteristics*
  - *Dependent specific software and data storage structure*
  - *Includes tables and columns*
  - *Includes physical database objects (triggers, stored procedures, tablespaces)*
  - *Includes referential integrity rules that restrict relationships between tables*

Module 4: > Topic 1: Types of Data Model in DW-BI

Logical versus Physical Data Model

- *Entity becomes Table*
- *Attribute becomes Column*
- *Alternate Key/Inversion Entry becomes Index*
- *Some objects are logical only; some are physical only*

EMPLOYEE

employee id

employee first name
employee last name
employee address
employee phone number

EMPLOYEE

employee_id: number(10)

Employee_first_name: varchar2(30)
Employee_last_name: varchar2(30)
Employee_address: varchar2(100)
Employee_phone_number: number

Module 4: > Topic 1: Types of Data Model in DW-BI Summary

- Having completed this topic, you should be able to:
  - Data Model Definition
  - Conceptual Data Model
  - Logical Data Model
  - Physical Data Model

Module 4: > Topic 1: Types of Data Model in DW-BI Review

Module 4: > Topic 2: ER – Relational Model

- ER – Relational Data Model – Definition
- Entity, Attribute & Relationship
- Primary Key, Foreign Key, Referential Integrity
- Normalization Process

Module 4: > Topic 2: ER – Relational Model

ER – Relational Model

- *It describes data as Entities, Attributes & Relationships*
- *ENTITY - Is a thing in the real world with independent existence*
  - *Entity can be an object with physical existence or conceptual existence. Example: Person, Job*

ENTITY

A person, place, thing, event, or concept about which the business keeps data

  - *Each ENTITY represents a set/collection of like individual objects called instances.*
  - *Each instance must be uniquely identifiable and distinct from all other instances*
  - *Each ENTITY should be named uniquely. Example -* **ORDER**

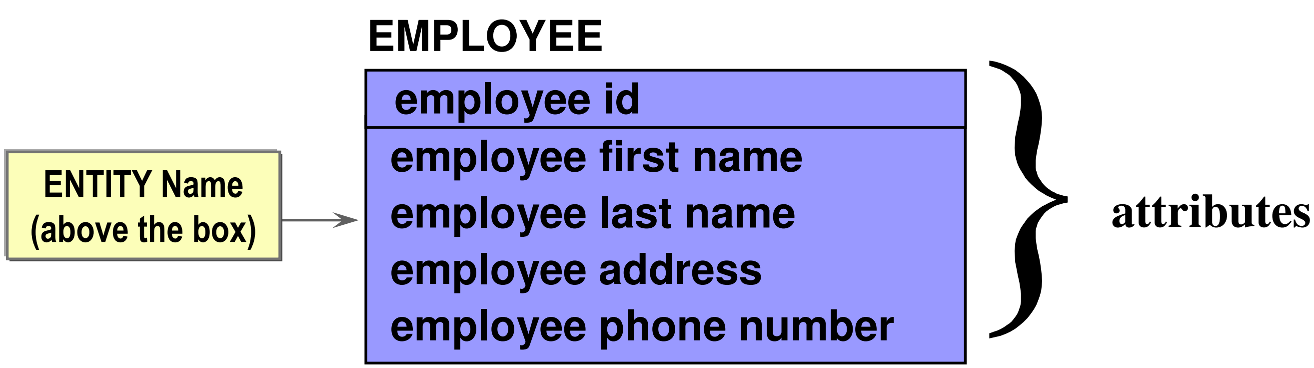
Module 4: > Topic 2: ER – Relational Model

ER – Relational Model

- *Attribute - Is a characteristic of an Entity describing it*

ATTRIBUTE

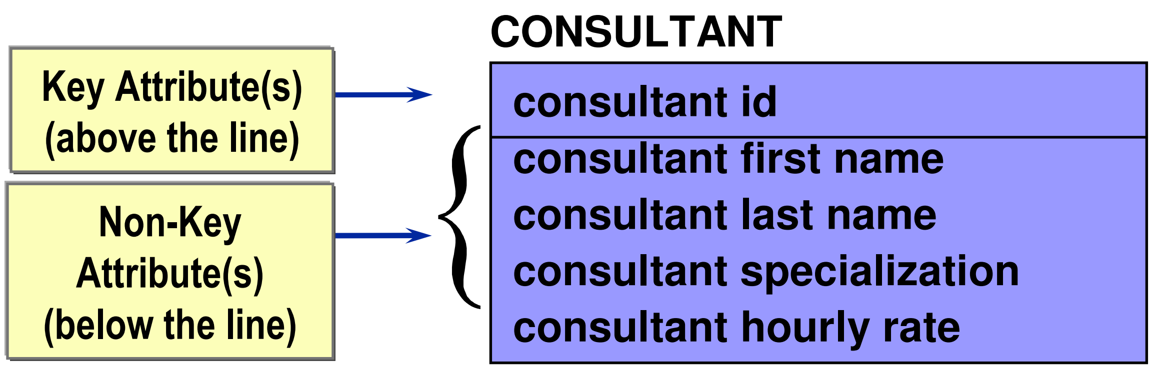
A distinct characteristic of an ENTITY for which data is maintained



Module 4: > Topic 2: ER – Relational Model

ER – Relational Model

- *Each Attribute must be represented with a unique business oriented name*
- *There are two types of attributes*
  - *Key*
  - *Non Key*



Module 4: > Topic 2: ER – Relational Model

ER – Relational Model

- *Primary Key - Is an unique identifier for an Entity*

PRIMARY KEY

An ATTRIBUTE or group of attributes that uniquely identifies an instance of the entity

- The primary key is always placed above the line in an Entity

BOOK

book isbn
author first name **author last name** **book title** **book edition** **book publisher** **book year published** **book lc catalog number**

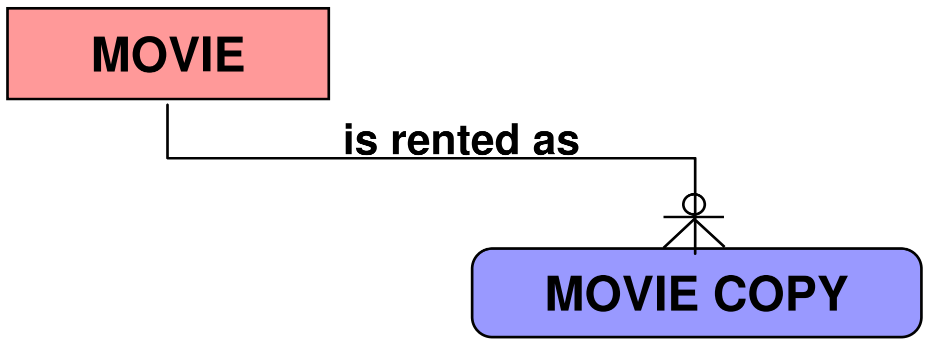
Module 4: > Topic 2: ER – Relational Model

ER – Relational Model

- *Relationship -*

RELATIONSHIP

A logical link between two entities that represents a business rule or constraint



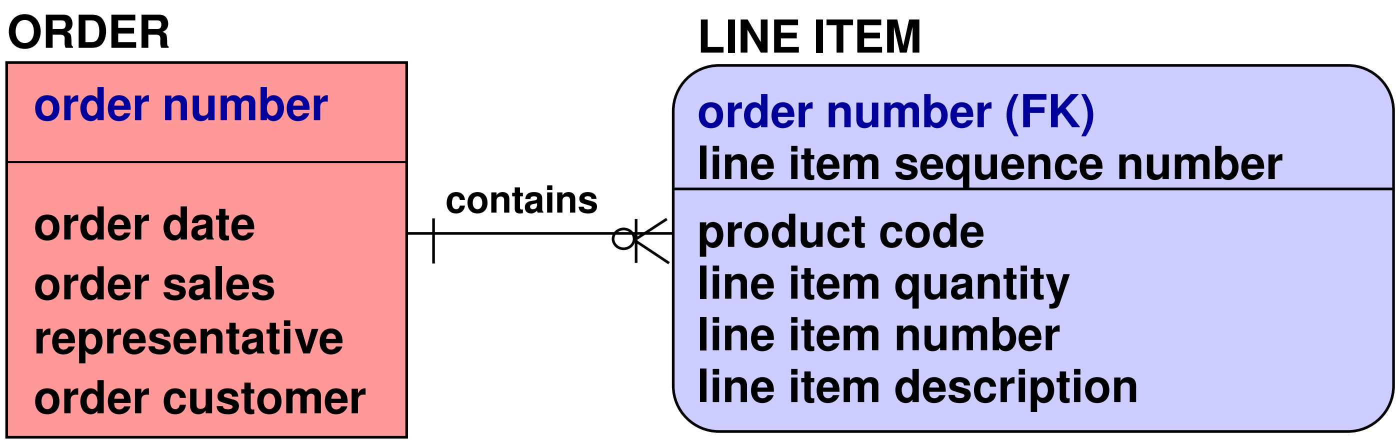
Module 4: > Topic 2: ER – Relational Model

ER – Relational Model

- *Foreign Key -*

FOREIGN KEY (FK)

A primary key of a parent entity that is contributed to a child entity across a relationship



Module 4: > Topic 2: ER – Relational Model

ER – Relational Model

- *Referential Integrity -*

REFERENTIAL INTEGRITY (RI)

Rules that determine what happens when a Parent or Child instance is inserted, updated or deleted

RI Option	What it does
RESTRICT	Does not allow Action
CASCADE	Duplicates Action in related tables
SET NULL	Allows Action and sets the Child foreign key to Null
SET DEFAULT	Allows Action & sets the Child foreign key to the Default value
NULL	No Restriction placed on Action

Module 4: > Topic 2: ER – Relational Model

ER – Relational Model

- *Normalization - Is a concept used in the relational model for storage of data in a database*

NORMALIZATION

A formal data modeling approach to examining and validating the model

- *Pros*
  - *Ensures that each attribute belongs to the entity to which it is assigned*
  - *Redundant storage of information is minimized*
- *Cons*
  - *Can adversely affect performance if rigorously implemented*
  - *Can adversely affect deadlines if rigorously implemented*

Module 4: > Topic 2: ER – Relational Model

ER – Relational Model

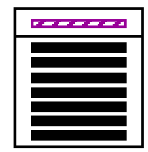
- *Rules of Normalization -*

- Dr. E. F. Codd identified ‘normal forms’ as the different states of a ‘normalized relational’ data model.
  - 1NF = No repeating groups
  - 2NF = No partial key dependencies
  - 3NF = No non-key interdependencies
  - 4NF = No independent multiple relationships
  - 5NF = No semantically related multiple relationships

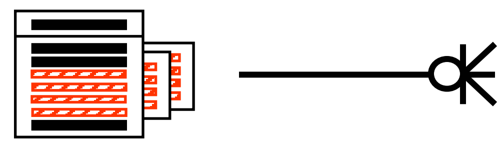
Module 4: > Topic 2: ER – Relational Model

ER – Relational Model – Normalization Process

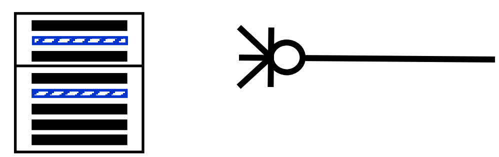
- Step 1. Before you normalize an entity, identify its Primary Key



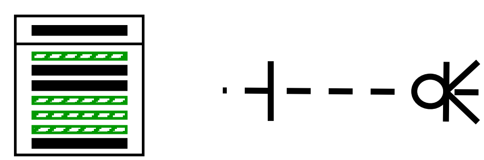
- Step 2. Identify and resolve violations of 1NF
  - make sure there are no repeating groups



- Step 3. Identify and resolve violations of 2NF
  - make sure that each non-key attribute depends on the entire key



- Step 4. Identify and resolve violations of 3NF
  - make sure that no non-key attribute depends on another non-key attribute



Module 4: > Topic 2: ER – Relational Model

ER – Relational Model – Normalization Example - Step 1

PUPPY

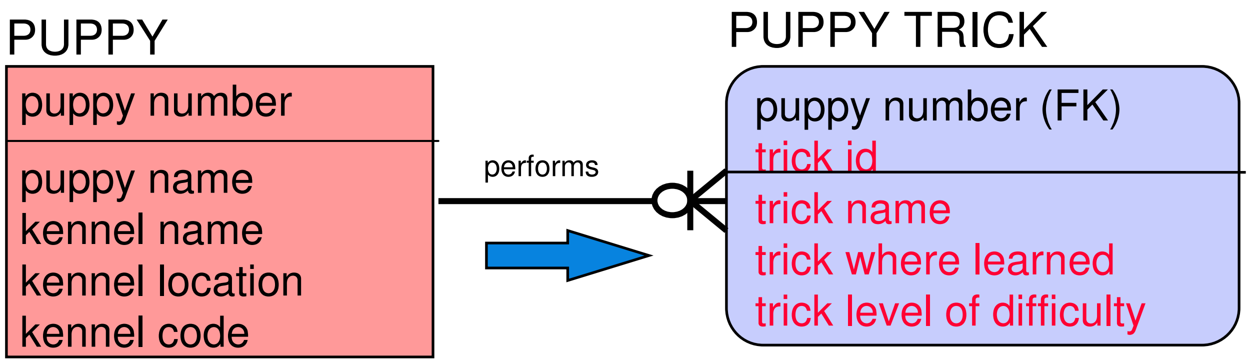
puppy number puppy name kennel name kennel location kennel code trick id 1 trick name 1 trick where learned 1 trick level of difficulty 1 trick id 2 trick name 2 trick where learned 2 trick level of difficulty 2

PUPPY

puppy number
puppy name kennel name kennel location kennel code trick id 1 trick name 1 trick where learned 1 trick level of difficulty 1 trick id 2 trick name 2 trick where learned 2 trick level of difficulty 2

Module 4: > Topic 2: ER – Relational Model

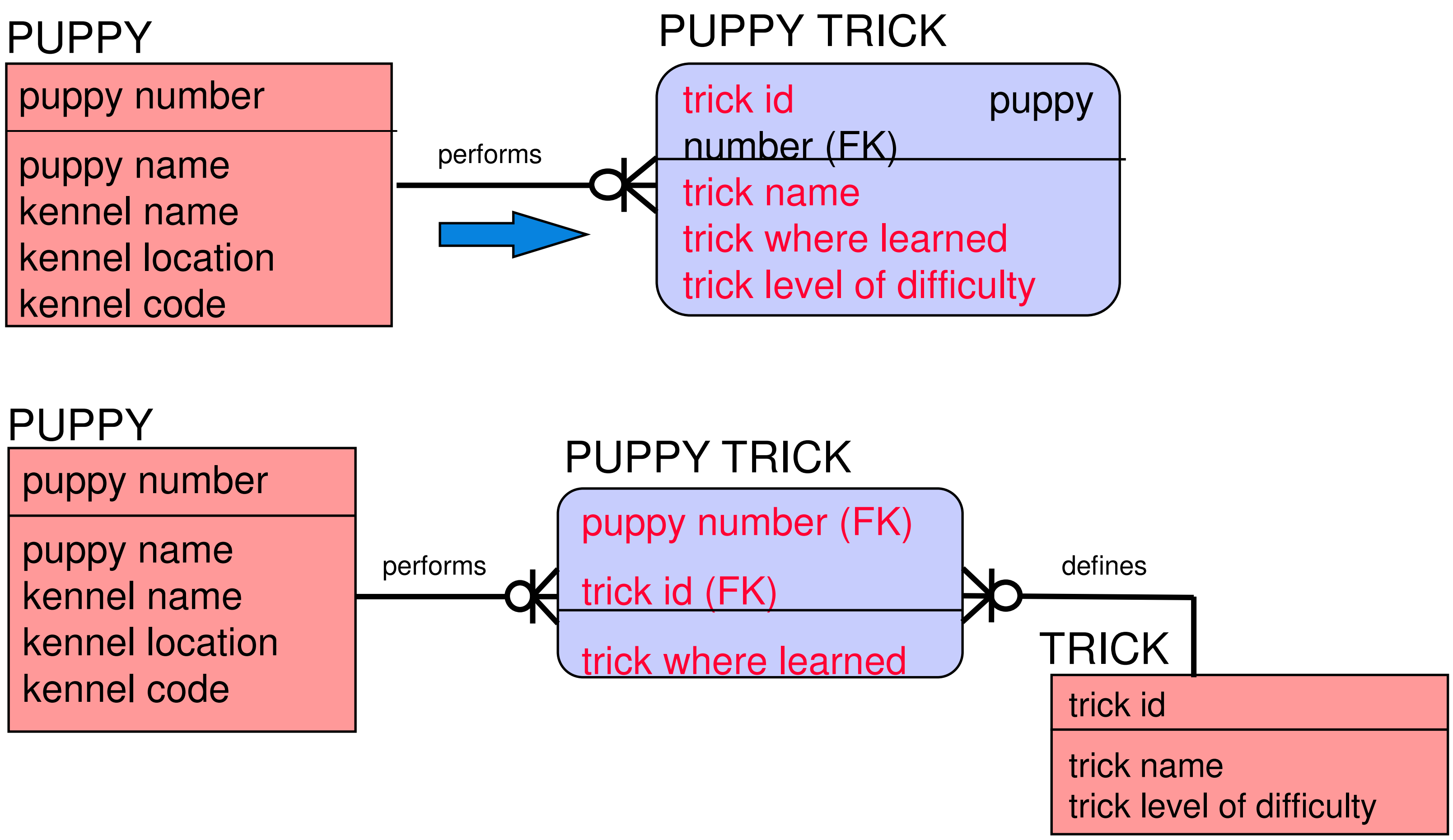
ER – Relational Model – Normalization Example - Step 2 – 1NF



The new entity should be named to reflect its intention, given a primary key, and the inherited foreign key will be present as part of the primary key.

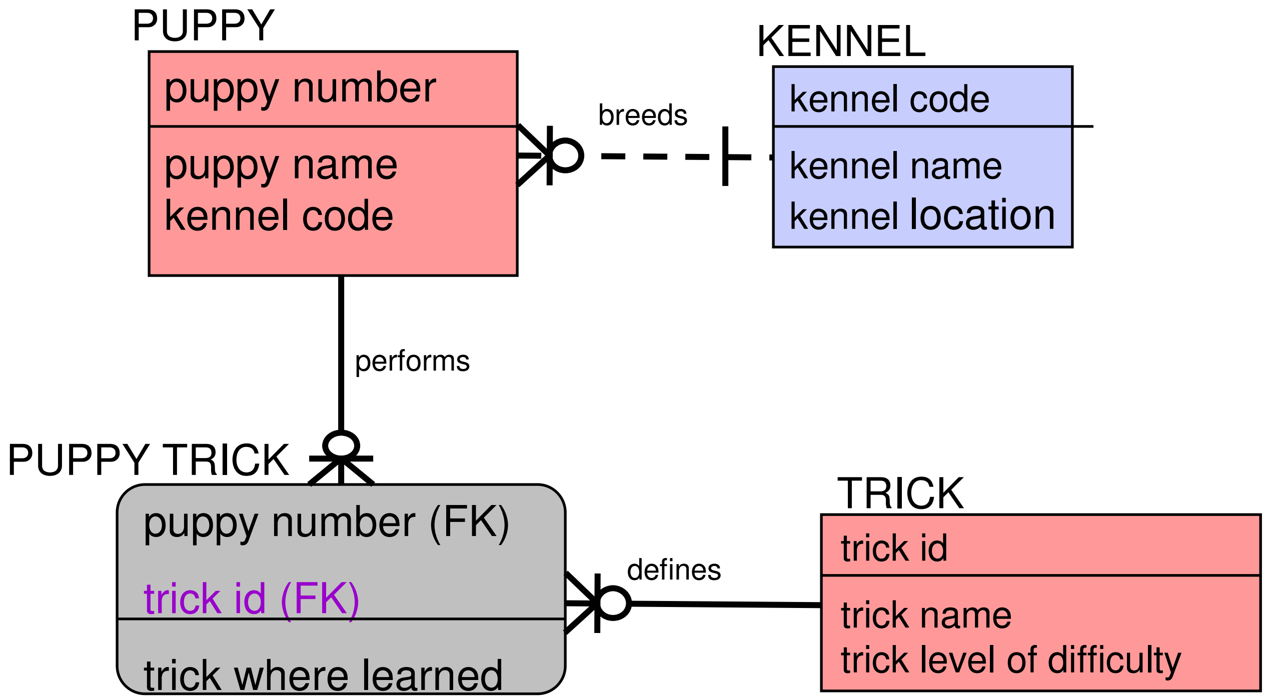
Module 4: > Topic 2: ER – Relational Model

ER – Relational Model – Normalization Example - Step 3 – 2NF



Module 4: > Topic 2: ER – Relational Model

ER – Relational Model – Normalization Example - Step 4 – 3NF



Separate the non-key attributes that depend on other non-key attributes (i.e., kennel location) into their own entity

Module 4: > Topic 2: ER – Relational Model

ER – Relational Model – Normalization – Results

- Data is easier to define
- Data interdependencies are identified
- Data ambiguities are resolved
- Data model can be more flexible
- Data model is easier to maintain

- Performance can be an issue
- The structure can be very complex
- Data Model is not business friendly

Module 4: > Topic 2: ER – Relational Model Summary

- Having completed this topic, you should be able to:
  - ER Modeling
  - Entities, Attributes & Relationships
  - Primary Key, & Foreign Key, Referential Integrity
  - Process of Normalization - 1NF, 2NF, & 3NF Normal Forms

Module 4: > Topic 2: ER – Relational Model Review

Module 4: > Topic 3: Dimensional Model

- Dimensional Model – Definition
- Dimension, Facts, Dimensional Model Terminology
- Dimensional Model – Process
- Star Schema
- Snow Flake Schema
- Multidimensional Schema
- Slowly Changing Dimensions – Type 1, Type 2, Type 3
- Type of Facts – Additive, Semi Additive, Non-Additive

Module 4: > Topic 3: Dimensional Model

Dimensional Model

- *Developed Bottom Up*
- *Depicts a business process through its relevant facts and dimensions*
- *Groups data into categories of business measure and characteristics*
- *More suitable for analytical applications where the focus is querying large sets of data*
- *The fundamental idea of dimensional modeling is that nearly every type of business data can be represented as a kind of cube of data, where the cells of the cube contain measured values and the edges of the cube define the natural dimensions of the data*

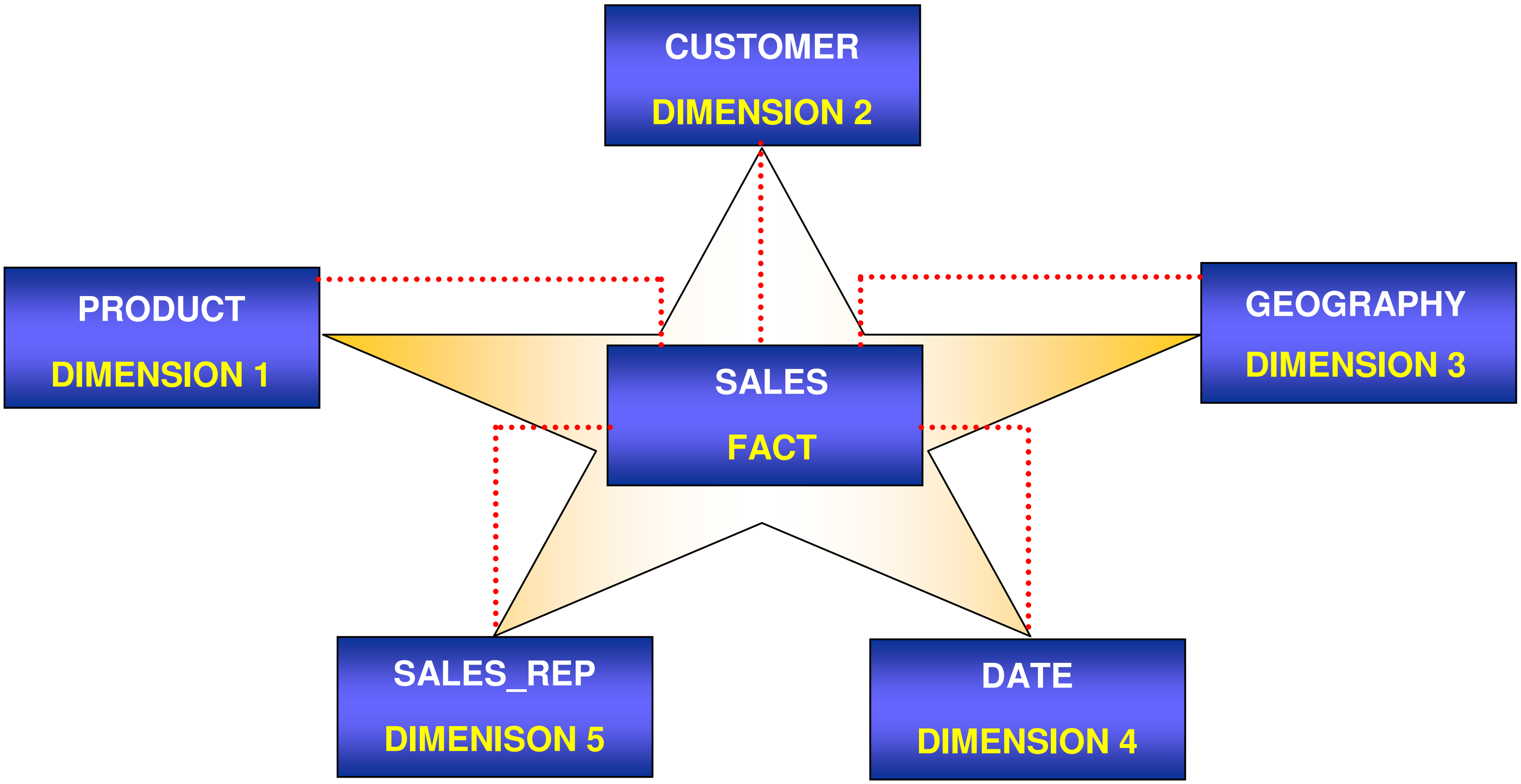
Module 4: > Topic 3: Dimensional Model

Dimensional Model

- *Every dimensional model is composed of one table with multipart key, called the fact table, and a set of smaller tables called dimension tables*
- *Each dimension table has a single part primary key that corresponds exactly to one of the components of the multipart key in the fact table*
- *The dimension tables contain textual attributes that describe the facts*
- *The fact table contains facts or measurements of the business*
- *The attributes in the dimension tables are used for constraining & grouping data within data warehouse queries*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Star Schema



Single FACT (measure) table surrounded by multiple DIMENSION (descriptive) tables like a 'STAR' structure

Module 4: > Topic 3: Dimensional Model

Dimensional Model – DIMENSION

DIMENSION

An entity by which the business views the measures (facts) & by which we try to model data in the real world

- *They are the entry points into the Data Warehouse*
- *They are groupings of similar data into larger category or subject area*
- *They are designed especially for selection & grouping under a common head*
- *They determine the contextual background for FACTs*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – DIMENSION – Example

PRODUCT

PRODUCT_KEY
PRODUCT_ID
PRODUCT_NAME
PRODUCT_DESC
PRODUCT_CATEGORY
PRODUCT_BRAND
BRAND_DESC
CLASS
SIZE

- *Dimensions provide descriptive information*
- *They can represent hierarchical relationships*
- *Common Dimensions are –*
  - *Date*
  - *Product*
  - *Location / Region*
  - *Customer*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – ATTRIBUTE

ATTRIBUTE

A distinct characteristic of an DIMENSION for which data is maintained

- *Fields within a dimension that describe the item associated with a dimension*
- *Acts as a source of query constraints*
- *Dimensions are made up of attributes*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – FACT

FACT

A measurement with reference to a criteria, generally additive in nature of the organization

- *Facts are typically described as the performance measures of the business*
- *Usually numerical and represent counts, dollar amounts, percentages or ratios*
- *Examples are sales, revenue, expenses, policies, and claims*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – FACT – Example

SALES FACT

DATE_KEY
PRODUCT_KEY
CUSTOMER_KEY
PROMOTION_KEY
LOCATION_KEY
SALES_QUANTITY
SALES $ AMOUNT
PROFIT

- *FACTs are where the numerical measurements of business are stored*
- *It consists of Foreign keys to Dimensional tables*
- *They are analyzed across Dimensions*
- *They serve as the Key Performance Indicators (KPI) of the business*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Characteristics

- *The Structure is organized across subject areas*
- *The Structure is integrated*
- *The structure has a standard set of keys, naming conventions, and field formats*
- *Connection paths are clearly defined*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Characteristics

- *The Structure is organized across subject areas*
- *The Structure is integrated*
- *The structure has a standard set of keys, naming conventions, and field formats*
- *Since OLAP is focused on query performance, it has characteristics of a Dimensional model*
  - Increased redundancy (de-normalization)
  - Increased index use
  - Increased storage space
  - Consolidation of inconsistent data
  - Increased maintenance issues (history)

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Terminology

- ***Grain*** *- Level of detail in fact and dimension tables*
- ***Hierarchy*** *- Represents the organizational structure of like data groupings. Implies levels and rollup*
- ***Fact*** *- Performance measure of the business*
- ***Dimension*** *- Descriptions, or views, of facts*
- ***Attribute*** *- Descriptions of dimensions or the data elements on the dimension tables*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Terminology

- ***Atomic Layer*** *- Dimensions and facts at the lowest level of detail (think ODS)*
- ***Summary Layer*** *- Dimensions and facts aggregated to intermediate values*
- ***Presentation Layer*** *- Dimensions, facts, and other tables altered specifically for presentation tool limitations*
- ***Reporting Layer*** *- Dimensions, facts, and other tables created or altered to improve reporting capabilities and performance*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Process

Step 1. Identify the Business Process

Step 2. Identify the grain of each Fact Table

Step 3. Identify the Dimension & its Attributes

Step 4. Identify the Dimensional Hierarchies

Step 5. Choose the Dimensions that apply to each Fact Table

Step 6. Identify the Measured Facts & Pre-Calculated Facts

Step 7. Determine Slowly Changing Dimensions

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Process – Identify the Business process

- A major operational process that is supported by some kind of legacy system(s) from which data can be collected for the purpose of the data warehouse
  - Example: orders, invoices, shipments, inventory, sales

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Process – Identify the Grain of each FACT Table

- Granularity defines the level of detailed data
- It must be determined prior to going forward in the modeling process
- Typical grains are individual transactions, time-based aggregation, and/or aggregations along a commonly used dimension

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Process – Identify the DIMENSIONS & its ATTRIBUTES

- For example, what should our time dimension look like? Should it have just ‘January for month’, or also ‘Jan’ and ‘1’?

- Should we store the code and the description, just the code, or just the description?

- What values will our users need to filter or report on?

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Process – Identify the DIMENSION Hierarchies

- A dimension such as time may have days rolling into months and then quarters, as well as days rolling into weeks which may cross months and quarters
- Sales geography may differ from physical geography
- Zip codes can cross city boundaries and cities are made up of multiple zip codes

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Process – Choose DIMENSIONS for each FACT Table

- Typical dimensions include time, product, policyholder, agent, and geography

- Remember to evaluate granularity when applying dimensions to facts

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Process – Identify Measured & Pre-Calculated FACTS

- Each aggregated and derived fact will need to be evaluated for inclusion in the model or calculation in the application.

- Trade-offs include storage and indexing and must be weighed against the access requirements

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Process – Determine Slowly Changing DIMENSIONS

- These are the dimensions that change over time
- If tracking these changes is important, the method must be decided
- Options: overwrite the existing record, store all records with effective dates, or a historical and current value tables

Module 4: > Topic 3: Dimensional Model

Dimensional Model – SCHEMAS

- *Dimensional Data Model falls into 3 types of schemas –*
  - *STAR Schema*
  - *SNOWFLAKE Schema*
  - *Multi-Dimensional Schema*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – SCHEMAS

- *Dimensional Data Model falls into 3 types of schemas –*
  - *STAR Schema*
  - *SNOFLAKE Schema*
  - *Multi-Dimensional Schema*

- *Several factors influence schema choice namely –*
  - *Presentation restrictions*
  - *Inconsistency of data*
  - *Complex queries and analysis*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – STAR Schema

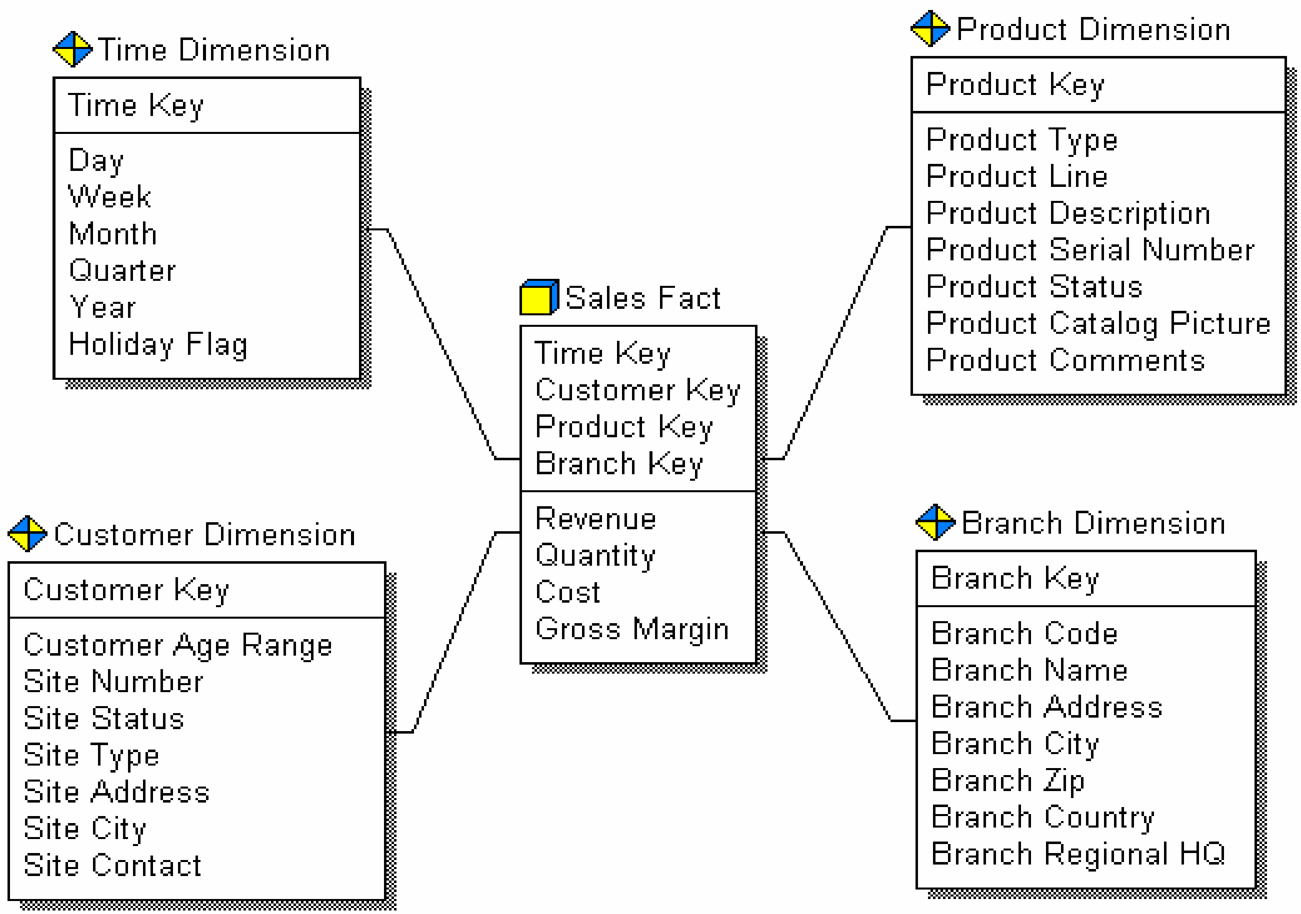
STAR SCHEMA

A database design that stores a central fact table surrounded by multiple dimension tables

- *Star schema represents a compromise between the fully normalized model and the de- normalized model*
- *Descriptive 'dimension' information is maintained in a set of de-normalized dimension tables*

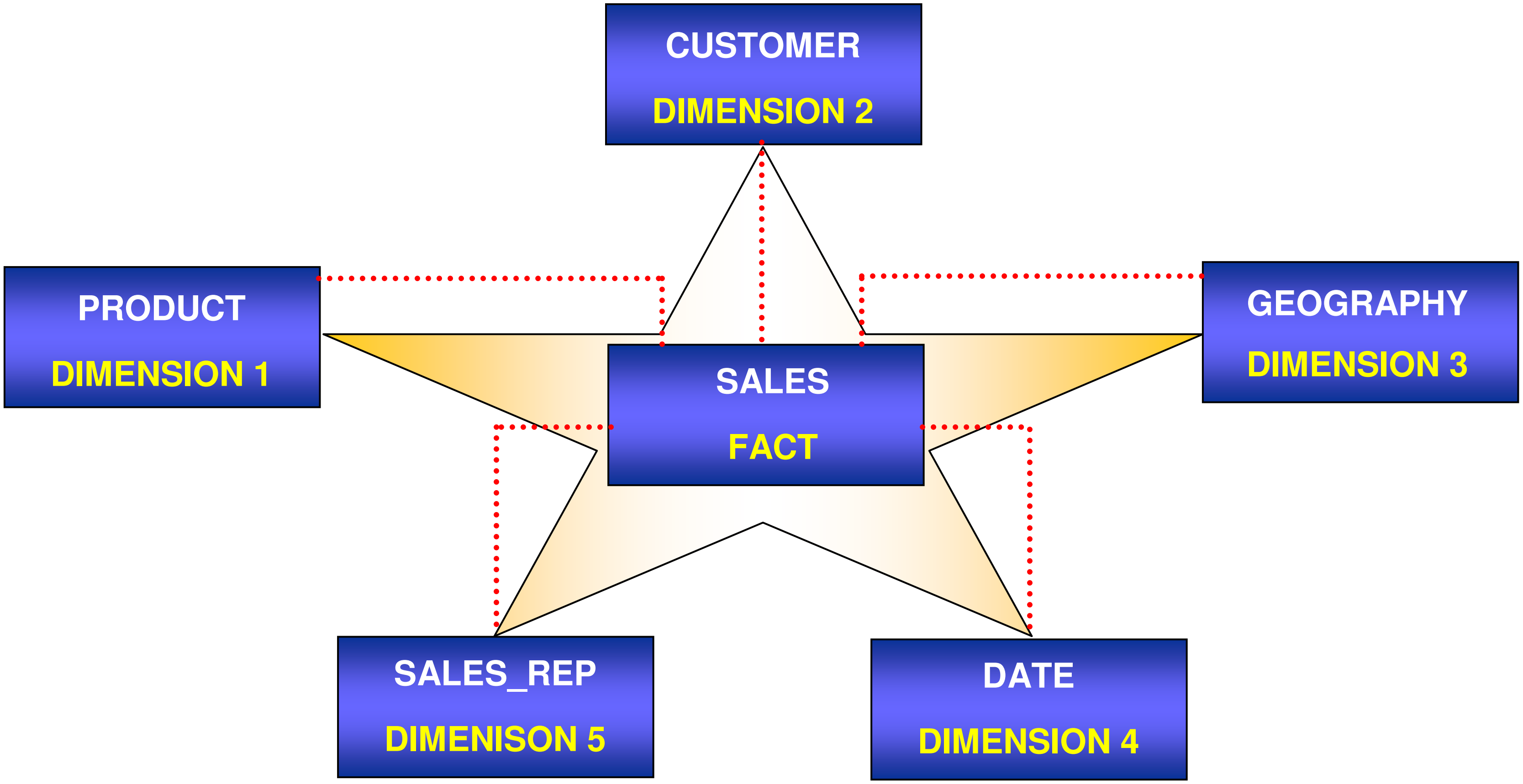
Module 4: > Topic 3: Dimensional Model

Dimensional Model – STAR Schema - Example



Module 4: > Topic 3: Dimensional Model

Dimensional Model – STAR Schema - Example



Module 4: > Topic 3: Dimensional Model

Dimensional Model – SNOWFLAKE Schema

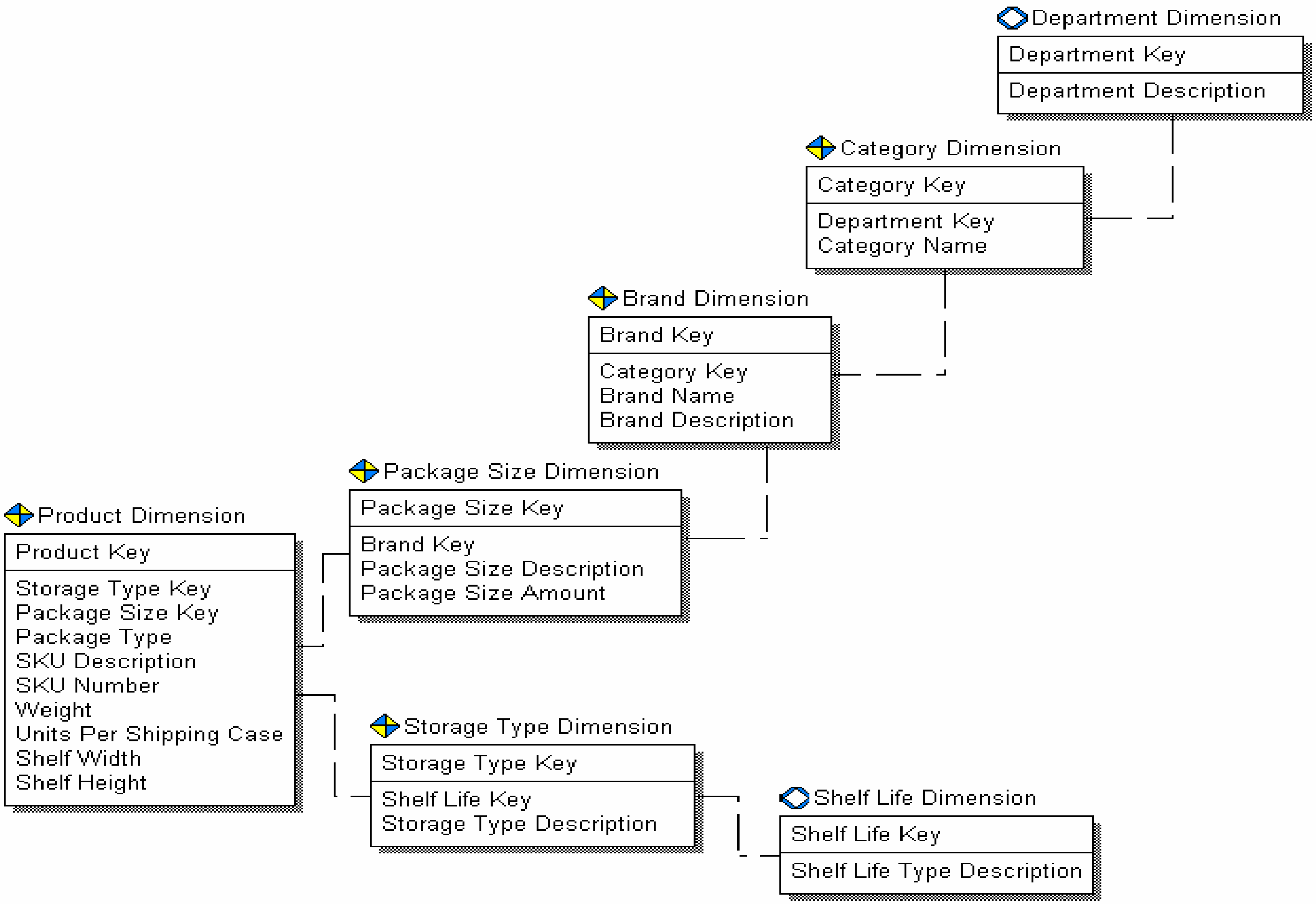
SNOWFLAKE SCHEMA

A database design that stores a central fact table surrounded by multiple dimension tables decomposed or normalized into one or more hierarchies

- *Snowflake schemas are most often used when dealing with large hierarchies that are static.*
- *Snowflake tables (look-up tables) may increase the speed of queries depending on the presentation tool (i.e. Micro Strategy)*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – SNOWFLAKE Schema - Example



Module 4: > Topic 3: Dimensional Model

Dimensional Model – STAR vs. SNOWFLAKE Schema

- *Star Schemas:*
  - *Easier to navigate*
  - *Improved performance*
  - *Supports 'browsing' dimensions*
  - *More widely used*
  - *Storage saved with snowflake becomes negligible*

- *Snowflake Schemas:*
  - *Hierarchical structure can be visualized*
  - *Easier for data modelers and DBA's to use because it is closer to normalized model*
  - *Some tools don't support snowflakes*
  - *Destroys 'browsing' speed and flexibility*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – MULTIDIMENSIONAL Schema

MULTIDIMENSIONAL SCHEMA

Hierarchical databases that consists of only one structure - a multi-dimensional array - that contains all the summarized data at higher levels in the array

- *Also known as MOLAP databases*
- *Stores and aggregates data at multiple levels in a hierarchy.*
- *Utilizes drill-up and drill-down to move around the hierarchy*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – SURROGATE KEY

SURROGATE KEY

A contrived, non-intelligent, single-attribute key used to replace a long composite key

CASH MACHINE TRANSACTION

account id
customer id
cash machine id
transaction date

Composite Key

CASH MACHINE TRANSACTION

transaction id
account id
customer id
cash machine id
transaction date

Surrogate Key

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Moving from Logical to Dimensional Model

- *Moving from a logical relational model to a dimensional model requires following the dimensional steps*
- *How do we take a relational table and 'dimensionalize' it?*
- *What issues are we concerned with when modeling new 'dimensions' from our existing logical model?*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Moving from Logical to Dimensional Model

- *Remember 'physical only' attributes:*
  - *Audit Columns*
  - *Code Table Foreign Keys*

- *Some attributes will need to be added to dimensions to handle cases that would not occur in an OLTP model.*
  - *Attributes denoting an 'event' in time*
  - *Demographical attributes that allow users to categorize customers by age, location, preferences*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Moving from Logical to Dimensional Model

- *If we are tracking history, each new record must have a new primary key in order to preserve referential integrity*

- *Remember 'surrogate' keys:*
  - *A contrived, non-intelligent, single-attribute key used to replace a long composite key or an id attribute if there might be multiple history records for the same id*

- *Customer Key might be added to the Customer dimension to capture this information*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Moving from Logical to Dimensional Model

- *Indicators/Flags are frequently used to identify special events*
- *Events can also be captured in their own dimension table*
- *Holidays may influence our sales or accident rates and some are not consistent dates*
- *'Holiday Flag', 'Store Event', and 'Promotion' are attributes that might be added to the Time dimension to capture this information*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Moving from Logical to Dimensional Model

- *Marketing analysts may need to browse data by age range (i.e. 21-30) for groups of customers*
- *Other examples include income, marital status, level of education*
- *Demographic attributes may be added to the Customer dimension to capture this information*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Moving from Logical to Dimensional Model

- *Issues arise around how to handle data that changes over time*
- *For example, price records for a product may change based on promotions or seasonal factors*
- *Effective dates and active row indicators may be added to a product or other slowly changing dimension to capture values changing over time*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Types of Dimensions -

- *Slowly Changing Dimensions - Type 1, 2 or 3*
- *Rapidly Changing or Volatile Dimensions*
- *Huge Dimensions and Mini-Dimensions*
- *Causal Dimensions*
- *Dirty Dimensions*
- *Degenerate Dimensions*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Slowly Changing Dimensions

- *Most dimensions change over time*
  - *Products change offered coverage or limits and deductibles.*
  - *Employees are promoted, fired, or change departments.*
  - *Customers change names and addresses*

- *What are our choices for tracking these changes over time?*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Slowly Changing Dimensions (SCD)

- *There are three types of slowly changing dimensions:*
  - ***Type 1****: Overwrites the old data for a record with new data. This eliminates the ability to track history over time*
  - ***Type 2****: Creates a new record with the new data at the type of the change. Accurately tracks history, but requires generalized key*
  - ***Type 3****: Tracks new and original values in separate fields at time of change. Intermediate values are lost*

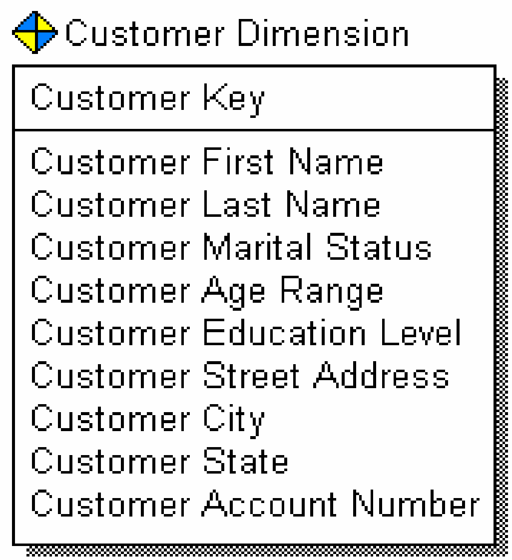
Module 4: > Topic 3: Dimensional Model

Dimensional Model – Type 1 SCD – Overwrite Old Values

- *Customer Lynnette Groves is changing her name to ?*

- *If there is no value in tracking this change, we will overwrite the First Name and Last Name fields with the new values*

- *'UPDATE' statement; 1 record is maintained*



Module 4: > Topic 3: Dimensional Model

Dimensional Model – Type 2 SCD – Create New Record

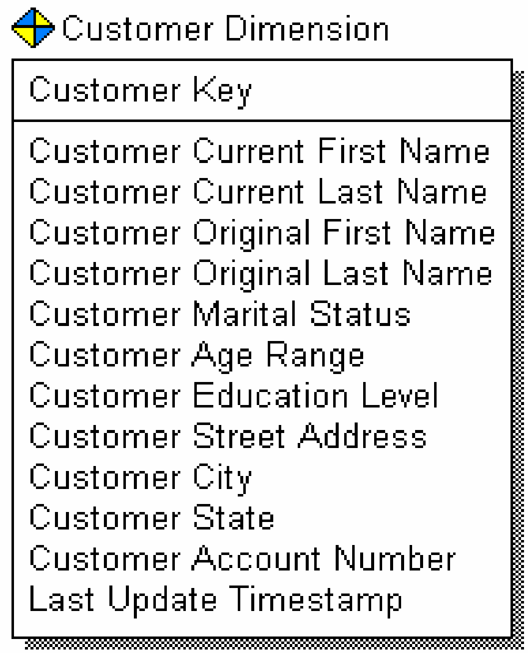
- *Lynnette Groves is changing her name and we want to track both values*
- *Add a second record with a new Customer Key and make it the active row*
- *'INSERT' statement for new, 'UPDATE' for active; 2 records are maintained*
- *New record for each change up to n records*

Customer Dimension
Customer Key
Customer First Name Customer Last Name Customer Marital Status Customer Age Range Customer Education Level Customer Street Address Customer City Customer State Customer Account Number Last Update Timestamp Active Row Indicator

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Type 3 SCD – Original & Current

- *We decide that no matter how many times she changes her name, we only want to track the current and the previous*
- *Before any changes, original and current are the same. Any name change updates 'current' fields*
- *UPDATE' statement; 1 record is maintained*



Module 4: > Topic 3: Dimensional Model

Dimensional Model – Types of FACTS

- *Understanding which facts can be added across which dimensions is an important data design issue*

- *Three Types of Facts:*
  - *Additive*
  - *Non-Additive*
  - *Semi-Additive*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Additive FACTS

ADDITIVE FACTS

Measurements in a fact table that can be added across all dimensions

- *Since aggregation is a key element in the usefulness of the dimensional model, its best utilized for facts that are additive, numeric values*

- *We can add revenue, cost, and quantity sold for all products, all stores, and any time period*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Semi-Additive FACTS

SEMI-ADDITIVE FACTS

Measurements in a fact table that can be added across some dimensions but not others

- *We cannot add risk exposure at the coverage level to get the number of policy level exposures*
- *We can add coverage level exposures across the customer dimension to determine exposure by gender or age range*

Module 4: > Topic 3: Dimensional Model

Dimensional Model – Non-Additive FACTS

NON-ADDITIVE FACTS

Measurements in a fact table that cannot be added across any dimensions, like ratios

- A new value will need to be calculated at each level for each level or for each set of data.
- It should be determined, at what levels, if any, the fact should be stored. Some values may need to be pre-calculated

Module 4: > Topic 3: Dimensional Model Summary

- Having completed this topic, you should be able to:
  - Dimensional Model
  - Dimension, Attribute, Fact, Surrogate Key
  - Process of Dimensionilzation
  - Star Schema, Snowflake Schema, Multidimensional Schema
  - Slowly Changing Dimensions – Type 1, Type 2, & Type 3
  - Type of Facts – Additive, Semi Additive, & Non Additive

Module 4: > Topic 3: Dimensional Model Review

References

- DM Review … www.dmreview.com
- Wikipedia … http://en.wikipedia.org
- Bill Inmon … www.billinmon.com
- Ralph Kimball … www.ralphkimball.com
- TDWI – The Data Warehousing Institute … www.tdwi.org