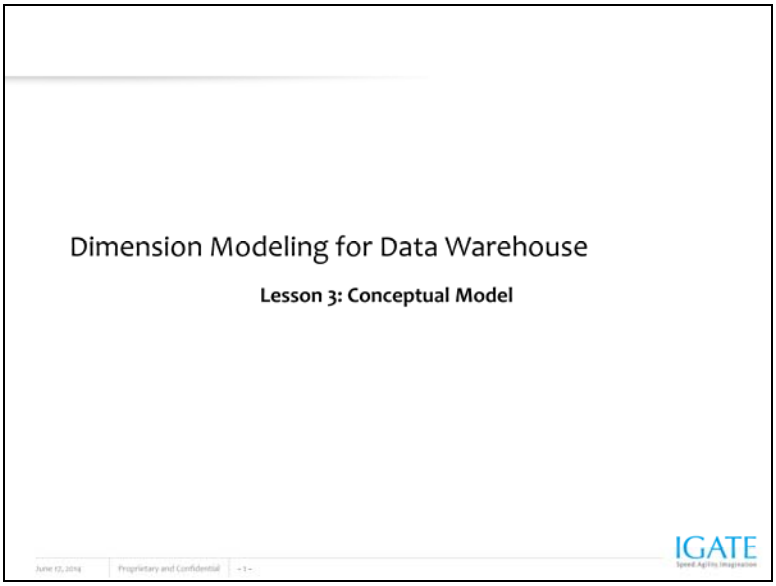


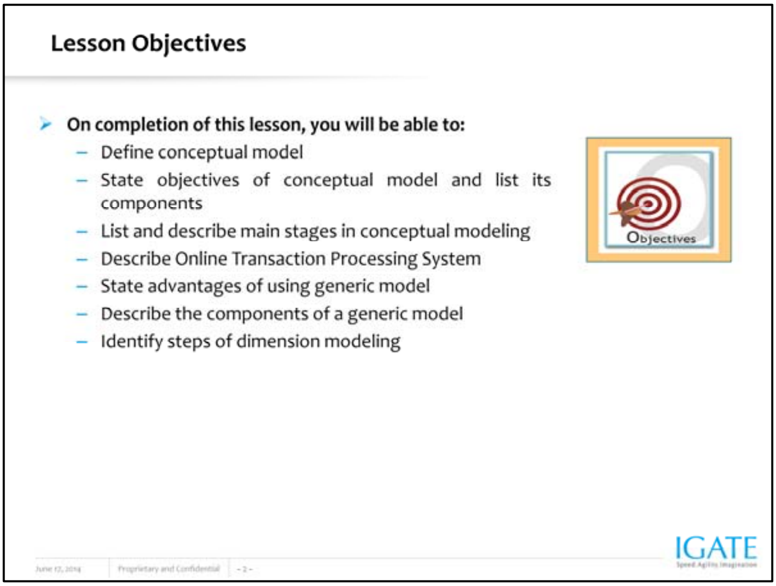
Dimension Modeling for Data Warehouse

Lesson 3: Conceptual Model

June 17, 2014 | Proprietary and Confidential | - 1 -

IGATE

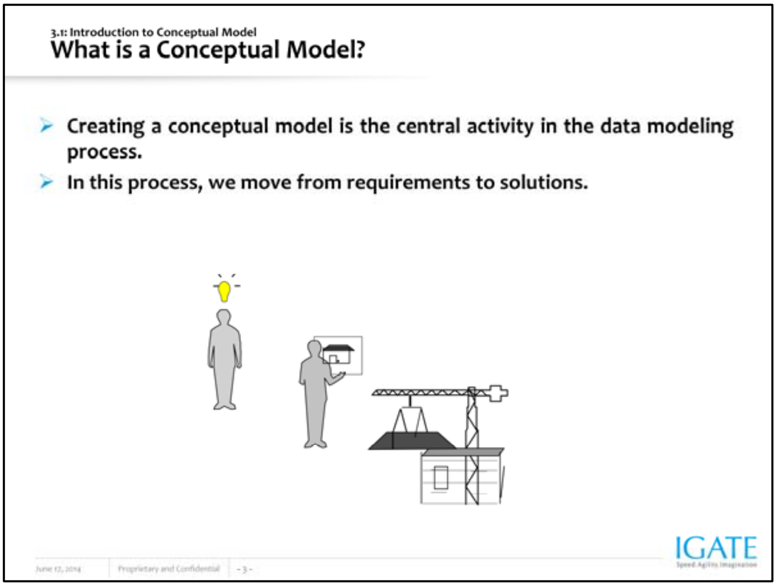
Speed Agility Imagination



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

June 17, 2014 | Proprietary and Confidential | - 2 -



3.1: Introduction to Conceptual Model

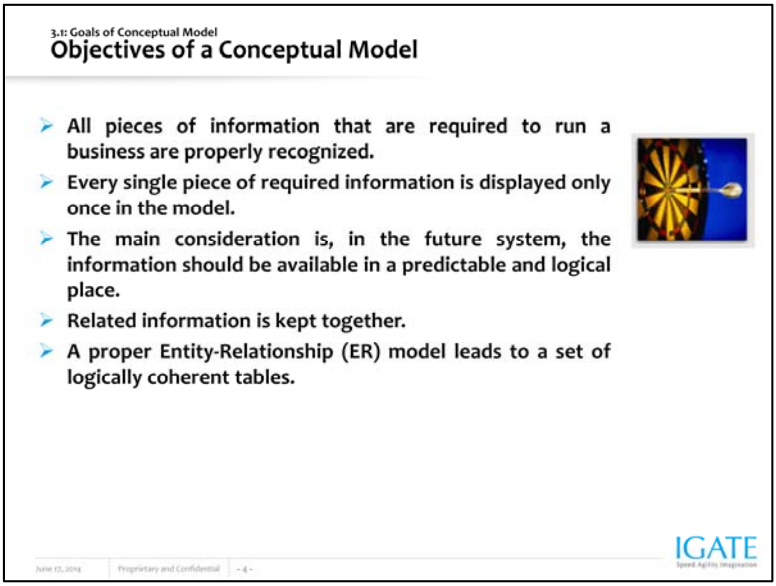
What is a Conceptual Model?

- Creating a conceptual model is the central activity in the data modeling process.
- In this process, we move from requirements to solutions.

June 17, 2014 | Proprietary and Confidential | - 3 -

IGATE
Speed Agility Imagination

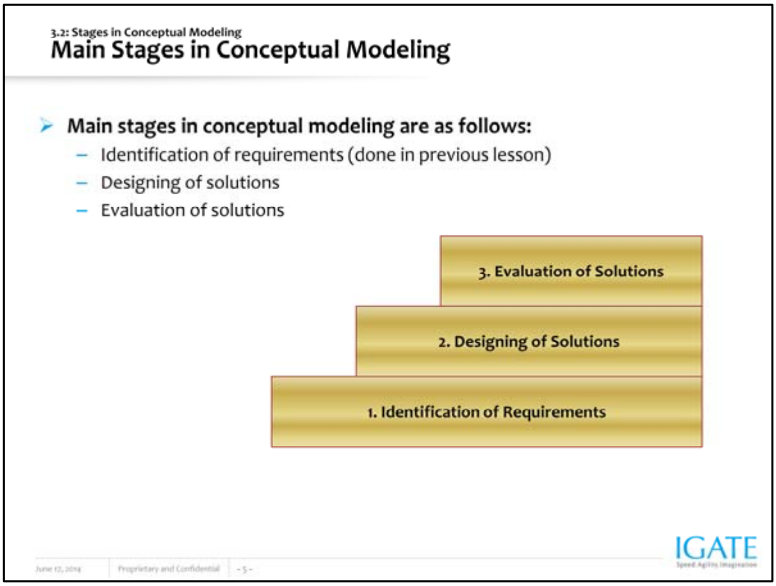
Conceptual data model includes all major entities and relationships and does not contain much detailed level of information about attributes and is often used in the INITIAL PLANNING PHASE. Conceptual data model is created by gathering business requirements from various sources like business documents, 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.



3.2: Stages in Conceptual Modeling

Main Stages in Conceptual Modeling

- Main stages in conceptual modeling are as follows:
  - Identification of requirements (done in previous lesson)
  - Designing of solutions
  - Evaluation of solutions

3. Evaluation of Solutions
2. Designing of Solutions
1. Identification of Requirements

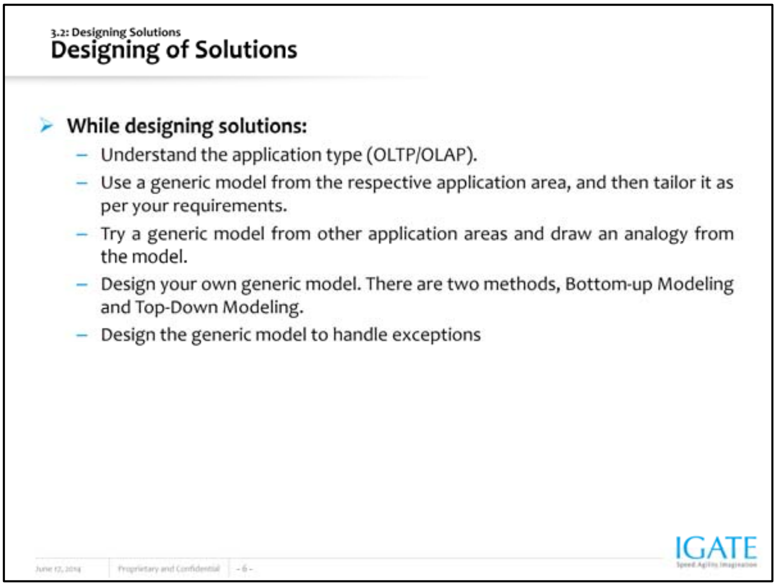
June 17, 2014 | Proprietary and Confidential | - 5 -

IGATE
Speed. Agility. Imagination

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.



3.2: Designing Solutions

Designing of Solutions

- **While designing solutions:**
  - Understand the application type (OLTP/OLAP).
  - Use a generic model from the respective application area, and then tailor it as per your requirements.
  - Try a generic model from other application areas and draw an analogy from the model.
  - Design your own generic model. There are two methods, Bottom-up Modeling and Top-Down Modeling.
  - Design the generic model to handle exceptions

June 17, 2014 | Proprietary and Confidential | - 6 -

IGATE
Speed Agility Imagination

Designing of Solutions:

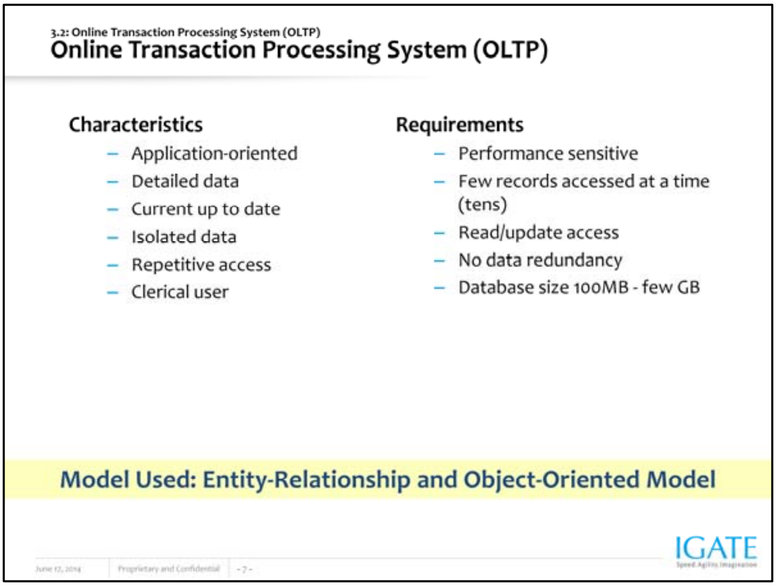
While designing, usually try to find a generic model that broadly meets the users' requirements, and then tailor it to suit a particular application, drawing on standard structures and adapting structures from other models as opportunities arise.

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

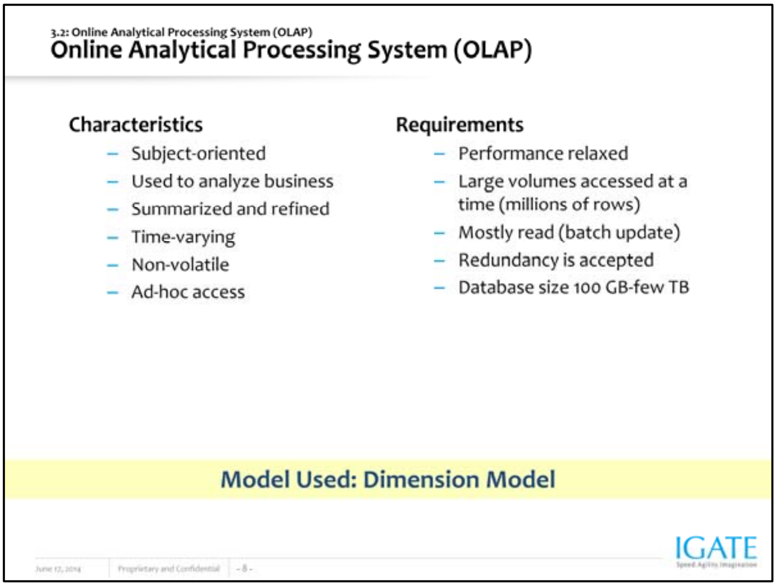
- Application-oriented
- Detailed data
- Current up to date
- Isolated data
- Repetitive access
- Clerical user

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

IGATE
Speed Agility Imagination



3.2: Online Analytical Processing System (OLAP)

Online Analytical Processing System (OLAP)

Characteristics

- Subject-oriented
- Used to analyze business
- Summarized and refined
- Time-varying
- Non-volatile
- Ad-hoc access

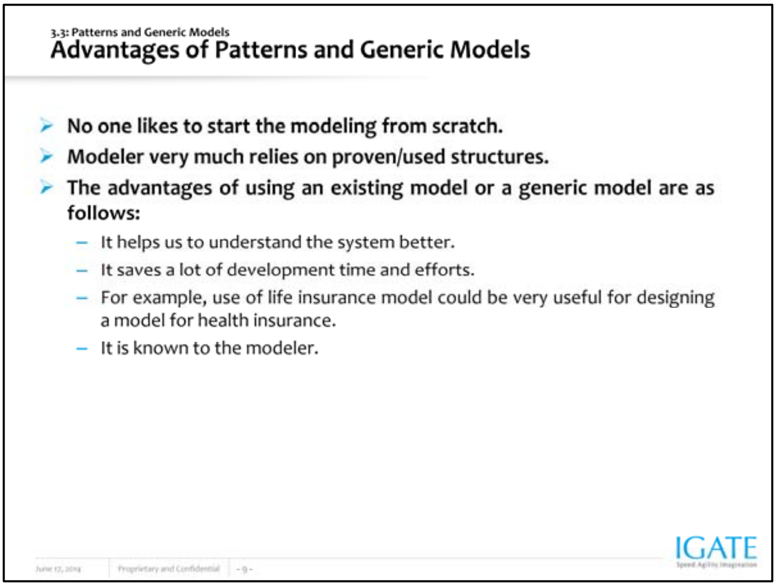
Requirements

- Performance relaxed
- Large volumes accessed at a time (millions of rows)
- Mostly read (batch update)
- Redundancy is accepted
- Database size 100 GB-few TB

Model Used: Dimension Model

June 17, 2014 | Proprietary and Confidential | - 8 -

IGATE
Speed. Agility. Imagination



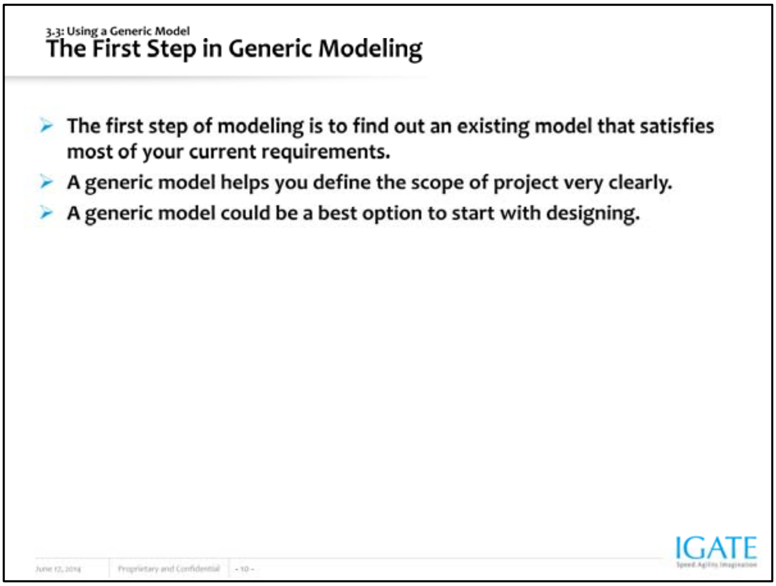
3.3: Patterns and Generic Models

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.

June 17, 2014 Proprietary and Confidential - 9 -

IGATE
Speed Agility Imagination



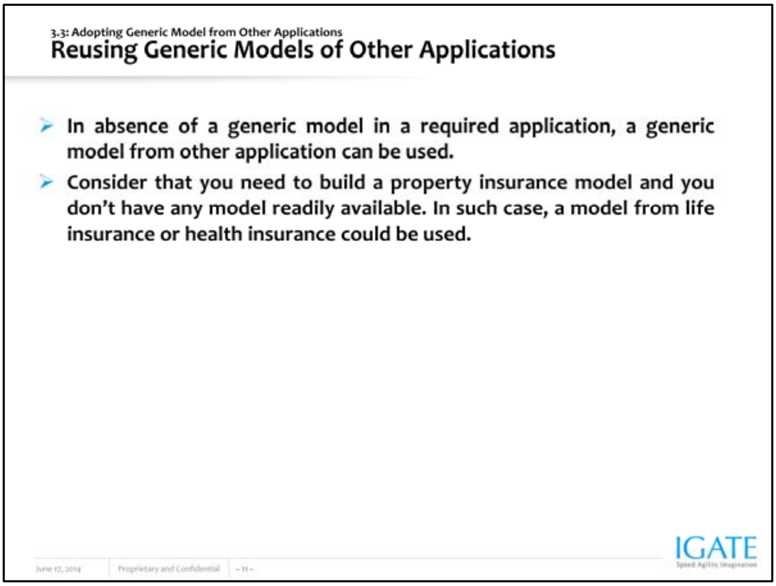
3.3: Using a Generic Model

The First Step in Generic Modeling

- The first step of modeling is to find out an existing model that satisfies most of your current requirements.
- A generic model helps you define the scope of project very clearly.
- A generic model could be a best option to start with designing.

June 17, 2014 | Proprietary and Confidential | - 10 -

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.

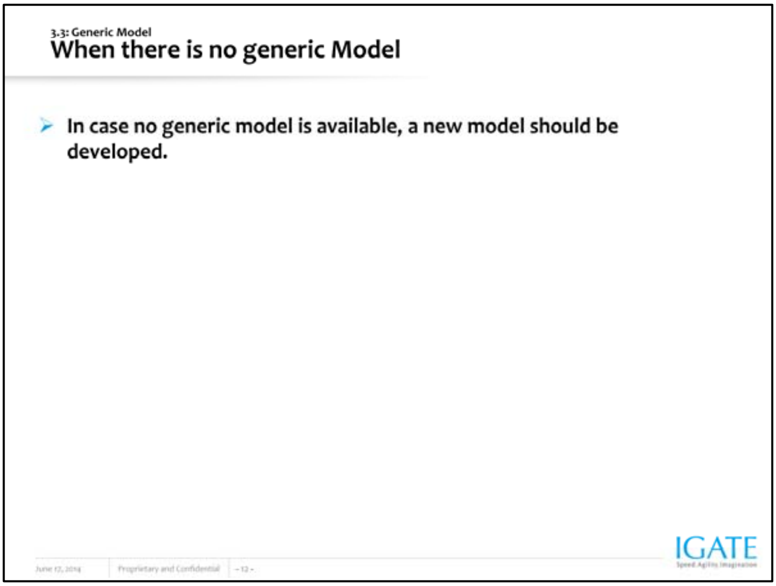


3.3: Adopting Generic Model from Other Applications

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.

IGATE
Speed Agility Imagination



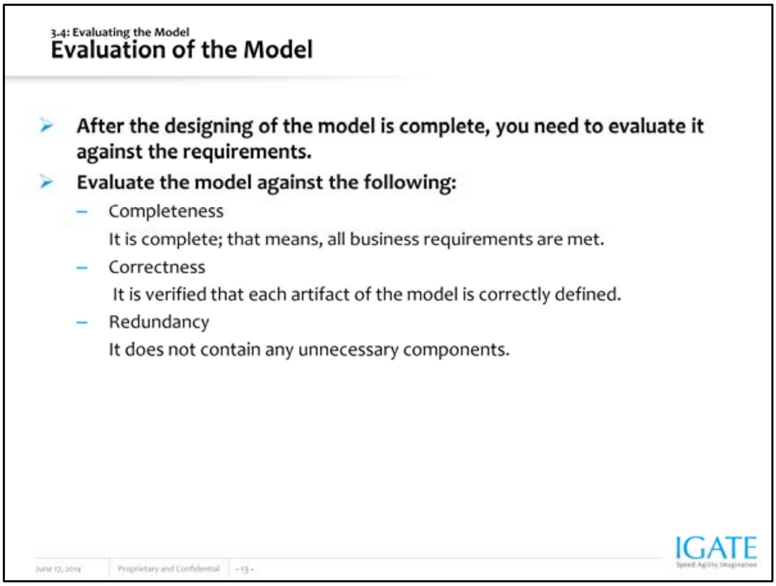
3.3: Generic Model

When there is no generic Model

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

June 17, 2014 | Proprietary and Confidential | - 12 -

IGATE
Speed. Agility. Imagination



3.4: Evaluating the Model

Evaluation of the Model

- After the designing of the model is complete, you need to evaluate it against the requirements.
- Evaluate the model against the following:
  - Completeness

    It is complete; that means, all business requirements are met.
  - Correctness

    It is verified that each artifact of the model is correctly defined.
  - Redundancy

    It does not contain any unnecessary components.

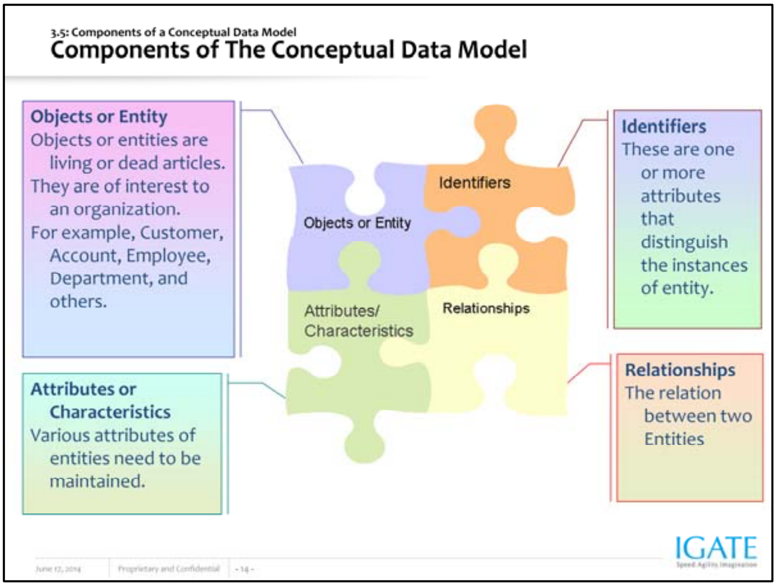
June 17, 2014 | Proprietary and Confidential | - 13 -

IGATE

Speed Agility Imagination

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.



3.5: Components of a Conceptual Data Model

Components of The Conceptual Data Model

Objects or Entity
Objects or entities are living or dead articles.
They are of interest to an organization.
For example, Customer, Account, Employee, Department, and others.

Attributes or Characteristics
Various attributes of entities need to be maintained.

Identifiers
These are one or more attributes that distinguish the instances of entity.

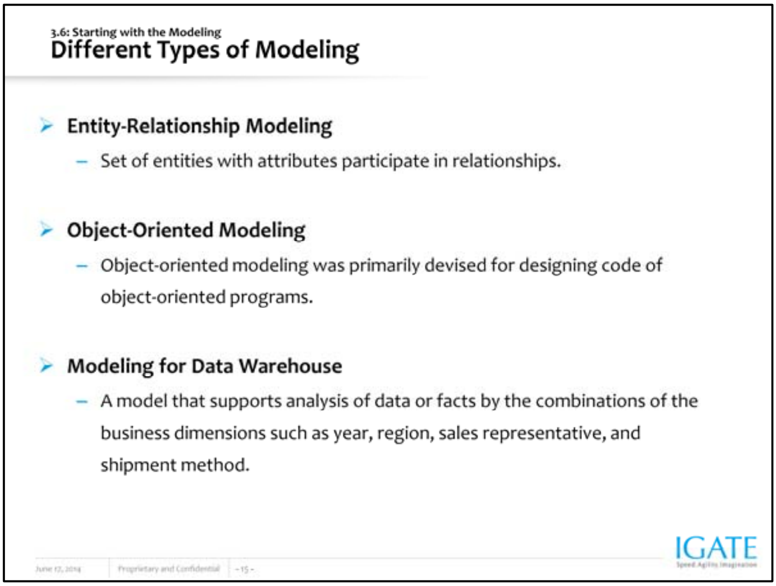
Relationships
The relation between two Entities

June 17, 2014 | Proprietary and Confidential | - 14 -

IGATE
Speed Agility Imagination

<u>Components of The Conceptual Data Model:</u>

- **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.



3.6: Starting with the Modeling

Different Types of Modeling

- Entity-Relationship Modeling
  - Set of entities with attributes participate in relationships.
- Object-Oriented Modeling
  - Object-oriented modeling was primarily devised for designing code of object-oriented programs.
- Modeling for Data Warehouse
  - A model that supports analysis of data or facts by the combinations of the business dimensions such as year, region, sales representative, and shipment method.

June 17, 2014 | Proprietary and Confidential | - 15 -

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.

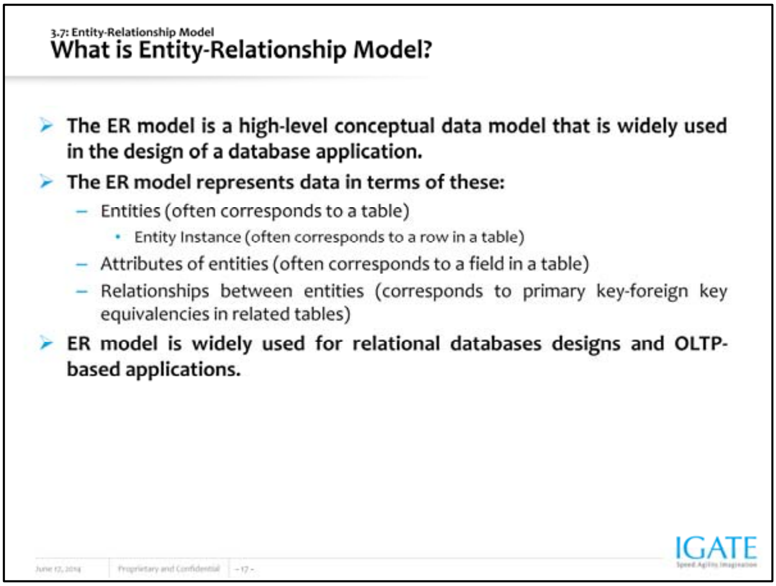
Object-Oriented Modeling: In this approach, both data and behavior are encapsulated within objects. Thus, object-oriented modeling was primarily devised for designing code of object-oriented programs. However, this modeling approach can be adapted for conceptual modeling and eventually for database design.

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.



3.7: Entity-Relationship Model

What is Entity-Relationship Model?

- The ER model is a high-level conceptual data model that is widely used in the design of a database application.
- The ER model represents data in terms of these:
  - Entities (often corresponds to a table)
    - Entity Instance (often corresponds to a row in a table)
  - Attributes of entities (often corresponds to a field in a table)
  - Relationships between entities (corresponds to primary key-foreign key equivalencies in related tables)
- ER model is widely used for relational databases designs and OLTP-based applications.

June 17, 2014 | Proprietary and Confidential | - 17 -

IGATE
Speed Agility Imagination

<u>What is Entity-Relationship Model?</u>

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

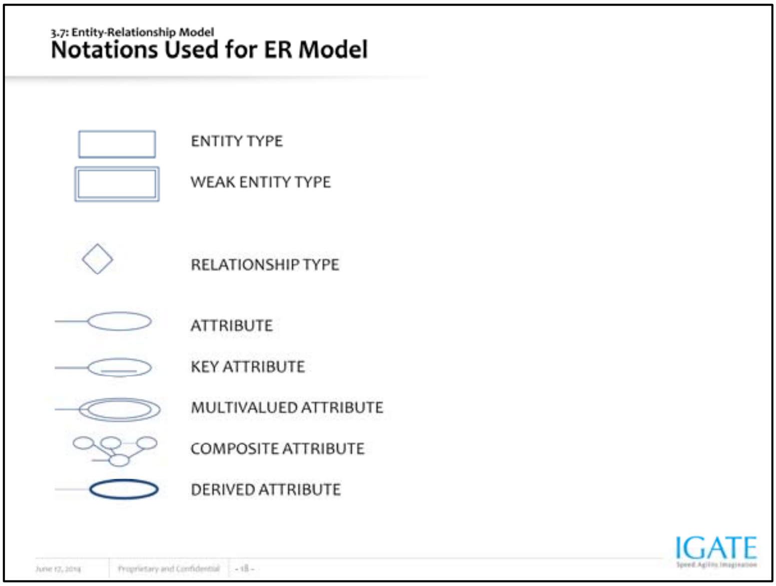
Attributes describe the entity with which they are associated. A particular instance of an attribute is a value. The domain of an attribute is the collection of all possible values an attribute can have. The domain of Name is a character string.

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)



3.7: Entity-Relationship Model

Notations Used for ER Model

ENTITY TYPE

WEAK ENTITY TYPE

RELATIONSHIP TYPE

ATTRIBUTE

KEY ATTRIBUTE

MULTIVALUED ATTRIBUTE

COMPOSITE ATTRIBUTE

DERIVED ATTRIBUTE

June 17, 2014 | Proprietary and Confidential | - 18 -

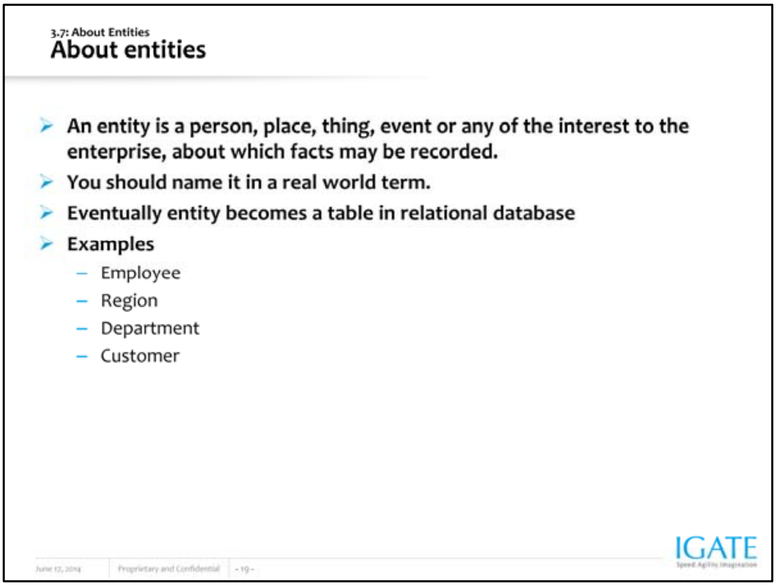
<u>**Notations Used for ER Model:**</u>

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

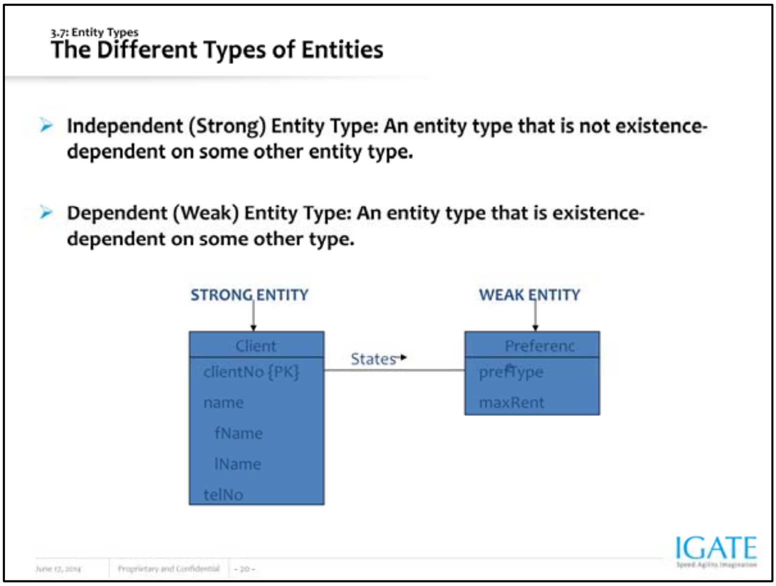
About entities

- 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

June 17, 2014 | Proprietary and Confidential | - 19 -

IGATE

Speed. Agility. Imagination



3.7: Entity Types

The Different Types of Entities

- Independent (Strong) Entity Type: An entity type that is not existence-dependent on some other entity type.
- Dependent (Weak) Entity Type: An entity type that is existence-dependent on some other type.

STRONG ENTITY

WEAK ENTITY

Client

clientNo {PK}

name

fName

lName

telNo

States

Preferenc

prefType

maxRent

June 17, 2014 | Proprietary and Confidential | - 20 -

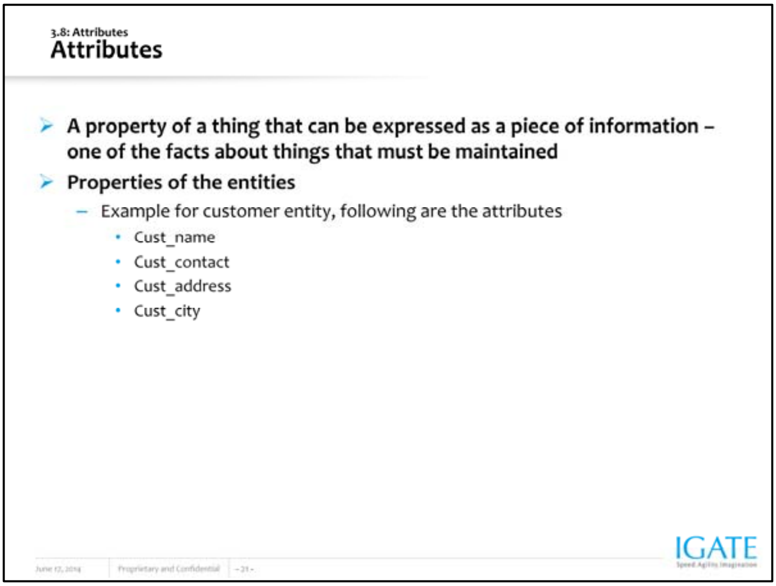
IGATE

Speed Agility Imagination

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.



3.8: Attributes

Attributes

- A property of a thing that can be expressed as a piece of information – one of the facts about things that must be maintained
- Properties of the entities
  - Example for customer entity, following are the attributes
    - Cust_name
    - Cust_contact
    - Cust_address
    - Cust_city

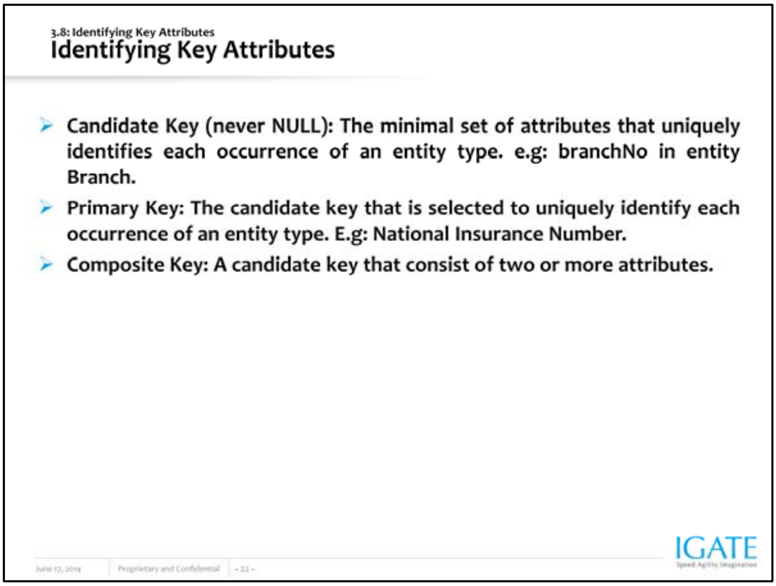
June 17, 2014 | Proprietary and Confidential | - 21 -

IGATE
Speed. Agility. Imagination

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.



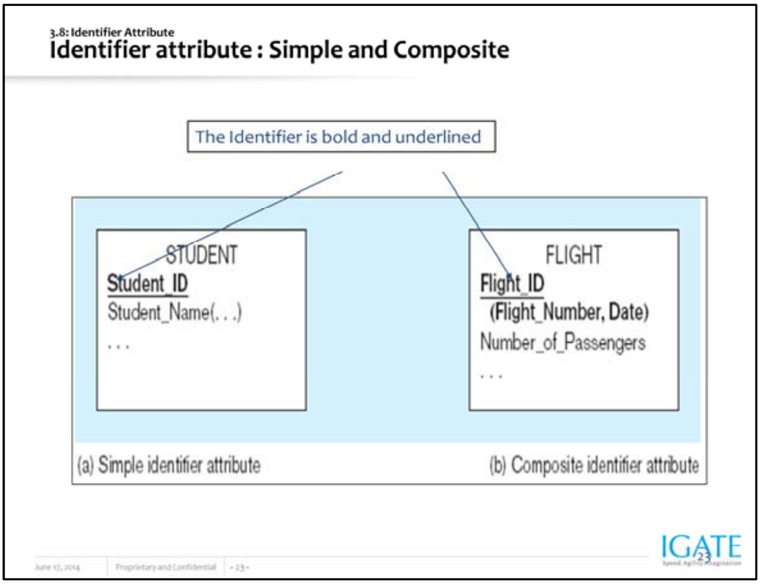
3.8: Identifying Key Attributes

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.

June 17, 2014 | Proprietary and Confidential | - 22 -

IGATE
Speed Agility Imagination



3.8: Identifier Attribute

Identifier attribute : Simple and Composite

The Identifier is bold and underlined

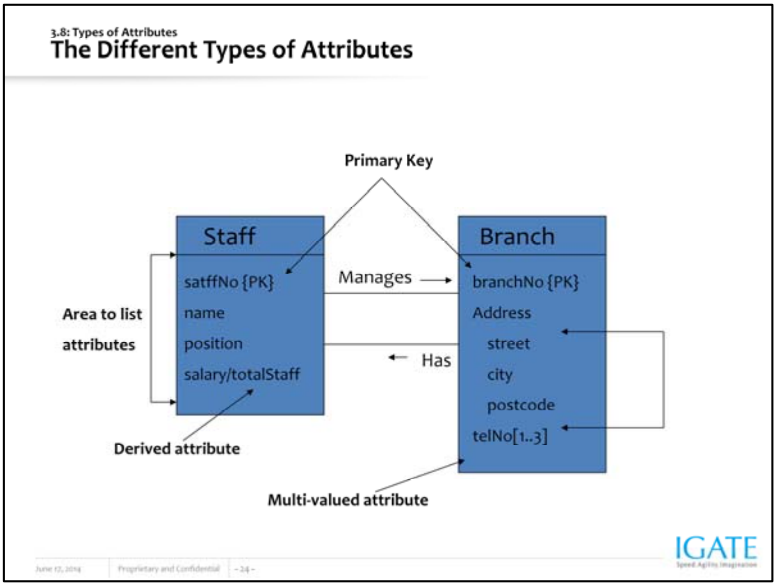
(a) Simple identifier attribute

(b) Composite identifier attribute

June 17, 2014 | Proprietary and Confidential | - 23 -

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



3.8: Types of Attributes

The Different Types of Attributes

Primary Key

Staff

satffNo {PK}

name

position

salary/totalStaff

Manages →

← Has

Branch

branchNo {PK}

Address

street

city

postcode

telNo[1..3]

Area to list attributes

Derived attribute

Multi-valued attribute

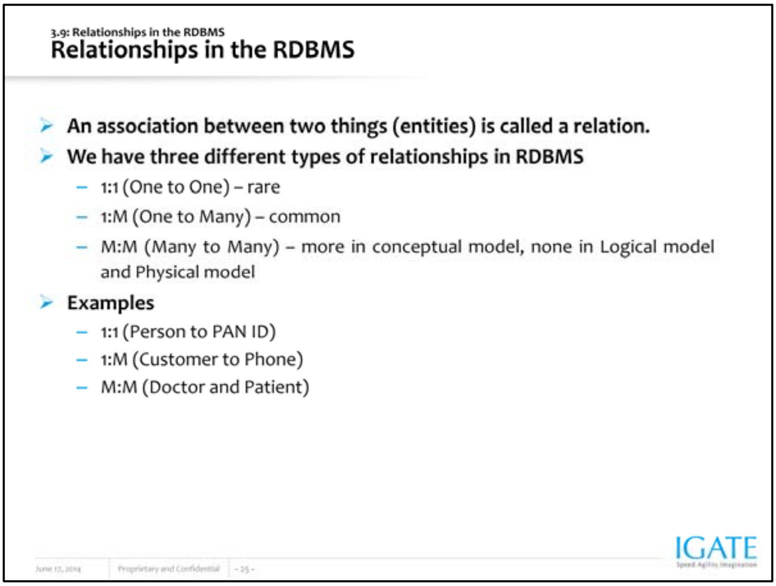
June 17, 2014 | Proprietary and Confidential | - 24 -

IGATE Speed. Agility. Imagination

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.

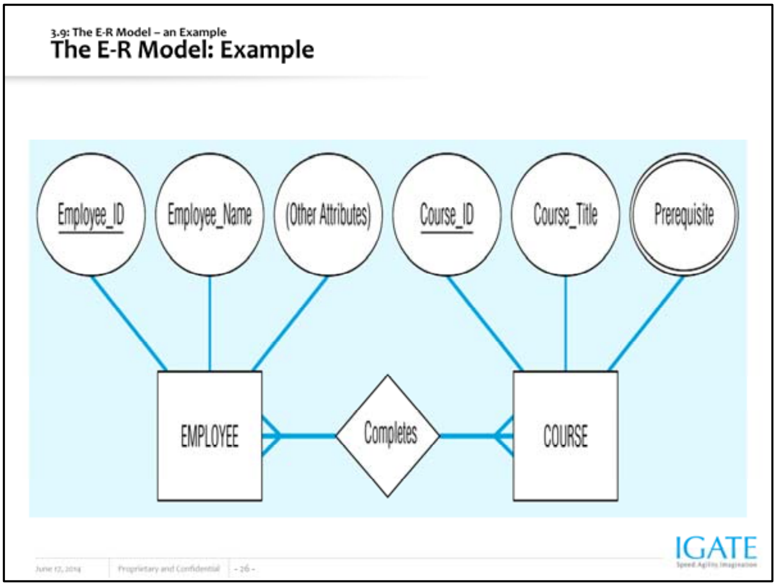


3.9: Relationships in the RDBMS

Relationships in the RDBMS

- An association between two things (entities) is called a relation.
- 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)

June 17, 2014 | Proprietary and Confidential | - 25 -



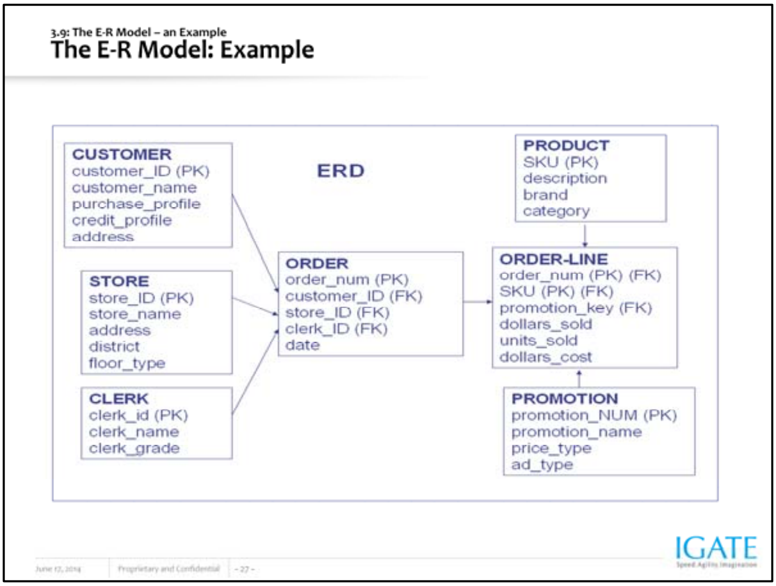
3.9: The E-R Model – an Example

The E-R Model: Example

IGATE

Speed. Agility. Imagination

June 17, 2014 Proprietary and Confidential - 26 -



3.9: The E-R Model – an Example

The E-R Model: Example

ERD

CUSTOMER
customer_ID (PK)
customer_name
purchase_profile
credit_profile
address

STORE
store_ID (PK)
store_name
address
district
floor_type

CLERK
clerk_id (PK)
clerk_name
clerk_grade

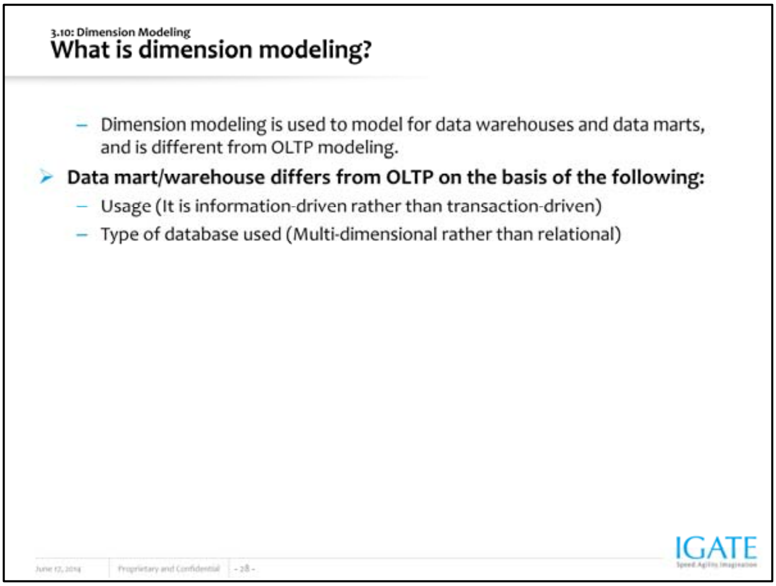
ORDER
order_num (PK)
customer_ID (FK)
store_ID (FK)
clerk_ID (FK)
date

PRODUCT
SKU (PK)
description
brand
category

ORDER-LINE
order_num (PK) (FK)
SKU (PK) (FK)
promotion_key (FK)
dollars_sold
units_sold
dollars_cost

PROMOTION
promotion_NUM (PK)
promotion_name
price_type
ad_type

June 17, 2014 | Proprietary and Confidential | - 27 -

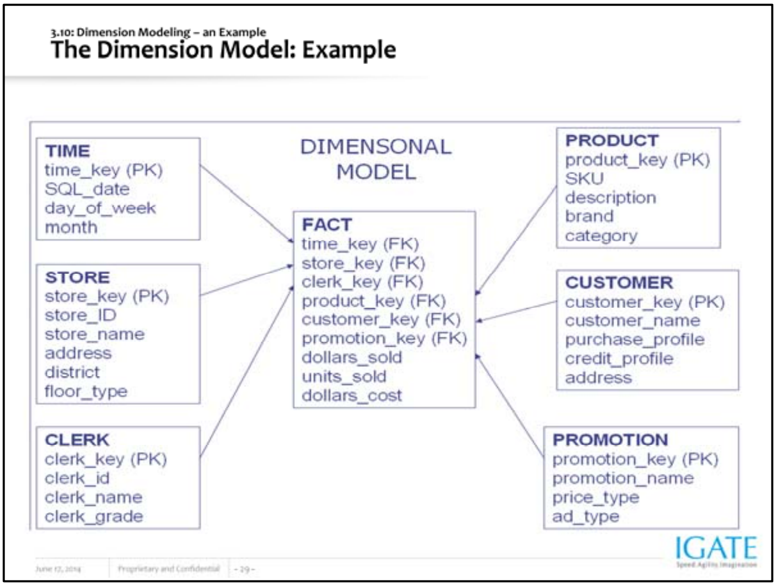


3.10: Dimension Modeling

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)

June 17, 2014 | Proprietary and Confidential | - 28 -



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

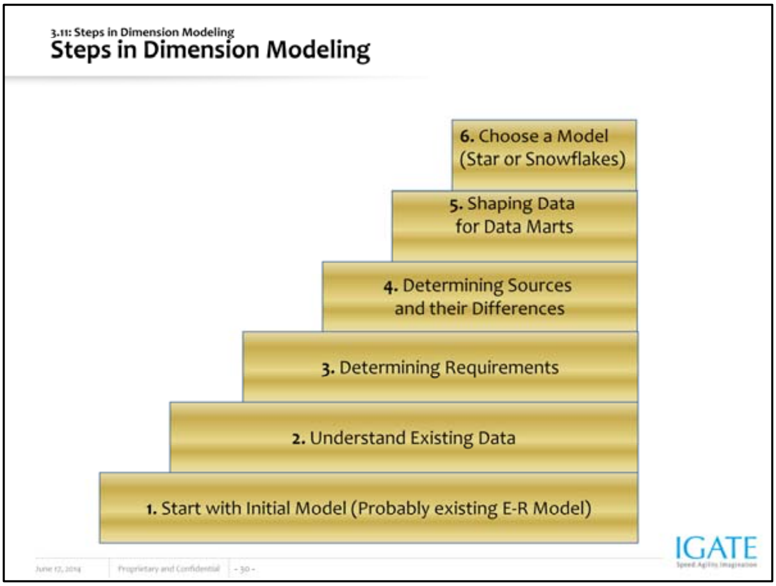
FACT
time_key (FK)
store_key (FK)
clerk_key (FK)
product_key (FK)
customer_key (FK)
promotion_key (FK)
dollars_sold
units_sold
dollars_cost

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

June 17, 2014 | Proprietary and Confidential | - 29 -

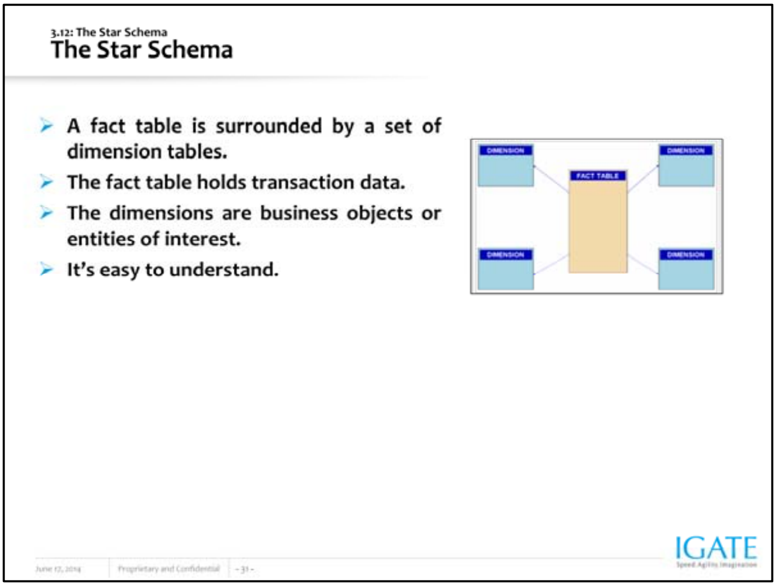


3.11: Steps in Dimension Modeling

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)

June 17, 2014 | Proprietary and Confidential | - 30 -



3.12: The Star Schema

The Star Schema

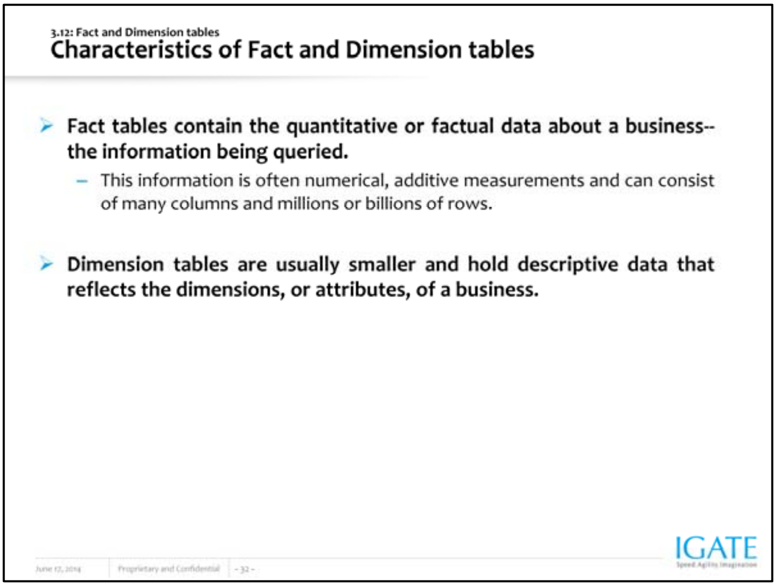
- A fact table is surrounded by a set of dimension tables.
- The fact table holds transaction data.
- The dimensions are business objects or entities of interest.
- It's easy to understand.

June 17, 2014 | Proprietary and Confidential | - 31 -

IGATE
Speed Agility Imagination

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.



3.12: Fact and Dimension tables

Characteristics of Fact and Dimension tables

- Fact tables contain the quantitative or factual data about a business--the information being queried.
  - This information is often numerical, additive measurements and can consist of many columns and millions or billions of rows.
- Dimension tables are usually smaller and hold descriptive data that reflects the dimensions, or attributes, of a business.

June 17, 2014 | Proprietary and Confidential | - 32 -

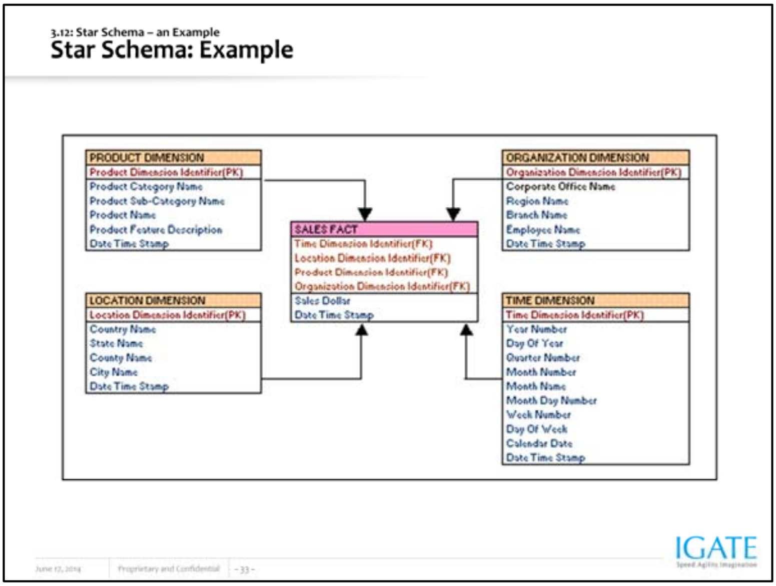
IGATE
Speed Agility Imagination

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

PRODUCT DIMENSION
Product Dimension Identifier(PK)
Product Category Name
Product Sub-Category Name
Product Name
Product Feature Description
Date Time Stamp

ORGANIZATION DIMENSION
Organization Dimension Identifier(PK)
Corporate Office Name
Region Name
Branch Name
Employee Name
Date Time Stamp

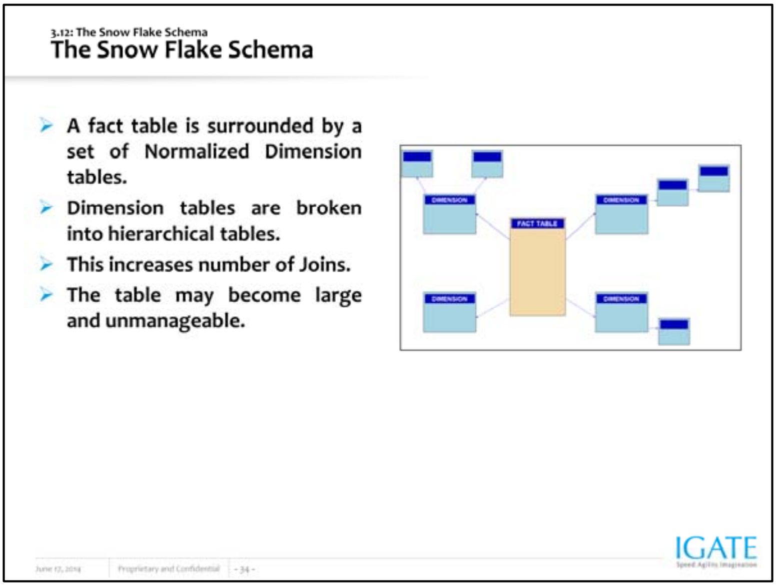
SALES FACT
Time Dimension Identifier(FK)
Location Dimension Identifier(FK)
Product Dimension Identifier(FK)
Organization Dimension Identifier(FK)
Sales Dollar
Date Time Stamp

LOCATION DIMENSION
Location Dimension Identifier(PK)
Country Name
State Name
County Name
City Name
Date Time Stamp

TIME DIMENSION
Time Dimension Identifier(PK)
Year Number
Day Of Year
Quarter Number
Month Number
Month Name
Month Day Number
Week Number
Day Of Week
Calendar Date
Date Time Stamp

June 17, 2014 | Proprietary and Confidential | - 33 -

IGATE
Speed Agility Imagination



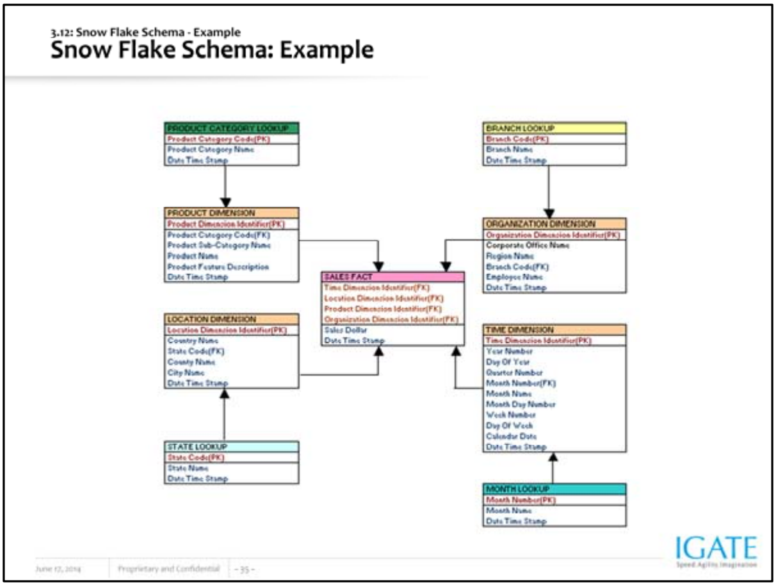
3.12: The Snow Flake Schema

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.

June 17, 2014 | Proprietary and Confidential | - 34 -

IGATE
Speed. Agility. Imagination



3.12: Snow Flake Schema - Example

Snow Flake Schema: Example

PRODUCT CATEGORY LOOKUP
Product Category Code(PK)
Product Category Name
Date Time Stamp

PRODUCT DIMENSION
Product Dimension Identifier(PK)
Product Category Code(FK)
Product Sub-Category Name
Product Name
Product Feature Description
Date Time Stamp

LOCATION DIMENSION
Location Dimension Identifier(PK)
Country Name
State Code(FK)
County Name
City Name
Date Time Stamp

STATE LOOKUP
State Code(PK)
State Name
Date Time Stamp

SALES FACT
Time Dimension Identifier(FK)
Location Dimension Identifier(FK)
Product Dimension Identifier(FK)
Organization Dimension Identifier(FK)
Sales Dollar
Date Time Stamp

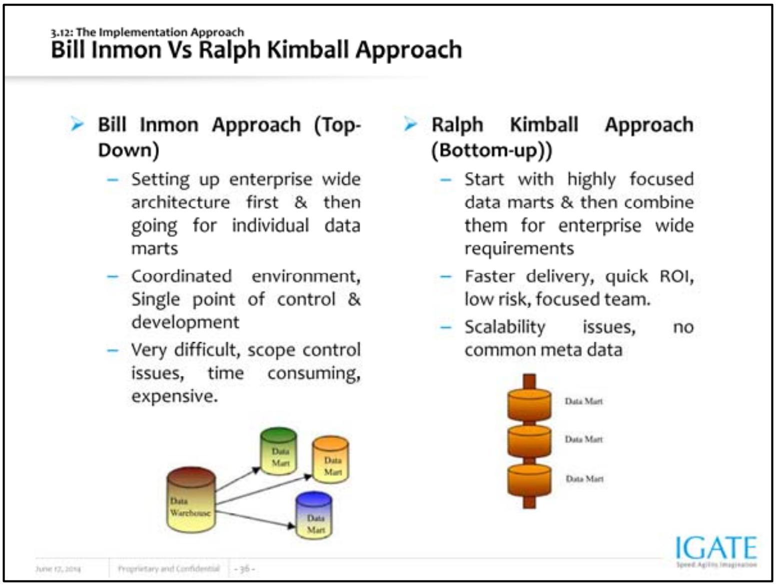
BRANCH LOOKUP
Branch Code(PK)
Branch Name
Date Time Stamp

ORGANIZATION DIMENSION
Organization Dimension Identifier(PK)
Corporate Office Name
Region Name
Branch Code(FK)
Employee Name
Date Time Stamp

TIME DIMENSION
Time Dimension Identifier(PK)
Year Number
Day Of Year
Quarter Number
Month Number(FK)
Month Name
Month Day Number
Week Number
Day Of Week
Calendar Date
Date Time Stamp

MONTH LOOKUP
Month Number(PK)
Month Name
Date Time Stamp

June 17, 2014 | Proprietary and Confidential | - 35 -



3.12: The Implementation Approach

Bill Inmon Vs Ralph Kimball Approach

- Bill Inmon Approach (Top-Down)
  - Setting up enterprise wide architecture first & then going for individual data marts
  - Coordinated environment, Single point of control & development
  - Very difficult, scope control issues, time consuming, expensive.
- Ralph Kimball Approach (Bottom-up))
  - Start with highly focused data marts & then combine them for enterprise wide requirements
  - Faster delivery, quick ROI, low risk, focused team.
  - Scalability issues, no common meta data

June 17, 2014 | Proprietary and Confidential | - 36 -

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.

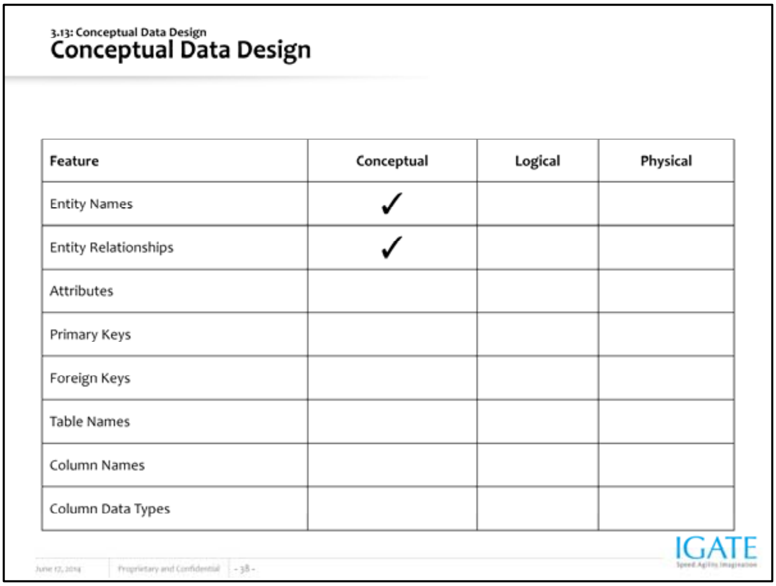
Ralph Kimball Approach

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.



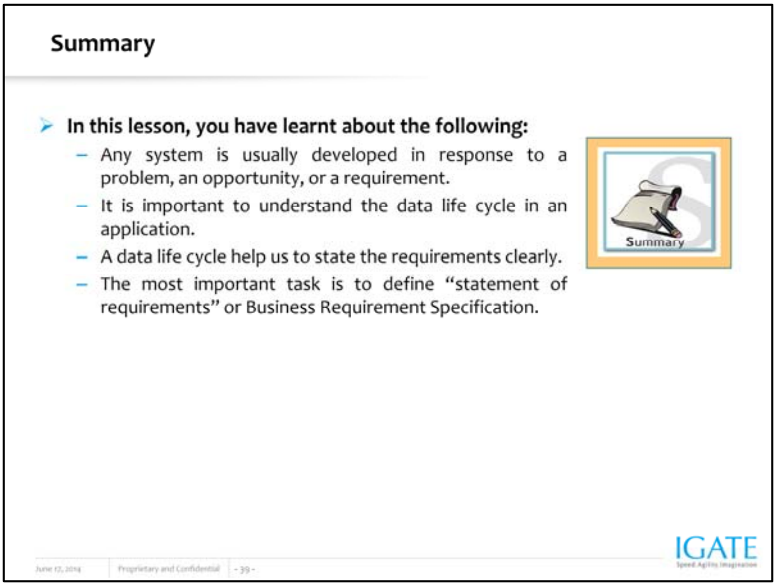
3.13: Conceptual Data Design

Conceptual Data Design

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

June 17, 2014 | Proprietary and Confidential | - 38 -

IGATE
Speed. Agility. Imagination



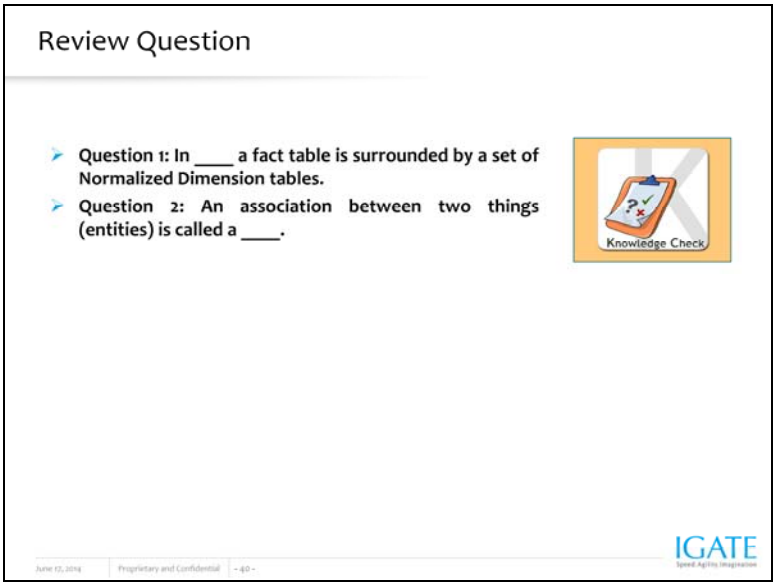
Summary

- In this lesson, you have learnt about the following:
  - Any system is usually developed in response to a problem, an opportunity, or a requirement.
  - It is important to understand the data life cycle in an application.
  - A data life cycle help us to state the requirements clearly.
  - The most important task is to define “statement of requirements” or Business Requirement Specification.

June 17, 2014 | Proprietary and Confidential | - 39 -

IGATE
Speed. Agility. Imagination

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 ____.

Knowledge Check

June 17, 2014 | Proprietary and Confidential | - 40 -

IGATE
Speed. Agility. Imagination