Transforming Data With Intelligence™

Previews of TDWI course books offer an opportunity to see the quality of our material and help you to select the courses that best fit your needs. The previews cannot be printed.

TDWI strives to provide course books that are content-rich and that serve as useful reference documents after a class has ended.

This preview shows selected pages that are representative of the entire course book; pages are not consecutive. The page numbers shown at the bottom of each page indicate their actual position in the course book. All table-of-contents pages are included to illustrate all of the topics covered by the course.

This page intentionally left blank.

TDWI Advanced Data Modeling Techniques

TABLE OF CONTENTS

COURSE OBJECTIVES

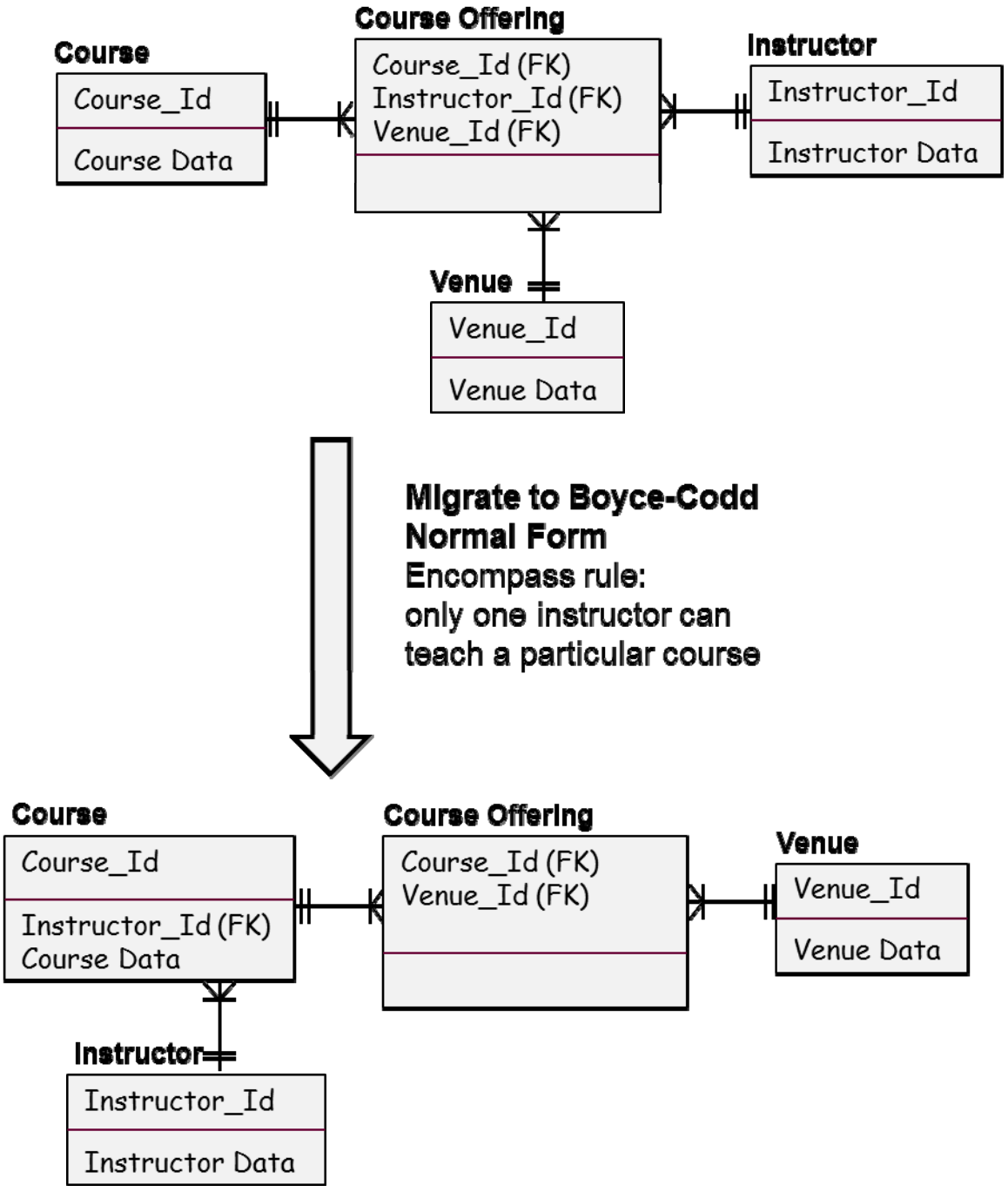
- ✓ ***Enterprise architecture approaches and how to apply them***
- ✓ ***How big data and analytics impact traditional approaches***
- ✓ ***Different data models and how they relate to each other***
- ✓ ***The role of modeling in analytics***
- ✓ ***Higher normalization forms***
- ✓ ***How to effectively apply generalization and specialization***
- ✓ ***The role of metadata management in data governance***
- ✓ ***State and time dependencies and how to handle them***
- ✓ ***How to validate the data model***
- ✓ ***How to transform the business data model into physical models based on the application***
- ✓ ***The implications of alternative storage approaches***
- ✓ ***The roles and structures of complementary models***
- ✓ ***How to deal with multiple time zones and currencies***

Module 1

Data Modeling Concepts

Higher Normal Forms

Boyce-Codd Normal Form



Higher Normal Forms

Boyce-Codd Normal Form

NORMALIZATION

Normalization is a formal, rule-based process of removing redundancy and dependency from data structures. It is performed as a step-by-step process. Normal forms one through three are related to redundancy and dependency of attributes. Normal forms four and five address redundancy and dependency of relationships.

Normalization is performed to optimize the database for update (at the expense of access and query) and to prevent update anomalies. In data warehousing, planned redundancy is often desirable and update anomalies are not a significant issue.

THIRD NORMAL FORM

The term "normalized" generally means that the data structure conforms with "third normal form" or "3NF". In 3NF, each non-key attribute depends on the key (1NF), the whole key (2NF), and nothing but the key (3NF).

HIGHER NORMAL FORMS

The higher normal forms apply in only a small percentage of the time. When they do, they provide additional opportunities to reduce redundancy and increase flexibility. They enable the data structure to enforce additional business rules that are not represented in third normal form.

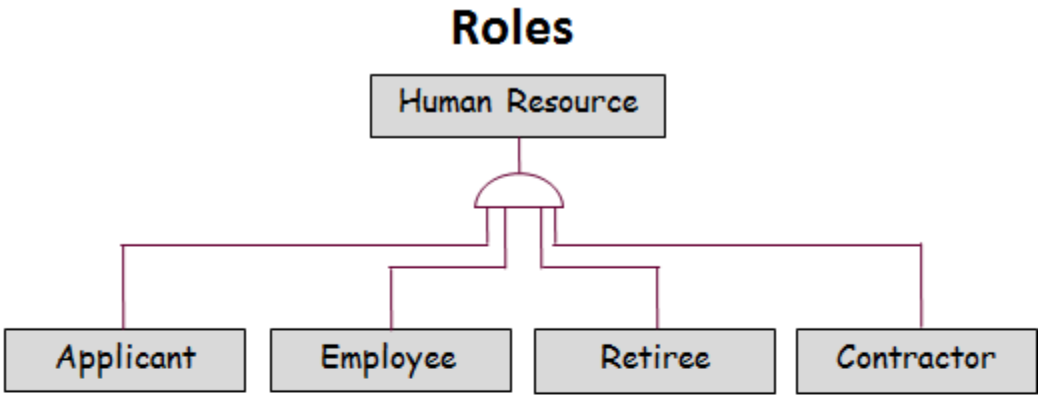
BOYCE-CODD NORMAL FORM

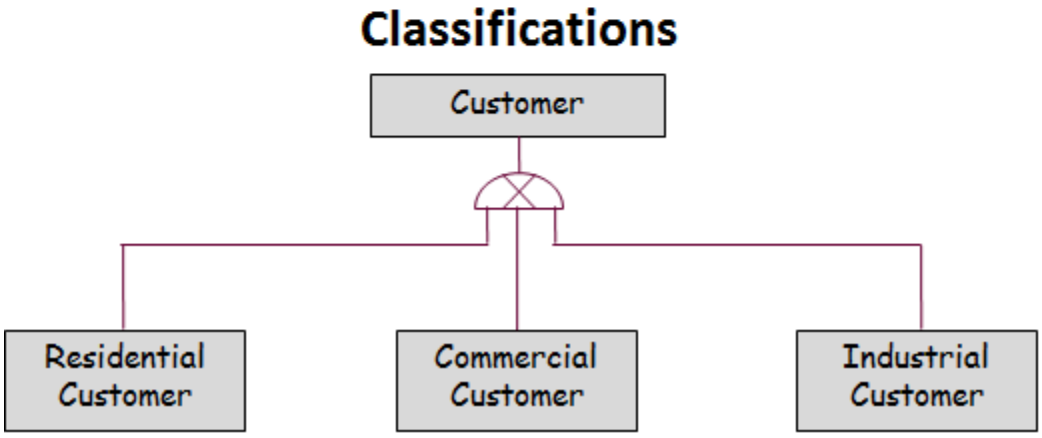
Boyce-Codd Normal Form was developed by Raymond F. Boyce and Edgar F. Codd in 1974. It moves beyond third normal form by applying the rigor that 3NF applies to non-key attributes to all data elements. Hence, composite alternate keys need to be examined to determine if there are any overlaps that need to be eliminated.

In the example shown, the three independent entities, course, venue, and instructor each contain appropriate information pertaining to these individually. When a particular course is delivered, this is done at a particular venue by a particular instructor, resulting in the third normal form structure shown. If there is a business rule that each course has only one instructor eligible to teach it, there are overlapping composite keys in the associative entity – a key consisting solely of course_id and venue_id is sufficient to uniquely identify each instance. With Boyce-Codd normal form, this is resolved by creating the structure shown.

Specialization and Generalization

Roles and Classifications





Specialization and Generalization

Roles and Classifications

DEFINITION

Generalization and specialization is a data modeling approach that results in supertype and subtype entities. The supertype entity is a generalized view; the subtype entity is a specialized view. All of the attributes and relationships of the supertype entity are inherited by the subtype entity. In addition, each subtype entity has attributes and relationships unique to it.

ROLES

Roles are subtypes that are not mutually exclusive. In the human resource example, if a retiree can simultaneously also be a contractor, the subtypes are roles. These are sometimes called conjoint relationships.

For roles, the sum of the number of occurrences of each of the subtypes is greater than or equal to the number of occurrences of the supertype. In building data marts, one must be very careful with dimensions that are roles to avoid duplicate counts.

CLASSIFICATIONS

Classifications are mutually exclusive subtypes. For example, a Customer may have subtypes of Residential Customer, Commercial Customer, and Industrial Customer. If the business rule is that the same customer cannot be both a residential and commercial customer, these subtypes are classifications. These are sometimes called disjoint relationships.

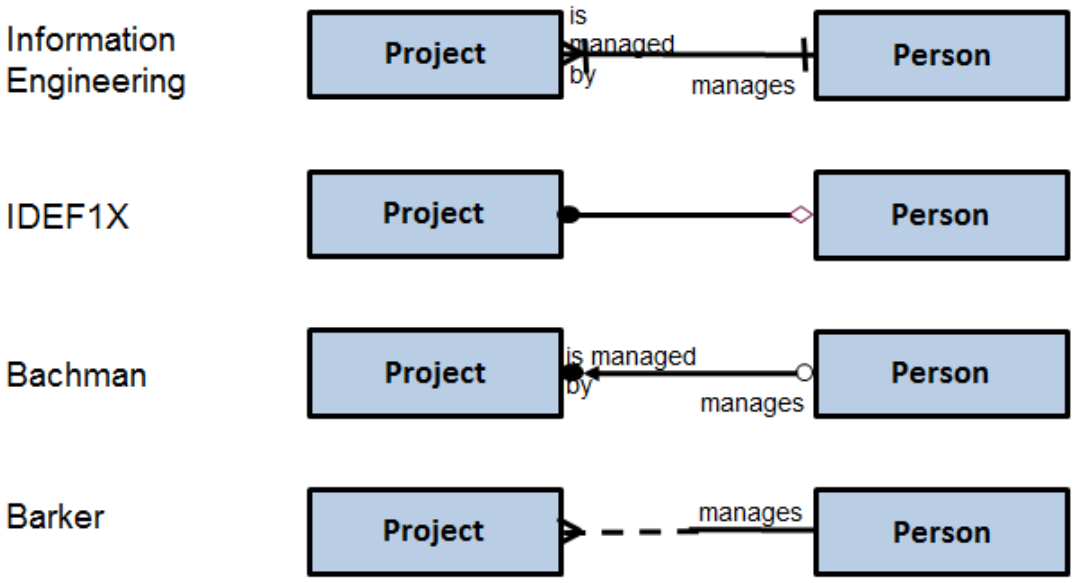
For classifications, the sum of the number of occurrences of each of the subtypes is the same as the number of occurrences of the supertype. Additionally, an attribute indicating the classification type is inserted in the supertype entity.

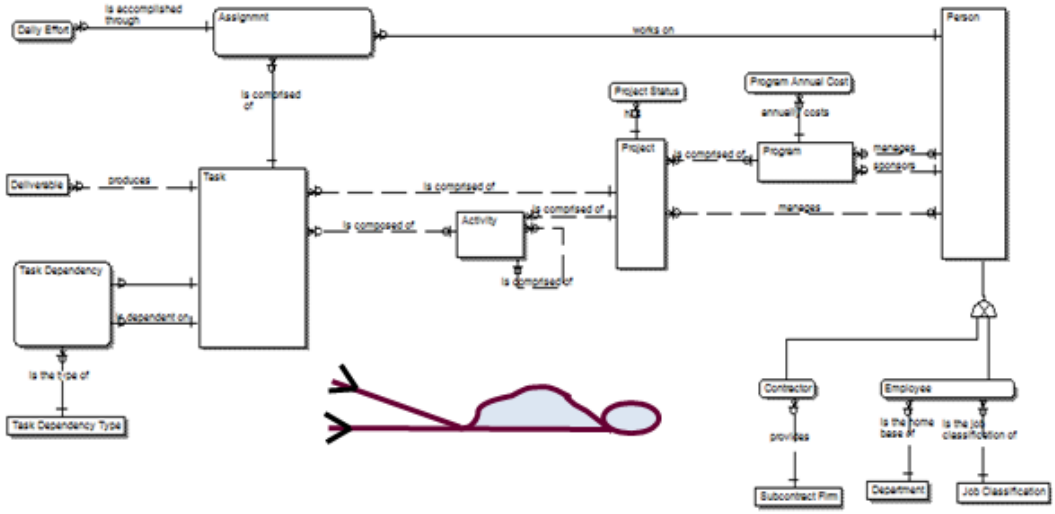
BI IMPACT

Within business intelligence, supertypes and subtypes often get transformed into dimensions within the star schema. Care must be taken to ensure that the relationships are structured to provide correct results. This requires distinguishing between two types of subtypes – classifications and roles.

Presentation

Diagramming Options





Presentation

Diagramming Options

CONVENTIONAL

Conventional data model diagrams relate entities to each other. As more and more entities are added, the diagram gets crowded, and additional effort is needed to make the display consumable. A common approach is to expend a lot of effort to avoid as many crossed lines as possible and to use views that provide subsets of the model.

NOTATIONS

There are many modeling conventions and data modeling tools often provide display options to the data modeler. Common notations include:

- Information Engineering (a.k.a., Crows' Feet)
- IDEF1X
- Bachman
- Barker

DEAD CROWS FLY EAST

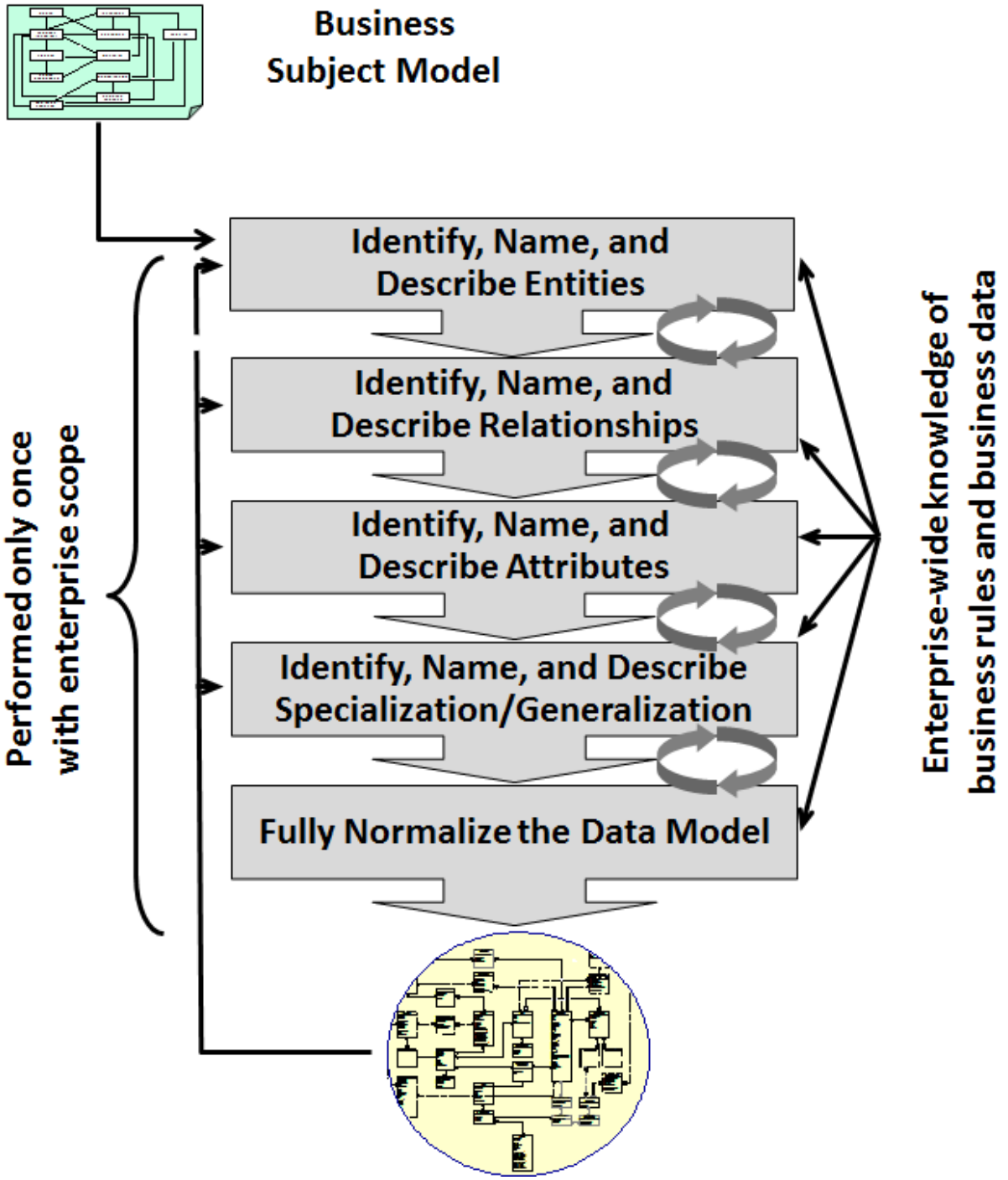
The dead crows fly east convention is a unique approach that applies information engineering ("crows feet") notation and structures the diagram such that all the crows feet point up and to the left. While this may appear silly on the surface, an understanding of the implication of the crows feet reveals the purpose. When two entities are related to each other, the crows feet will be adjoined to the dependent entity. Hence, if this convention is followed, independent entities tend to be on the bottom right, with stronger and stronger dependencies occurring as one migrates to the top left of the diagram.

Module 2

Business Data Model Development

Business Data Model Development Approaches

Top-Down



Business Data Model Development Approaches

Top-Down

MODEL DEVELOPMENT OPTIONS

Once an organization decides to develop the business data model, it has to determine the approach to be used. There are two fundamental options (top-down and bottom-up). In addition, there are hybrid solutions and variations in the way the information for the data model is collected.

TOP-DOWN DEVELOPMENT

Top-down business data model development entails transitioning from the top row of the Zachman Framework (list of major subjects) into the third-normal form model that represents the business.

The most significant decision to be made for top-down data model development is scope. At one extreme, the scope encompasses the entire enterprise, while at the other extreme, the scope is limited to what needs to be implemented. Both of these approaches are risky, and the best practice is to define data model increments that go beyond a single application and can be developed in a reasonable timeframe.

- Developing a full business data model may take 6–12 months, and the business will not see a payback until after it's applied. The risk for this approach is that the effort will be derailed.
- Confining the scope to a single application imposes that system's constraints on the business data model. The risk for this approach is that it does not portray the enterprise view.

PROCESS

The recommended process consists of the following major activities:

- Determine the scope.
- Determine the entities of interest, and establish a business-oriented name and definition for each.
- Determine relationships among entities, and establish the verb phrase describing the relationship, the optionality, and the cardinality for each.
- Determine the attributes of interest, and establish a business-oriented name and definition for each.
- Determine appropriate generalizations and specializations.
- Normalize the model to at least third normal form.

PROS AND CONS

The major advantage of the top-down approach is that it describes the business free from influence by the technical environment, organizational constraints, or process constraints. The major disadvantage is that the team must ferret out the information to build the model.

Data Modeling Roles

Functions, Traits, and Challenges

Role	Function	Traits	Challenges
Business Stakeholder	• Provide resources, direction, & support • Resolve conflicts	• Visibility and respect • Authority	• Provide funding • Maintain interest
Business Data Steward	• Identify and define entities and attributes • Provide business rules	• Communication • Analysis	• Resolve differences • Influence business processes
Subject Matter Expert	• Provide advisory support • Clarify nuances • Verify model	• Business and data knowledge • Respect within business area	• Determine involvement areas
Business Analyst	• Conduct or participate in interviews • Work with modeler	• Communications • Analysis	• Balance needs and wants
Data Modeler	• Lead model development • Translate business information into models • Ensure model quality	• Data modeling • Analysis	• Balance theory with practicality • Know when to stop

Data Modeling Roles

Functions, Traits, and Challenges

BUSINESS STAKEHOLDER

The business stakeholders are the highest level business representatives who are sponsoring the business intelligence effort or are significantly impacted by it. These typically consist of officer or director level people in the major business units involved in the business intelligence effort.

BUSINESS DATA STEWARD

The data stewards (sometimes called data owners) are people within the business areas who are responsible for a group of data elements. These are typically analysts familiar with the data and business processes for acquiring, managing, and providing the data. When the role does not formally exist, this role is performed by the business analysts or subject matter experts.

They are active participants in the development of the business data model and in data profiling activities. Since these people provide these rules and policies about the data, they need to be knowledgeable of the governing business practices and processes and have the authority to speak on behalf of their area concerning the impacted data.

SUBJECT MATTER EXPERT

The subject matter experts are people, within the business areas, who are recognized as the most knowledgeable people about a topic. They may be at any level of the organization, and often wield substantial informal authority even if they are not in a high managerial role. Subject matter experts are often used to perform roles of data stewardship and/or business analysts.

BUSINESS ANALYST

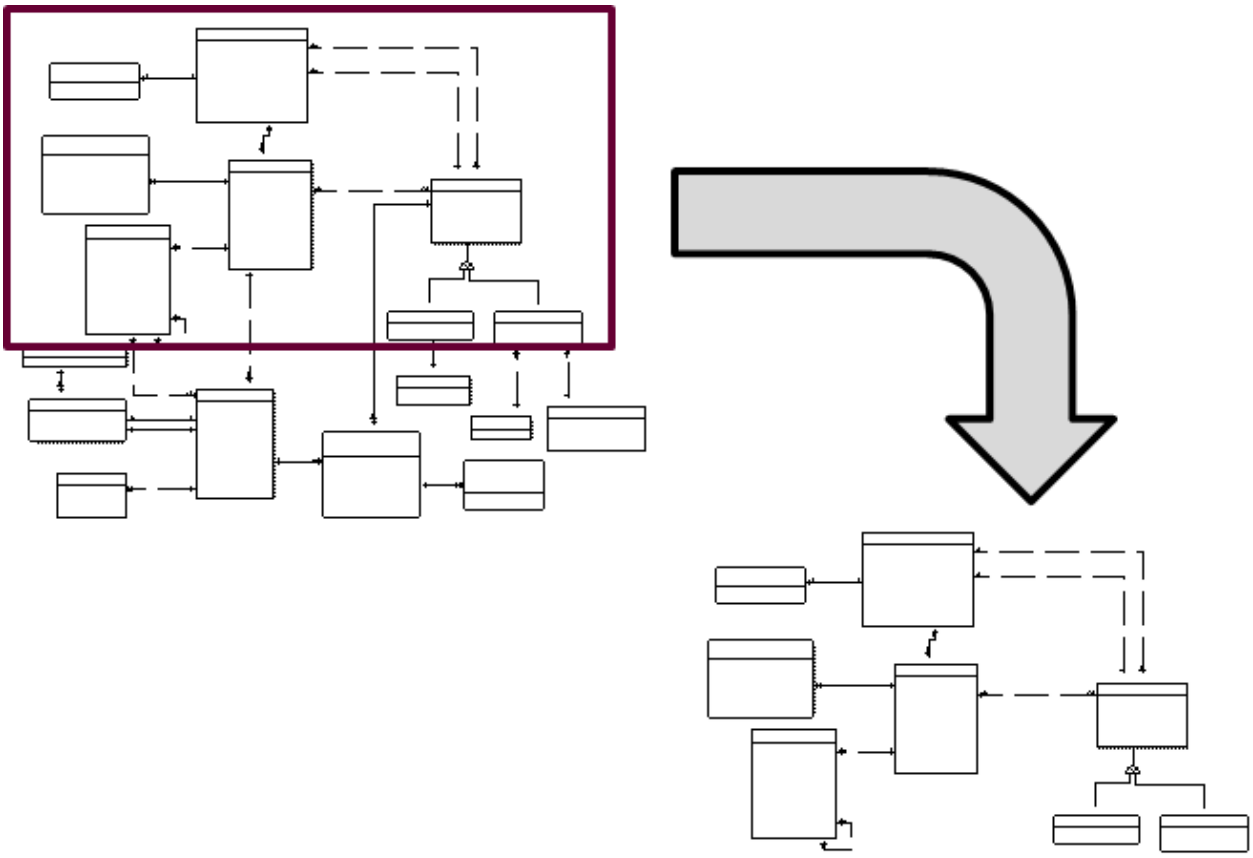
The business analysts are the people who provide or interpret the information gleaned from the business units to ensure that the requirements correctly reflect the needs and priorities. Often, these people join the data analysts in soliciting requirements from the business areas. While they serve as liaisons between the business area they represent and IT, direct access to the users should also be provided.

DATA MODELER

The data modeler (a.k.a. data analyst) translates the information gathered during the interviews and facilitated sessions into the business data model. This person needs to understand the role of the business data model and be highly skilled in information gathering, in modeling techniques, and in the modeling tool(s) being used.is the IT person responsible for developing and maintaining the data models.

Business Data Model Application

Basis for System Data Model



- Incorporate additional constraints if needed
- Remove unneeded attributes
- Add derived attributes
- Replicate data (denormalize) for performance

Business Data Model Application

Basis for System Data Model

DESCRIPTION

One of the major purposes of the business data model is to become the basis for designing databases for application systems and for the data warehouse. This is the model that transforms into the system and subsequently technology model within the Zachman Framework. Each database that needs to be designed should begin with an extract of the business data model.

PARTICIPANTS

The system data model is typically developed by a data modeler working on the project team.

BENEFITS

When all system databases are consistent with the business data model, the definition and business rules for all of the databases are also consistent. This significantly streamlines interfaces among systems and simplifies the migration of data into an integrated data store such as the data warehouse. While there will be data redundancy, the redundancy is managed and created to achieve operational efficiencies and data protection.

PROCESS

Development of the system data model begins with a subset of the business data model that contains the needed data. Additional constraints may be added based on the system's purpose and some attributes may be removed. If there are attributes missing, care must be taken not to add them into the system data model unless they are derived elements or are inserted as part of the denormalization process to improve performance.

TIPS

The system and technology data models are explored in greater depth in Module 3 of this course.

Module 3

System and Physical Data Model Development

Data Modeling Roles

Functions, Traits, and Challenges

Role	Function	Traits	Challenges
Data modeler	• Lead data modeling • Understand data in scope • Understand access needs • Transform business data model into system data model	• Recognition of data as an asset • Data modeling expertise (including tools) • Analysis	• Determine model type • Determine denormalization level • Know when to stop
Business Analyst	• Conduct or participate in interviews • Work with modeler • Verify model	• Communication • Analysis • Willingness to compromise	• Balance needs and wants • Grasp mode nuances
Database Administrator	• Lead technology model development • Transform system data model into technology data model • Adjust schema for performance	• Database expertise (DBMS & tools) • Data modeling expertise (normalized and dimensional)	• Meet performance and security objectives
Developer	• Consider development & operation implications • Provide insight based on physical environment	• Analysis • Technical knowledge	• Grasp model nuances

Data Modeling Roles

Functions, Traits, and Challenges

DATA MODELER

The data analyst/data modeler is the person who transforms the business data model into the systems model and supports the database administrator in creating the technology model.

BUSINESS ANALYST

The business analysts are the people who provide or interpret the business requirements that need to be satisfied. They may be directly providing the requirements or may participate in the interviews and facilitated sessions with other business people to provide them.

DATABASE ADMINISTRATOR

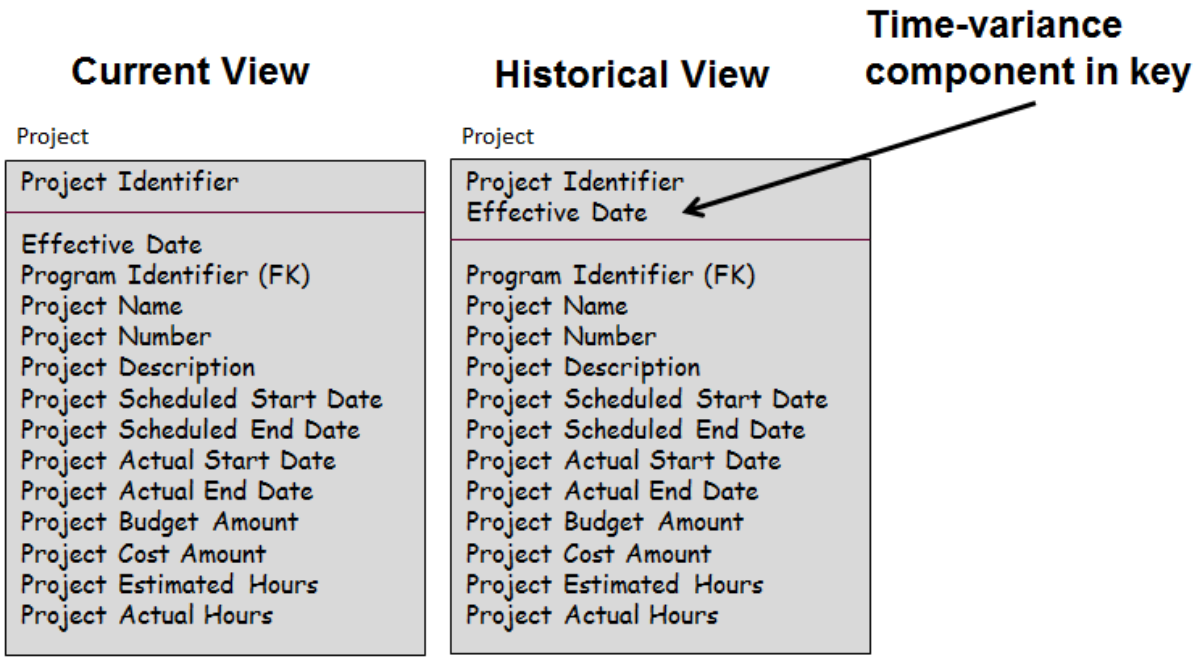
The database administrator (DBA) is responsible for the physical database design and its security, protection, and performance. This person is often located in a shared services group and is not on the project team on a full-time basis.

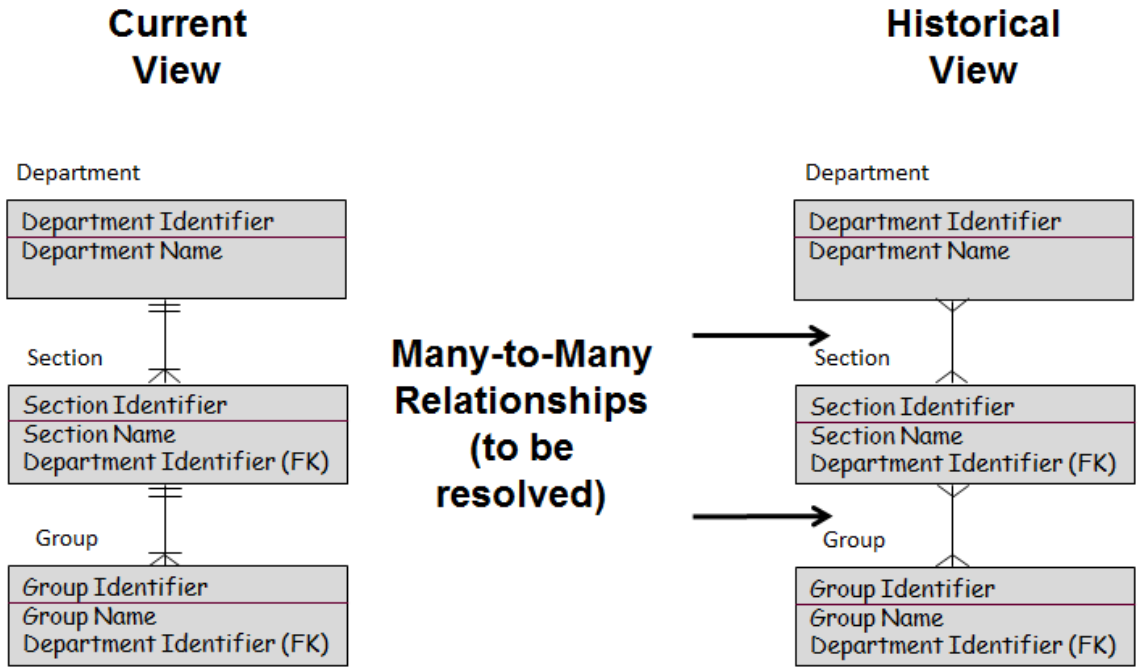
DEVELOPER

The developers are the people who design and build the programs needed to migrate the data through the business intelligence environment.

Time-Variance

History





Time-Variance

History

PURPOSE

The data warehouse contains current and historical data. Therefore, it needs to provide a time-dependent representation of the data.

NORMALIZED MODEL

In the normalized model, history may require substantial model transformations.

- History within a single record adds a component to the key to store unique historical information for each instance.
- If history is retained in the base entity with the concatenated key, each update to this entity also requires an update to its dependent entities due to the cascading of the primary key as a foreign key. Another option is to retain the current data in a base entity, with history being stored in a second table.
- Relationships may change over time, and therefore one-to-many relationships become many-to-many relationships, and these must be resolved.

DIMENSIONAL MODEL

In the dimensional model, history is typically handled using a Type II slowly changing dimension. This dimension contains a new record each time a data value or hierarchical relationship changes. The key to the dimension becomes a new surrogate key that reflects the concatenation of the original key and the date of the change.

SUPPORTING "AS-OF" REPORTING

Data warehouse often need to deal with a perception that history changes. In fact, history cannot change, since the event already took place. What may change, however, is our perception of the event. Hence, it is sometimes useful to include two dates for an event – the date of the event itself, and the data of our view of the event.

Depending on the business needs, it is often unnecessary to propagate the historical views of the event. They should still be retained for audit and control purposes and to be able to recreate views of data as of a specific date.

AUDIT TRAIL

Audit trail records can be handled similar to history records. Since this information will rarely be used in normal processing, however, segregating this data to a separate set of tables simplifies the environment.

DISCUSSION

Discuss the data model and ETL implications for handling history.

Non-Relational Data Structures

Columnar Databases

Data

Project Number	Project Name	Budget Amount	Cost To Date
1	CRM	1000000	780000
2	SCM	2500000	3200000
3	EDW	750000	500000

Row-oriented data storage

1, CRM, 1000000, 780000, 2, SCM, 2500000, 3200000, 3, EDW, 750000, 500000

Column-oriented data storage

1, 2, 3, CRM, SCM, EDW, 1000000, 2500000, 750000, 780000, 3200000, 500000

Non-Relational Data Structures

Columnar Databases

DESCRIPTION

Data warehouses typically retrieve data sequentially from the disk in 32k blocks. A columnar database management system stores data by column rather than row. Hence, when data is retrieved, the columnar value from multiple rows is retrieved with each block of data.

While this approach is inefficient for transaction processing, which process full transactions (i.e., a full row), it is very efficient for data retrieval, since queries typically return data from a limited number of columns across multiple rows.

BENEFITS

In a data warehousing environment, column-oriented databases have several benefits, largely due to reductions in disk seek-time, including:

- Data retrieval – A smaller subset of the data is retrieved to provide aggregates and satisfy queries.
- Column-oriented updates – When there are massive column value replacement (e.g., update to unit prices), the data to be updated is concentrated in one place on the disk, and hence can be updated faster.
- Structure supports greater data compression

DISADVANTAGES

In an OLTP environment, traditional row-oriented databases are advantages. Reasons include:

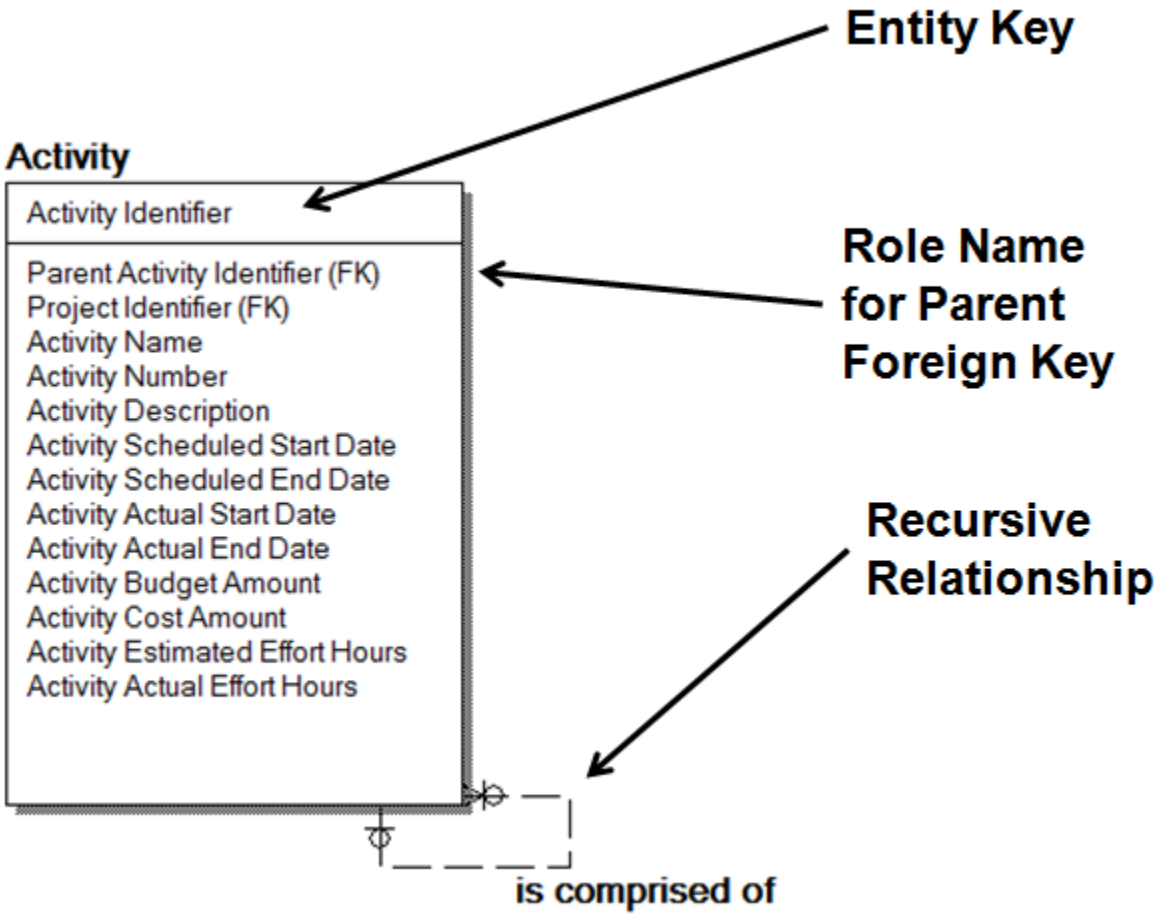
- Entire rows are often retrieved for use, and when the row-size is relatively small, the data could be retrieved with a single seek.
- The data for multiple columns within a row is often updated simultaneously.
- Conversion of data to columnar structure is done during the load.

Module 4

Additional Concepts

Recursive Relationships

Normalized Approach



Recursive Relationships

Normalized Approach

DESCRIPTION A recursive relationship is a relationship between an entity and itself.

APPLICATION Recursive relationships occur frequently with generalized entities. The hierarchical relationship among activities is typically represented as a recursive relationship for the entity "Activity."

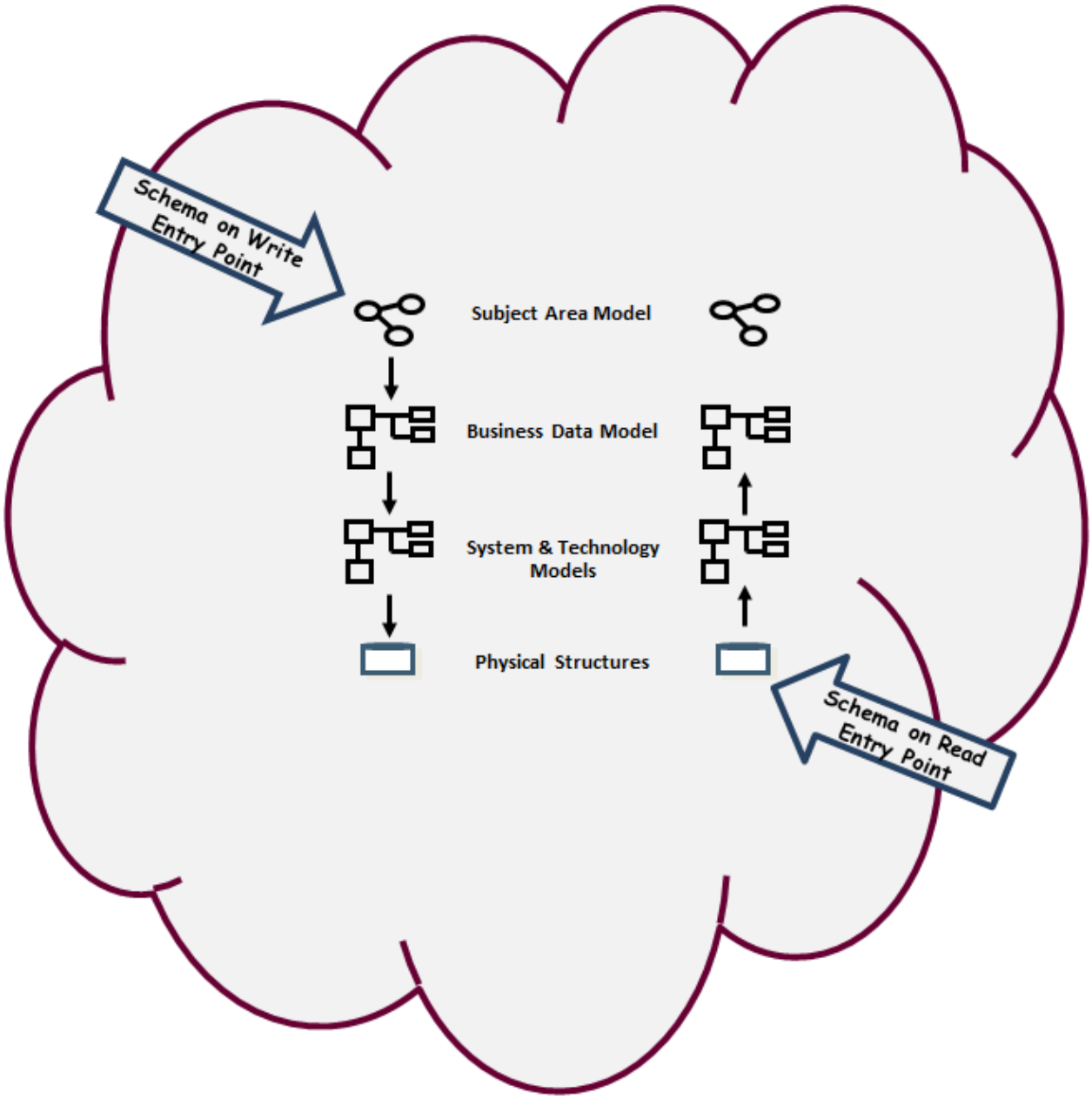
CHARACTERISTICS Recursive relationships are always optional in both directions. The reason for this is that there is always a top to the implied hierarchy (which means there is no parent for that instance) and there is always a bottom (which means there is no child for that instance). Further, if it is a generalization of a hierarchical relationship, the cardinality is one-to-many, that a child cannot have multiple parents.

BI IMPLICATIONS Recursive relationships are acceptable within the normalized hub (in a hub-and-spoke architecture), but cannot be directly deployed in a star schema. Consideration should be given to the data mart requirements in designing the data warehouse.

TIPS Recursive relationships present a minor challenge when it comes to attribute names. Since the parent child relationship exists, a foreign key (Employee Key in the example above) is generated. The problem is that the primary key with the same name already exists and a single entity cannot have two attributes with the same name. To resolve this anomaly, the generated foreign key should be given a role name, such as "Manager Employee Key."

Cloud

Modeling Implications



Cloud

Modeling Implications

CLOUD

Cloud storage is an approach in which companies place their data in virtualized pools of storage, generally hosted by third parties. There are different types of cloud implementations, and there are advantages and disadvantages for each.

MODELING IMPLICATIONS

The cloud option manifests itself at the physical level, and hence with a relational data structure, from a modeling viewpoint, the impact begins with the technology model. In some cases, the technology model is left up to the third party.

Cloud solutions also support a sandbox for performing business analytics. In that environment, they apply the schema on read philosophy.

