System Analysis and Design

Eighth Edition

Alan Dennis, Barbara Wixom, Roberta M. Roth

Chapter 5

Data Modeling

Objectives

- Explain the rules and style guidelines for creating entity relationship diagrams (ERDs).
- Create an ERD.
- Describe the use of a data dictionary and metadata.
- Explain how to balance ERDs and data flow diagrams.
- Describe the process of normalization.

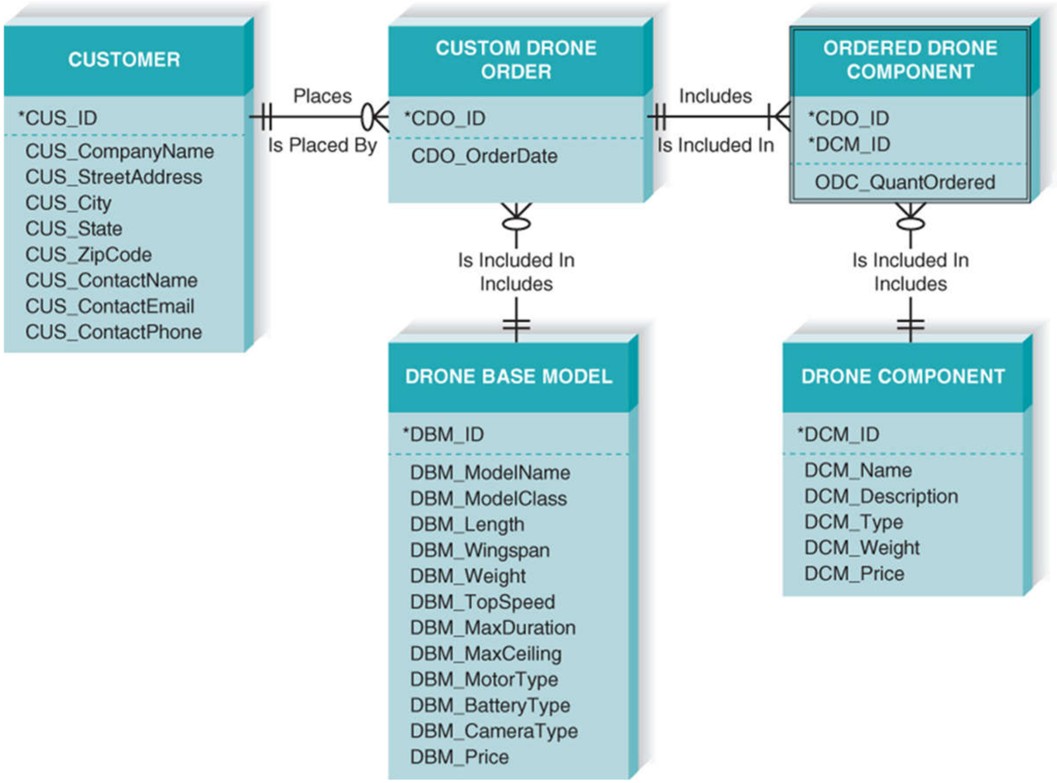
Introduction

- A ***data model*** is a formal way of representing the data that are used and created by a business system
  - Shows the people, places and things about which data is captured and the relationships among them
- ***Logical data model*** shows the organization of data without indicating how it is stored, created, or manipulated
- ***Physical data model*** shows how the data will actually be stored in databases or files

The Entity Relationship Diagram

- ***Entity relationship diagram*** (ERD) is a popular way to depict the data model
- On an ERD, similar kinds of information are listed together and placed inside boxes called ***entities***
  - Lines are drawn between entities to represent relationships among the data
  - Special symbols are added to the diagram to communicate high-level business rules that need to be supported by the system
- The ERD implies no order

Reading an Entity Relationship Diagram



CUSTOMER

*CUS_ID

CUS_CompanyName
CUS_StreetAddress
CUS_City
CUS_State
CUS_ZipCode
CUS_ContactName
CUS_ContactEmail
CUS_ContactPhone

Places
Is Placed By

CUSTOM DRONE ORDER

*CDO_ID

CDO_OrderDate

Includes
Is Included In

ORDERED DRONE COMPONENT

*CDO_ID
*DCM_ID

ODC_QuantOrdered

Is Included In
Includes

DRONE BASE MODEL

*DBM_ID

DBM_ModelName
DBM_ModelClass
DBM_Length
DBM_Wingspan
DBM_Weight
DBM_TopSpeed
DBM_MaxDuration
DBM_MaxCeiling
DBM_MotorType
DBM_BatteryType
DBM_CameraType
DBM_Price

Is Included In
Includes

DRONE COMPONENT

*DCM_ID

DCM_Name
DCM_Description
DCM_Type
DCM_Weight
DCM_Price

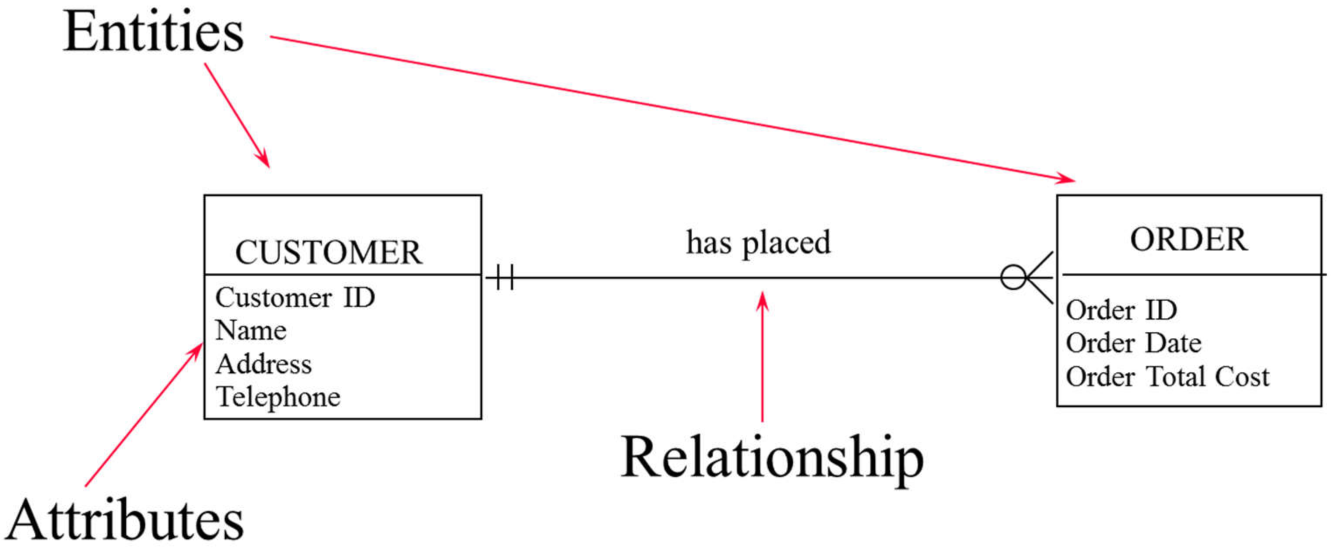
Using the ERD to Show Business Rules

- Business rules are constraints that are followed when the system is in operation
- ERD symbols can show when one instance of an entity must exist for an instance of another to exist
  - The customer instance must exist before custom drone order instances can be created for that customer
  - The drone component instance must exist before ordered drone components instances can be created for that drone component

Using the ERD to Show Business Rules Continued

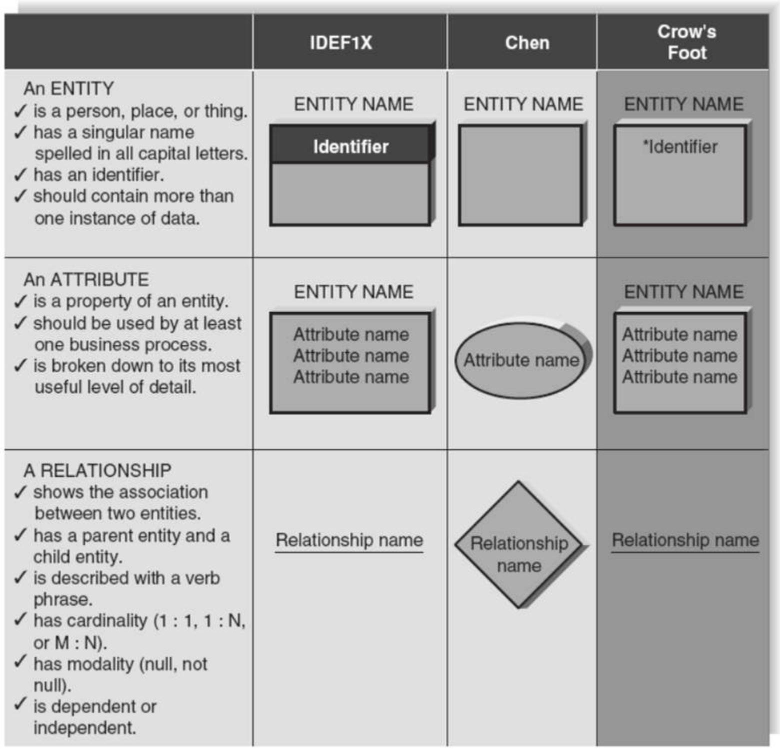
- ERD symbols can show when one instance of an entity can be related to only one or to many instances of another entity
  - One customer instance can create many custom drone order instances; each custom drone order instance is created by only one customer instance
  - A drone can be included on many chemical requests; a chemical request is for only one chemical
- ERD symbols show when the existence of an entity instance is optional for a related entity instance
  - A customer instance may exist without being included on any custom drone order instances

An ERD Example



Elements of an Entity Relationship Diagram

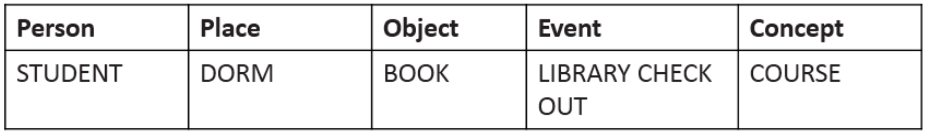
- Entity
- Attribute
- Relationships
- Cardinality
- Modality



	IDEF1X	Chen	Crow's Foot
An ENTITY ✓ is a person, place, or thing. ✓ has a singular name spelled in all capital letters. ✓ has an identifier. ✓ should contain more than one instance of data.	ENTITY NAME Identifier	ENTITY NAME	ENTITY NAME *Identifier
An ATTRIBUTE ✓ is a property of an entity. ✓ should be used by at least one business process. ✓ is broken down to its most useful level of detail.	ENTITY NAME Attribute name Attribute name Attribute name	Attribute name	ENTITY NAME Attribute name Attribute name Attribute name
A RELATIONSHIP ✓ shows the association between two entities. ✓ has a parent entity and a child entity. ✓ is described with a verb phrase. ✓ has cardinality (1 : 1, 1 : N, or M : N). ✓ has modality (null, not null). ✓ is dependent or independent.	Relationship name	Relationship name	Relationship name

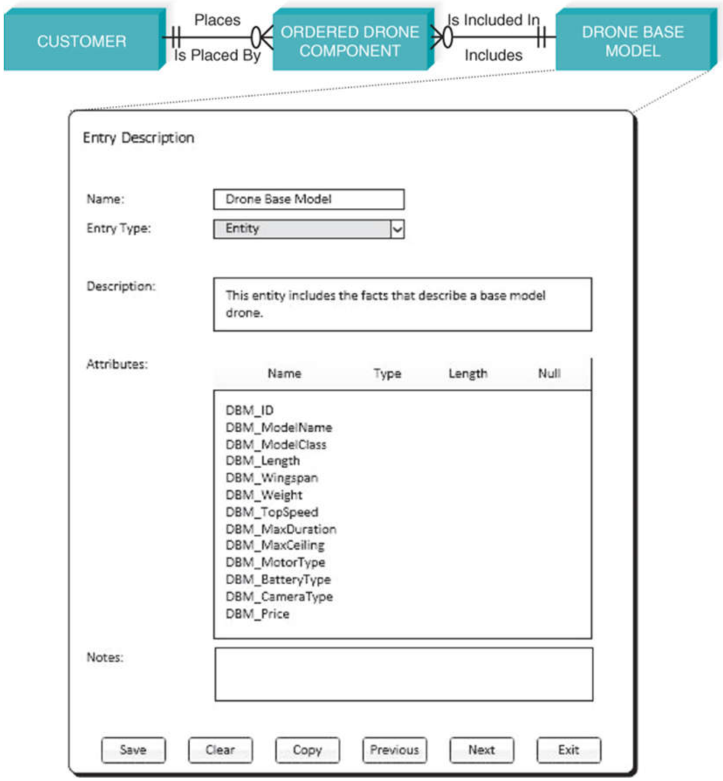
Entity

- A person, place, event, or thing about which data is collected
- Must be multiple occurrences to be an entity



Person	Place	Object	Event	Concept
STUDENT	DORM	BOOK	LIBRARY CHECK OUT	COURSE

CASE Entry for Entity



Entry Description

Name:	Drone Base Model
Entry Type:	Entity
Description:	This entity includes the facts that describe a base model drone.

Attributes:

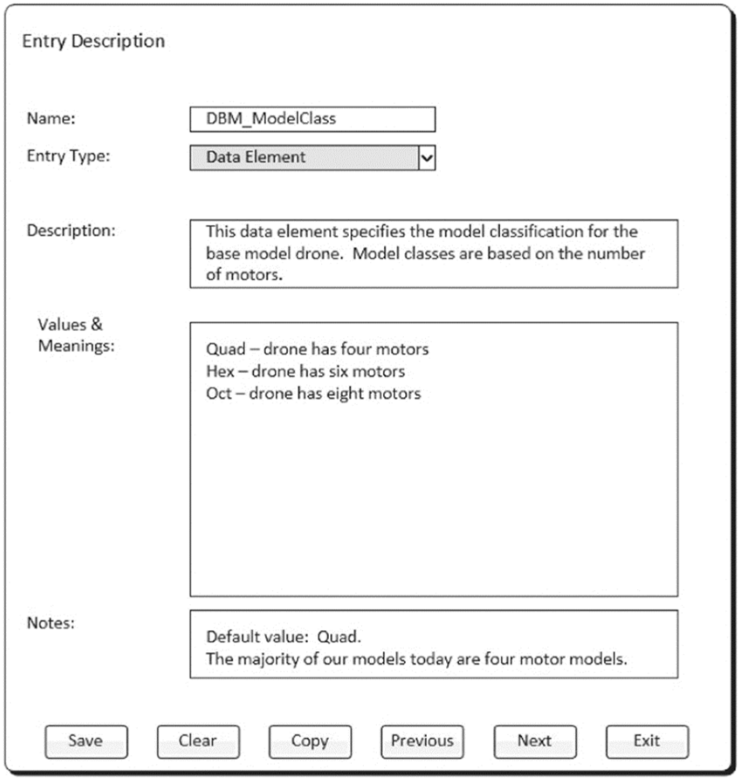
Name	Type	Length	Null
DBM_ID			
DBM_ModelName			
DBM_ModelClass			
DBM_Length			
DBM_Wingspan			
DBM_Weight			
DBM_TopSpeed			
DBM_MaxDuration			
DBM_MaxCeiling			
DBM_MotorType			
DBM_BatteryType			
DBM_CameraType			
DBM_Price			

Notes:

Attributes

- Information captured about an entity
- Only those used by the organization should be included in the model
- Attribute names are nouns
- Sometimes entity name is added at the beginning of the attribute name for clarity

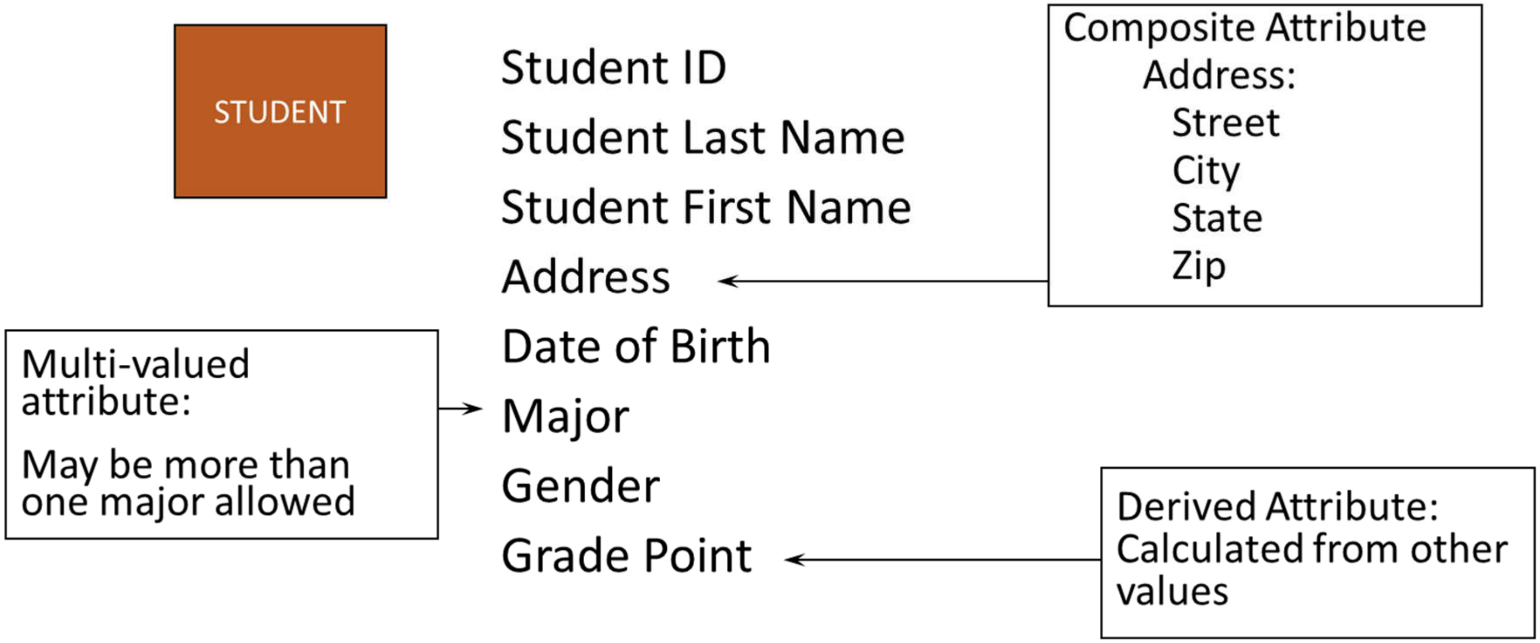
CASE Entry for Attribute



Entry Description

Name:	DBM_ModelClass
Entry Type:	Data Element
Description:	This data element specifies the model classification for the base model drone. Model classes are based on the number of motors.
Values & Meanings:	Quad – drone has four motors Hex – drone has six motors Oct – drone has eight motors
Notes:	Default value: Quad. The majority of our models today are four motor models.

Attributes – Understand Them!



STUDENT

Student ID

Student Last Name

Student First Name

Address

Date of Birth

Major

Gender

Grade Point

Composite Attribute
Address:
Street
City
State
Zip

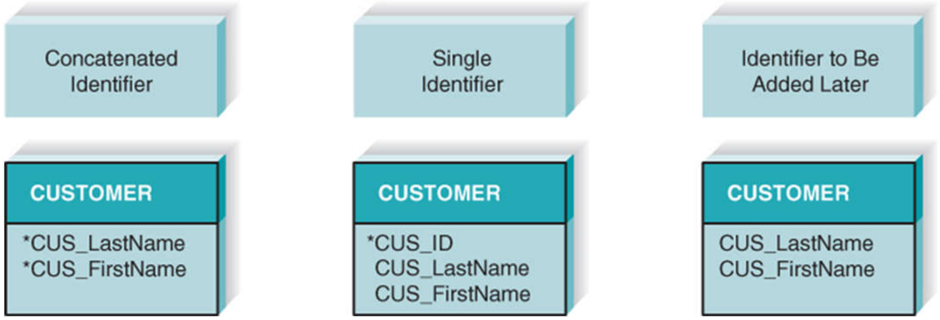
Multi-valued attribute:

May be more than one major allowed

Derived Attribute: Calculated from other values

Identifier Types

- One or more attributes can serve as the entity ***identifier***, uniquely identifying each entity instance
- Concatenated identifier consists of several attributes
- An identifier may be “artificial,” such as creating an I D number
- Final decision on identifiers may postponed to the Design Phase



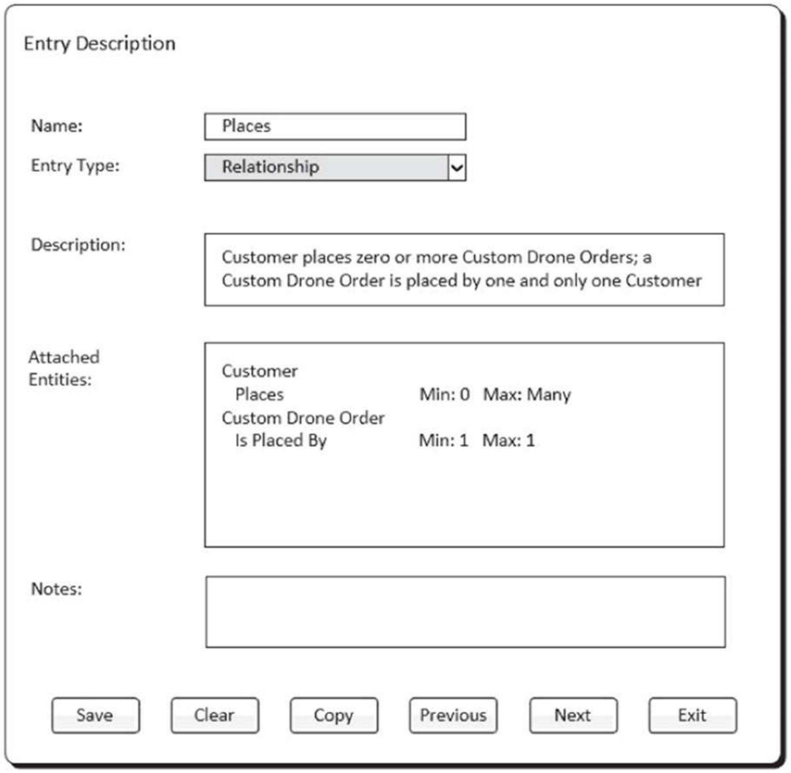
Relationships

- Associations between entities
- The first entity in the relationship is the ***parent*** entity; the second entity in the relationship is the ***child*** entity
- Relationships should have active verb names
- Relationships go in both directions

Cardinality

- Cardinality refers to the number of times instances in one entity can be related to instances in another entity
- One instance in an entity refers to one and only one instance in the related entity ***(1:1)***
- One instance in an entity refers to one or more instances in the related entity ***(1:N)***
- One or more instances in an entity refer to one or more instances in the related entity ***(M:N)***

CASE Entry for Relationships



Entry Description

Name: Places

Entry Type: Relationship

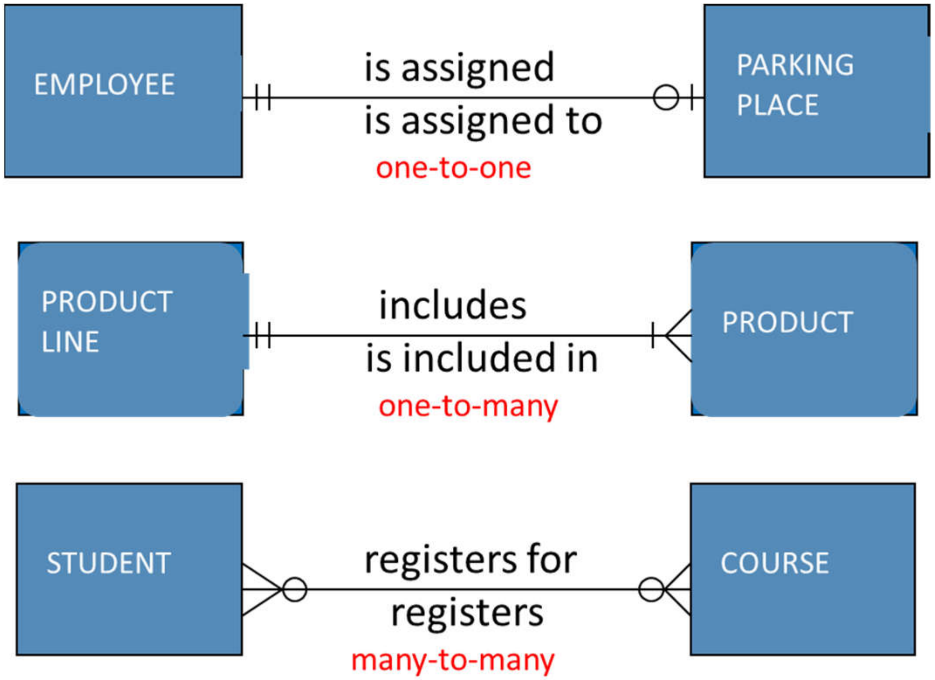
Description: Customer places zero or more Custom Drone Orders; a Custom Drone Order is placed by one and only one Customer

Attached Entities:

Customer		
Places	Min: 0	Max: Many
Custom Drone Order		
Is Placed By	Min: 1	Max: 1

Notes:

Binary Relationships

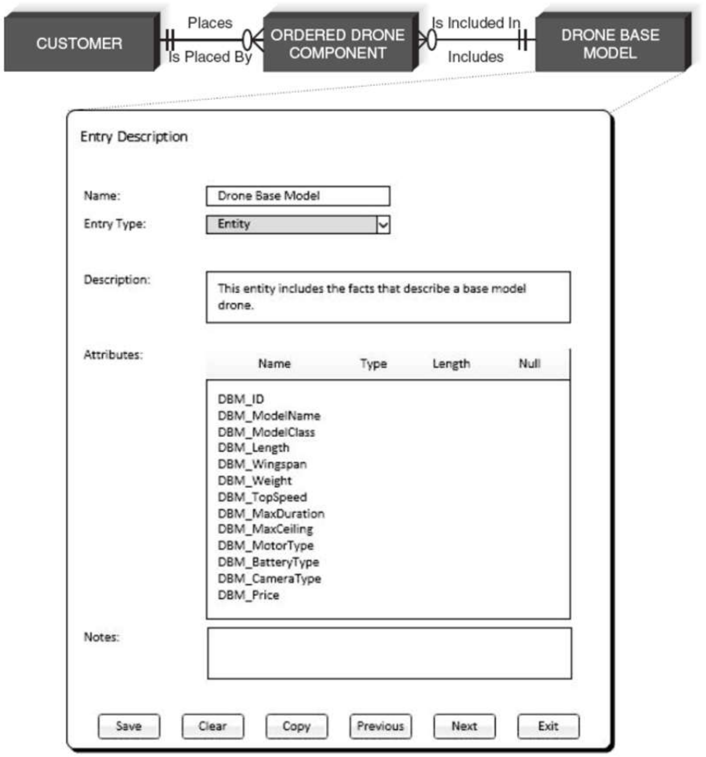


Modality

- Refers to whether or not an instance of a child entity can exist without a related instance in the parent entity
- ***Not Null*** - an instance in the related entity must exist for an instance in another entity to be valid
- ***Null*** - no instance in the related entity is necessary for an instance in another entity to be valid

The Data Dictionary and Metadata

- Every CASE tool has a ***data dictionary***
  - Where the analyst goes to define or look up information about the entities, attributes, and relationships on the ERD
- The information you see in the data dictionary is called ***metadata***
  - Data about data
- Metadata are stored in the data dictionary so that they can be shared



Entry Description

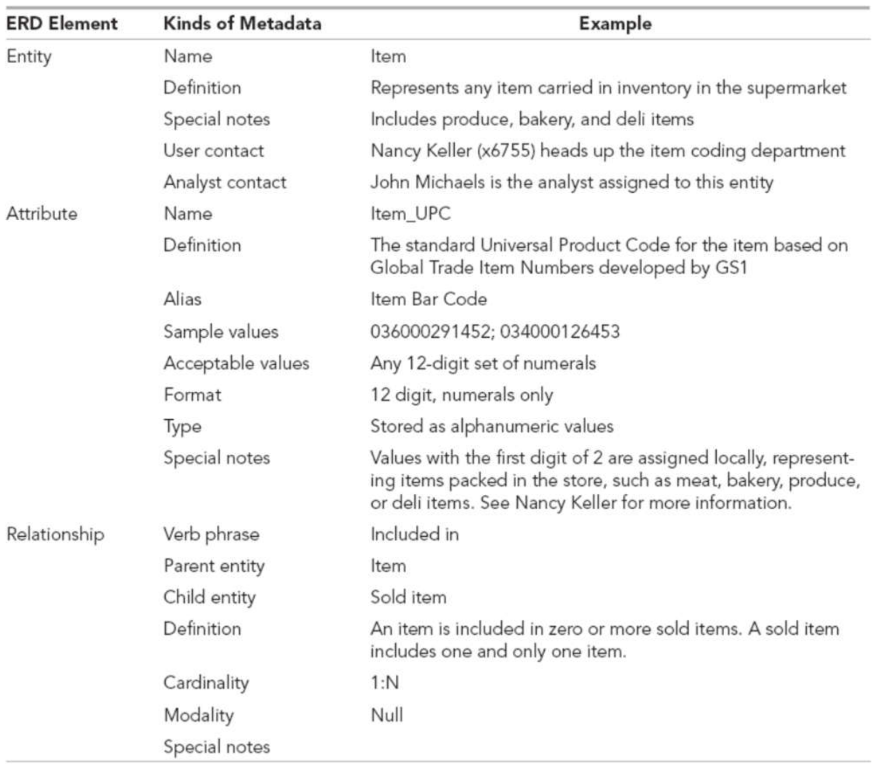
Name:	Drone Base Model
Entry Type:	Entity
Description:	This entity includes the facts that describe a base model drone.

Attributes:

Name	Type	Length	Null
DBM_ID			
DBM_ModelName			
DBM_ModelClass			
DBM_Length			
DBM_Wingspan			
DBM_Weight			
DBM_TopSpeed			
DBM_MaxDuration			
DBM_MaxCeiling			
DBM_MotorType			
DBM_BatteryType			
DBM_CameraType			
DBM_Price			

Notes:

Types of Metadata Captured by the Data Dictionary



ERD Element	Kinds of Metadata	Example
Entity	Name	Item
	Definition	Represents any item carried in inventory in the supermarket
	Special notes	Includes produce, bakery, and deli items
	User contact	Nancy Keller (x6755) heads up the item coding department
	Analyst contact	John Michaels is the analyst assigned to this entity
Attribute	Name	Item_UPC
	Definition	The standard Universal Product Code for the item based on Global Trade Item Numbers developed by GS1
	Alias	Item Bar Code
	Sample values	036000291452; 034000126453
	Acceptable values	Any 12-digit set of numerals
	Format	12 digit, numerals only
	Type	Stored as alphanumeric values
	Special notes	Values with the first digit of 2 are assigned locally, representing items packed in the store, such as meat, bakery, produce, or deli items. See Nancy Keller for more information.
Relationship	Verb phrase	Included in
	Parent entity	Item
	Child entity	Sold item
	Definition	An item is included in zero or more sold items. A sold item includes one and only one item.
	Cardinality	1:N
	Modality	Null
	Special notes	

Creating an Entity Relationship Diagram

- Drawing the ERD is an iterative process of trial and revision
- ERDs can become quite complex
- Steps in building ERDs…
  1. Identify the entities
  2. Add appropriate attributes for each entity
  3. Draw the relationships that connect associated entities

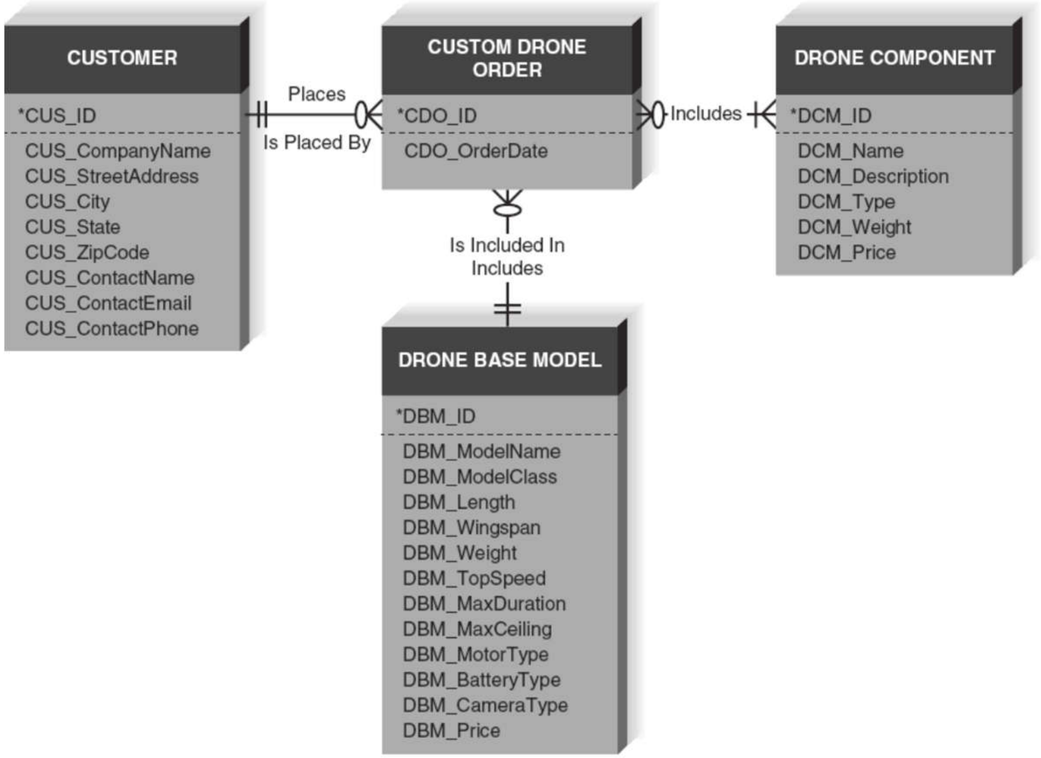
Identify the Entities

- Identify major categories of information
  - If available, check the process models for data stores, external entities, and data flows
  - Check the major inputs and outputs from the use cases
- Verify that there is more than one instance of the entity that occurs in the system

Add Attributes and Assign Identifiers

- Identify attributes of the entity that are relevant to the system under development
  - Check the process model repository entries for details on data flows and data stores
  - Check the data requirements of the requirements definition
  - Interview knowledgeable users
  - Perform document analysis on existing forms and reports
- Select the entity's candidate identifier (final decision may be postponed until Design phase)

Notice the Identifiers Selected for each Entity



Identify Relationships

- Start with an entity and identify all entities with which it shares relationships
- Describe the relationship with the appropriate verb phrase
- Determine the cardinality and modality by discussing the business rules with knowledgeable users

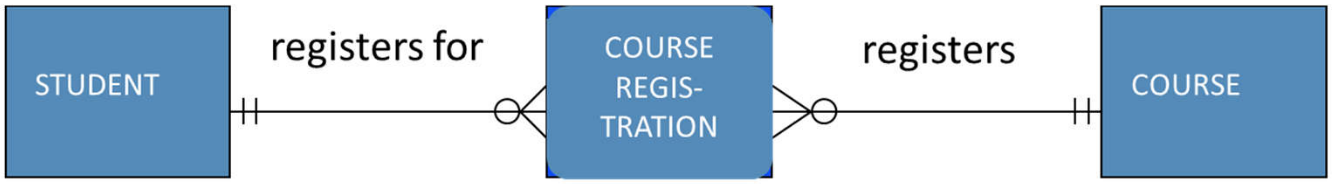
ERD Building Tips

- Data stores of the D F D generally correspond to entities
- Only include entities with more than one instance
- Do not include entities associated with implementation of the system (e.g., archive files of older data)
  - They will be added later

Advanced Syntax - Intersection Entities

- A new entity is created to store information about two entities sharing an M:N relationship
- Remove the M:N relationship between two entities and insert new entity between them
- Create two 1:N relationships: original entities are parents to the new child intersection entity
- Name the intersection entity
- Migrate parent entity primary keys to new entity as foreign keys (possibly also concatenated primary key)

Resolving M:N with an Intersection Entity

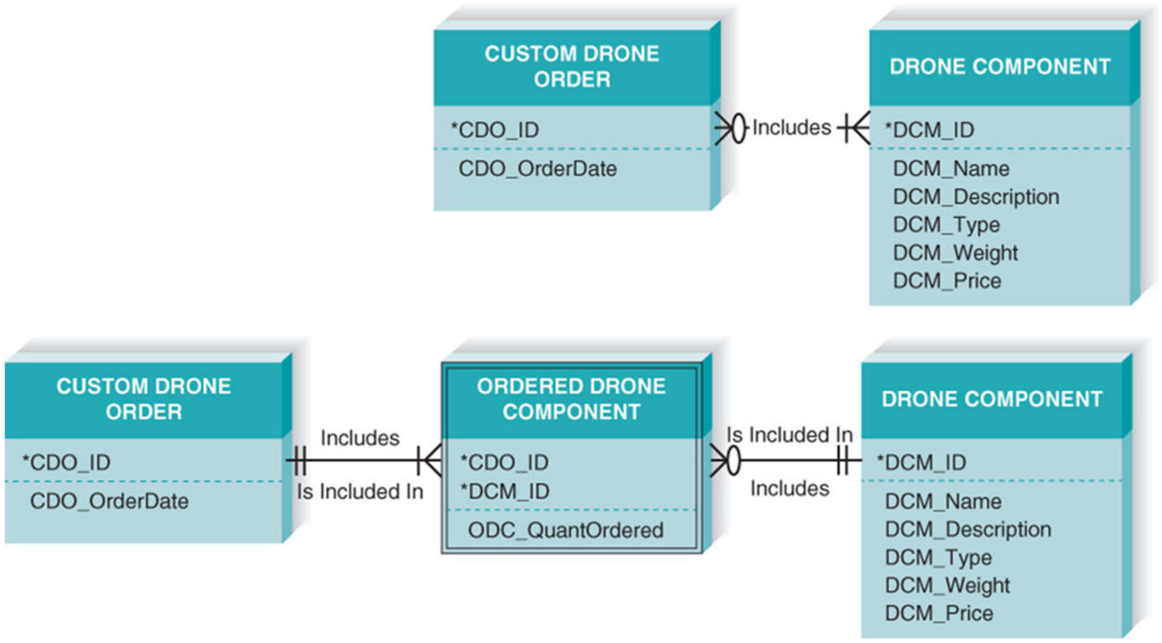


Student ID - PK
Student Name
Student Address

Student ID (PK FK)
Course ID (PK FK)
Semester (PK)
Final Grade

Course ID - PK
Course Name
Course Descrip

Resolving M:N Relationship



CUSTOM DRONE ORDER

*CDO_ID

CDO_OrderDate

Includes

DRONE COMPONENT

*DCM_ID

DCM_Name
DCM_Description
DCM_Type
DCM_Weight
DCM_Price

CUSTOM DRONE ORDER

*CDO_ID

CDO_OrderDate

Includes

Is Included In

ORDERED DRONE COMPONENT

*CDO_ID
*DCM_ID

ODC_QuantOrdered

Is Included In

Includes

DRONE COMPONENT

*DCM_ID

DCM_Name
DCM_Description
DCM_Type
DCM_Weight
DCM_Price

Other Syntax

- An ***independent entity*** is an entity that can exist without the help of another entity
- When a relationship includes an independent child entity, it is called a ***non-identifying relationship***
- There are situations when a child entity requires attributes from the parent entity to uniquely identify an instance
  - Called a ***dependent entity***
- When relationships have a dependent child entity, they are called ***identifying relationships***

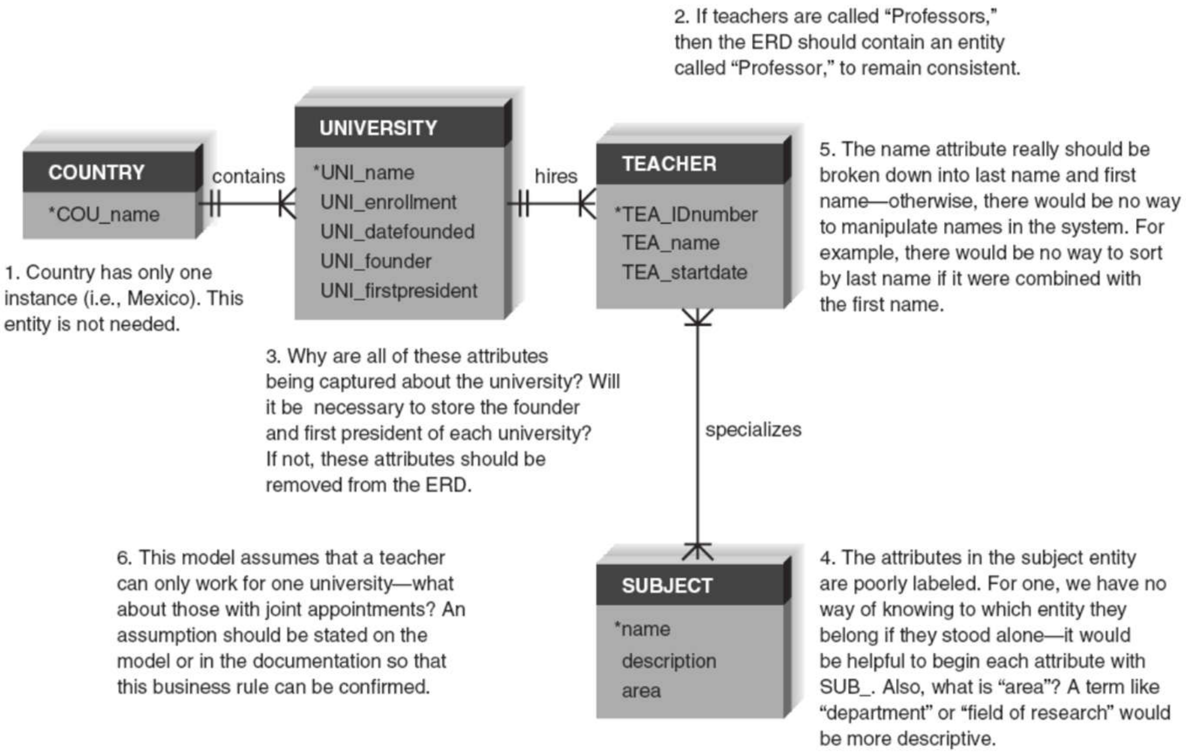
Validating an Entity Relationship Diagram

- Creating ERDs is pretty tough
- There are some general design guidelines that you can keep in mind as you build ERDs
- Once the ERDs are drawn, you can use a technique called ***normalization*** to validate that your models are well formed

Design Guidelines

- Best practices rather than rigid rules
- Entities should have many occurrences
- Avoid unnecessary attributes
- Clearly label all components
- Apply correct cardinality and modality
- Break attributes into lowest level needed
- Labels should reflect common business terms
- Assumptions should be clearly stated

Data Modeling Guidelines Summary



1. Country has only one instance (i.e., Mexico). This entity is not needed.

2. If teachers are called "Professors," then the ERD should contain an entity called "Professor," to remain consistent.

3. Why are all of these attributes being captured about the university? Will it be necessary to store the founder and first president of each university? If not, these attributes should be removed from the ERD.

4. The attributes in the subject entity are poorly labeled. For one, we have no way of knowing to which entity they belong if they stood alone—it would be helpful to begin each attribute with SUB_. Also, what is "area"? A term like "department" or "field of research" would be more descriptive.

5. The name attribute really should be broken down into last name and first name—otherwise, there would be no way to manipulate names in the system. For example, there would be no way to sort by last name if it were combined with the first name.

6. This model assumes that a teacher can only work for one university—what about those with joint appointments? An assumption should be stated on the model or in the documentation so that this business rule can be confirmed.

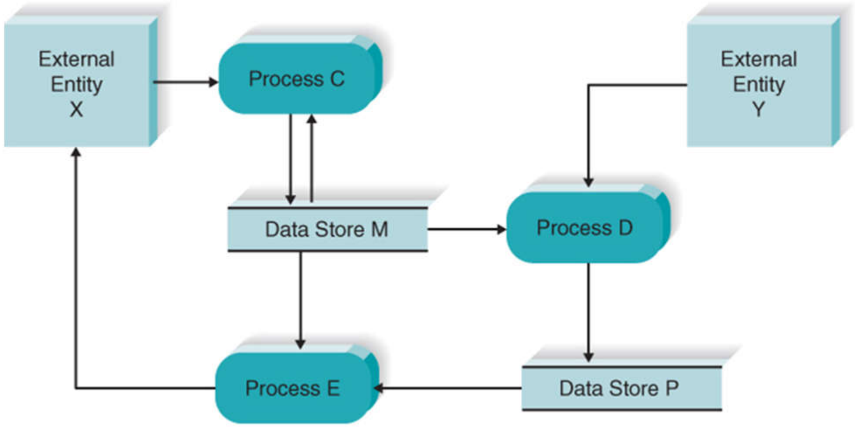
Normalization

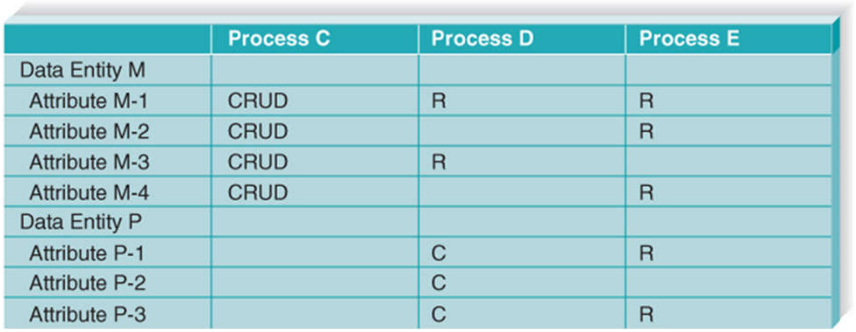
- ***Normalization*** is a process whereby a series of rules are applied to a logical data model or a file to determine how well formed it is
- Normalization rules help analysts identify entities that are not represented correctly in a logical data model, or entities that can be broken out from a file
- Three normalization rules described in the appendix

Balancing ERDs with Data Flow Diagrams

- All analysis activities are interrelated
- Process models contain two data components
- Data flows and data stores
- The D F D data components need to balance the ERD’s data stores (entities) and data elements (attributes)
- Many CASE tools provide features to check for imbalance
- Check that all data stores and elements correspond between models
- Data that is not used is unnecessary
- Data that has been omitted results in an incomplete system
- Do not follow thoughtlessly -- check that the models make sense!

Use of a CRUD Matrix





	Process C	Process D	Process E
Data Entity M			
Attribute M-1	CRUD	R	R
Attribute M-2	CRUD		R
Attribute M-3	CRUD	R	
Attribute M-4	CRUD		R
Data Entity P			
Attribute P-1		C	R
Attribute P-2		C	
Attribute P-3		C	R

Chapter Review

- Define the meaning and purpose of the entity and relationship shown on an entity relationship diagram (ERD).
- Explain the meaning and purpose of attributes included in a data model.
- Explain what is meant by an entity’s identifier.
- Explain the meaning of the cardinality and modality of a relationship.
- Explain the concept of metadata and how it is compiled in the project repository.
- Discuss the process used to create a data model.
- Describe how to ensure that the process model and data model are balanced through the use of the CRUD matrix.
- Discuss how the normalization process is performed and how it contributes to the quality of the data model (from chapter appendix).

Key Terms

- 1:1 relationships
- 1:N relationships
- Assumptions
- Attribute
- Balance
- Business rules
- Cardinality
- Child entity
- Clients
- Concatenated identifier
- Create, read, update, delete (CRUD) matrix
- Data dictionary
- Data model
- Dependent
- Dependent entity
- Derived attributes
- Entity
- Entity relationship diagram (ERD)
- First normal form (1NF)
- Identifier
- Identifying relationships
- Independent entity
- Instances
- Intersection entity
- Logical data model
- Metadata
- M:N relationship
- Non-identifying relationship
- Normalization
- Parent entity
- Partial dependency
- Physical data model
- Relationships
- Repeating attributes
- Repeating attribute groups
- Second normal form
- (2NF)
- Subject area
- Third normal form (3NF)
- Transitive dependency