Data Modeling for Business Intelligence

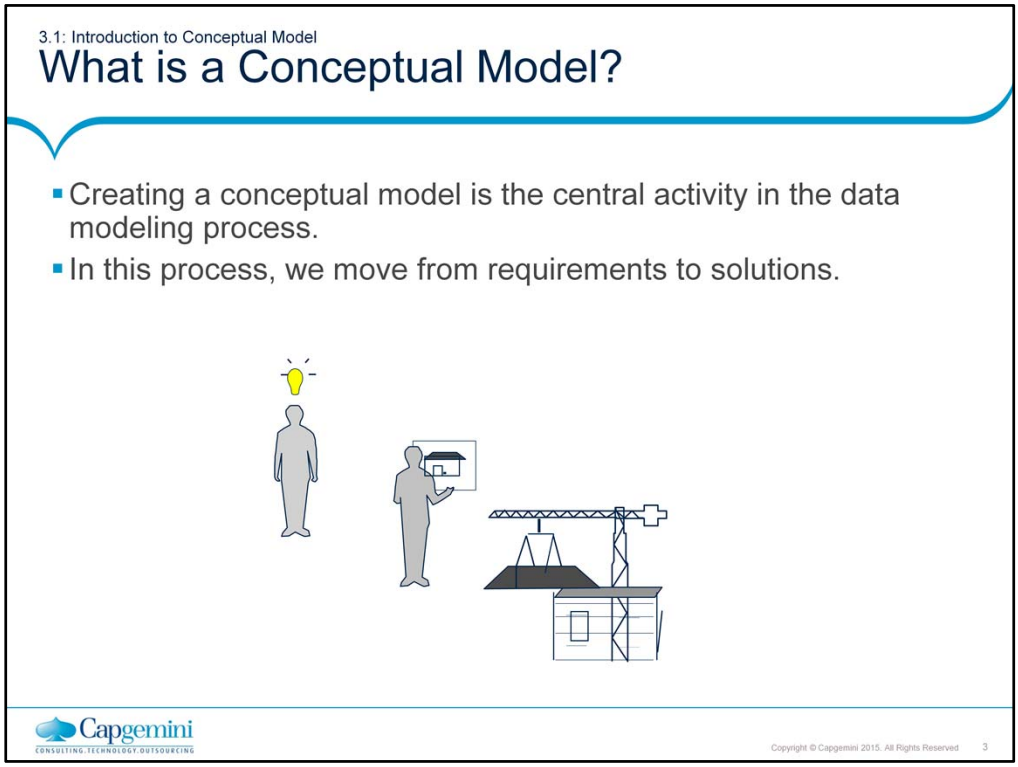
Lesson 3: Conceptual Model

Lesson Objectives

- On completion of this lesson, you will be able to:
  - Define conceptual model
  - State objectives of conceptual model and list its components
  - List and describe main stages in conceptual modeling
  - Describe Online Transaction Processing System
  - State advantages of using generic model
  - Describe the components of a generic model
  - Identify steps of dimension modeling

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 2



Conceptual data model includes all major entities and relationships and does not contain much detailed level of information about attributes and is often used in the INITIAL PLANNING PHASE. Conceptual data model is created by gathering business requirements from various sources like business documents, discussion with functional teams, business analysts, smart management experts and end users who do the reporting on the database.

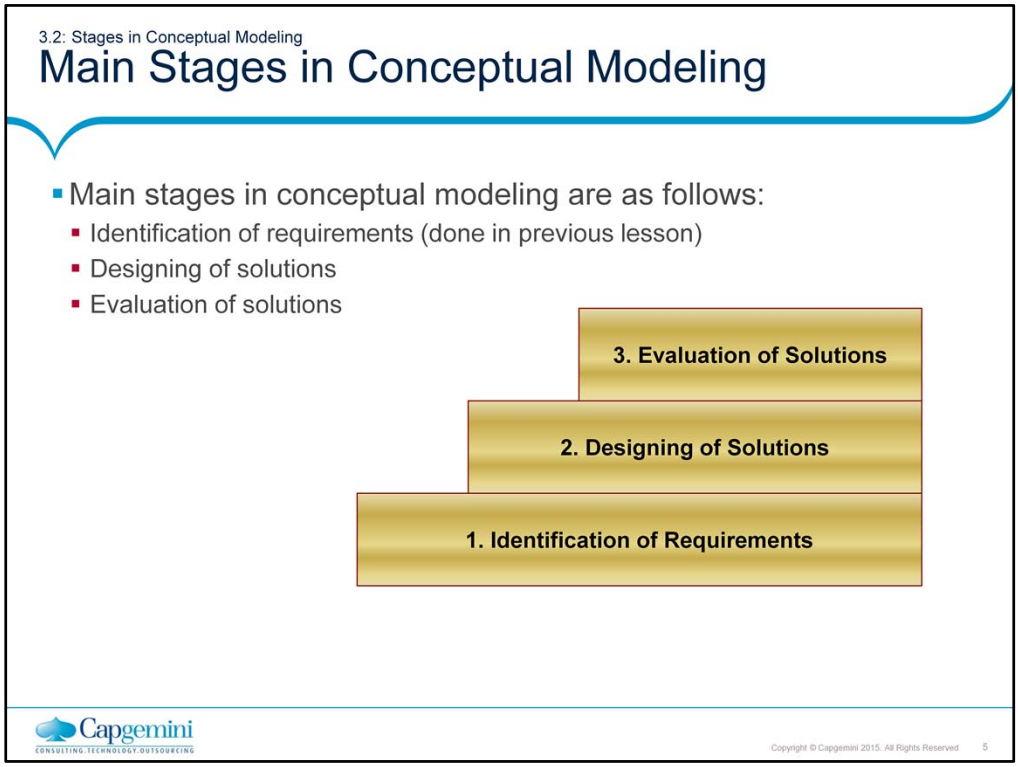
3.1: Goals of Conceptual Model

Objectives of a Conceptual Model

- All pieces of information that are required to run a business are properly recognized.
- Every single piece of required information is displayed only once in the model.
- The main consideration is, in the future system, the information should be available in a predictable and logical place.
- Related information is kept together.
- A proper Entity-Relationship (ER) model leads to a set of logically coherent tables.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

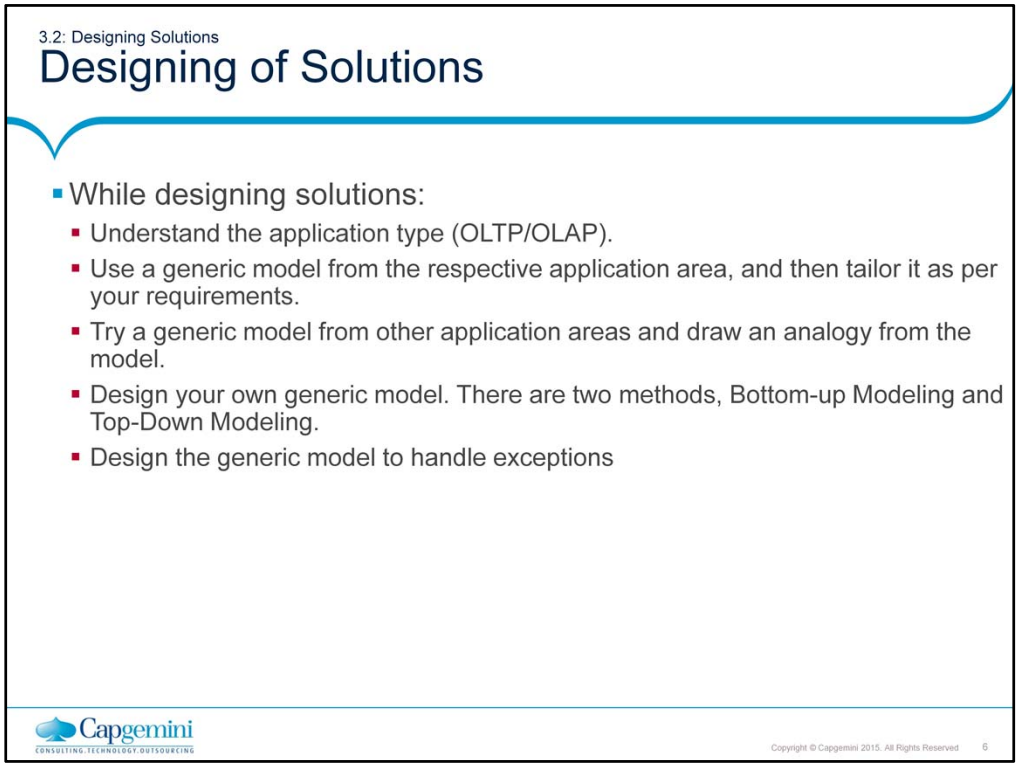
 4



Main Stages in Conceptual Modeling:

Conceptual modeling has various stages, starting from identification of requirements to design of solutions and finally to their evaluation. All these stages provide basic inputs to conceptual modeling process, which gets fine-tuned in the later stages of modeling.

Refer to lesson 2 for Identification of Requirements.



Designing of Solutions:

While designing, usually try to find a generic model that broadly meets the users' requirements, and then tailor it to suit a particular application, drawing on standard structures and adapting structures from other models as opportunities arise.

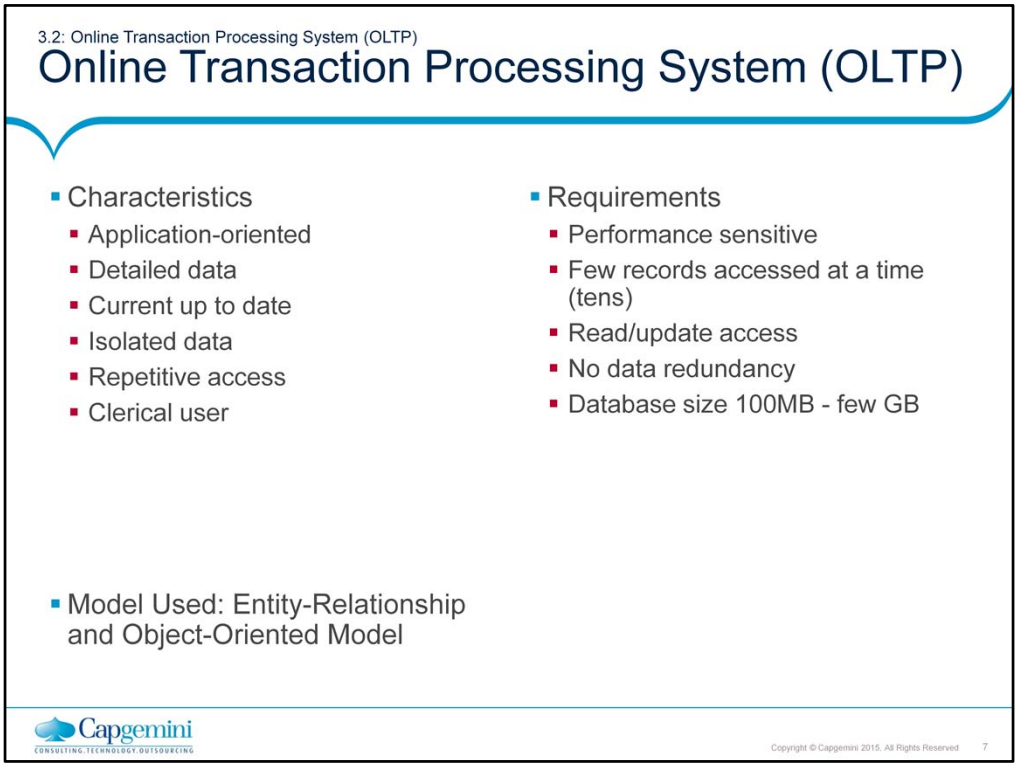
Sometimes, you may not have an explicit generic model available, however, you can draw an analogy with a model from a different field. Try using Life Insurance model for Health Insurance System.

There are two methods for designing a generic model, Bottom-up Modeling and Top-Down Modeling.

The Bottom-up approach: You initially develop a very "literal" model, based on existing data structures and terminology. Then, you use subtyping and super typing to move towards other options. You need not be creative; however, the model should be improvable over a period of time.

The Top-Down approach: We simply use a model that is generic enough to cover at least the main entity classes in any business or organization.

While designing model, try to make is flexible. Add necessary structures required to handle it. Try to optimize the common situations to handle it.


3.2: Online Transaction Processing System (OLTP)
Online Transaction Processing System (OLTP)
Characteristics
Application-oriented
Detailed data
Current up to date
Isolated data
Repetitive access
Clerical user
Requirements
Performance sensitive
Few records accessed at a time (tens)
Read/update access
No data redundancy
Database size 100MB - few GB
Model Used: Entity-Relationship and Object-Oriented Model
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
7

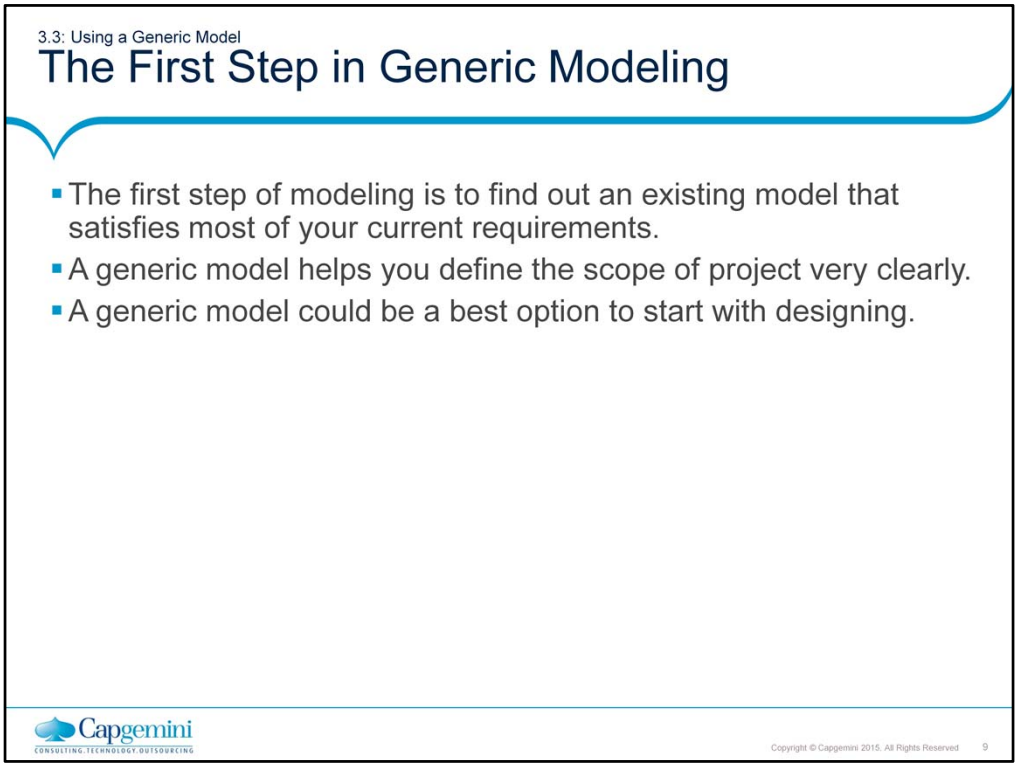
3.3: Patterns and Generic Models

Advantages of Patterns and Generic Models

- No one likes to start the modeling from scratch.
- Modeler very much relies on proven/used structures.
- The advantages of using an existing model or a generic model are as follows:
  - It helps us to understand the system better.
  - It saves a lot of development time and efforts.
  - For example, use of life insurance model could be very useful for designing a model for health insurance.
  - It is known to the modeler.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 8



For example, we may need to develop a data model to support human resource management. Suppose we have seen successful human resources models in the past, and have (explicitly or just mentally) generalized these to produce a generic model.

3.3: Adopting Generic Model from Other Applications

Reusing Generic Models of Other Applications

- In absence of a generic model in a required application, a generic model from other application can be used.
- Consider that you need to build a property insurance model and you don't have any model readily available. In such case, a model from life insurance or health insurance could be used.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 10

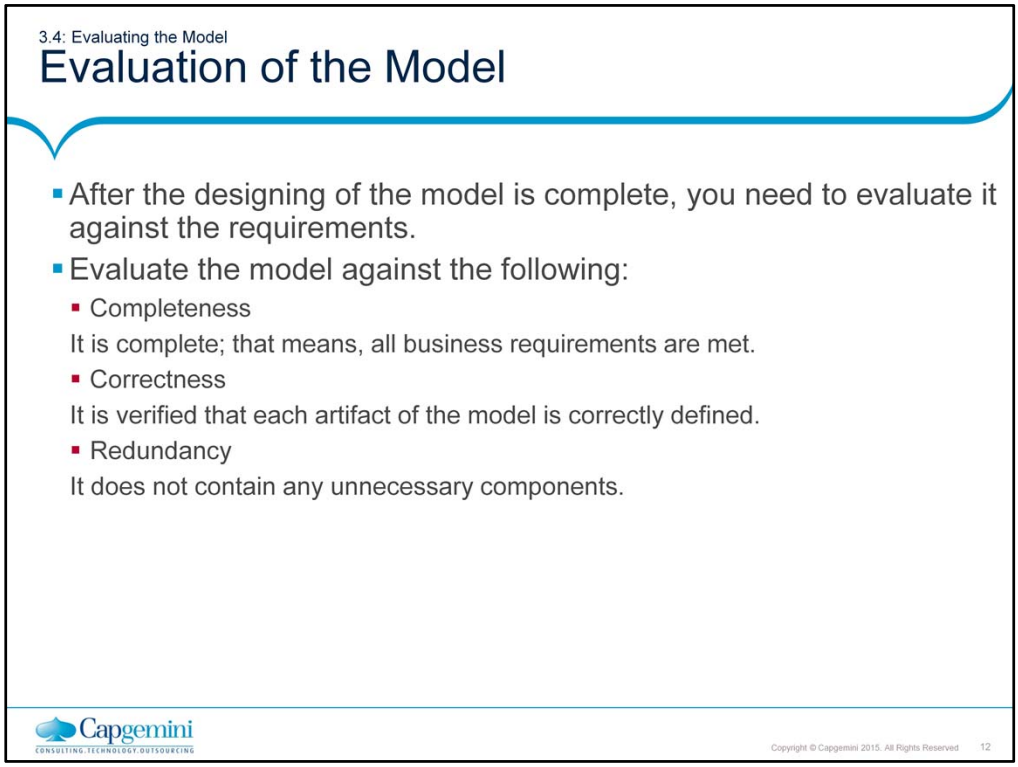
3.3: Generic Model

When there is no generic Model

- In case no generic model is available, a new model should be developed.

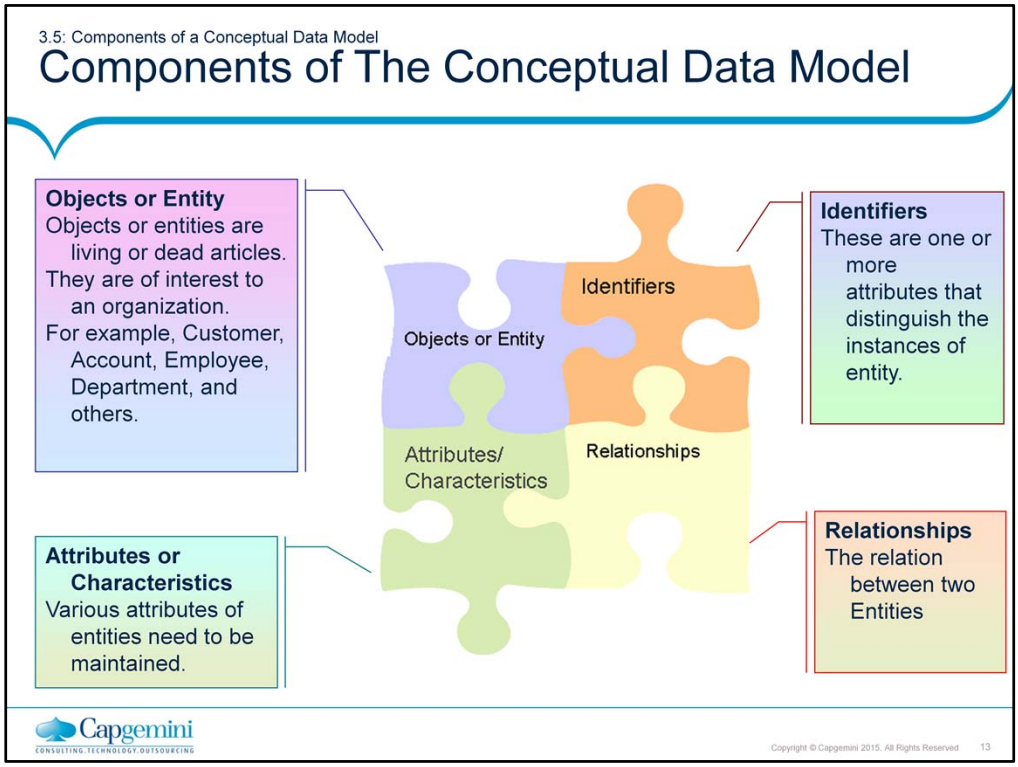
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 11



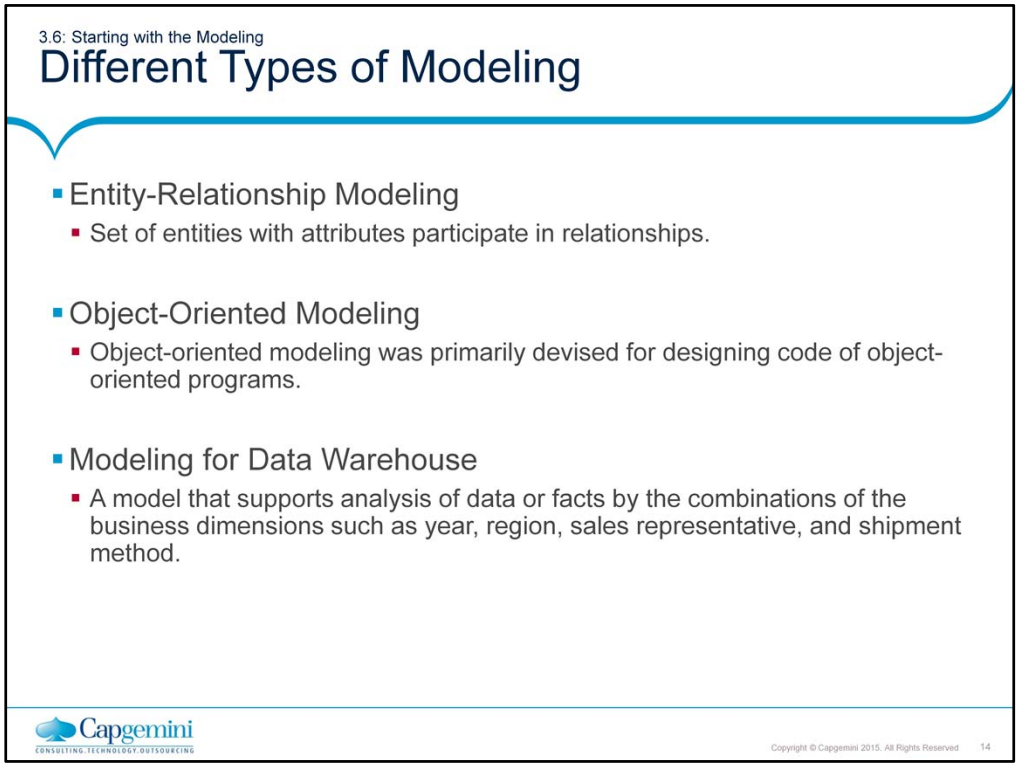
Evaluation of the Model:

Having developed one or more candidate conceptual models, you need to select the most appropriate alternative and verify that it meets the business needs. Perform the evaluation thoroughly at this step. You will then require to only review the design decisions that you make as you proceed from the conceptual to logical and to physical models, rather than reviewing the later models in their entirety.



Components of The Conceptual Data Model:

- **Objects or Entity:**
  When you analyze the information requirements of a company, you will notice that the company needs information about the business objects significant for it.
- **Attributes:**
  Each customer has an intrinsic characteristic known as Customer Name. Every customer has a specific name. Every customer has other inherent or intrinsic characteristics such as Customer Address, Customer Phone Number, Customer Balance, and so on.



Different Types of Modeling:

Entity-Relationship Modeling:

This approach, introduced by Peter Chen in 1976, is still the most popular and widely-used technique. Vendors have produced several computer-aided software engineering (CASE) tools to support this method. This method perceives and portrays the information requirements of an organization as a set of entities with their attributes participating in relationships.

The ER model portrays the information domain of an organization in a way that it is free from any considerations of database software or hardware. Because of this independency, this method is well-suited for conceptual data modeling. It does not burden the domain experts with unnecessary details. However, an ER data model diagram has its shortcomings. The diagram does not clearly indicate constraints in the relationships.

Fact-Oriented Modeling

Not all domain experts are comfortable with the notations in the ER model. Some of them find some of the notations, especially those for relationships, incomplete and imprecise. The fact-oriented data modeling approach attempts to overcome some of the deficiencies of the ER approach.

In the 1970s, an approach to data modeling arose by viewing the information domain in terms of objects playing roles. A role is the part played by an object in a relationship. Object-role modeling (ORM) is such a fact-oriented modeling approach. This is perhaps the only major fact-oriented modeling technique with fairly wide industry support.

Compared with ORM, ER has the following shortcomings:

•It is not closer to natural language for validation by domain experts.

•ER techniques generally support only two-way relationships. N-way relationships in ER are broken down into two-way relationships by introducing intersection identities. However, these intersection identities seem arbitrary and not understood by domain experts.

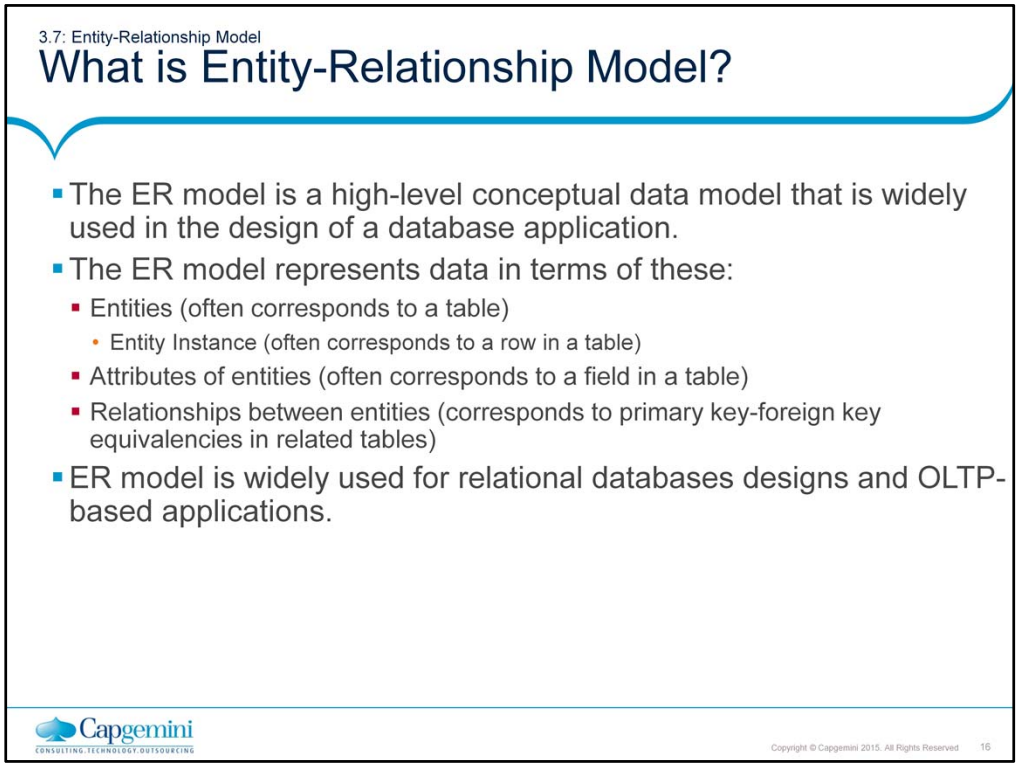
Object-Oriented Modeling: In this approach, both data and behavior are encapsulated within objects. Thus, object-oriented modeling was primarily devised for designing code of object-oriented programs. However, this modeling approach can be adapted for conceptual modeling and eventually for database design.

Till today, the most popular and widely used object-oriented approach is the Unified Modeling Language (UML). The Unified Modeling Language has an array of diagram types, and class diagrams form one important type. Class diagrams can represent data structures and may be considered as extensions of the ER technique.

Data Warehousing Model

As businesses grow, data management becomes more complex. Business executives desperately seek information to stay competitive, improve the bottom line and, importantly, to make strategic decisions. Companies accumulate vast quantities of data in their OLTP systems, but these systems themselves could not support intricate queries and analysis for providing strategic information.

A data warehouse must contain data extracted from OLTP systems — data that can be viewed and modeled for querying and analysis.



What is Entity-Relationship Model?

Entities are the principal data objects about which information is to be collected. Entities are usually recognizable concepts, either concrete or abstract, such as person, places, things, or events which have relevance to the database. Some specific examples of entities are EMPLOYEES, PROJECTS, INVOICES. An entity is analogous to a table in the relational model.

An *entity occurrence* (also called an instance) is an individual occurrence of an entity. An occurrence is analogous to a row in the relational table.

Attributes

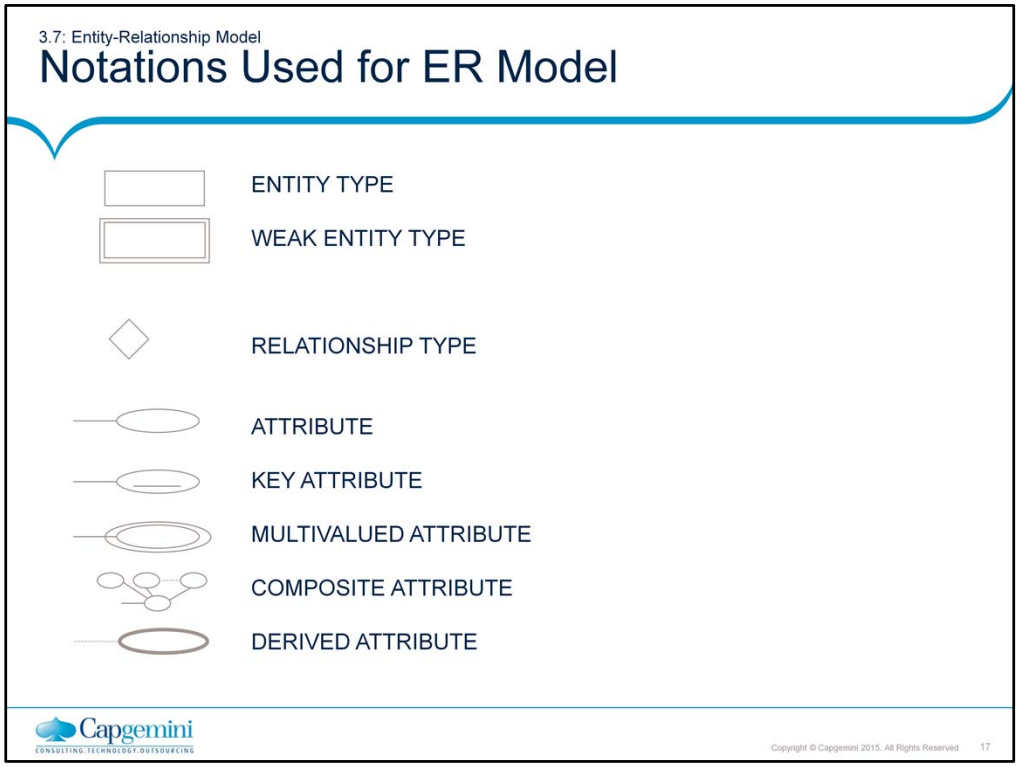
Attributes describe the entity with which they are associated. A particular instance of an attribute is a value. The domain of an attribute is the collection of all possible values an attribute can have. The domain of Name is a character string.

Attributes can be classified as identifiers or descriptors. Identifiers, more commonly called keys, uniquely identify an instance of an entity. A descriptor describes a non-unique characteristic of an entity instance.

Attribute–property or characteristic of an entity or relationship type (often corresponds to a field in a table)

Relationships:

Relationship instance–link between entities (corresponds to primary key-foreign key equivalencies in related tables)



Notations Used for ER Model:

Entities are represented by labeled rectangles. The label is the name of the entity. Entity names should be singular nouns.
Relationships are represented by a solid line connecting two entities. The name of the relationship is written above the line. Relationship names should be verbs.

Attributes, when included, are listed inside the entity rectangle. Attributes which are identifiers are underlined. Attribute names should be singular nouns.

Cardinality of many is represented by a line ending in a crow's foot. If the crow's foot is omitted, the cardinality is one.

Existence is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the bar (looks like the number 1) next to the entity that has a mandatory instance. Optional existence is shown by placing a circle next to the entity that is optional.

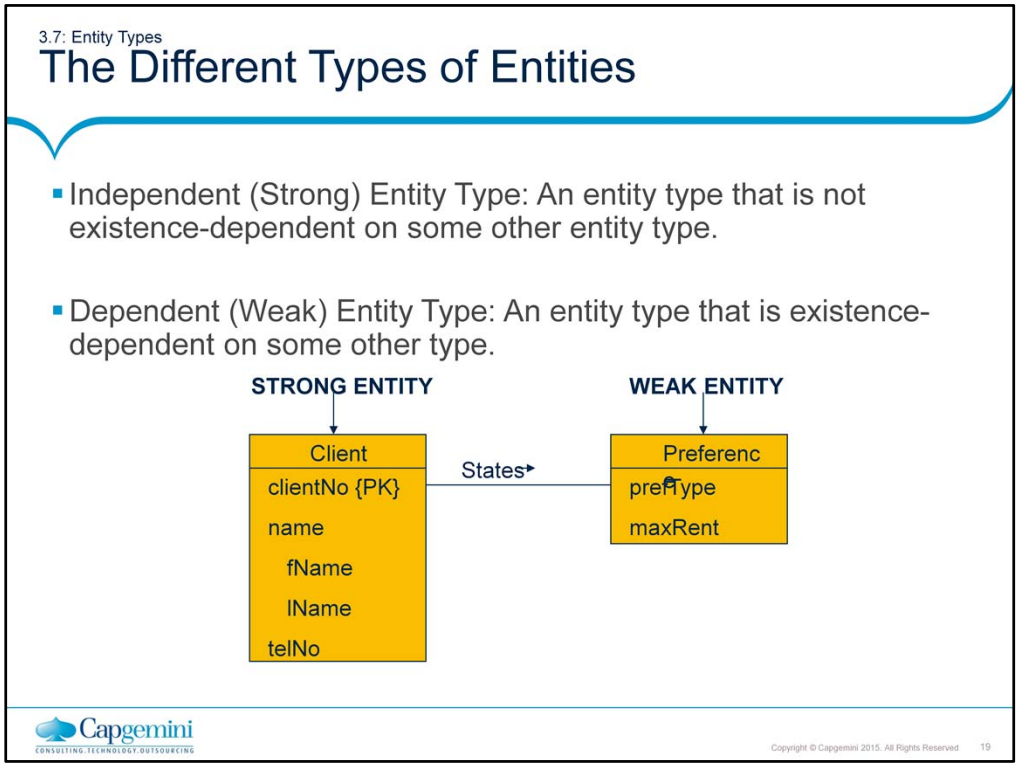
3.7: About Entities

About entities

- An entity is a person, place, thing, event or any of the interest to the enterprise, about which facts may be recorded.
- You should name it in a real world term.
- Eventually entity becomes a table in relational database
- Examples
  - Employee
  - Region
  - Department
  - Customer

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

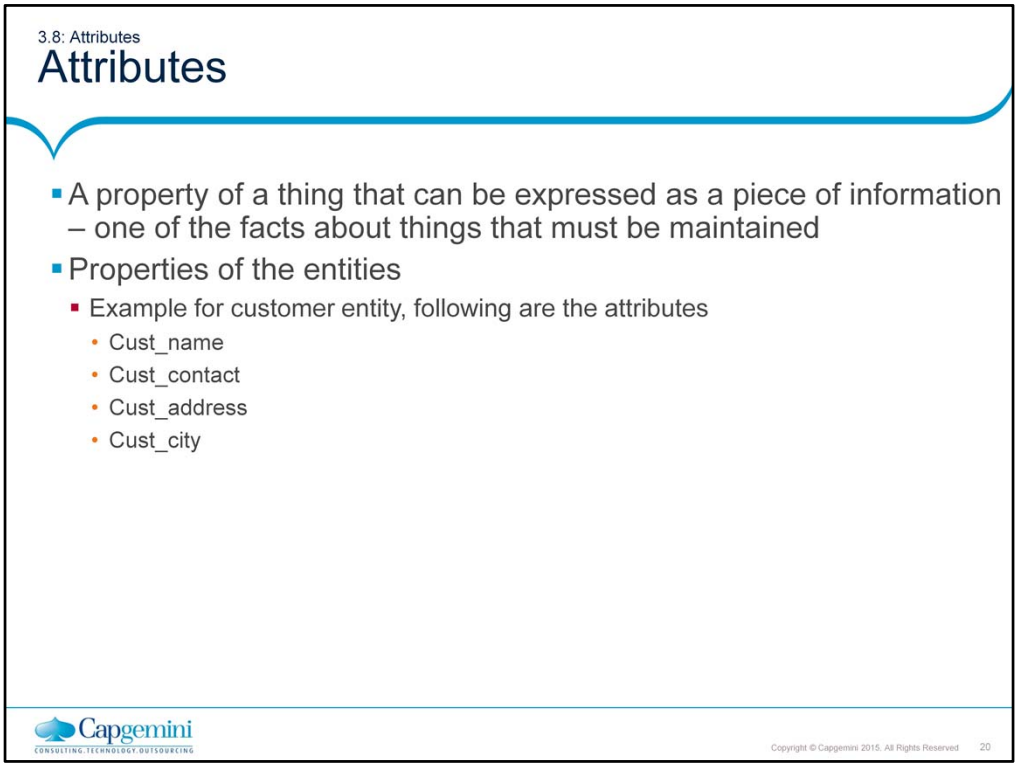
 18



Entity Types

Entities are classified as independent or dependent (in some methodologies, the terms used are strong and weak, respectively). An *independent entity* is the one that does not rely on another for identification. A *dependent entity* is the one that relies on another for identification.

An *entity occurrence* (also called an instance) is an individual occurrence of an entity. An occurrence is analogous to a row in the relational table.



Attributes

Attributes describe the entity with which they are associated. A particular instance of an attribute is a value. For example, “S Ranjan" is one of the values of the attribute Name. The domain of an attribute is the collection of all possible values an attribute can have. The domain of Name is a character string.

Attributes can be classified as identifiers or descriptors. Identifiers, more commonly called keys, uniquely identify an instance of an entity. A descriptor describes a non-unique characteristic of an entity instance.

Types of Attributes

Simple Attributes	
•Each entity has a single atomic value for the attribute.	E.g. SSN or Sex
Composite Attributes	
•The attribute may be composed of several components. •Composition may form a hierarchy where some components are themselves composite	E.g.: Address (Apt#, House#, Street, City, State, Zip_Code, Country) or Name (First_Name, Middle_Name, Last_Name).
Multi-valued Attributes	
•An entity may have multiple values for the attribute.	E.g.: Color of a CAR or Previous Degrees of a STUDENT.
Nested Attributes	
In general, composite and multi-valued attributes may be nested arbitrarily to any number of levels although this is rare.	E.g.: Previous Degrees of a STUDENT is a composite multi-valued attribute denoted by {Previous Degrees (College, Year, Degree, Field)}.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 21

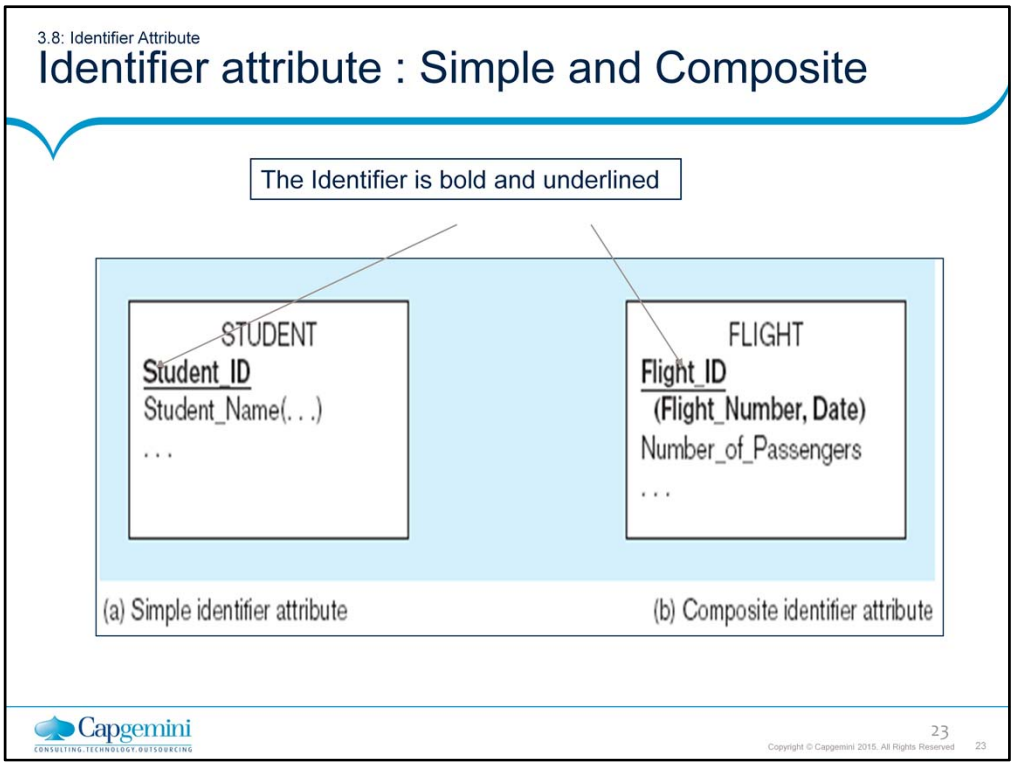
3.8: Identifying Key Attributes

Identifying Key Attributes

- Candidate Key (never NULL): The minimal set of attributes that uniquely identifies each occurrence of an entity type. e.g: branchNo in entity Branch.
- Primary Key: The candidate key that is selected to uniquely identify each occurrence of an entity type. E.g: National Insurance Number.
- Composite Key: A candidate key that consist of two or more attributes.

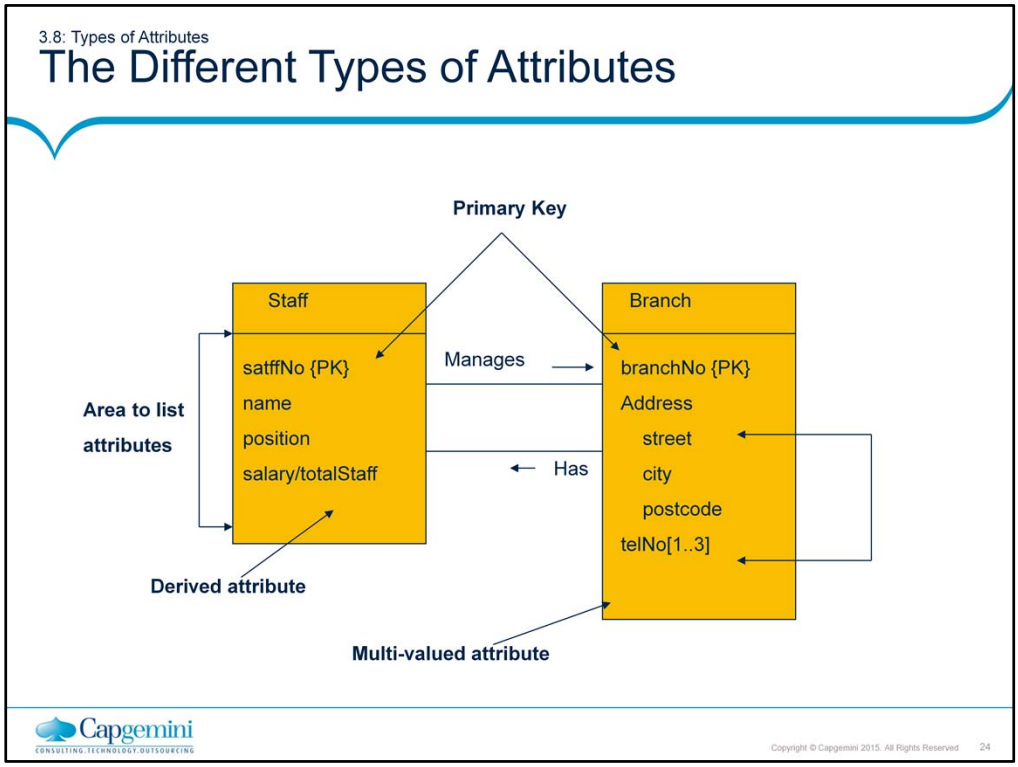
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 22



Simple Attribute: An attribute composed of a single component with an independent existence. E.g position and salary of the Staff entity.

Composite Attribute: An attribute composed of multiple components, each with an independent existence. E.g address attribute of the branch entity that can be subdivided into street, city and postcode attributes



Single-Valued Attribute: An attribute that holds a single value for each occurrence. e.g. EmpId

Multi-Valued Attributes: An attribute that holds multiple values for each occurrence. e.g PhoneNo

Derived Attributes: An attribute that represents a value that is derivable from the value of a related attribute or set of attributes, not necessarily in the same entity type. e.g attribute Age whose value is derived from the CurrentDate and DateOfBirth attributes.

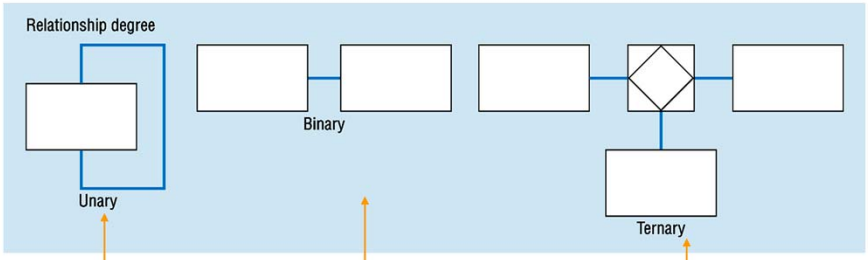
Classifying relationships

- An association between two things (entities) is called a relation.
- Relations are classified by their
  - Degree
  - Connectivity
  - Cardinality
  - Direction
  - Existence.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 25

Degree of Relationship



One entity related to another of the same entity type

Entities of two different types related to each other

Entities of three different types related to each other

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 26

3.9: Relationships in the RDBMS

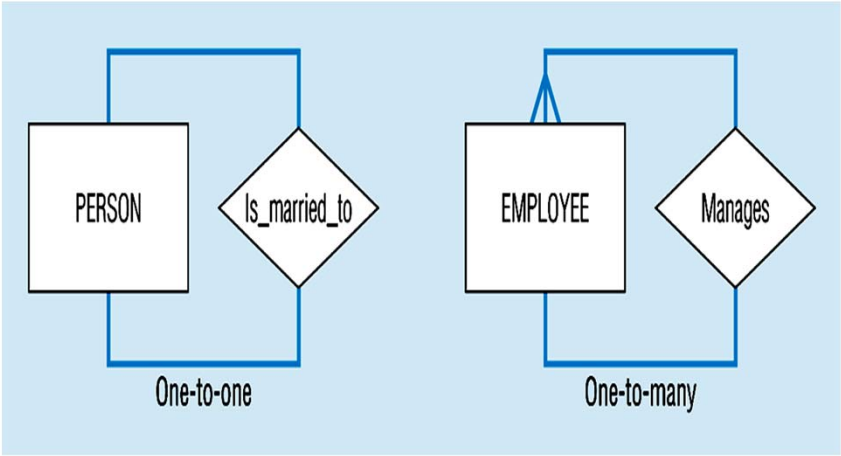
Connectivity and Cardinality

- We have three different types of relationships in RDBMS
  - 1:1 (One to One) – rare
  - 1:M (One to Many) – common
  - M:M (Many to Many) – more in conceptual model, none in Logical model and Physical model
- Examples
  - 1:1 (Person to PAN ID)
  - 1:M (Customer to Phone)
  - M:M (Doctor and Patient)

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 27

Cardinality…

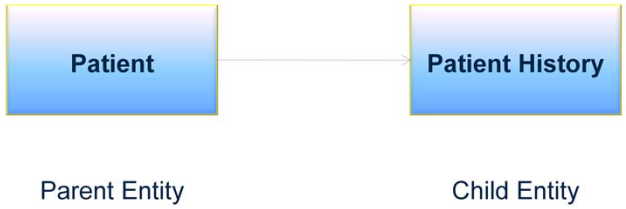


Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 28

direction

- The direction of a relationship indicates the originating entity of a binary relationship.
- The entity from which a relationship originates is the parent entity.
- The entity where the relationship terminates is the child entity.



Parent Entity Child Entity

Existence

- Denotes whether the existence of an entity instance is dependent upon the existence of another, related, entity instance.
- Either mandatory or optional.
- Mandatory - “Every project must be managed by a single department".
- Optional - "employees may be assigned to a BU".

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 30

Constraints on Relationships

- Cardinality Constraints - the number of instances of one entity that can or must be associated with each instance of another entity.
- Minimum Cardinality(also called participation constraint or existence dependency constraints)
  - If zero, then optional participation, not existence-dependent
  - If one or more, then mandatory, existence-dependent
- Maximum Cardinality

The maximum number

One-to-one (1:1)

One-to-many (1:N) or Many-to-one (N:1)

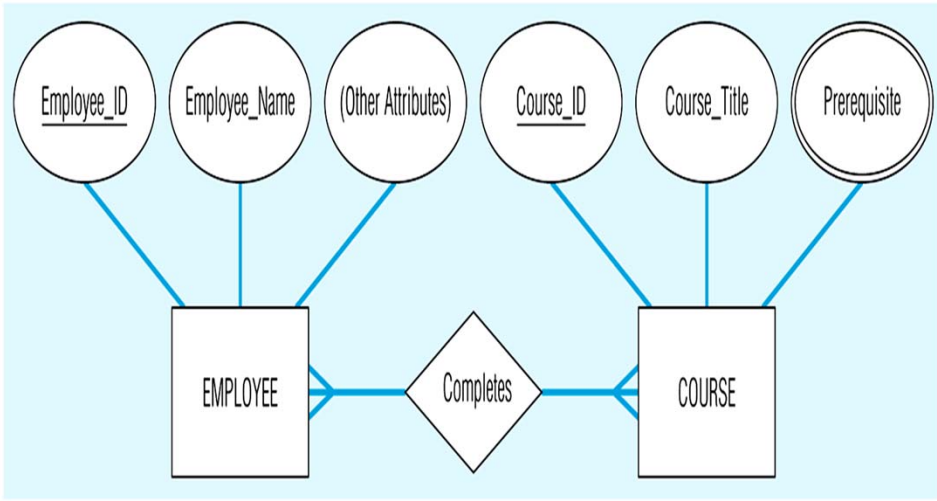
Many-to-many

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 31

3.9: The E-R Model – an Example

The E-R Model: Example

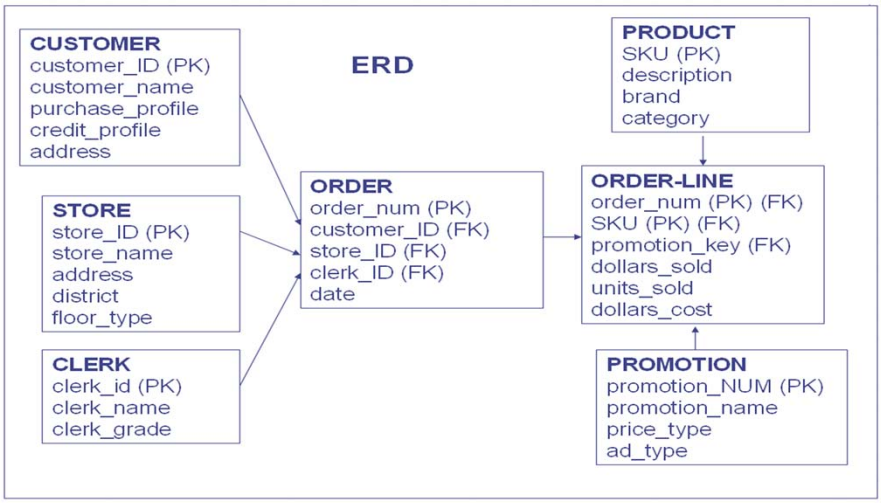


Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 32

3.9: The E-R Model – an Example

The E-R Model: Example



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 33

Case Study

- The XYZ Company wants Capgemini India Pvt Ltd., to design and develop a database system for its regular operations.
- The database should record information about the departments, projects, employees and their dependant. The company is organized into departments. Employees work for a department and may work on many projects. Departments control the project which are being operated from that location. Department has to be managed by someone.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

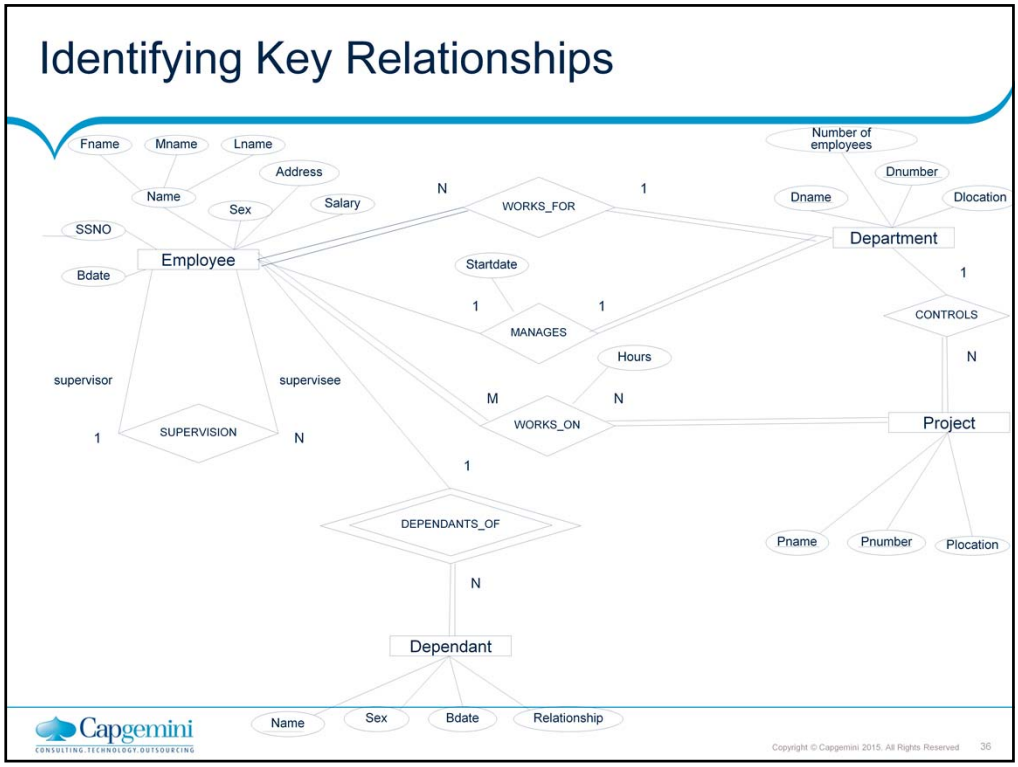
 34

Case Study

- There are managers who manages and monitors the work done by the employees. Suppose an employee is assigned to a project, the hours are calculated based on number of hours the employee is scheduled to work on a project.
- Although most employees have managers, senior staff. The date on which a manager started managing the department could be stored as an attribute of department.
- A department may be spread over many locations. The department name and number are unique for the department. Employee may have number of dependants.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 35


Identifying Key Relationships
Fname
Mname
Lname
Name
Address
Sex
Salary
SSNO
Bdate
Employee
N
WORKS_FOR
1
Number of employees
Dnumber
Dname
Dlocation
Department
Startdate
1
1
MANAGES
1
CONTROLS
N
Hours
supervisor
supervisee
M
N
WORKS_ON
Project
1
SUPERVISION
N
1
DEPENDANTS_OF
Pname
Pnumber
Plocation
N
Dependant
Name
Sex
Bdate
Relationship
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
36

1-to-1 Entity Relationship

- The relationship between these two entities is 1 to 1 because in this company, only 1 manager is allowed to manage a single department.
- Every department is required to have an assigned manager.
- What kind of table design does this suggest?
- A single table: for the Department entity that includes the Manager Entity.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 37

One-to-many (1:N) RELATIONSHIP

- The relationship between these two entities is 1 to Many because there can be 1 or more employees in each department.
- Every department is required to have at least one employee, and no employee can belong to more than one department.
- What kind of table design does this suggest?
- A single table for each entity: the Department Table and Employee Table.

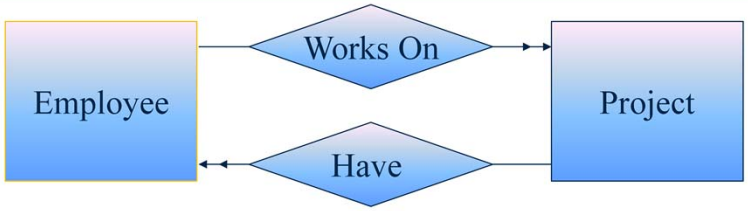
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 38

Many-to-one (N:1) RELATIONSHIP

- The relationship between these two entities is Many to 1 because there can be 1 or more dependants for each employee.
- What kind of table design does this suggest?
- A single table for each entity: the Dependants Table and Employee Table.

Many- to – Many (N:M) Relationship

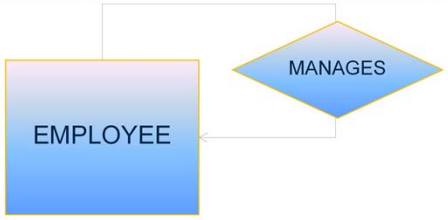


- These 2 entities have 2 relationships - 1 to many in each direction - resulting in a many-many relationship.
- Employees are optionally assigned to one or more Projects, as appropriate. A Project must have at least 1 employee.
- What kind of table design does this suggest?
- 2 Tables plus a table with a column for each entity. (Employee, Project, Employee_Project)

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 40

Recursive Relationships

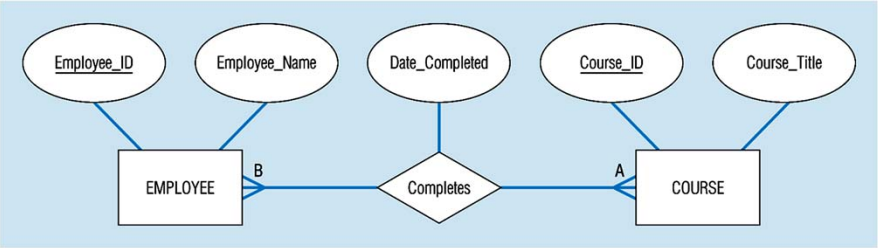


- We can also have a recursive relationship type.
- Both participations are same entity type in different roles.
  E.g.: SUPERVISION (MANAGES) relationships between EMPLOYEE (in role of supervisor or boss) and (another) EMPLOYEE (in role of subordinate or worker).
- In ER diagram, need to display role names to distinguish Participations.

Capgemini
CONSULTING. TECHNOLOGY. OUTSOURCING

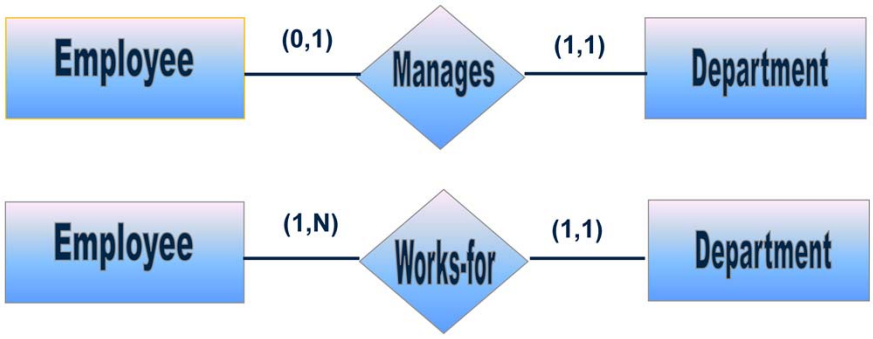
 41

Attributes of Relationship types



- Here, the date completed attribute pertains specifically to the employee's completion of a course…it is an attribute of the relationship

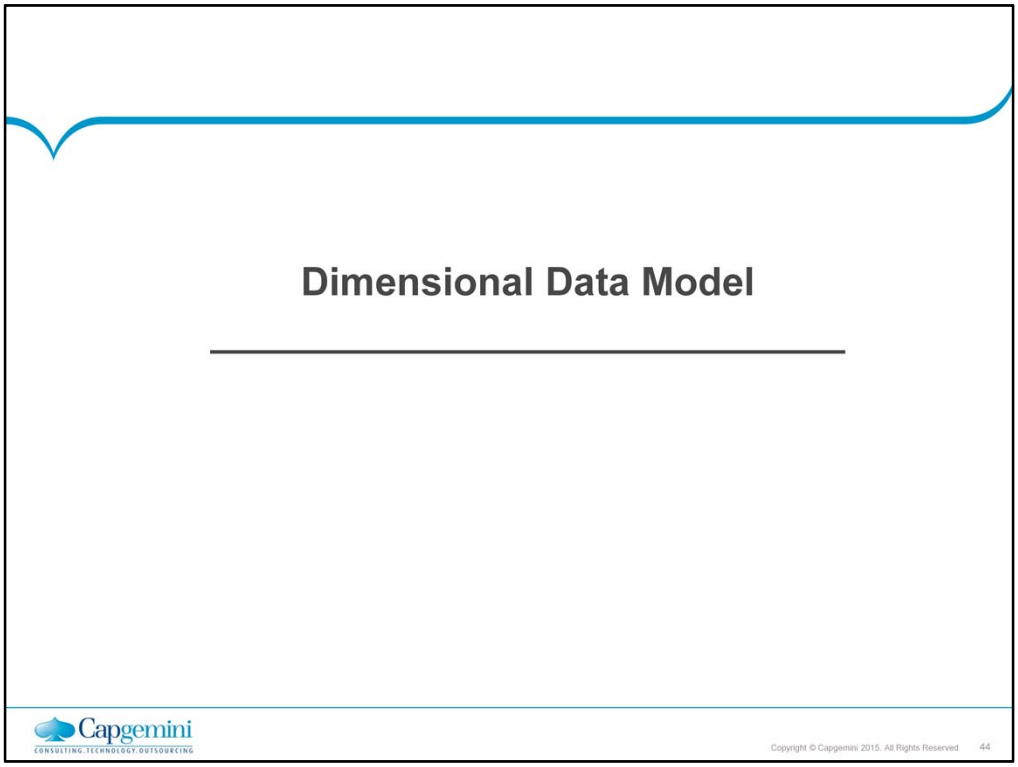
Notation



- The (min, max) notation relationship constraints

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 43


Dimensional Data Model
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
44

3.10: Dimension Modeling

What is dimension modeling?

- Dimension modeling is used to model for data warehouses and data marts, and is different from OLTP modeling.
- Data mart/warehouse differs from OLTP on the basis of the following:
  - Usage (It is information-driven rather than transaction-driven)
  - Type of database used (Multi-dimensional rather than relational)

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 45

Dimensional Model

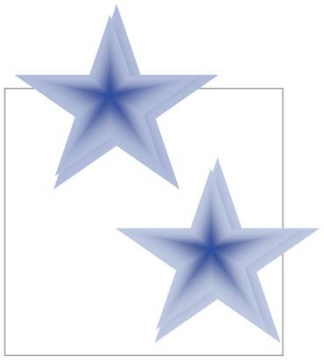
- Definition
  - Logical data model used to represent the measures and dimensions that pertain to one or more business subject areas
  - Dimensional Model = Star Schema
- Serves as basis for the design of a relational database schema
- Can easily translate into multi-dimensional database design if required
- Overcomes OLTP design shortcomings

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 46

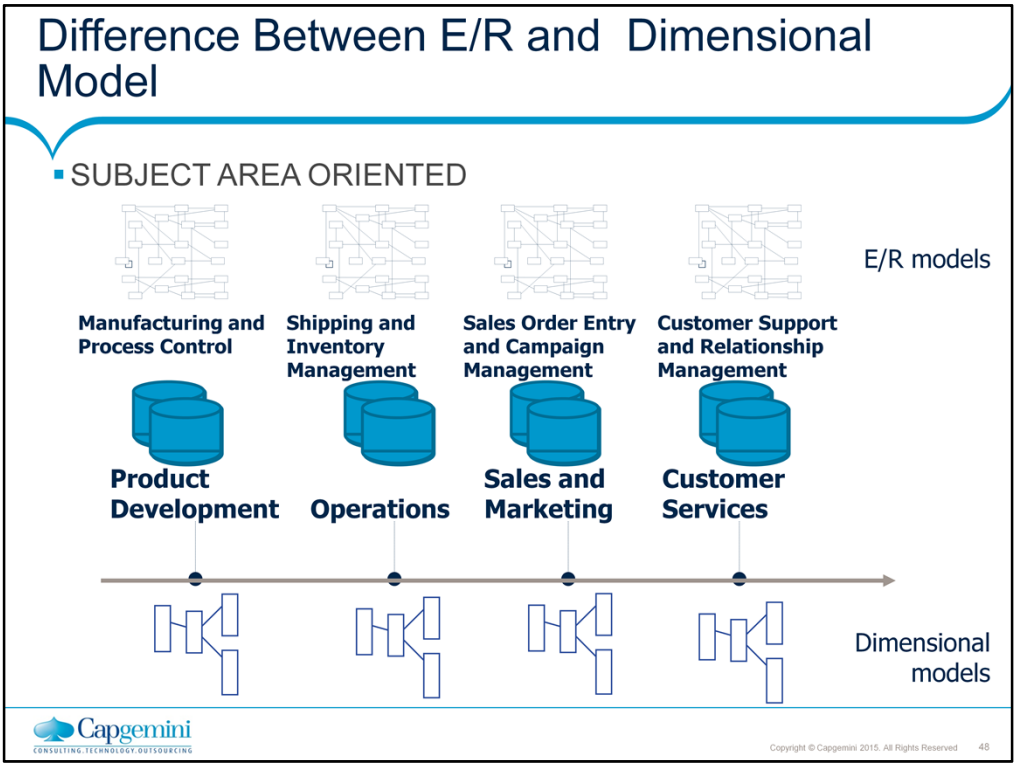
Dimensional Model Advantages

- Understandable
- Systematically represents history
- Reliable join paths
- High performance query
- Enterprise scalability



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

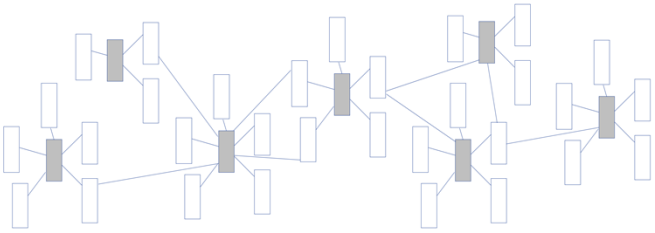
 47


Difference Between E/R and Dimensional Model
SUBJECT AREA ORIENTED
E/R models
Manufacturing and Process Control
Shipping and Inventory Management
Sales Order Entry and Campaign Management
Customer Support and Relationship Management
Product Development
Operations
Sales and Marketing
Customer Services
Dimensional models
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
48

Difference Between E/R and Dimensional Model

- ENTERPRISE SCOPE FOR MODELS

E/R model



Dimensional model

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 49

Process Measurement

- Measures
  - Metrics or indicators by which people evaluate a business process
  - Referred to as “Facts”
- Examples
  - Margin
  - Inventory Amount
  - Sales Dollars
  - Receivable Dollars
  - Return Rate

Coffee Maker Fulfillment Report

Brand	Product	Units Sold	Units Shipped	% Shipped
Captain Coffee	Standard Coffee Maker	5,000	3,800	76%
	Thermal Coffee Maker	2,400	1,632	68%
	Deluxe Coffee Maker	2,073	1,658	80%
	All Products	9,473	**7,090**	75%

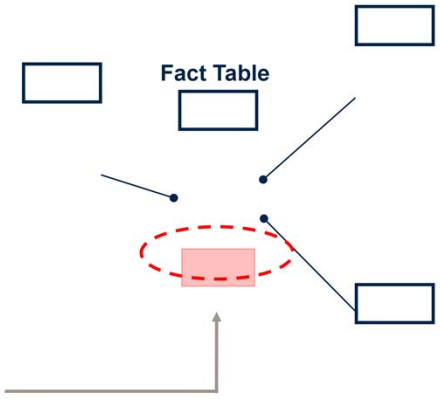
Facts

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 50

Fact Table Grain

- Grain
  - The level of detail represented by a row in the fact table
  - Must be identified early
  - Cause of greatest confusion during design process
- Example
  - Each row in the fact table represents the daily item sales total



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 51

Process Perspectives

- Dimensions
  - The parameters by which measures are viewed
  - Used to break out, filter or roll up measures
  - Often found after the word “by” in a business question
  - Descriptive business terms
- Examples
  - Product
  - Warehouse
  - Customer
  - Supplier

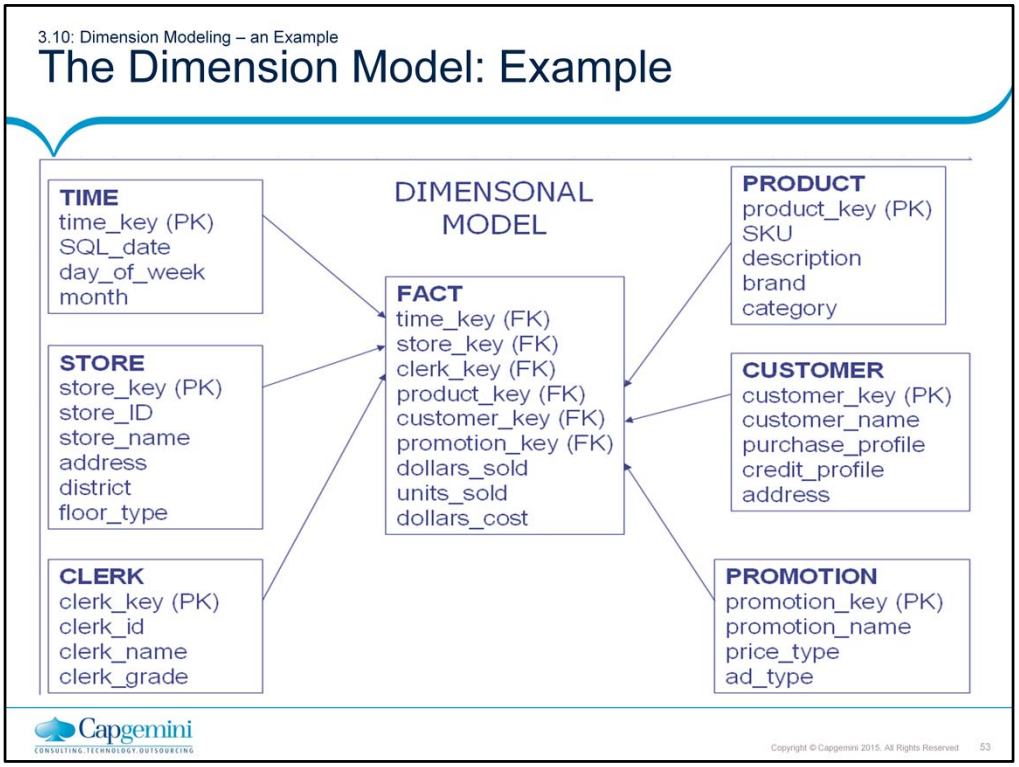
Coffee Maker Fulfillment Report

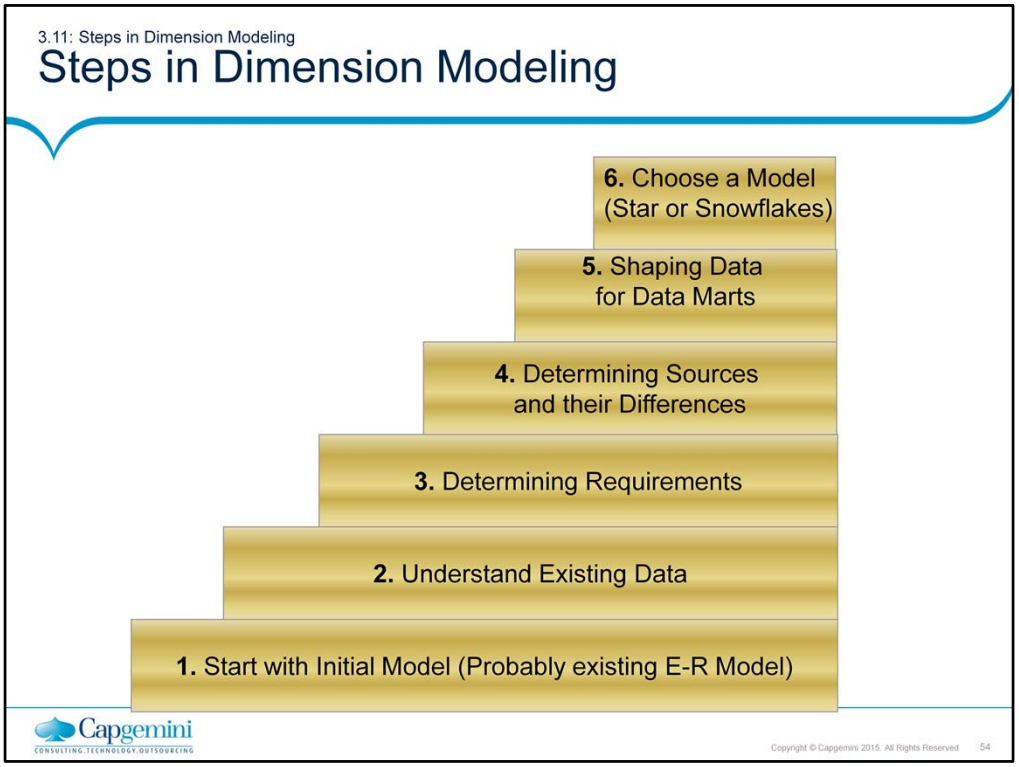
Brand	Product	Units Sold	Units Shipped	% Shipped
Captain Coffee	Standard Coffee Maker	5,000	3,800	76%
	Thermal Coffee Maker	2,400	1,632	68%
	Deluxe Coffee Maker	2,073	1,658	80%
	All Products	**9,473**	**7,090**	**75%**

Dimensions

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 52

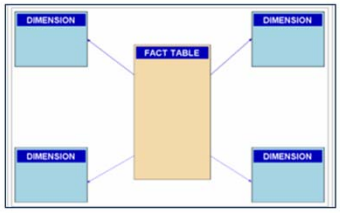

3.10: Dimension Modeling – an Example
The Dimension Model: Example
DIMENSONAL MODEL
TIME
time_key (PK)
SQL_date
day_of_week
month
STORE
store_key (PK)
store_ID
store_name
address
district
floor_type
CLERK
clerk_key (PK)
clerk_id
clerk_name
clerk_grade
FACT
time_key (FK)
store_key (FK)
clerk_key (FK)
product_key (FK)
customer_key (FK)
promotion_key (FK)
dollars_sold
units_sold
dollars_cost
PRODUCT
product_key (PK)
SKU
description
brand
category
CUSTOMER
customer_key (PK)
customer_name
purchase_profile
credit_profile
address
PROMOTION
promotion_key (PK)
promotion_name
price_type
ad_type
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
53


3.11: Steps in Dimension Modeling
Steps in Dimension Modeling
6. Choose a Model (Star or Snowflakes)
5. Shaping Data for Data Marts
4. Determining Sources and their Differences
3. Determining Requirements
2. Understand Existing Data
1. Start with Initial Model (Probably existing E-R Model)
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
54

3.12: The Star Schema

The Star Schema

- A fact table is surrounded by a set of dimension tables.
- The fact table holds transaction data.
- The dimensions are business objects or entities of interest.
- It's easy to understand.



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 55

In addition to the measurements, a fact table contains foreign keys for the dimension tables. These foreign keys are connected to the primary keys of the dimension table. . In star schema each dimension is represented by a single table.

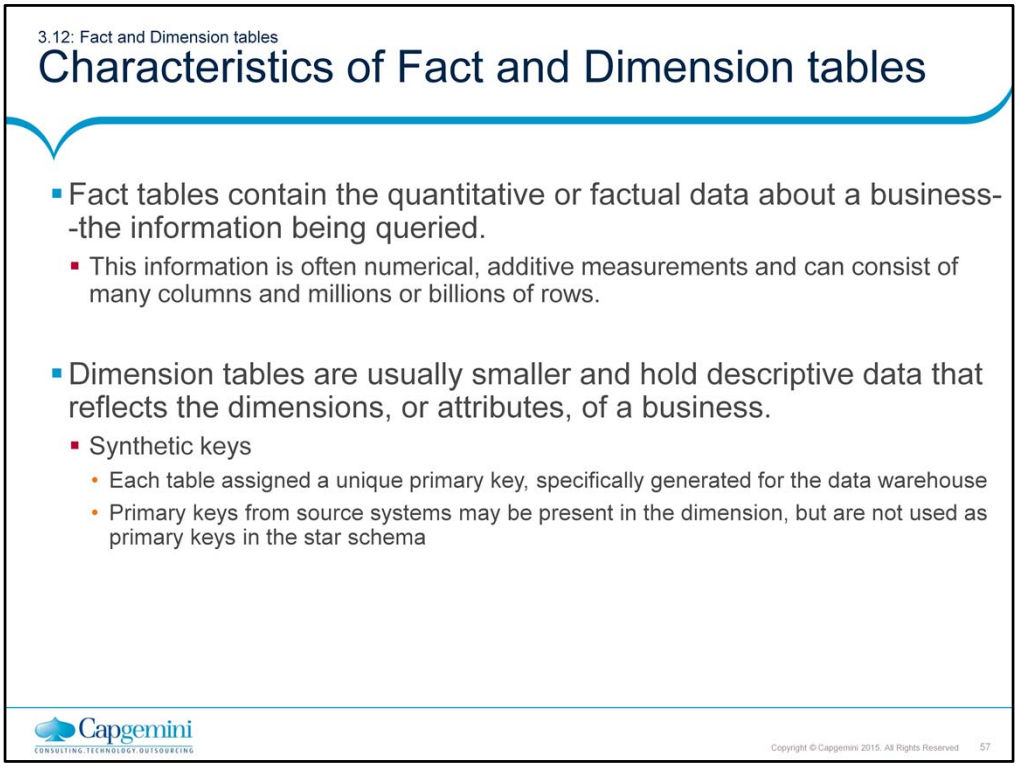
Star Schema is a relational database schema for representing multidimensional data. It is the simplest form of data warehouse schema that contains one or more dimensions and fact tables. It is called a star schema because the entity-relationship diagram between dimensions and fact tables resembles a star where one fact table is connected to multiple dimensions. The center of the star schema consists of a large fact table and it points towards the dimension tables.

Designing a Star Schema

- Based on Kimball's six steps
  - Start designing in order
  - Re-visit and adjust over project life
  - Five initial design steps
    - Identify fact table
    - Identify fact table grain
    - Identify dimensions
    - Select facts
    - Identify dimensional attributes

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 56



Fact table characteristics

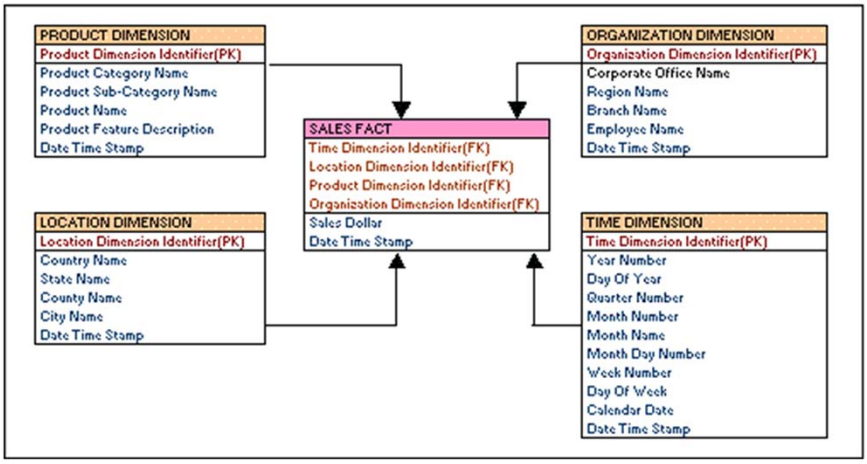
- The fact table contains numerical values of what you measure. For example, fact value of 20 might mean that 20 widgets have been sold.
- Each fact table contains the keys to associated dimension tables. These are called *foreign keys* in the fact table.
- Fact tables typically contain a small number of columns.
- Compared to dimension tables, fact tables have a large number of rows.
- The information in a fact table has characteristics, such as:
  - It is numerical and used to generate aggregates and summaries.
  - Data values need to be additive, or semi-additive, to enable summarization of a large number of values.

Dimension table characteristics

- Dimension tables contain the details about the facts. That, as an example, enables the business analysts to better understand the data and their reports.
- The dimension tables contain descriptive information about the numerical values in the fact table. That is, they contain the attributes of the facts. For example, the dimension tables for a marketing analysis application might include attributes such as time period, marketing region, and product type.
- Since the data in a dimension table is denormalized, it typically has a large number of columns.
- The dimension tables typically contain significantly fewer rows of data than the fact table.
- The attributes in a dimension table are typically used as row and column headings in a report or query results display. For example, the textual
- descriptions on a report come from dimension attributes.

3.12: Star Schema – an Example

Star Schema: Example



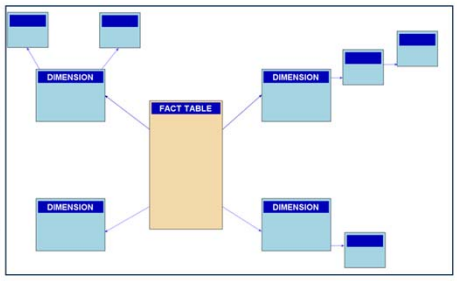
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 58

3.12: The Snow Flake Schema

The Snow Flake Schema

- A fact table is surrounded by a set of Normalized Dimension tables.
- Dimension tables are broken into hierarchical tables.
- This increases number of Joins.
- The table may become large and unmanageable.

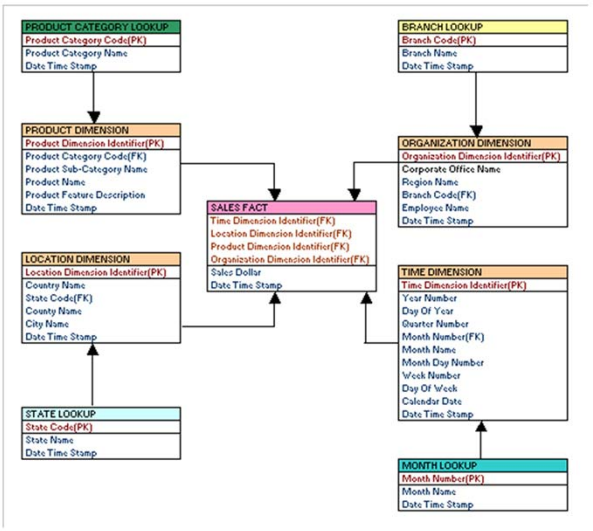


Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 59

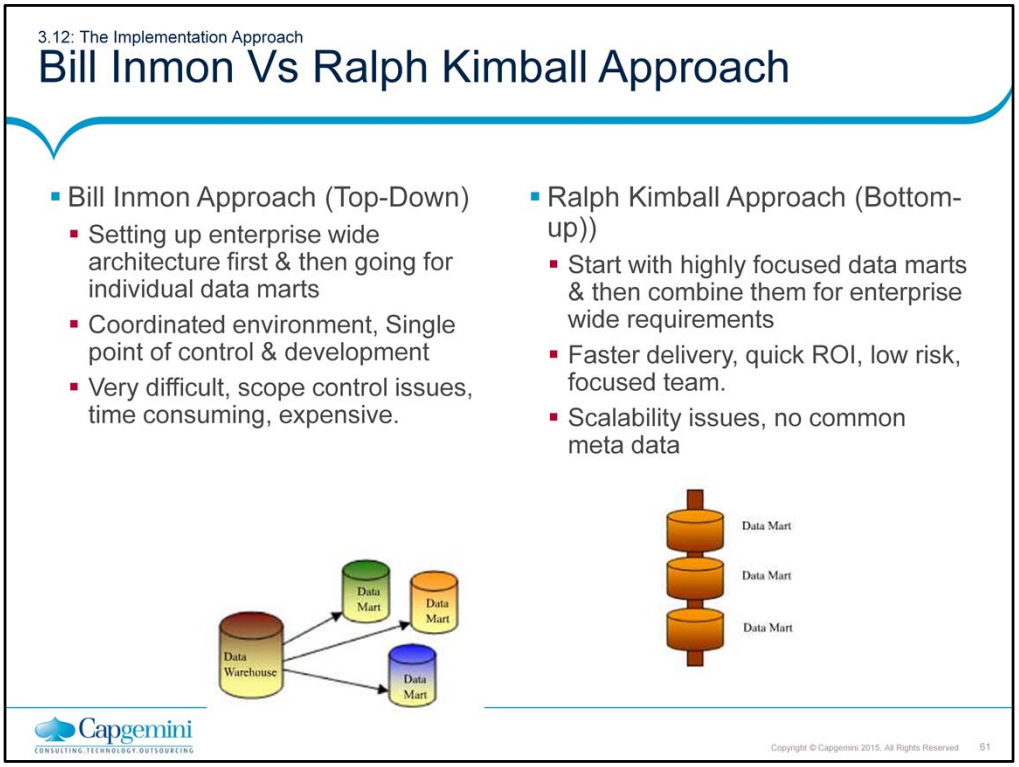
3.12: Snow Flake Schema - Example

Snow Flake Schema: Example



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 60



Both experts are of the opinion that the success of the warehouse/marts depends on effectively gathering the business requirements first. These requirements drive the design of the mart which, in turn, drives the data required in the warehouse. Both experts agree that business-user validation of the data mart design ensures that expectations are managed.

The initial model is the starting point for the design of the staging area (or warehouse). This is where the referential integrity rules are applied (via the DBMS or software validation) and transformation of disparate values is performed. Kimball calls it the backroom, and Inmon calls it the data warehouse.

Inmon advocated a “dependent data mart structure” whereas Kimball advocated the “data warehouse bus structure”.

Bill Inmon Approach
Transfer of data happens from diverse OLTP systems into a centralized place where the data could be used for analysis. Warehouse needs to be build first and data should be made accessible at detailed atomic levels by drilling down or at summarized levels by drilling up. The data marts are treated as sub sets of the data warehouse. Each data mart is built for an individual department and is optimized for their analysis needs.

This data is loaded into the staging area and validated and consolidated for ensuring a level of accuracy and then transferred to the **optional** Operational Data Store (ODS). Data is also loaded into the Data warehouse in a parallel process to avoid extracting it from the ODS. Once the Data warehouse building processes are complete, the data mart refresh cycles will extract the data from the Data warehouse into the staging area and perform a new set of transformations on them. This helps in organizing the data in particular structures required by data marts.

Ralph Kimball Approach
Ralph Kimball suggested the data warehouse with the data marts connected to it with a bus structure. The bus structure contained all the common elements that are used by data marts such as conformed dimensions, measures etc defined for the enterprise as a whole. According to him, by using these conformed elements, users can query all data marts together. This architecture makes the data warehouse more of a virtual reality than a physical reality.
The bottom-up approach reverses the positions of the Data warehouse and the Data marts. Data marts are directly loaded with the data from the operational systems through the staging area.

Hybrid Approach
Start with data mart having focus on enterprise wide scope. It aims to harness the speed and user orientation of the Bottom up approach to the integration of the top-down approach. The Hybrid approach begins with an Entity Relationship diagram of the data marts and a gradual extension of the data marts to extend the enterprise model in a consistent, linear fashion. The data from the various data marts are then transferred to the data warehouse and query tools are reprogrammed to request summary data from the marts and atomic data from the data warehouse.

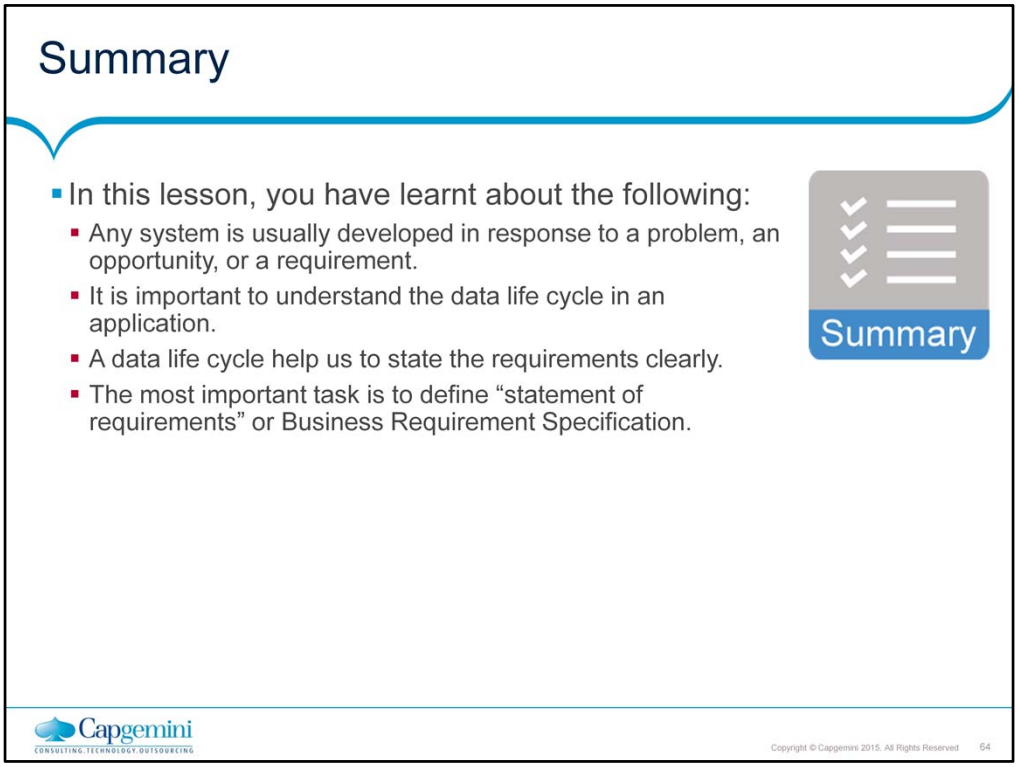
3.13: Conceptual Data Design

Conceptual Data Design

Feature	Conceptual	Logical	Physical
Entity Names	✓		
Entity Relationships	✓		
Attributes			
Primary Keys			
Foreign Keys			
Table Names			
Column Names			
Column Data Types			

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 63



Add the notes here.

Review Question

- Question 1: In ____ a fact table is surrounded by a set of Normalized Dimension tables.
- Question 2: An association between two things (entities) is called a ____.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 65