Data Modeling for Business Intelligence

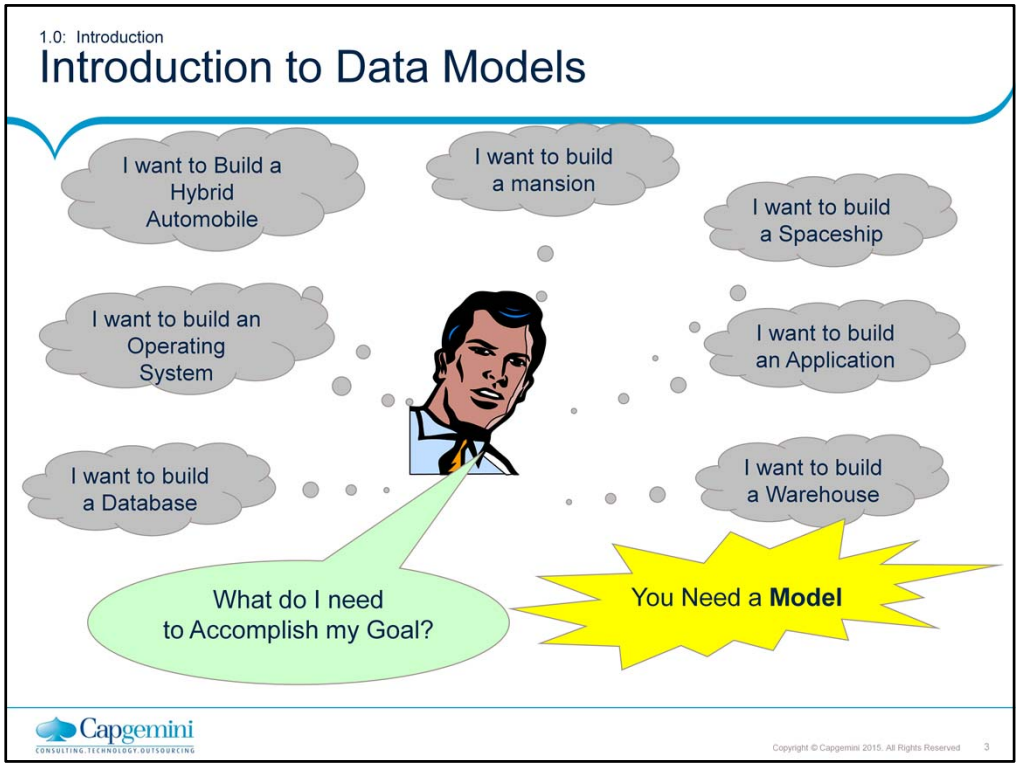
Lesson 1: Introduction to Data Modeling

Lesson Objectives

- On completion of this lesson on Data Modeling, you will be able to:
  - State the importance of data modeling
  - Identify features of a good data model
  - Identify who should be involved in data modeling
  - List the database design stages and deliverables
  - Explain classification of information

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

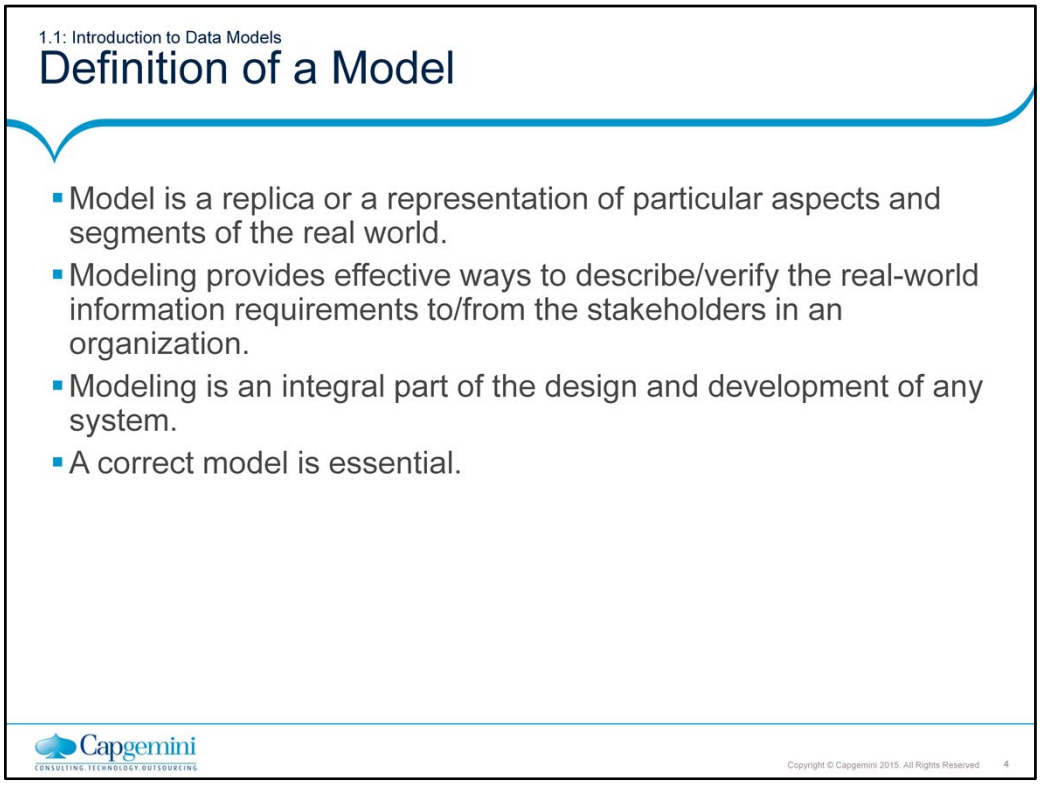
 2



Introduction:

In order to do any of the above, First, you need to create a model of the requirement.

Without a proper model of a requirements, an adequate system cannot be correctly designed and implemented. A good model of high quality forms an essential prerequisite for any successful system.



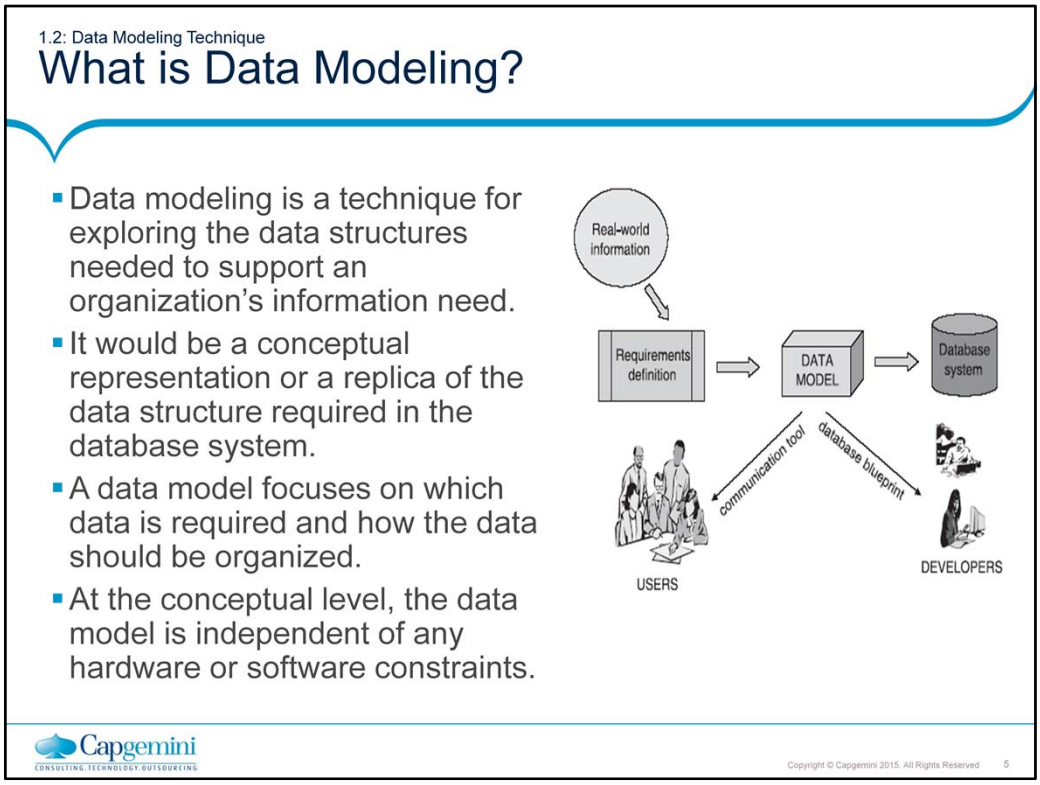
What is a model?

A model serves two primary purposes:

1) As a true representation of some aspects of the real world, a model enables clearer communication about those aspects.
2) A model serves as a blueprint to shape and construct the proposed structures in the real world.

So, what is a data model? A data model is an instrument that is useful in the following ways:

1) A model helps the users or stakeholders clearly understand the database system that is being implemented. It helps them understand the system with reference to the information requirements of an organization.
2) It enables the database practitioners to implement the database system exactly conforming to the information requirements.



What is Data Modeling?

- At this level, the data model is generic; it does not vary whether you want to implement an object-relational database, a relational database, a hierarchical database, or a network database.
- At the next level down, a data model is a logical model relating to the particular type of database relational, hierarchical, network, and so on. This is because in each of these types, data structures are perceived differently.
- If you proceed further down, a data model is a physical model relating to the particular database management system (DBMS) you may use to implement the database.

1.3: Simple Data Model

Example of a Simple Data Model

- The data is divided into two tables: one for policy data and one for customer data.

POLICY TABLE

Policy Number	Date Issued	Policy Type	Customer Number	Commission Rate	Maturity Date
V213748	02/29/1989	E20	HAYES01	12%	02/29/2009
N065987	04/04/1984	E20	WALSH01	12%	04/04/2004
W345798	12/18/1987	WOL	ODEAJ13	8%	06/12/2047
W678649	09/12/1967	WOL	RICHB76	8%	09/12/2006
V986377	11/07/1977	SUI	RICHB76	14%	09/12/2006

CUSTOMER TABLE

Customer Number	Name	Address	Postal Code	Gender	Age	Birth Date
HAYES01	S Hayes	3/1 Collins St	3000	F	25	06/23/1975
WALSH01	H Walsh	2 Allen Road	3065	M	53	04/16/1947
ODEAJ13	J O'Dea	69 Black Street	3145	M	33	06/12/1967
RICHB76	B Rich	181 Kemp Rd	3507	M	59	09/12/1941

Capgemini CONSULTING.TECHNOLOGY.OUTSOURCING

 6

A closer look at the model might suggest some questions:

- The meaning of customer is not clear whether he/she is the person insured or the beneficiary of the policy, or the person who pays the premiums?

Could a customer be more than one person, for example, a couple? If so, how would we interpret Age, Gender, and Birth Date?

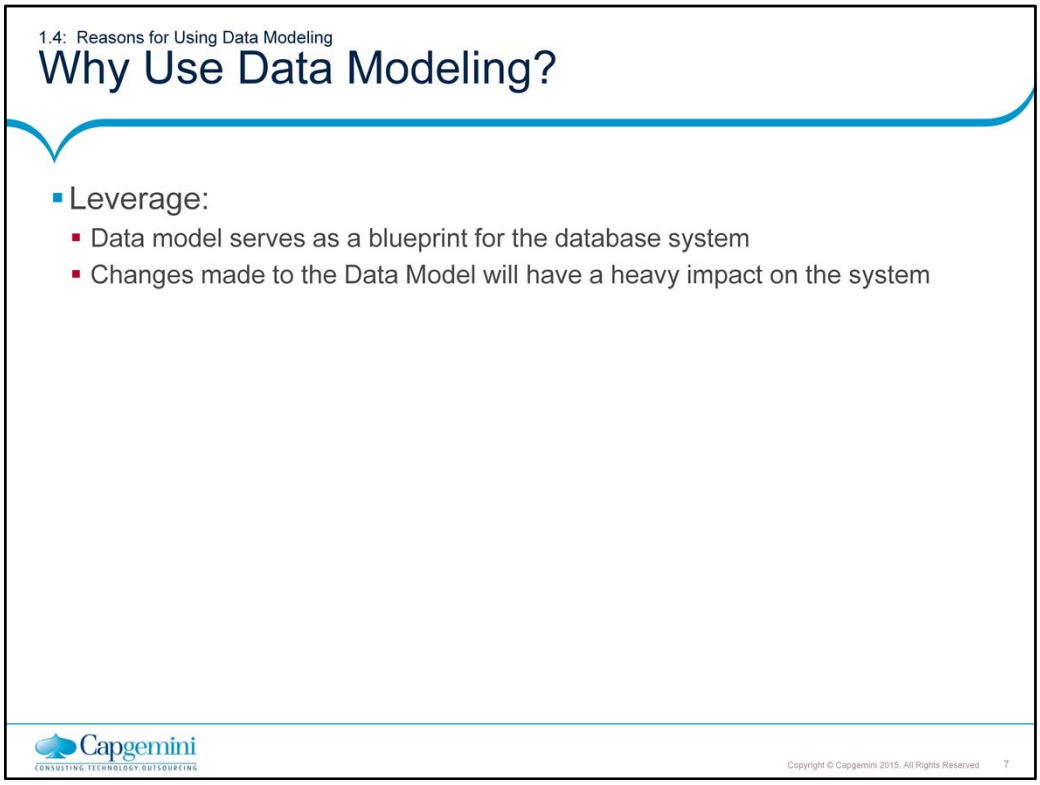
- There may not be any requirement for storing the customers ages. It will be easier to calculate it from Birthdate.
- Is there a relationship between Commission Rate and a Policy Type?. Do policies of type E20 always earn 12% commission?

This will imply recording the same rate many times. How do we record the Commission Rate for a new type of policy if we have not yet sold any policies of that type?

- Customer Number appears to consist of an abbreviated surname, initial,and a two-digit "tie-breaker" to distinguish customers who would otherwise have the same numbers.

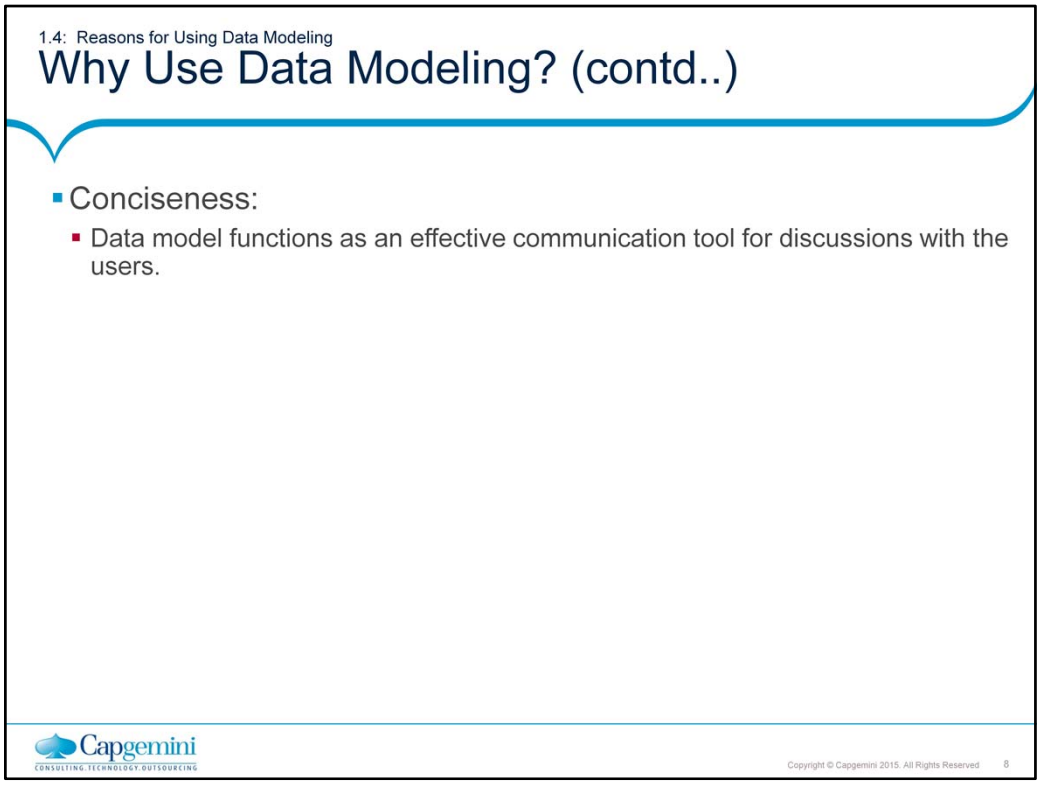
Is this a good choice?

- Would it be better to hold customers' initials in a separate column from their family names?
- "Road" and "Street" have not been abbreviated consistently in the Address column. Should we impose a standard?



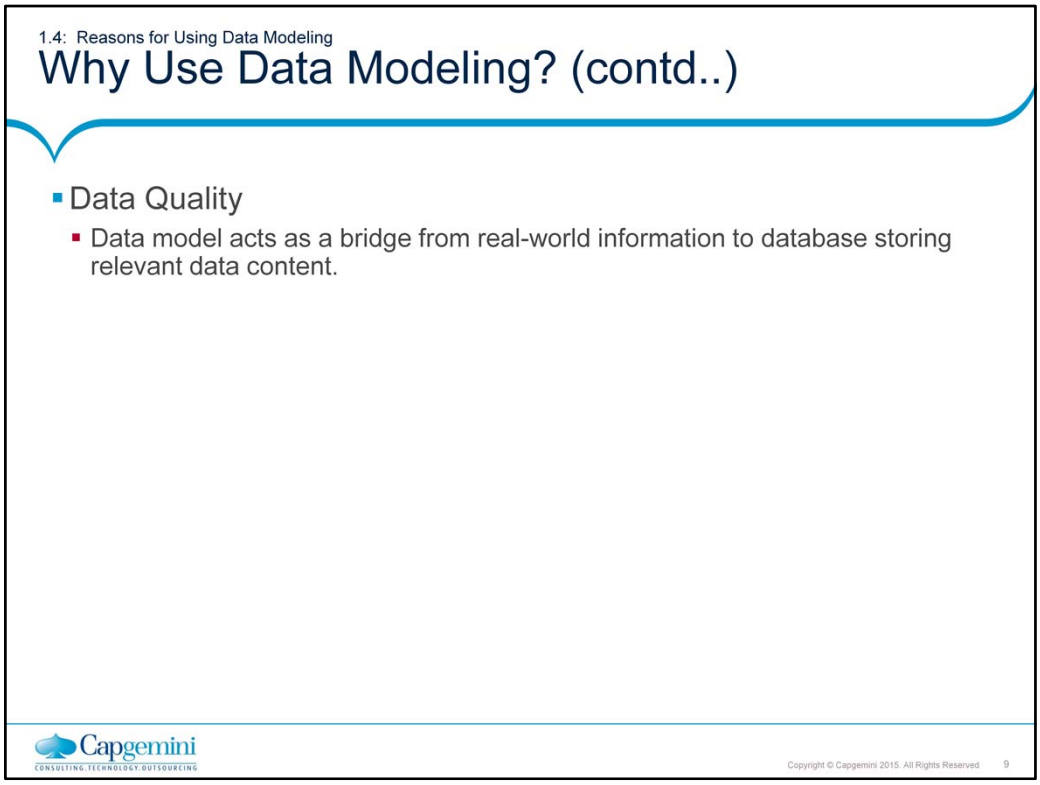
Why Use Data Modeling?

Leverage: The key reason for giving special attention to data organization is the leverage. A small change to a data model may have a major impact on the whole system. Therefore, you can opt for modifying the data model instead of the system. For the most commercial information systems, the programs are far more complex. Also, considerable time is consumed in specifying and constructing them, as compared to the database. However, their contents and structures are heavily influenced by the database design. In the insurance example, imagine that we need to change the rule that each customer can have only one address. The change to the data model may well be reasonably straightforward. Perhaps we will need to add a further two or three address columns to the Policy table. With modern database management software, the database can probably be reorganized to reflect the new model without much difficulty. But the real impact is on the rest of the system. Report formats will need to be redesigned to allow for the extra addresses; screens will need to allow input and display of more than one address per customer; programs will need loops to handle a variable number of addresses; and so on. Changing the shape of the database may in itself be straightforward, but the costs come from altering each program that uses the affected part. In contrast, fixing a single incorrect program, even to the point of a complete rewrite, is a (relatively) simple, contained exercise.



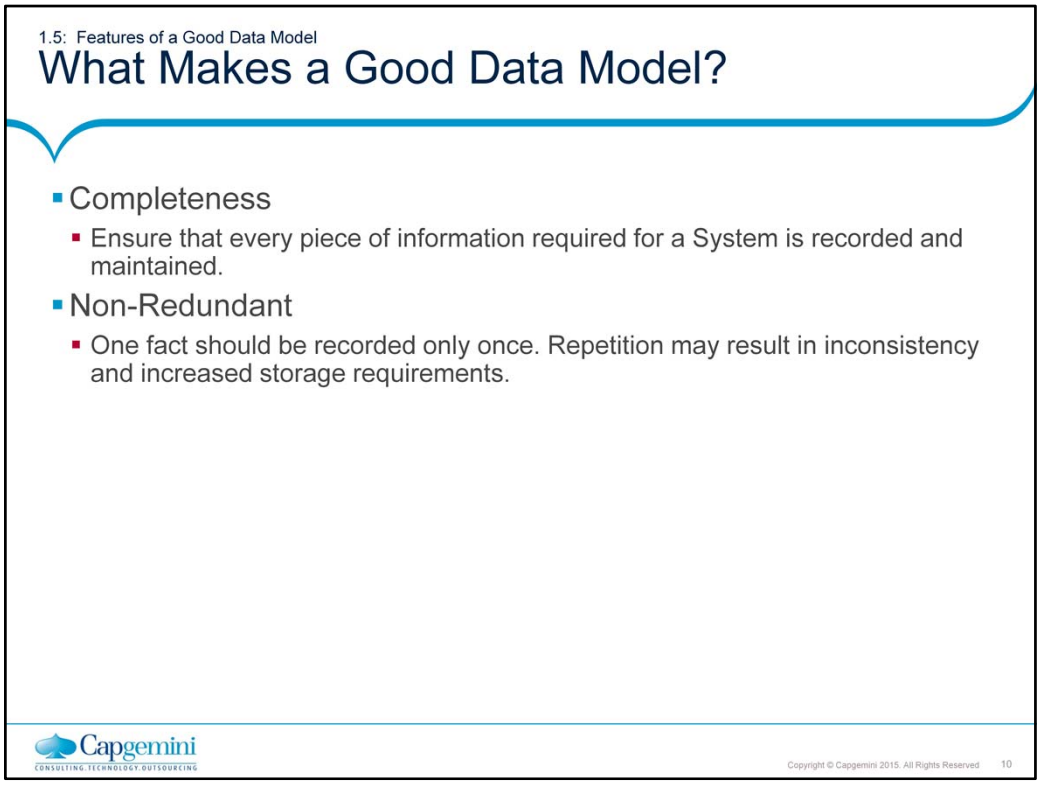
Why Use Data Modeling?

- **Conciseness**: A data model is a very powerful tool for establishing requirements and capabilities of information systems. Its valuable because of its *conciseness*. It implicitly defines a whole set of screens, reports, and processes needed to capture, update, retrieve, and delete the specified data. The data modeling process can tremendously facilitate our understanding of the essence of business requirements.



Why Use Data Modeling?

- **Data Quality**: The data held in a database is usually a valuable business asset built up over a long period. Inaccurate data (poor **data quality**) reduces the value of the asset and can be expensive or impossible to correct. Frequently, problems with data quality can be traced back to a lack of consistency in (a) defining and interpreting data, and (b) implementing mechanisms to enforce the definitions.
  In the insurance example, is Birth Date in U.S. or European date format (mm/dd/yyyy or dd/mm/yyyy)? Inconsistent assumptions here by people involved in data capture and retrieval could render a large proportion of the data unreliable.

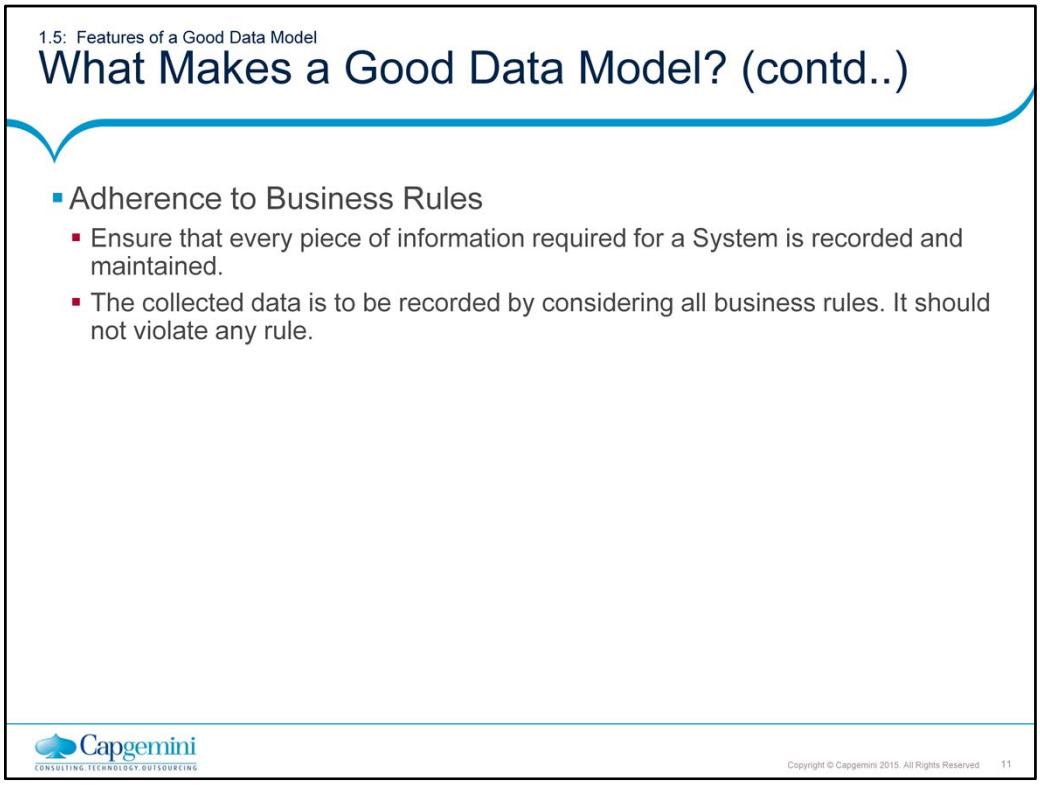


What Makes a Good Data Model?

•**Completeness**: The data model must support all the necessary data. A loss of small piece of information could result in significant loss to the company.

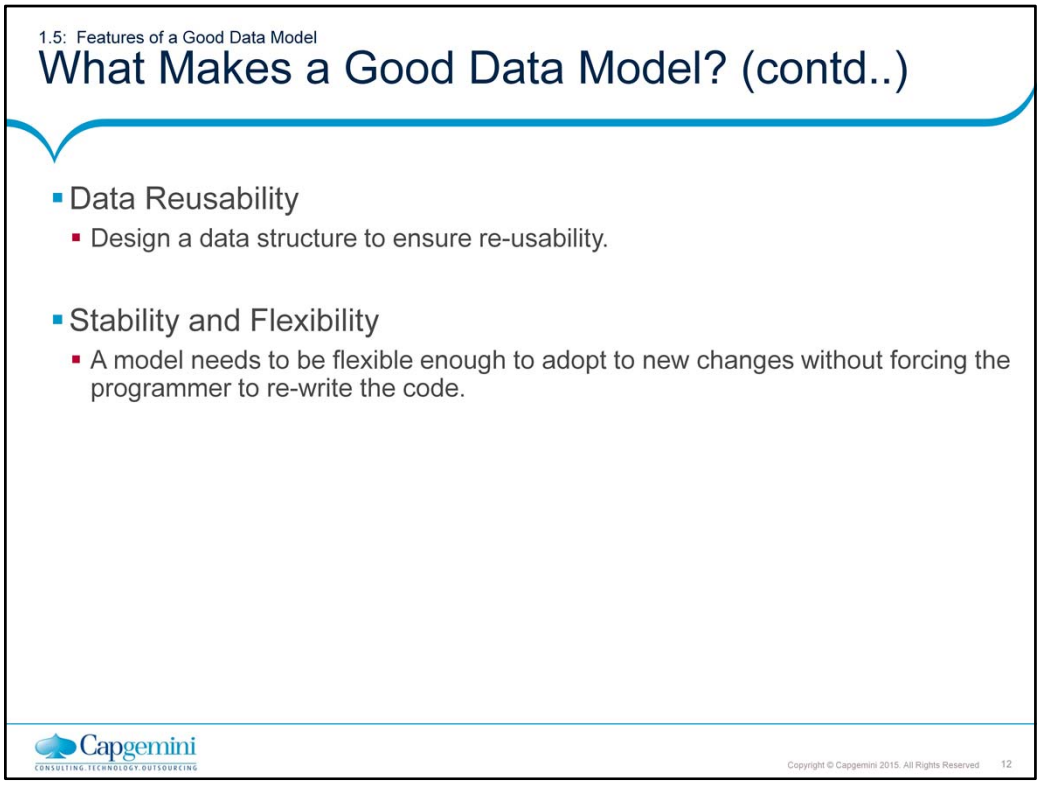
E.g. The insurance model lacks, does not have a column to record a customer's occupation and a table to record premium payments. If such data is required by the system, then these are serious omissions. Also, we have noted that we might be unable to register a commission rate if no policies had been sold at that rate.

•**Non-redundant**: Is the same information recorded more than once? In the example, the same commission rate could be held in many rows of the Policy table. The Age column records the same fact as Birth Date, in a different form. If we added another table to record insurance agents, we could end up holding data about people who happened to be both customers and agents in two places. Recording the same data more than once increases the amount of space needed to store the database.



What Makes a Good Data Model?

Adherence to Business Rules: The data model should accurately reflect and enforce the rules that apply to the business' data. The insurance model enforces the rule that each policy can be owned by only one customer, as there is provision for only one Customer Number in each row of the Policy table. No user or even programmer of the system will be able to break this rule: there is simply no place to record more than one customer against a policy (except extreme measures as holding a separate row of data in the Policy table for each customer associated with a policy). If this rule correctly reflects the business requirement, the resulting database will be a powerful tool in enforcing correct practice, and in maintaining data quality. On the other hand, any misrepresentation of business rules in the model may be very difficult to correct later (or to code around).

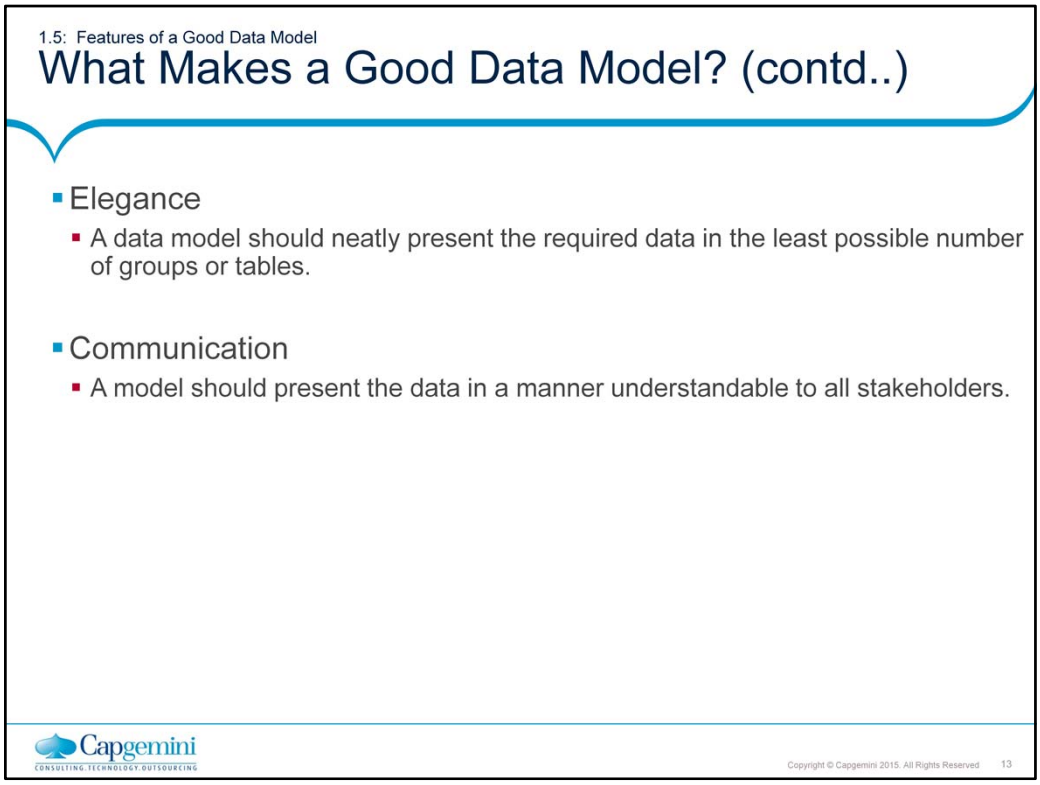


What Makes a Good Data Model? (contd.):

Data Reusability: The data stored in the database be should be reusable for purposes beyond those anticipated in the process model. Once an organization has captured data to for specific requirement, other potential uses and users emerge. An insurance company might initially record data about policies to support the billing function. The sales department then wants to use the data to calculate commissions; the marketing department wants demographic information; regulators require statistical summaries. Seldom can all of these needs be predicted in advance. If data has been organized with one particular application in mind, it is often difficult to use for other purposes. If the system users who have been into capture and storage of data are told that it cannot be made available to suit a new information requirement without extensive and costly reorganization, it could be very frustrating for them. Hence, as far as possible, data should be organized independently of any specific application.

•**Stability and Flexibility**: Regarding the stability and flexibility of the model, the following aspects should be considered:

- ➢Is the model able to cope with possible changes to the business requirements?
- ➢Are the existing tables able to accommodate any new data required to support such changes.
- ➢Alternatively, will simple extensions suffice?
- ➢Or else, will we be forced to make major structural changes, with corresponding impact on the rest of the system?
- ➢A data model is **stable** if we do not need to modify it at all, even if there is a change in requirements. A data model is **flexible** if it can be readily extended to accommodate probable new requirements with only minimal impact on the existing structure.

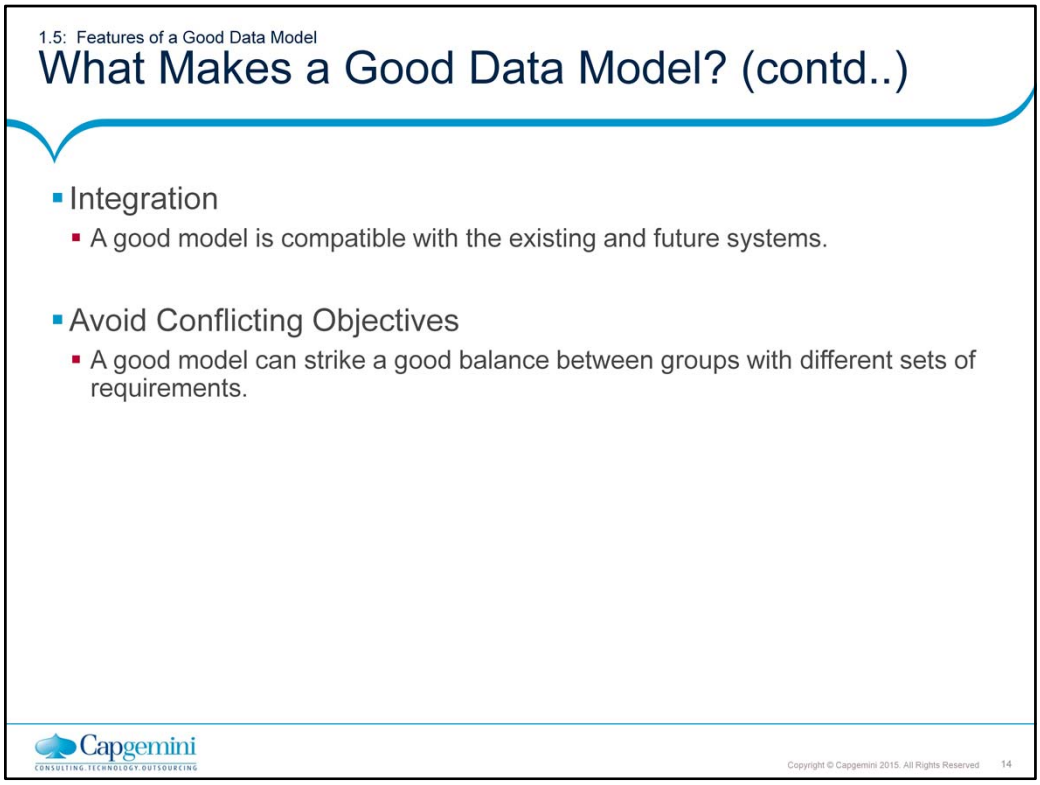


What Makes a Good Data Model? (contd.):

Elegance: Regarding the elegance of the model, the following aspect should be considered: Does the data model provide a reasonably neat and simple classification of the data? If the Customer table were to include only insured persons and not beneficiaries, we might need a separate Beneficiary table. To avoid recording facts about the same person in both tables, we would need to exclude beneficiaries who were already recorded as customers. Our Beneficiary table would then contain "beneficiaries who are not otherwise customers," an inelegant classification that would very likely lead to a clumsy system.

•**Communication**: Regarding the communication, the following aspects should be considered:

- ➢How effective is the model in supporting communication among the various stakeholders in the design of a system?
- ➢Do the tables and columns represent business concepts that the users and business specialists are familiar with and can easily verify?
- ➢Will programmers interpret the model correctly?



Features of a Good Data Model (contd.):

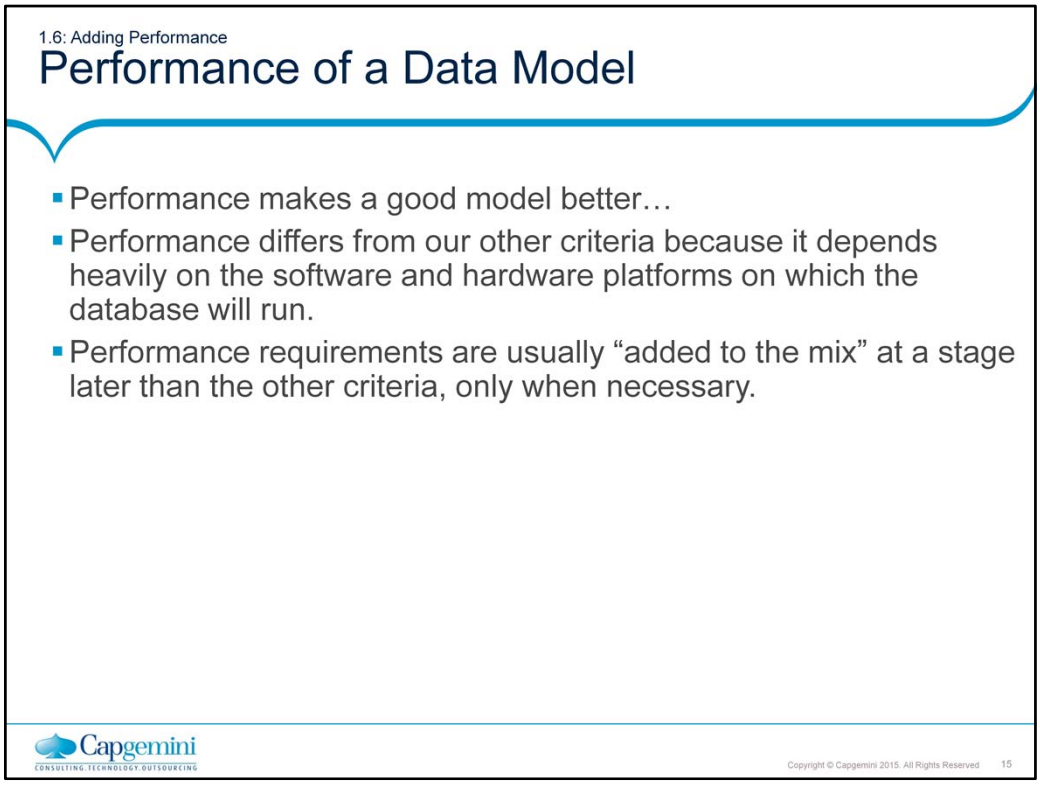
•**Integration**: Regarding integration of the model, the following aspect should be considered:

➢How will the proposed database fit in the organization's existing and future databases?

Even when individual databases are well designed, it is common for the same data to appear in more than one database and for problems to arise in collating together data from multiple databases.

•**Conflicting Objectives:** In many cases, the above aims conflict with one another. An elegant but radical solution may be difficult to communicate to conservative users. We may be so attracted to an elegant model that we exclude requirements that do not fit. A model that accurately enforces a large number of business rules will be unstable if some of those rules change. A model may be easy to understand because it reflects the perspectives of the immediate system users. However, it may not support reusability or may not integrate well with other databases.

Our overall goal is to develop a model that provides the best balance among these possibly conflicting objectives.



Considering the Performance at the initial stage could affect the natural process of data modeling. It may get biased towards a particular technology or database and make the design process biased.

Though it is a good idea to start considering about performance issues from the beginning, however, it must be just recorded for now and later implemented in physical design phase.

Performance can be introduced, once the logical model is freeze and selection of a particular technology & database is done. Since each technology will have it’s own methods to improve performance, the recorded requirement would be very handy while implementing the same.

Where Data Models are used ?

- Operational Systems
  - Traditional Applications designed to run the day-to-day business of the Enterprise
- External Systems ***
  - Data used within an Enterprise that is obtained from outside sources
- Staging Areas ***
  - Created to aid in the collection and transformation of data that is targeted for a Data Warehouse

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 16

Where Data Models are used ?

- Operational Data Store ***
  - W. H. Inmon and Claudia Imhoff definition: “A subject-oriented, integrated, volatile, current valued data store containing only corporate detailed data”.
- Data Warehouse (DW)
  - W. H. Inmon definition: “A subject-oriented, integrated, non-volatile, time-variant collection of data organized to support management needs”.
- Data Mart (DM)
  - TDWI definition: “A data structure that is optimized for access. It is designed to facilitate end-user analysis of data. It typically supports a single analytic application used by a distinct set of workers.”
- *** - Not discussed here

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

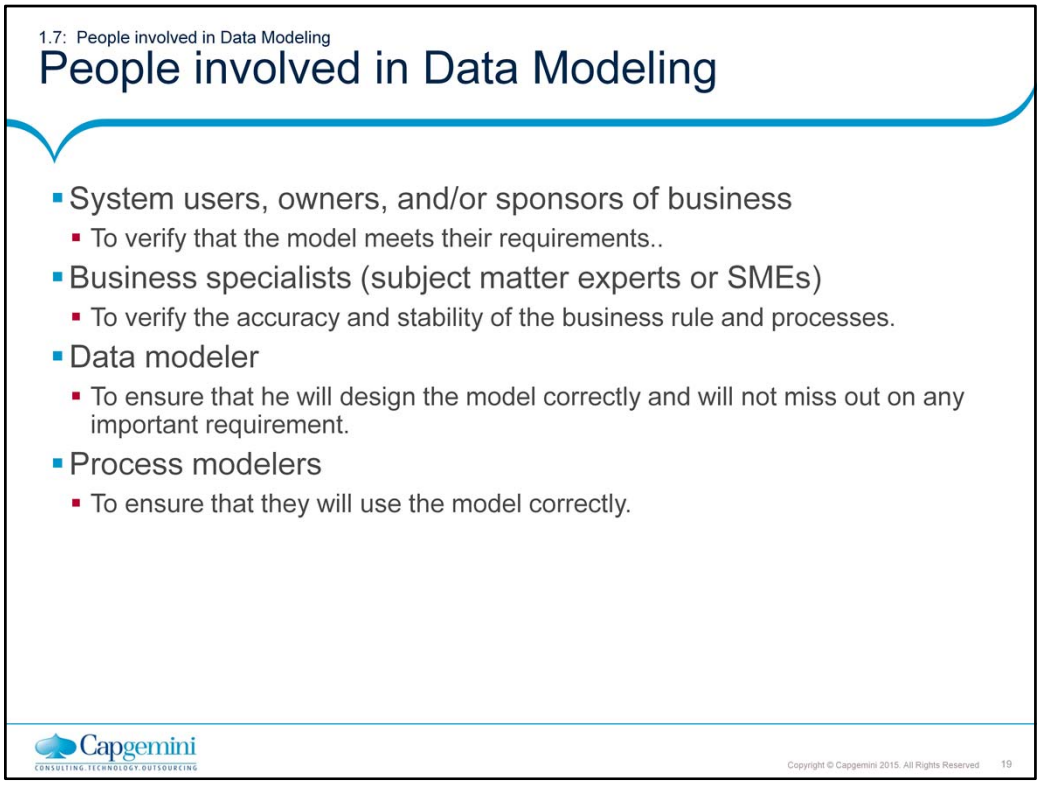
 17

What Data Modeling is not…

- A waste of time!
- A one time effort
- The ultimate IT application development cure
- A quick process
- A function solely performed and understood by and for IT professionals

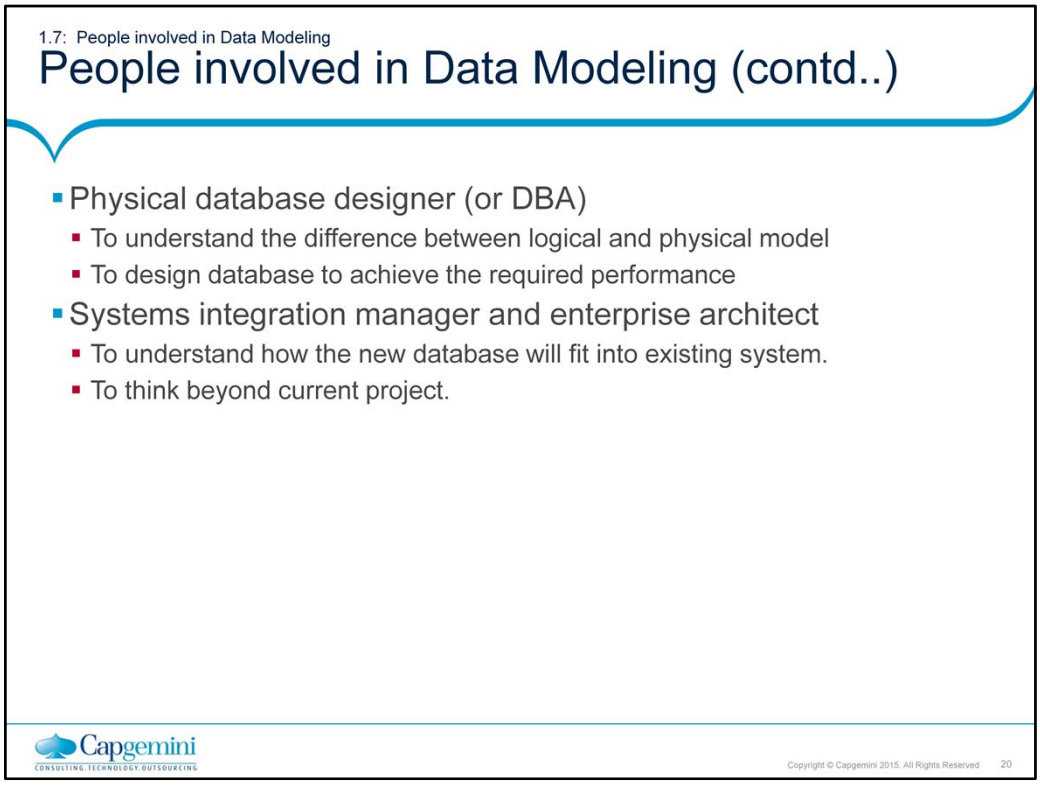
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 18



Who should be involved in data modeling?:

•**System users, owners, and/or sponsors of business:** The system users, owners, and/or sponsors need to verify that the model meets their requirements. Our ultimate aim is to produce a model as the most cost-effective solution for the business. The users' *informed* agreement is an important measure taken towards achieving this aim.
•**Business specialists:** Business specialists (sometimes called Subject Matter Experts or SMEs) may be called upon to verify the accuracy and stability of business rules incorporated in the model. They themselves may not have any immediate interest in the system. For example, we might involve strategic planners to assess the likelihood of various changes to the organization's product range.
•**Data Modeler**: The data modeler has overall responsibility for developing the model and ensuring that other stakeholders are fully aware of its implications for them: "Do you realize that any change to your rule that each policy is associated with only one customer will be very expensive to implement later?"
•**Process modelers:** Process modelers and program designers need to specify programs to run against the database. They want to verify that the data model supports all the required processes without requiring unnecessarily complex or sophisticated programming. In doing so, they need to gain an understanding of the model to ensure that they use it correctly.

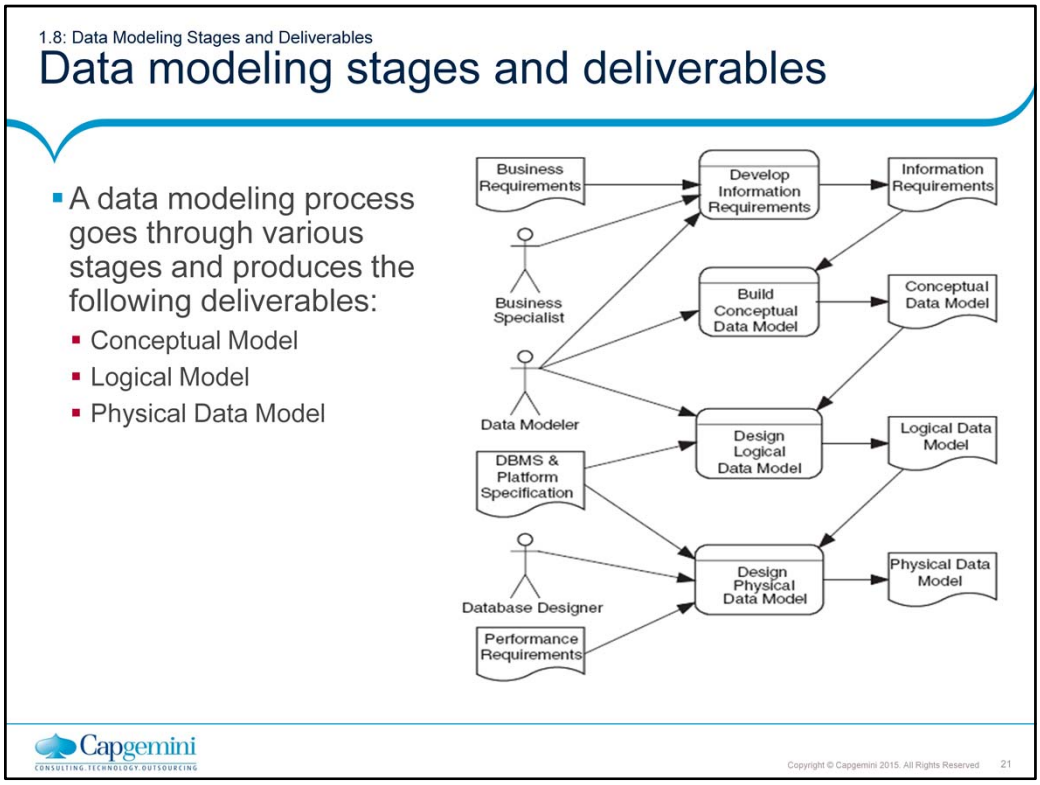


Who should be involved in data modeling? (contd.):

•**Physical Database Designer:** The physical database designer (often an additional role given to the database administrator) will need to assess whether the physical data model needs to differ substantially from the logical data model to achieve adequate performance, and, if so, propose and negotiate such changes. This person (or persons) will need to have an in-depth knowledge of the capabilities of the chosen DBMS.

•**Systems Integration Manager:** The systems integration manager (or other person with that responsibility, possibly an enterprise architect, data administrator, information systems planner, or chief information officer) is interested in how the new database fits into the bigger picture:

- ➢Are there any overlaps with other databases?
- ➢Does the coding of data follow organizational or external standards?
- ➢Have other users of the data been considered?
- ➢Are names and documentation in line with standards?
- ➢In encouraging consistency, sharing, and reuse of data, the integration manager represents business needs beyond the immediate project.



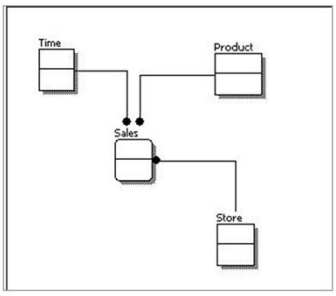
What are data modeling stages and deliverables?

- **Conceptual Model:** The conceptual data model is a (relatively) technology-independent specification of the data to be deposited and maintained in the database. The conceptual model is the focus of communication between the data modeler and business stakeholders, and it is usually presented as a diagram with supporting documentation.
- **Logical Model**: The logical data model is a translation of the conceptual model into structures that can be implemented using a Database Management System (DBMS). Today, that usually means that this model specifies tables and columns, as we saw in our first example. These are the basic building blocks of relational databases, which are implemented using a Relational Database Management System (RDBMS).
- **Physical Data Model**: The physical data model incorporates any changes necessary to achieve adequate performance and is also presented in terms of tables and columns, together with a specification of physical storage (which may include data distribution) and access mechanisms.

1.8: Data Modeling Stages and Deliverables

Conceptual Data Model

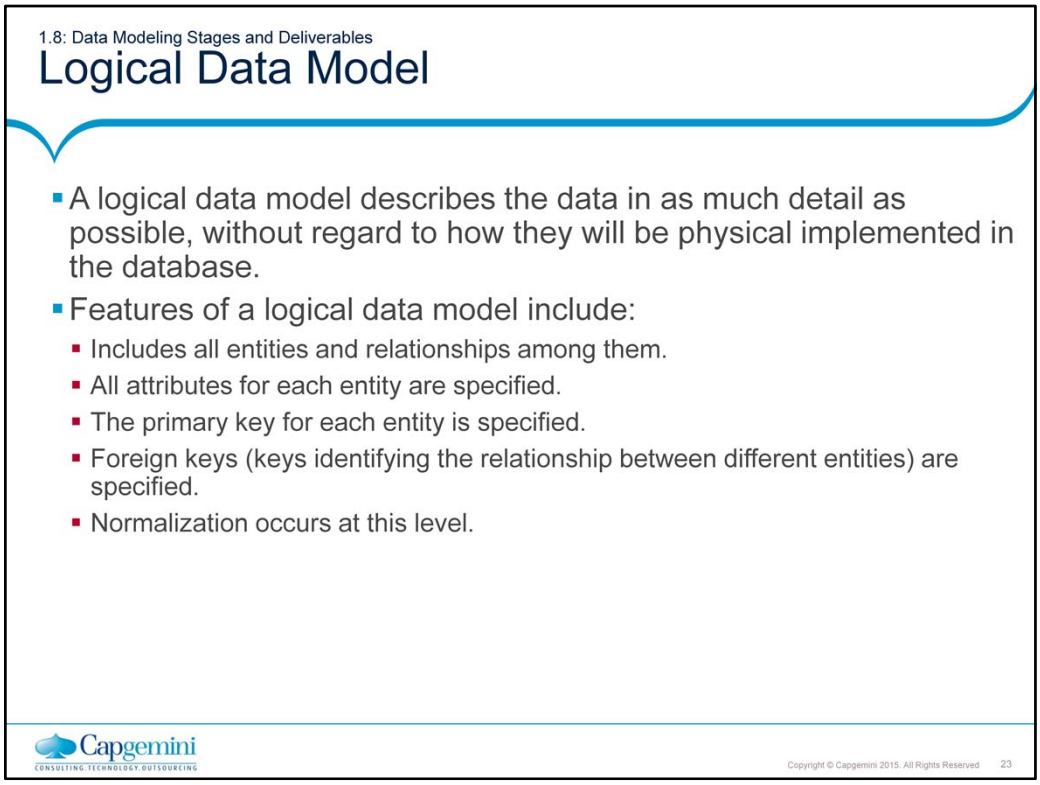
- A conceptual data model identifies the highest-level relationships between the different entities
  - Features of conceptual data model include:
  - Includes the important entities and the relationships among them.
  - No attribute is specified.
  - No primary key is specified



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 22

From the figure above, we can see that the only information shown via the conceptual data model is the entities that describe the data and the relationships between those entities. No other information is shown through the conceptual data model.



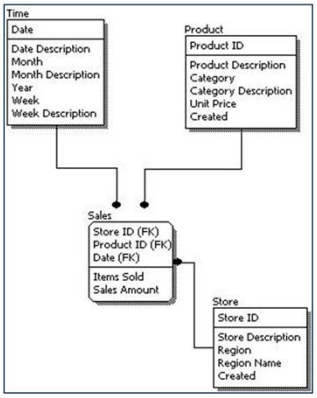
Comparing the logical data model shown above with the conceptual data model diagram, we see the main differences between the two:

- In a logical data model, primary keys are present, whereas in a conceptual data model, no primary key is present.
- In a logical data model, all attributes are specified within an entity. No attributes are specified in a conceptual data model.
- Relationships between entities are specified using primary keys and foreign keys in a logical data model. In a conceptual data model, the relationships are simply stated, not specified, so we simply know that two entities are related, but we do not specify what attributes are used for this relationship.

1.8: Data Modeling Stages and Deliverables

Logical Data Model (contd..)

- The steps for designing the logical data model are as follows:
  - Specify primary keys for all entities.
  - Find the relationships between different entities.
  - Find all attributes for each entity.
  - Resolve many-to-many relationships.
  - Normalization.



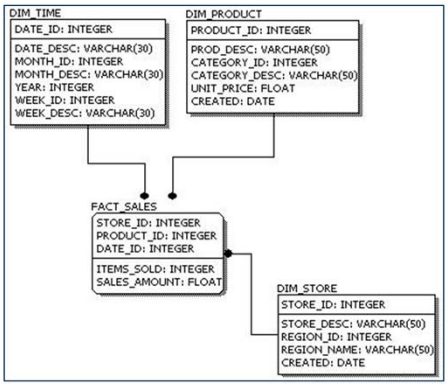
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 24

1.8: Data Modeling Stages and Deliverables

Physical Data Model (contd..)

- Physical data model represents how the model will be built in the database.
- A physical database model shows all table structures, including column name, column data type, column constraints, primary key, foreign key, and relationships between tables.



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 25

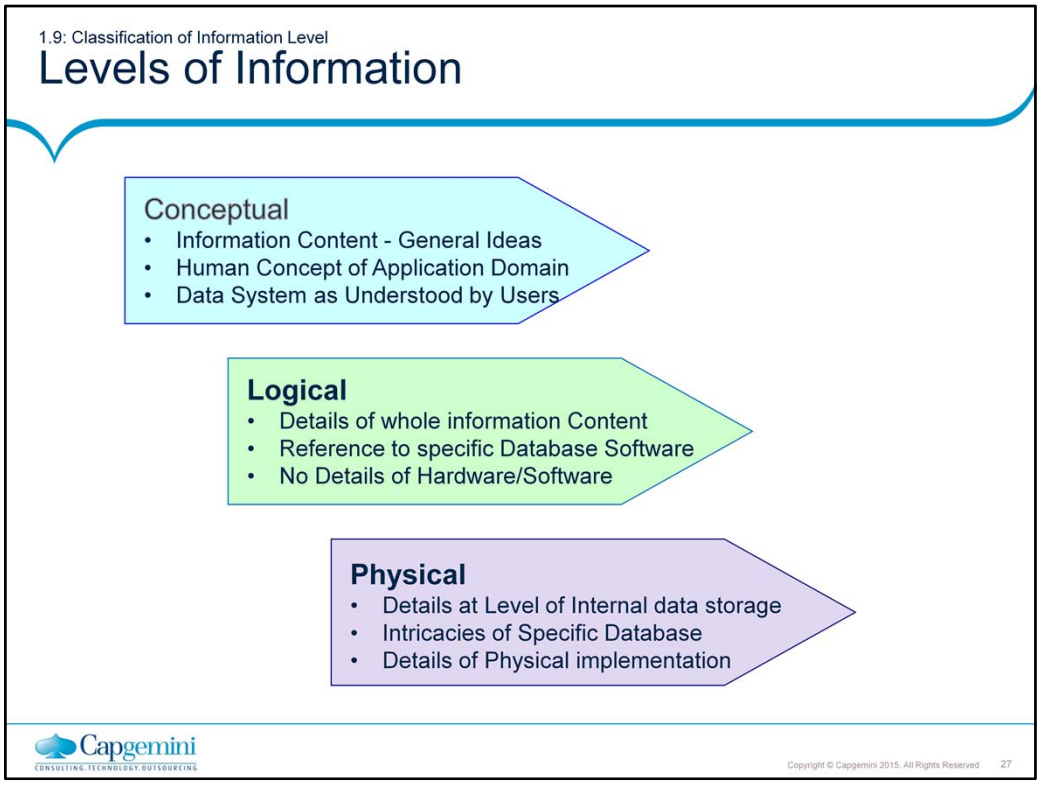
1.8: Data Modeling Stages and Deliverables

Physical Data Model (contd..)

- Features of a physical data model include:
  - Specification of all tables and columns.
  - Foreign keys are used to identify relationships between tables.
  - Demoralization may occur based on user requirements.
  - Physical considerations may cause the physical data model to be quite different from the logical data model.
  - Physical data model will be different for different RDBMS. For example, data type for a column may be different between MySQL and SQL Server

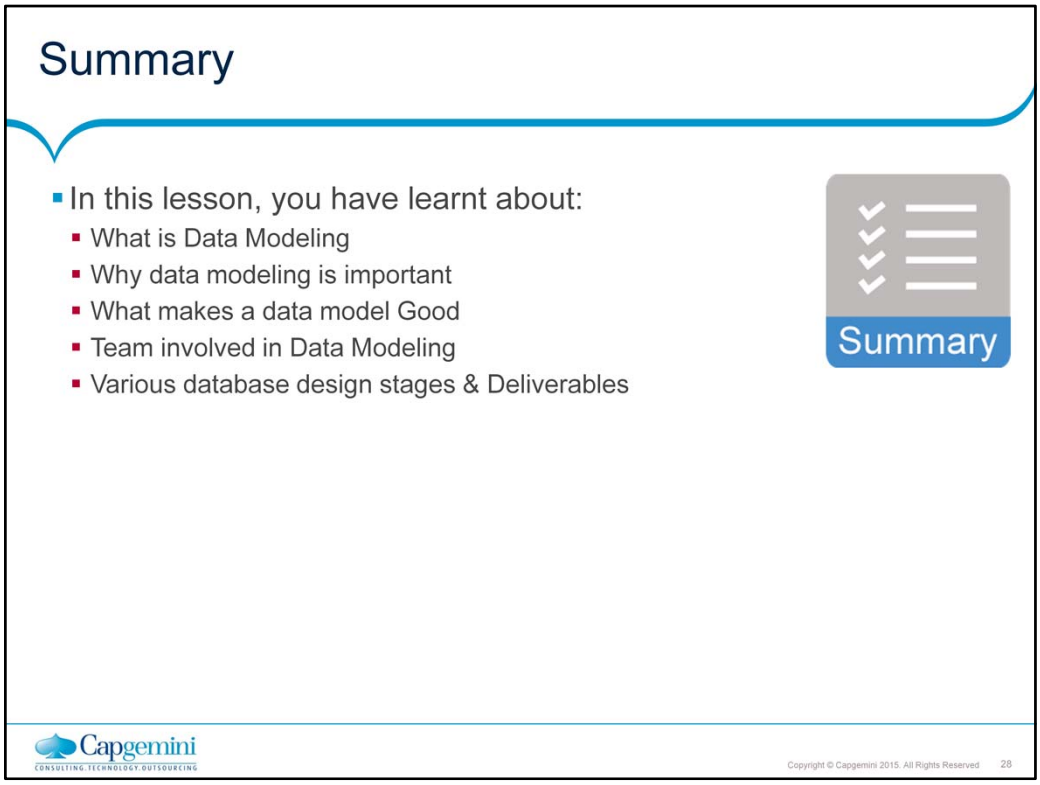
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 26



Levels of Information:

- **Conceptual Level:** This is the highest level consisting of general ideas about the information content. At the conceptual level, the data model represents the information requirements of the entire set of user groups in the organization. At this level, you have the description of application domain in terms of human concepts. This is the level at which the users are able to understand the data system. This is a comprehensive, complete and stable information level.
- **Logical Level:** At this level, the domain concepts and their relationships are explored further. This level accommodates more details about the information content. Still, storage and physical considerations are not part of this level. Considerations of a specific DBMS **may not** find a place at this level.
- **Internal or Physical Level:** This information level deals with the implementation of the database on secondary storage. Considerations of storage management, access management, and database performance apply at this level. Here intricacy and complex details of the particular database are relevant. The intricacies of the particular DBMS are taken into account at the physical level.



Add the notes here.

Review Question

- Question 1: _____ is a replica or a representation of particular aspects and segments of the real world.
- Question 2: _____ data model represents how the model will be built in the database.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 29