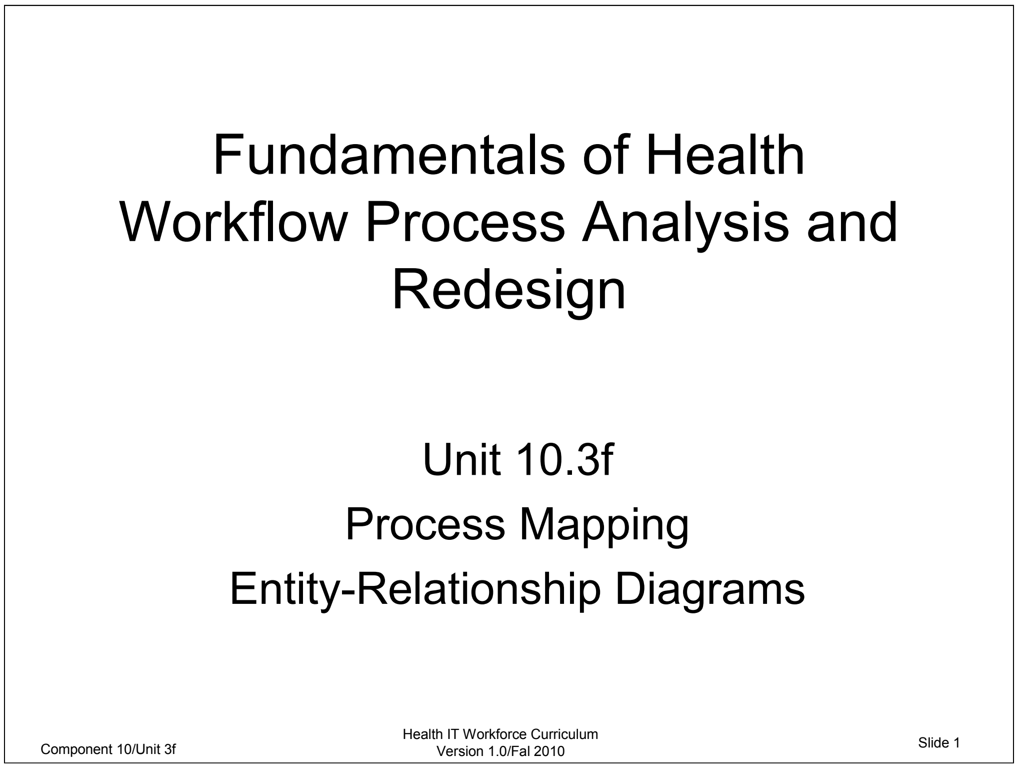


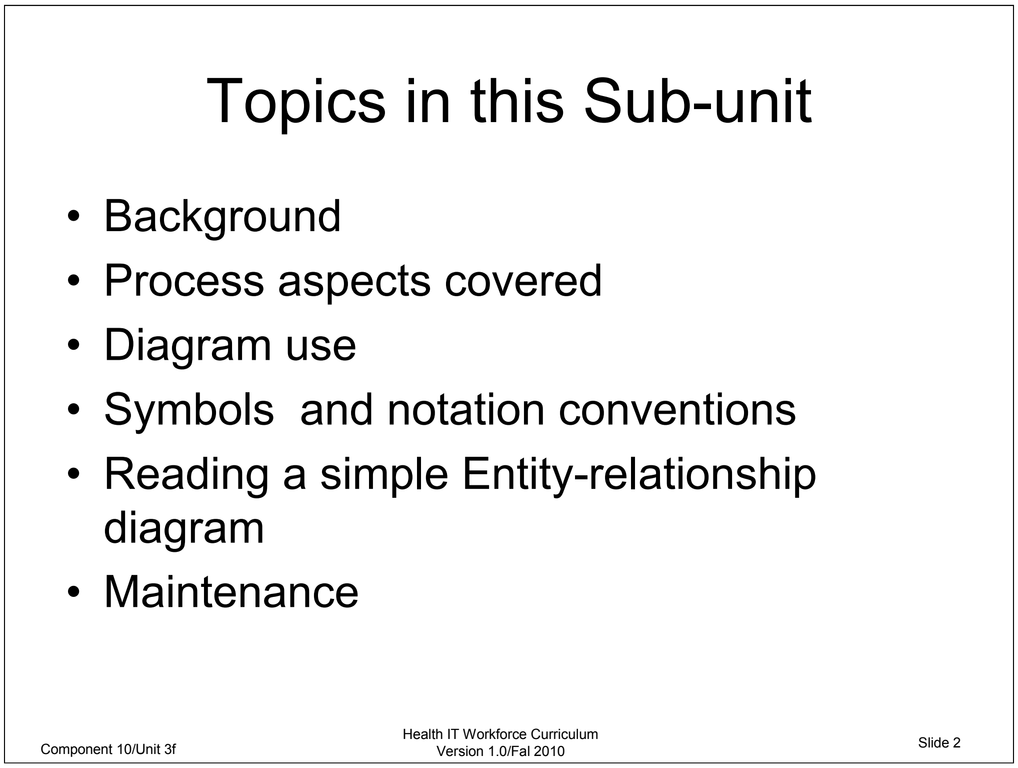
Fundamentals of Health Workflow Process Analysis and Redesign

Unit 10.3f

Process Mapping

Entity-Relationship Diagrams

Welcome to the Entity-Relationship Diagrams Subunit. This is the fifth and final Subunit of of the Process Mapping Unit.



Topics in this Sub-unit

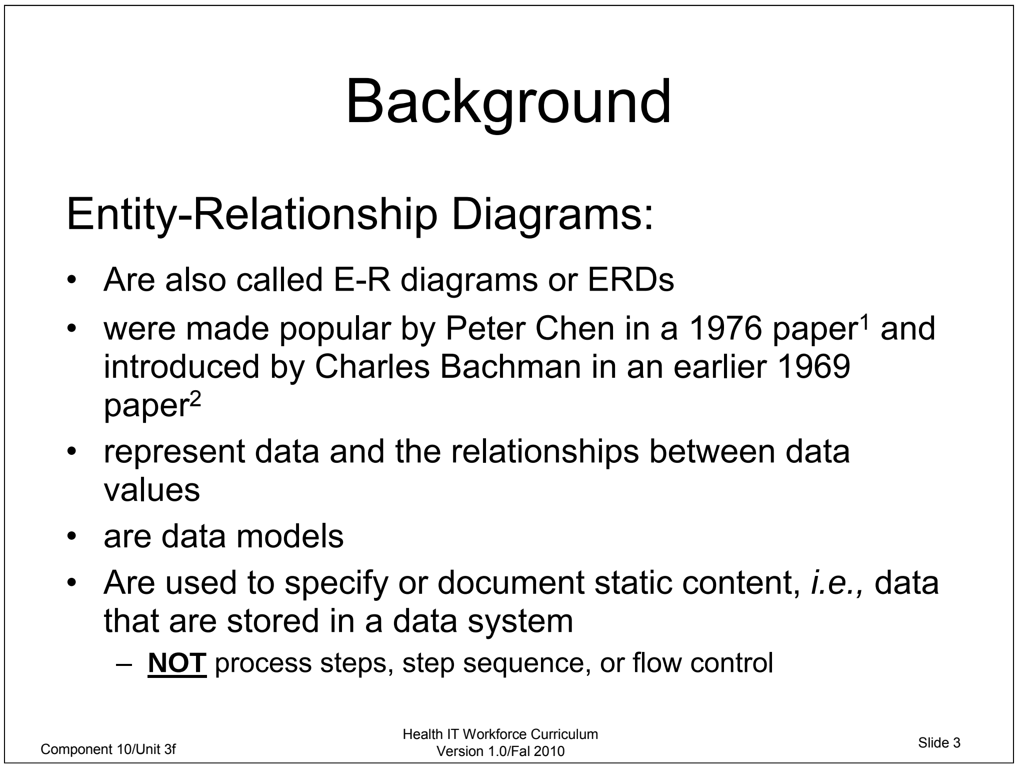
- Background
- Process aspects covered
- Diagram use
- Symbols and notation conventions
- Reading a simple Entity-relationship diagram
- Maintenance

Component 10/Unit 3f

Health IT Workforce Curriculum
Version 1.0/Fal 2010

Slide 2

Topics in this sub-unit include the background and development of entity-relationship diagrams), the process aspects that are covered by ERDs, how ERDs are used, the symbol set, and notation conventions. This sub-unit also covers how to read an ERD (rather than how to create one).



Background

Entity-Relationship Diagrams:

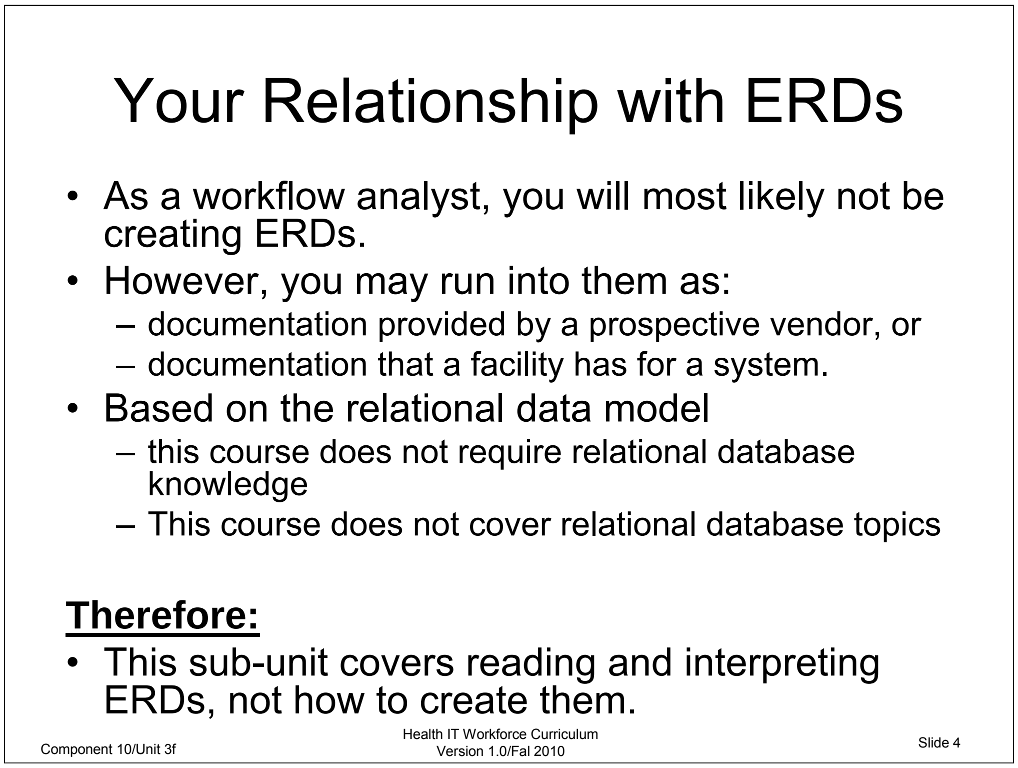
- Are also called E-R diagrams or ERDs
- were made popular by Peter Chen in a 1976 paper[1] and introduced by Charles Bachman in an earlier 1969 paper[2]
- represent data and the relationships between data values
- are data models
- Are used to specify or document static content, *i.e.,* data that are stored in a data system
  - **NOT** process steps, step sequence, or flow control

Component 10/Unit 3f — Health IT Workforce Curriculum Version 1.0/Fal 2010 — Slide 3

Entity-relationship diagrams are also called E-R diagrams and ERDs. The concept for ERDs was introduced in a 1976 paper by Peter Chen just six years after E.F. Codd published his seminal work defining the relational model of data. The citation for the papers are included on the references slide at the end of this lecture. Chen's notation provided a way to graphically shows relationships between data values, thus, it is best described as a data model. More specifically a logical data model because it describes how the data are related, not the physical locations of the data values within computer memory or how they are stored in a database.

Logical data models (referred to as just data models for the rest of the presentation) are used to represent or document the data that are needed, collected, stored or otherwise used for some business need or needs. A data model is a description of the data rather than how the data are used or how the data move through a system. Thus, such descriptions are called "static" models, or models of information content. Descriptions of how data move through a system would be called "dynamic" or "behavioral" data models. ERDs and for that matter static data models in general were not designed to , and in fact can not model process steps, step sequence, or flow control. Static data models convey "just the data".

ERDs are used in projects where the data content for a system needs to be documented.



Your Relationship with ERDs

- As a workflow analyst, you will most likely not be creating ERDs.
- However, you may run into them as:
  - documentation provided by a prospective vendor, or
  - documentation that a facility has for a system.
- Based on the relational data model
  - this course does not require relational database knowledge
  - This course does not cover relational database topics

Therefore:

- This sub-unit covers reading and interpreting ERDs, not how to create them.

Component 10/Unit 3f

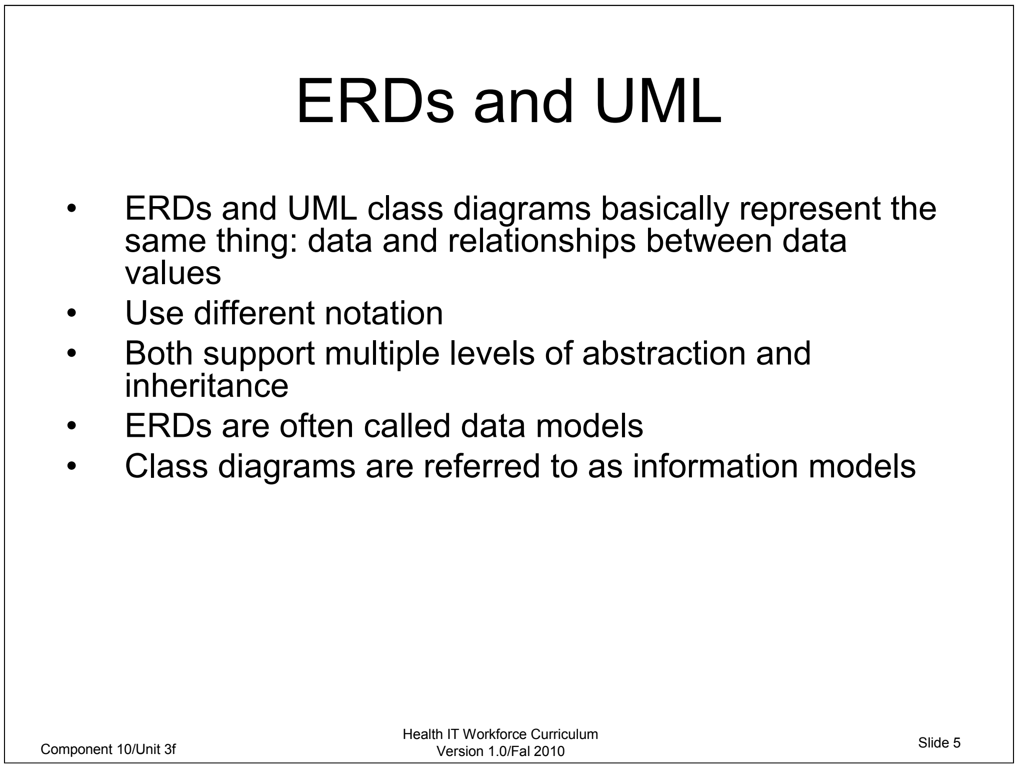
Health IT Workforce Curriculum
Version 1.0/Fal 2010

Slide 4

As a workflow analyst, you will most likely not be creating ERDs. However, you may run into them as either documentation provided by a prospective vendor that documents the data that their system stores, or documentation that a healthcare facility has for an existing system. ERDs are based on the relational data model (E. F. Codd). Briefly, the relational data model describes fundamental ways in which data values may be related to each other, and as such, guides the design of a relational database in such a way as to keep those relationships in tact.

Full appreciation of the information conveyed on an ERD requires knowledge of the relational data model. IMPORTANTLY: this course does not require knowledge of the relational data model or of databases built based on the relational model. Most community colleges and universities have courses on relational databases and relational database management that cover the relational data model in depth.

This sub-unit covers reading and interpreting ERDs, not how to create them.

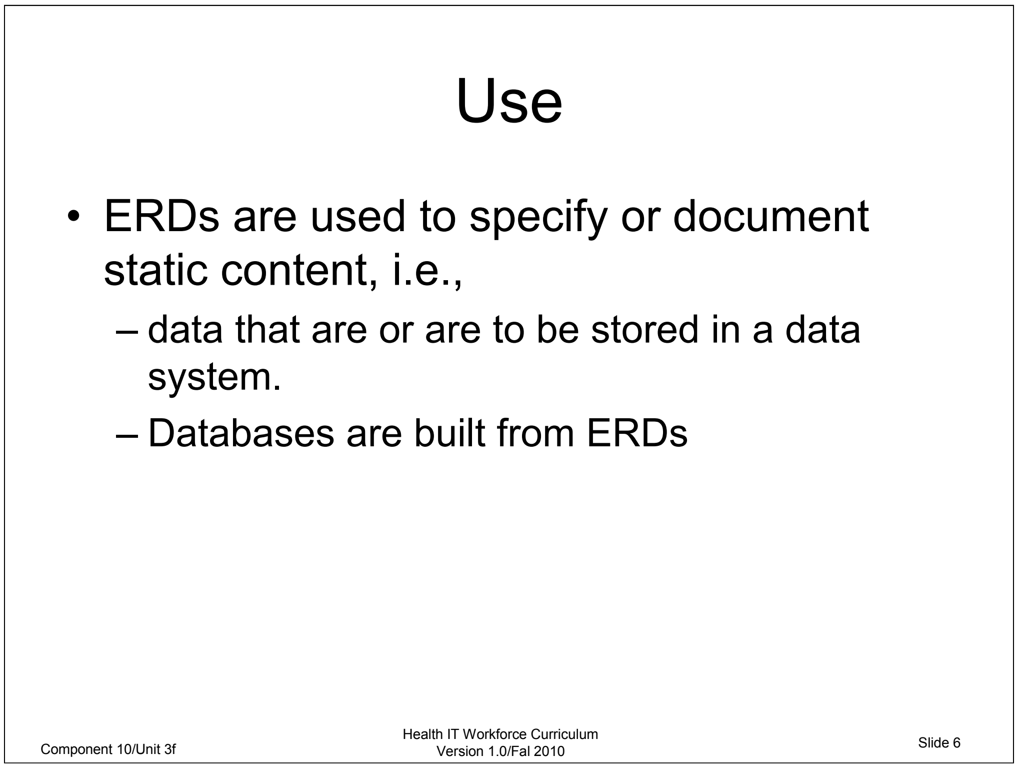


ERDs and UML

- ERDs and UML class diagrams basically represent the same thing: data and relationships between data values
- Use different notation
- Both support multiple levels of abstraction and inheritance
- ERDs are often called data models
- Class diagrams are referred to as information models

Component 10/Unit 3f Health IT Workforce Curriculum Version 1.0/Fal 2010 Slide 5

ERDs are similar to Unified Modeling Language (UML) class diagrams; they are both static data models. ERDs predate the development of the UML standard by over a decade, and the influence of ERD notation and methodology on UML class diagrams is quite visible. Although class diagrams represent additional relationships that ERDs do not, they both can represent multiple levels of abstraction, and the incorporation of hierarchy where the lower level classes “inherit” attributes from higher level classes. This is referred to as inheritance. Class diagrams can be used to show individual pieces of data, their attributes and relationships. If just this data content is needed, ERDs are equivalent to UML class diagrams. Class diagrams are covered in a separate sub-unit.



Use

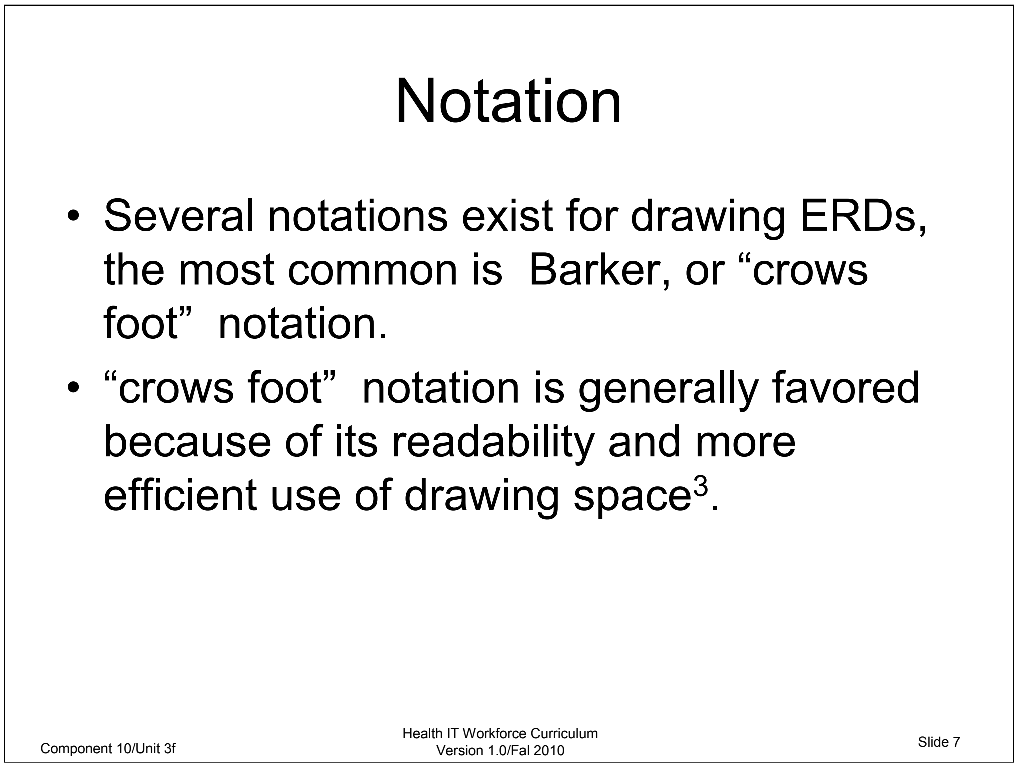
- ERDs are used to specify or document static content, i.e.,
  - data that are or are to be stored in a data system.
  - Databases are built from ERDs

Component 10/Unit 3f

Health IT Workforce Curriculum
Version 1.0/Fal 2010

Slide 6

To summarize the important points so far, ERDs are used to specify or document static content, i.e., data that are, or are to be stored in a data system. As such, ERDs are often used as the starting point in the design of relational databases. ERDs are included in this unit because data and information are used, collected and stored in many work processes, especially those in information intensive settings such as healthcare. Data are a part of healthacre workflow and need to be considered and understood along with the workflow steps. Thus, the workflow analysis and process redesign specialist may encounter ERDs. Further, the workflow analysis and process redesign specialist will need to know when data are important to a clinical workflow, and when to take these data into account as part of process redesign.



Notation

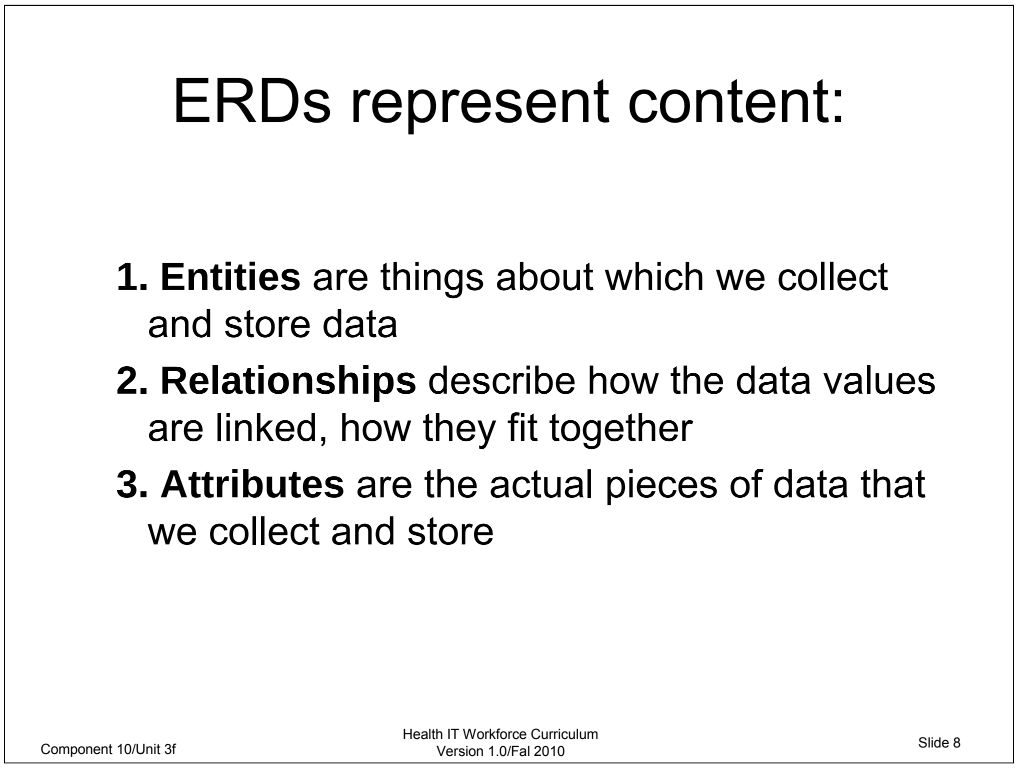
- Several notations exist for drawing ERDs, the most common is Barker, or “crows foot” notation.
- “crows foot” notation is generally favored because of its readability and more efficient use of drawing space[3].

Component 10/Unit 3f

Health IT Workforce Curriculum
Version 1.0/Fal 2010

Slide 7

Several notations exist for drawing ERDs, the most common is Barker or “crows foot” notation. “Crows foot” notation is favored over the original Chen style of ERD modeling because of its readability and more efficient use of drawing space. Data model diagrams provided by software vendors often use crow’s foot notation. We will cover only crow’s foot notation in this sub-unit.



ERDs represent content:

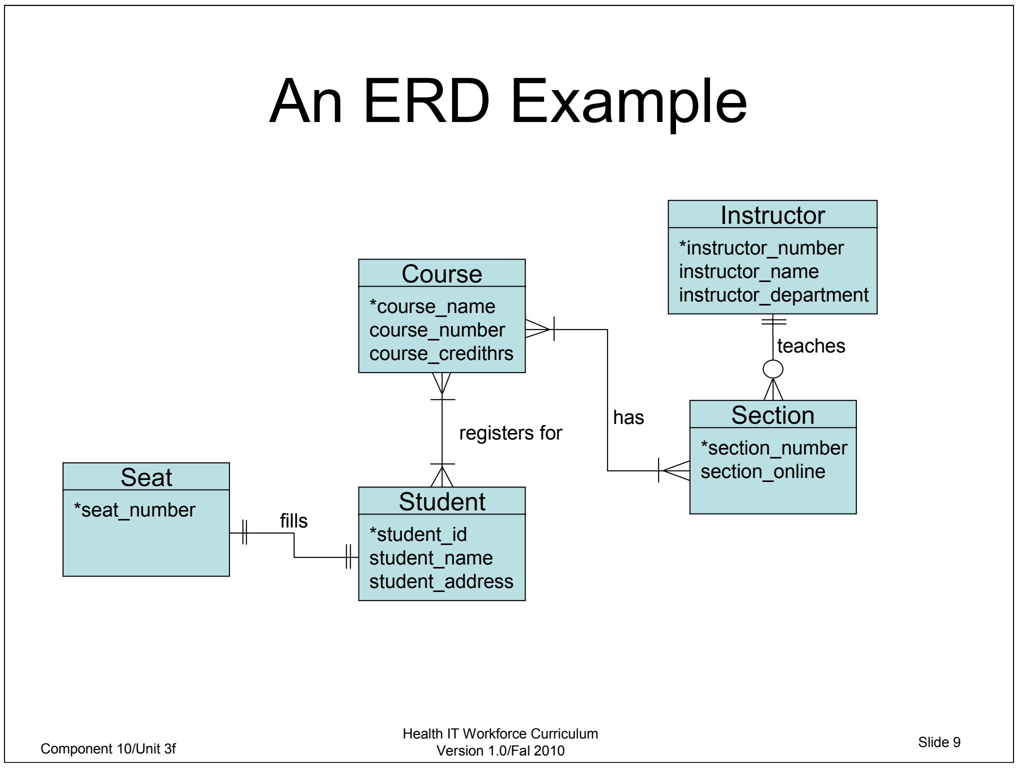
1. **Entities** are things about which we collect and store data
2. **Relationships** describe how the data values are linked, how they fit together
3. **Attributes** are the actual pieces of data that we collect and store

Component 10/Unit 3f

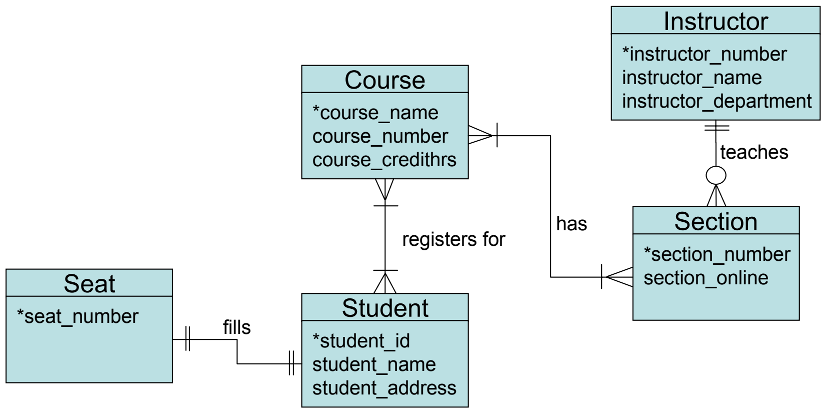
Health IT Workforce Curriculum
Version 1.0/Fal 2010

Slide 8

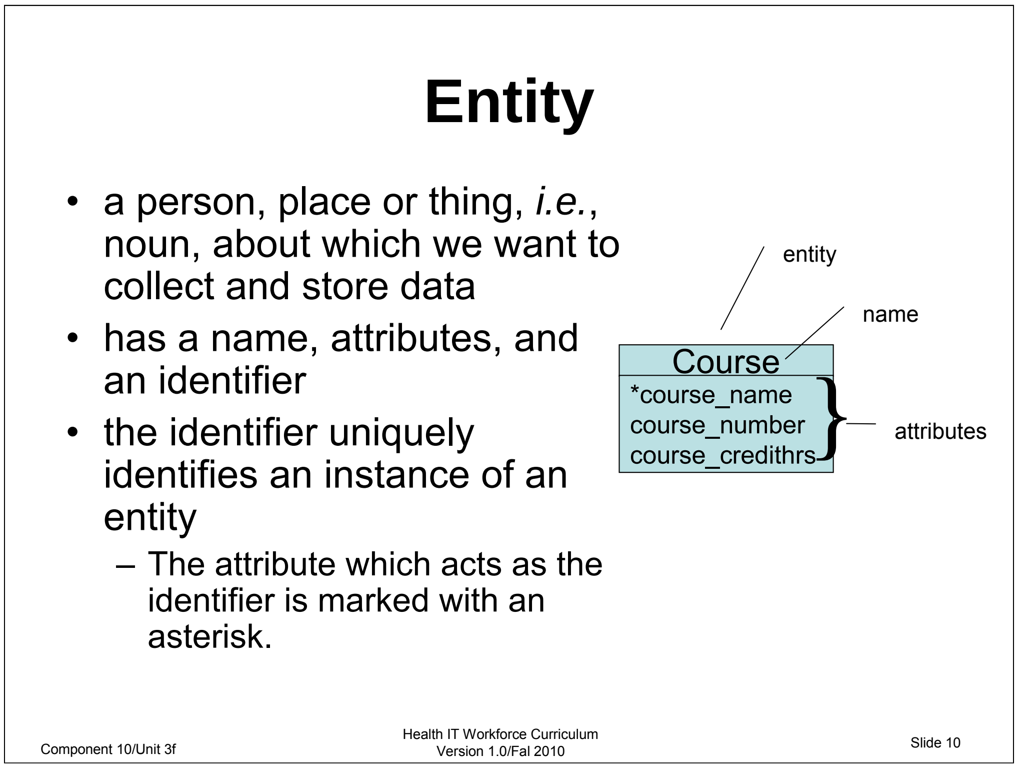
We collect and store data about things in the real world. When these data values have meaning, i.e., definition, context such as what the value is about, when it was collected, etc., we call the data information. Often the meaning is assumed and the word data is used interchangeably with the word information. Nonetheless, data models including ERDs represent three things 1) entities or things about which we collect data, e.g., cars, people, organizations. 2) relationships between the data values, for example, a person can have no cars, one car, or many cars. And 3) attributes, or the data that we collect about the entities. For example, we may be interested in the make and model of a car, or the name and address about a person. Make, model, name and address are considered attributes of the entities cars and people.



An ERD Example

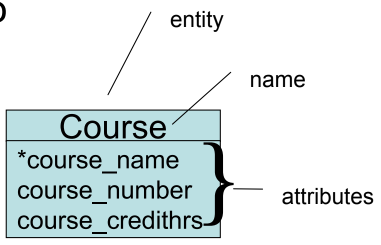


An ERD looks like this. There is only one type of E-R diagram, a data model. The ERD in the example describes the data that might be used to manage courses and student registration. A data system for managing student course registration would need data about courses, instructors, course sections, course seats, and students; these are entities. The course registration management process might require course name, number and credit hours; these are attributes. The course registration management process might also require instructor name, number and department; these are also attributes. In ERD notation, the Entity name appears in the top of the box. The attributes are listed in the box below the name. The lines connecting the boxes convey information about the relationships between the data. These relationships may be a model of reality, or may document constraints or business rules required by a process or data system. The statement every student must have a student ID is one such constraint that may or may not reflect reality.



Entity

- a person, place or thing, *i.e.*, noun, about which we want to collect and store data
- has a name, attributes, and an identifier
- the identifier uniquely identifies an instance of an entity
  - The attribute which acts as the identifier is marked with an asterisk.



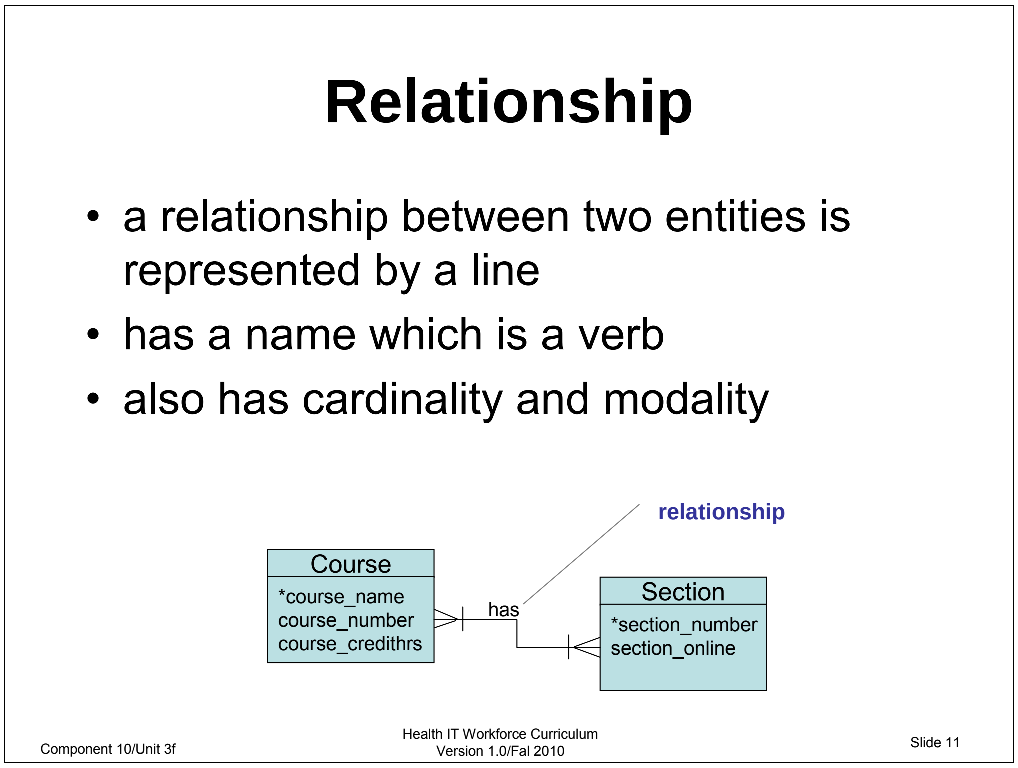
Component 10/Unit 3f

Health IT Workforce Curriculum
Version 1.0/Fal 2010

Slide 10

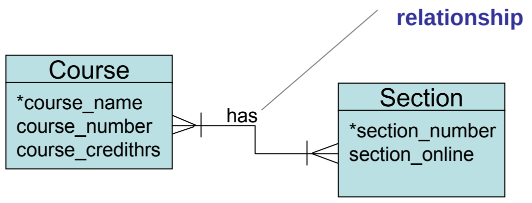
In ERD crow's foot notation, an entity is represented by a two-part box, i.e., the top section that holds the entity name AND the bottom section where the attributes are listed. An entity has a name, attributes, and an identifier that uniquely identifies instances of the entity. Unique identifiers are similar to social security numbers, each number is distinct and is assigned to one and only one person. Thus if we had a list of 12 social security numbers, we could be assured that the list represented 12 different people. An attribute is added to the entity that acts as the unique identifier, or selected from the attributes that meet the all important criteria of being capable of serving as a unique identifier. Sometimes on an ERD you may see boxes that have an entity name but no attributes. This is done on draft diagrams as a place holder to show that information about the entity is needed but has not yet been specified.

For the Course entity shown on the slide, a database would store the course name, number and credit hours. IMPORTANTLY: the Course entity box signifies that the database holds data values about courses. Each ACTUAL course for which data exist is called an instance of the course entity. If there were a person entity, you and I would be considered instances because we are real and exist. The generic entity, person, or course on the slide, are called universals because they represent the existence of instances.

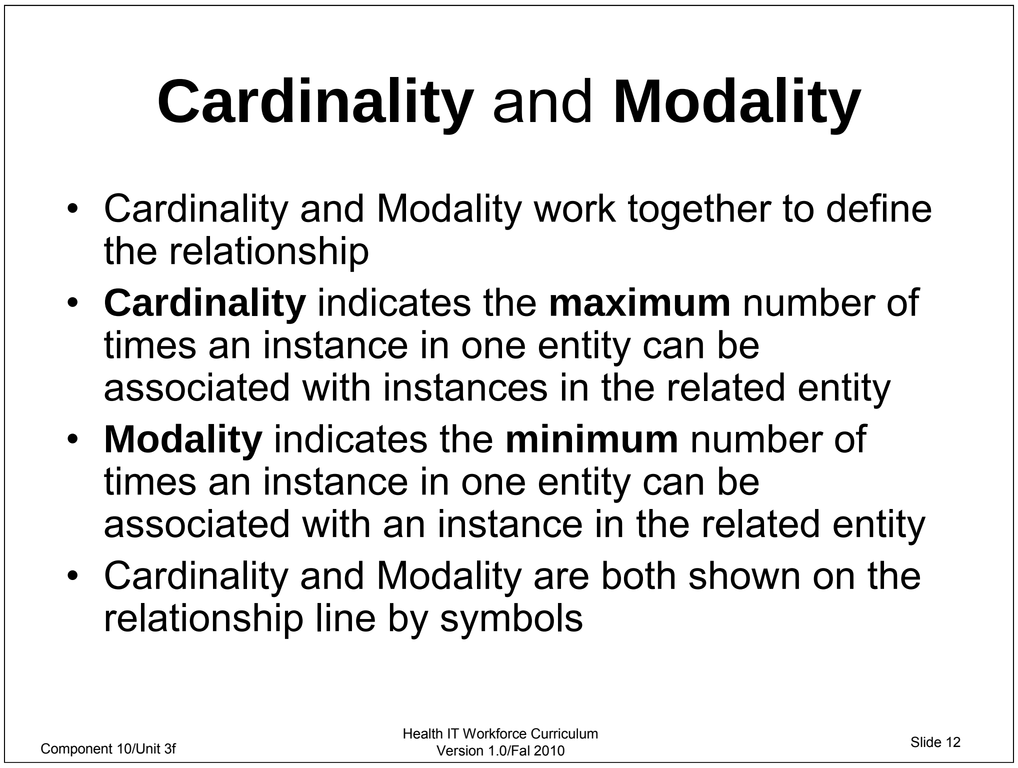


Relationship

- a relationship between two entities is represented by a line
- has a name which is a verb
- also has cardinality and modality



Relationships between instances of two entities are represented by a line. Relationships have a name which is a verb. ERDs only show a few types of relationships (UML class diagrams are capable of representing additional types of relationships). The types of relationships represented are 1) the existence of a relationship, denoted by the presence of a line between two entities, and 2) the maximum and minimum number of times an instance (actual occurrence) in one entity can be associated with instances in an entity with which it has a relationship. These maximum and minimum numbers are called cardinality and modality respectively. These are the **ONLY** types of relationships represented in data models.



Cardinality and **Modality**

- Cardinality and Modality work together to define the relationship
- **Cardinality** indicates the **maximum** number of times an instance in one entity can be associated with instances in the related entity
- **Modality** indicates the **minimum** number of times an instance in one entity can be associated with an instance in the related entity
- Cardinality and Modality are both shown on the relationship line by symbols

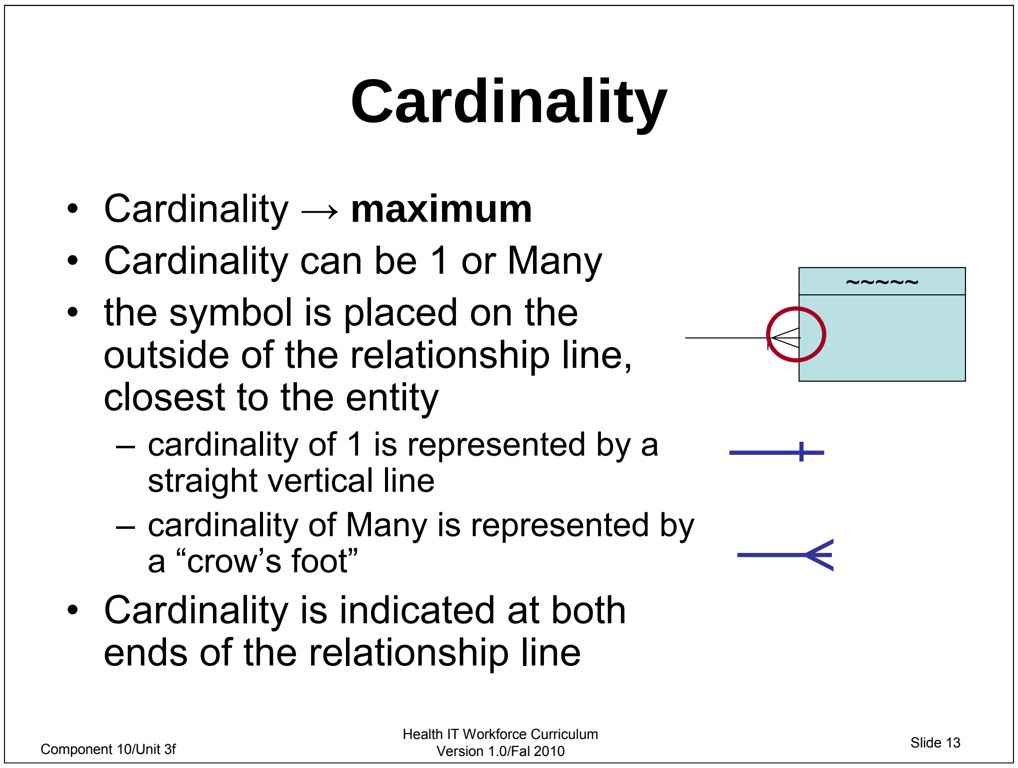
Component 10/Unit 3f

Health IT Workforce Curriculum
Version 1.0/Fal 2010

Slide 12

Cardinality and Modality work together to define the relationship. **Cardinality** indicates the **maximum** number of times an instance in one entity can be associated with instances in the related entity. **Modality** indicates the **minimum** number of times an instance in one entity can be associated with an instance in the related entity. Thus, Modality is also called participation because it denotes whether or not an instance on an entity MUST participate in the relationship.

Cardinality and Modality are both shown on the relationship line by symbols. We will go over each of the symbols and how to interpret them.

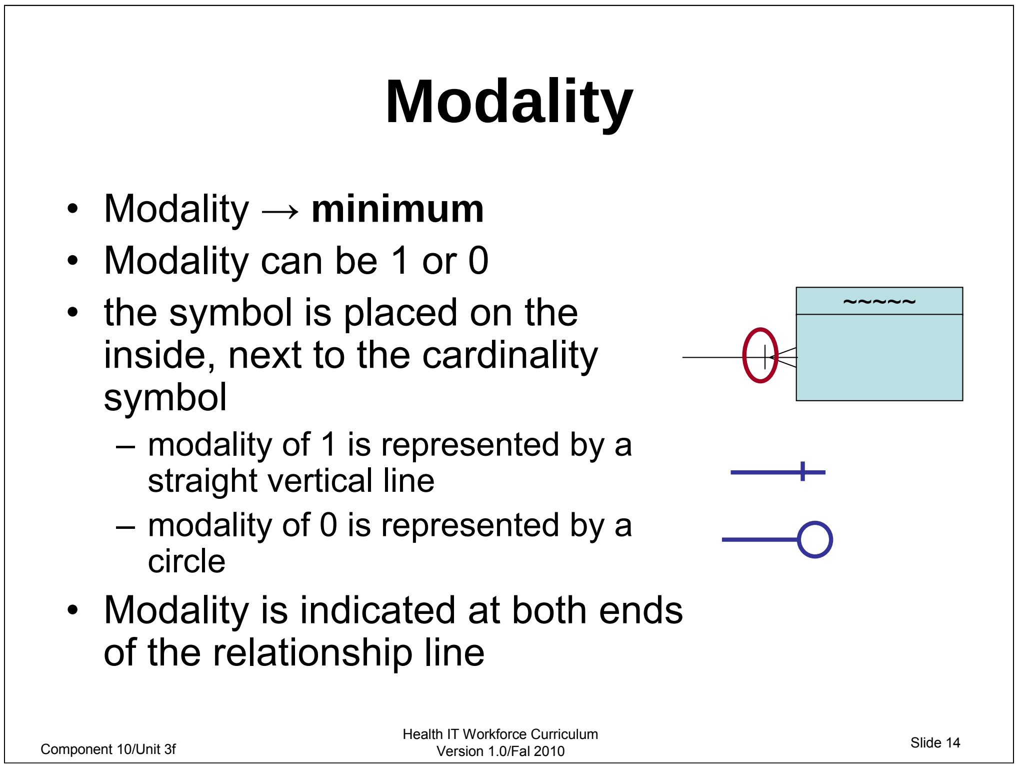


Cardinality

- Cardinality → **maximum**
- Cardinality can be 1 or Many
- the symbol is placed on the outside of the relationship line, closest to the entity
  - cardinality of 1 is represented by a straight vertical line
  - cardinality of Many is represented by a "crow's foot"
- Cardinality is indicated at both ends of the relationship line

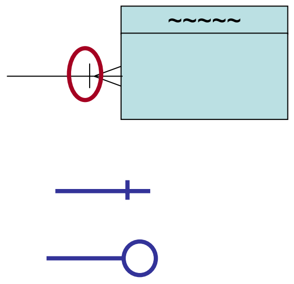
Component 10/Unit 3f | Health IT Workforce Curriculum Version 1.0/Fal 2010 | Slide 13

Cardinality indicates the **maximum** number of times an instance of one entity can be associated with instances in the related entity. Cardinality can have the values of one or many, no more detail than that. It is either one or more than one. On the relationship line, the cardinality is the closest to the entity box. The cardinality symbol in the diagram on the slide is in the red circle. Cardinality is indicated at BOTH ends of the relationship line, so there is a left to right cardinality and a right to left cardinality.



Modality

- Modality → **minimum**
- Modality can be 1 or 0
- the symbol is placed on the inside, next to the cardinality symbol
  - modality of 1 is represented by a straight vertical line
  - modality of 0 is represented by a circle
- Modality is indicated at both ends of the relationship line

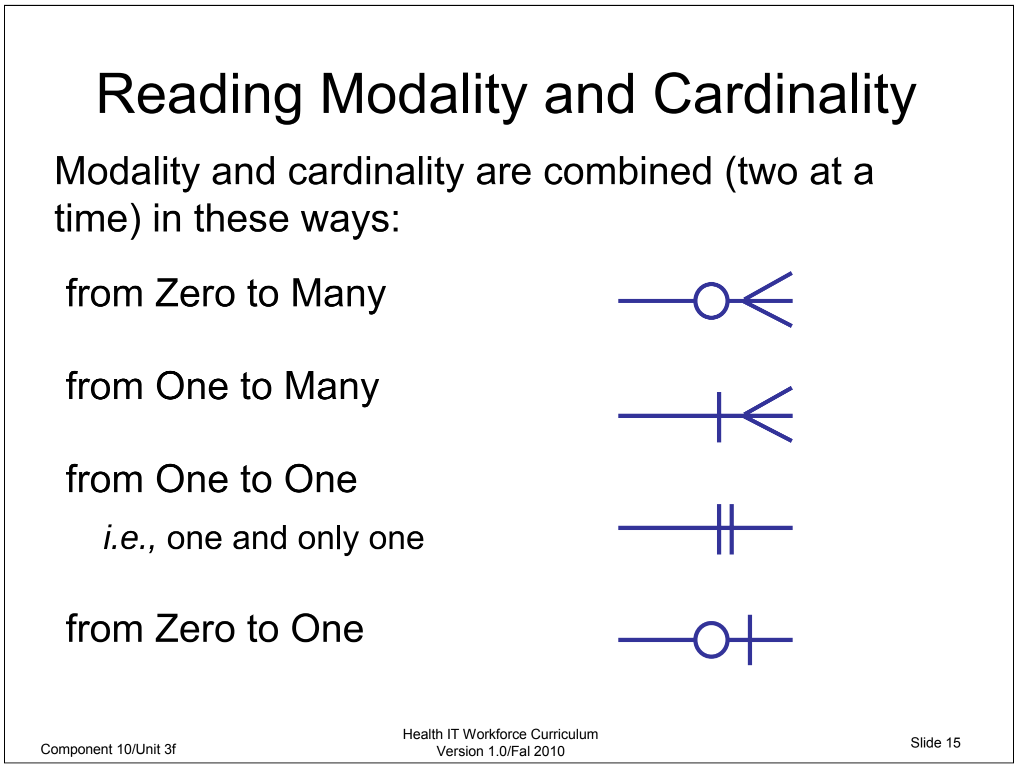


Component 10/Unit 3f

Health IT Workforce Curriculum
Version 1.0/Fal 2010

Slide 14

Modality indicates the **minimum** number of times an instance in one entity can be associated with an instance in the related entity. Modality can have the values of zero or one, two or three are not allowed. The modality symbol is located next to the cardinality symbol, on the inside, i.e., NOT next to the entity box. A Modality of one is denoted by a straight vertical line and a modality of zero is denoted by a circle. Like Cardinality, modality is indicated at both ends of the relationship.



Reading Modality and Cardinality

Modality and cardinality are combined (two at a time) in these ways:

from Zero to Many

from One to Many

from One to One

i.e., one and only one

from Zero to One

Component 10/Unit 3f

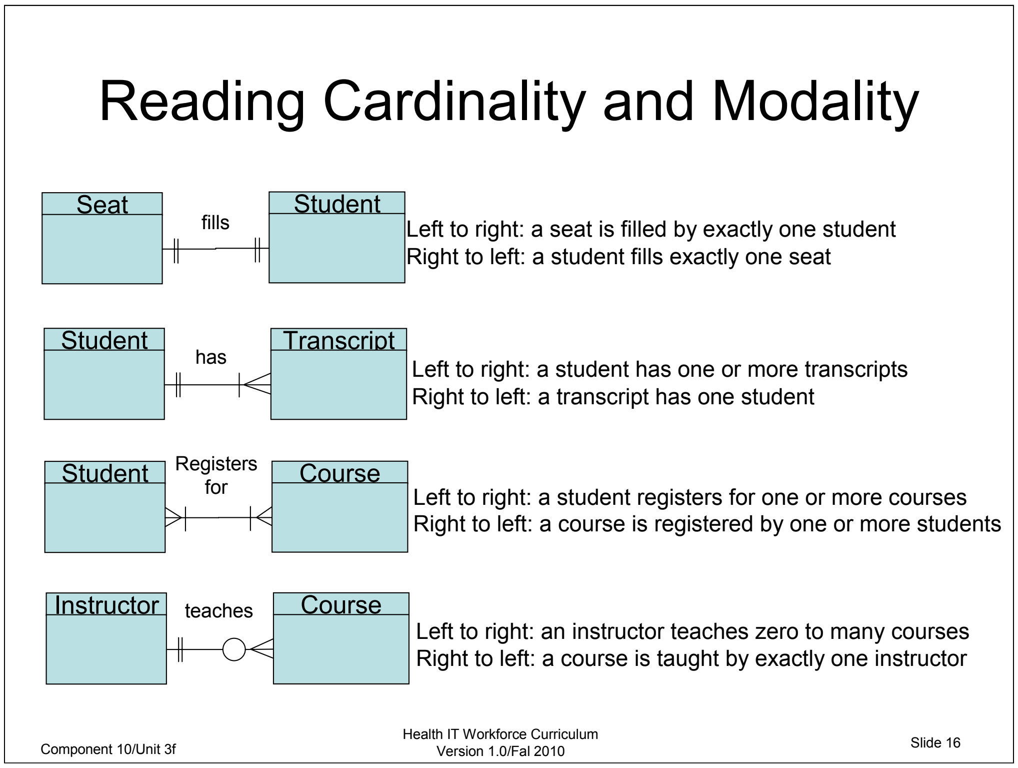
Health IT Workforce Curriculum
Version 1.0/Fal 2010

Slide 15

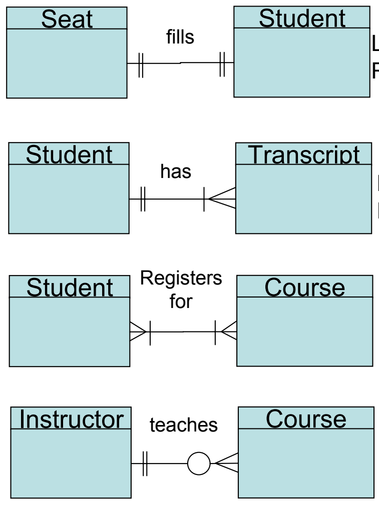
Once you have the three symbols (zero-circle, one-vertical line, and many-crow's foot) committed to memory, reading them is easy. Lets go over them starting from the top of the slide. Modality zero and cardinality of many means that instances of the entity on the left of the relation can have or be associated with from zero to many instances of the entity on the right. Modality one and cardinality many similarly means that instances of the entity on the left of the relation can have or be associated with from one to many instances of the entity on the right.

Modality one and cardinality one means that instances of the entity on the left of the relation can have or be associated with one and only one of instances of the entity on the right. Modality of zero and cardinality of one means that instances of the entity on the left of the relation can have or be associated with from zero to one instances of the entity on the right.

Remember, cardinality and modality specify a range !



Reading Cardinality and Modality



Left to right: a seat is filled by exactly one student
Right to left: a student fills exactly one seat

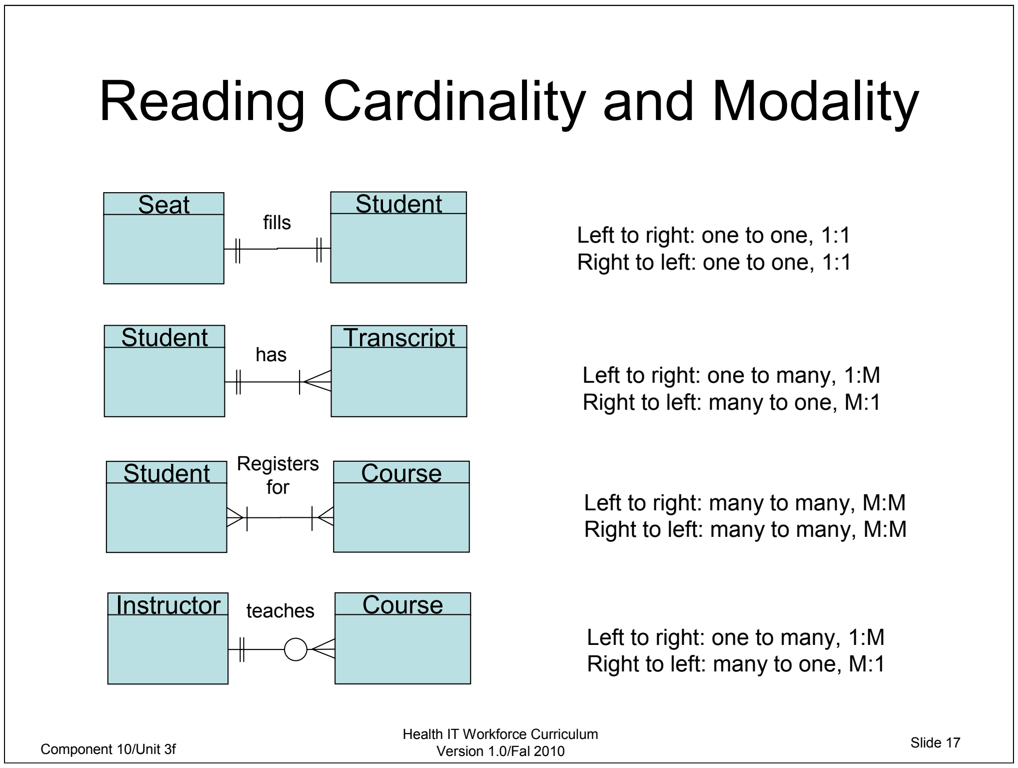
Left to right: a student has one or more transcripts
Right to left: a transcript has one student

Left to right: a student registers for one or more courses
Right to left: a course is registered by one or more students

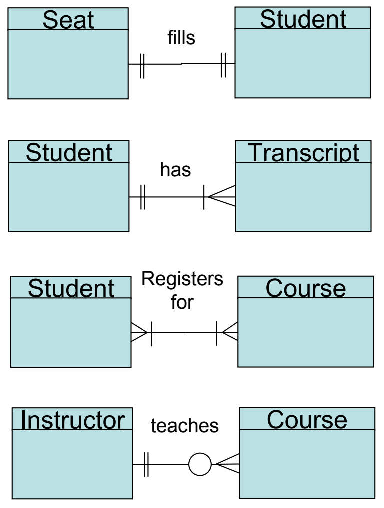
Left to right: an instructor teaches zero to many courses
Right to left: a course is taught by exactly one instructor

Component 10/Unit 3f Health IT Workforce Curriculum Version 1.0/Fal 2010 Slide 16

On an ERD, modality and cardinality look like the examples in the slide. Remember that thse are shown on both sides of a relationship, thus, there is a left-to-right AND a right-to-left reading direction! Pause the slides and read through each of the examples.



Reading Cardinality and Modality



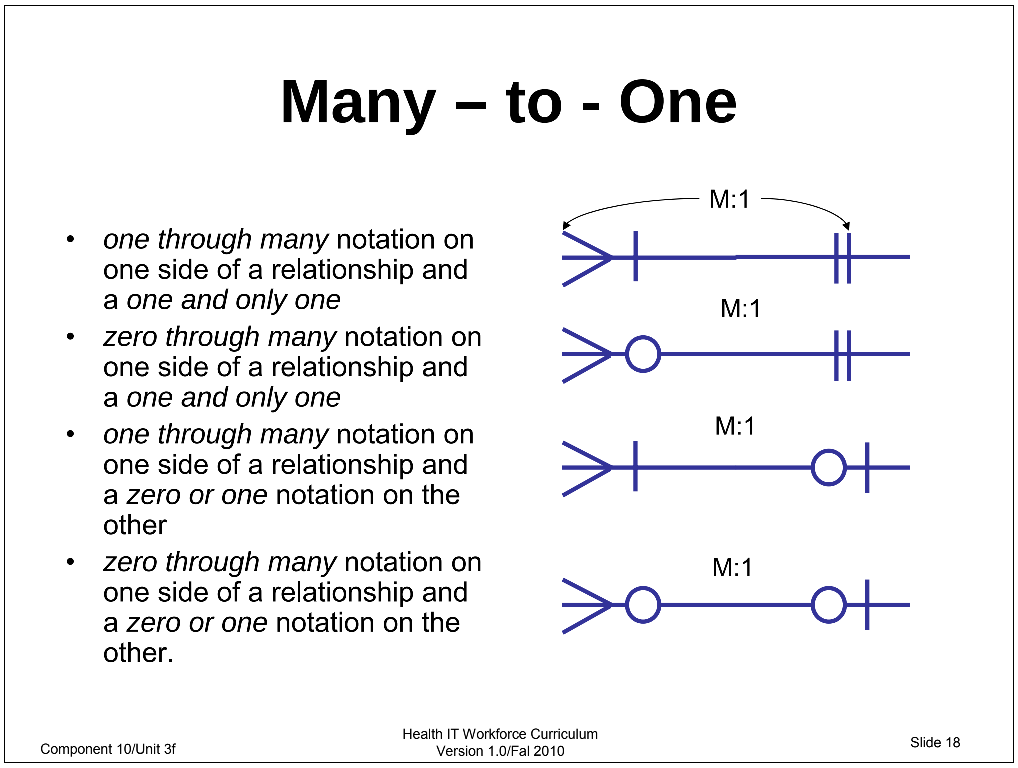
Left to right: one to one, 1:1
Right to left: one to one, 1:1

Left to right: one to many, 1:M
Right to left: many to one, M:1

Left to right: many to many, M:M
Right to left: many to many, M:M

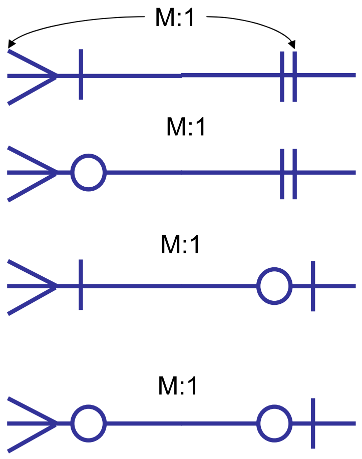
Left to right: one to many, 1:M
Right to left: many to one, M:1

When people talk about relationships between entities, they use phrases like one-to-one, many-to-one, or many-to-many. When people refer to the relationships in this way, they are calling out the cardinality (maximum number allowed) only, i.e., not mentioning modality (the minimum). Pause the slide and read through the relationships for each example. Note that the cardinality is used in both reading directions.



Many – to - One

- *one through many* notation on one side of a relationship and a *one and only one*
- *zero through many* notation on one side of a relationship and a *one and only one*
- *one through many* notation on one side of a relationship and a *zero or one* notation on the other
- *zero through many* notation on one side of a relationship and a *zero or one* notation on the other.

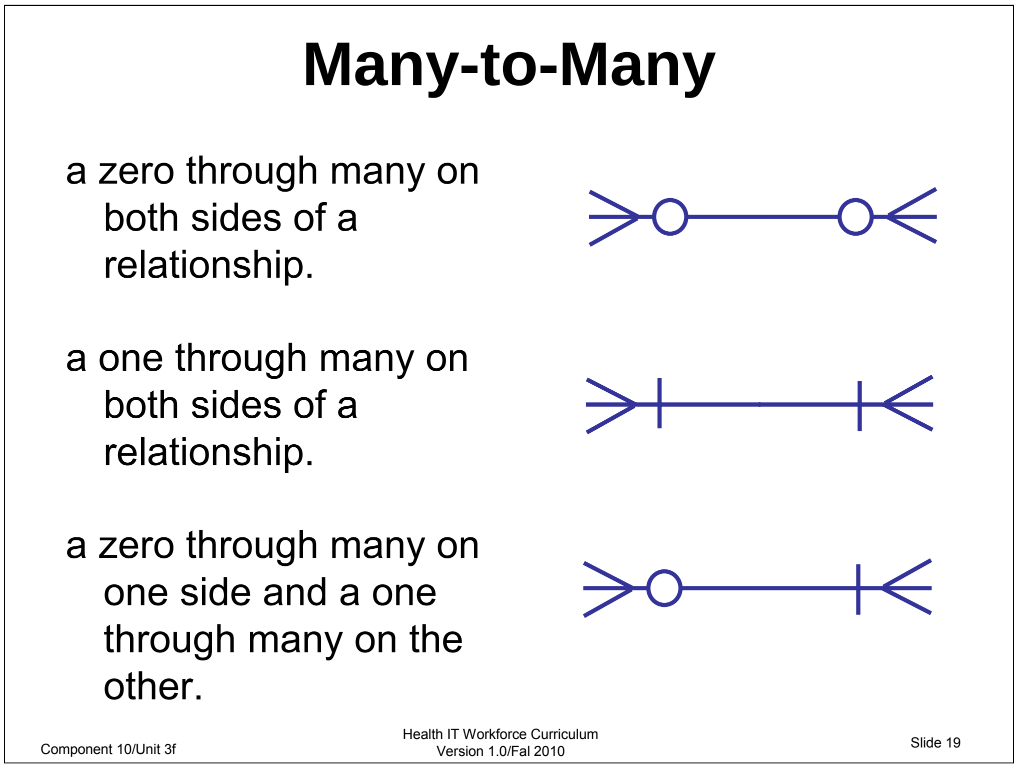


Component 10/Unit 3f

Health IT Workforce Curriculum
Version 1.0/Fal 2010

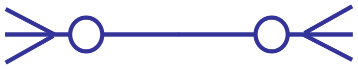
Slide 18

Because the x-to-x way of referring to relationships does not account for the modality, there are four different ways that a Many-to-One relationship can occur bassed on different combinations of modality. The four varieties of a Many-to-One relationship are shown on the slide.

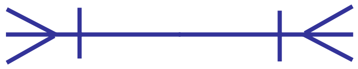


Many-to-Many

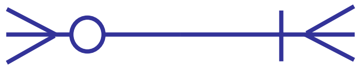
a zero through many on both sides of a relationship.



a one through many on both sides of a relationship.



a zero through many on one side and a one through many on the other.

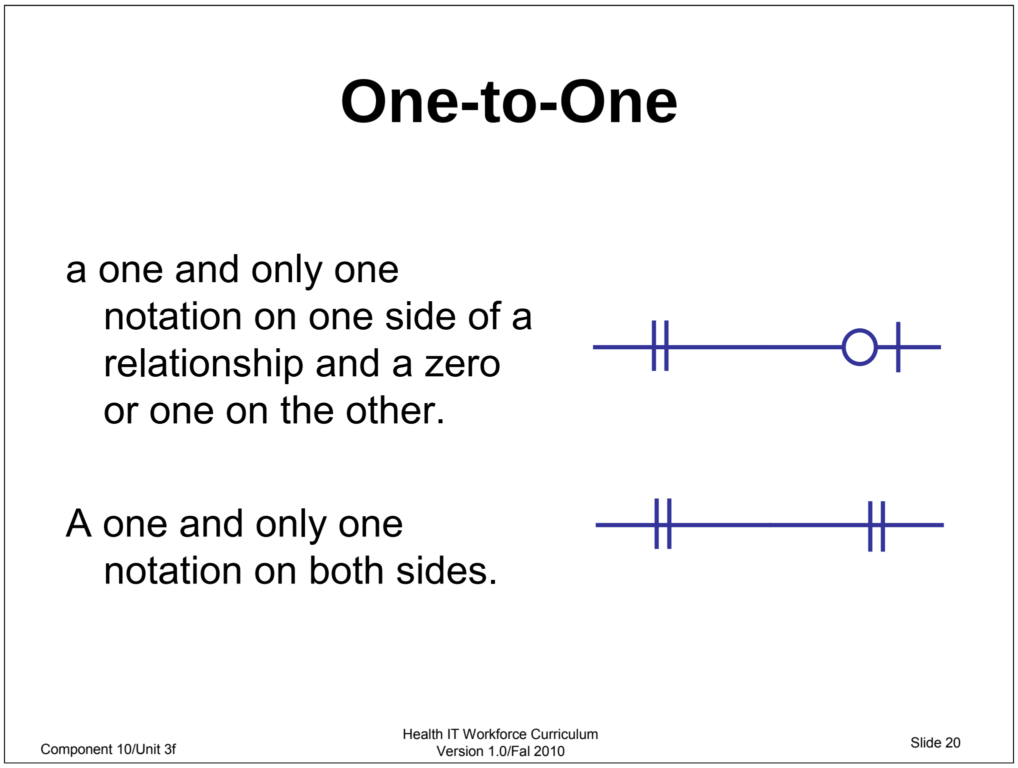


Component 10/Unit 3f

Health IT Workforce Curriculum
Version 1.0/Fal 2010

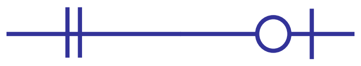
Slide 19

Similarly, because the x-to-x way of referring to relationships does not account for the modality, there are three different ways that a Many-to-Many relationship can occur based on different combinations of modality. The three varieties of a Many-to-Many relationship are shown on the slide.

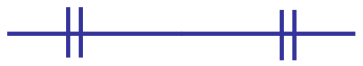


One-to-One

a one and only one notation on one side of a relationship and a zero or one on the other.



A one and only one notation on both sides.

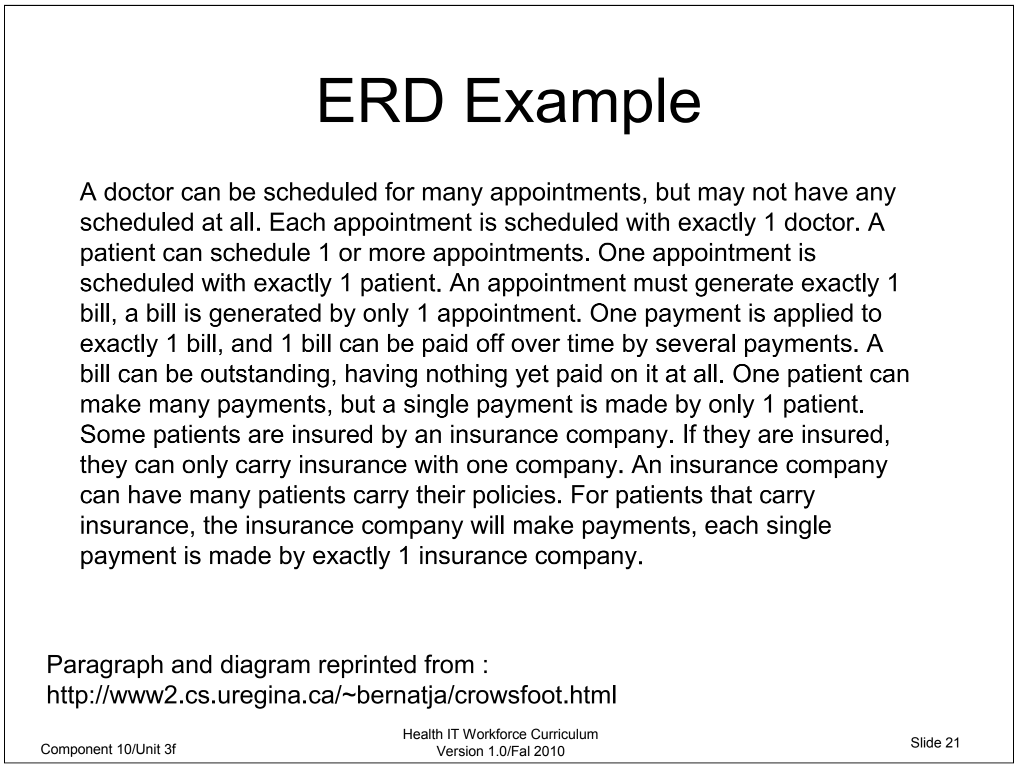


Component 10/Unit 3f

Health IT Workforce Curriculum
Version 1.0/Fal 2010

Slide 20

And finally, because the x-to-x way of referring to relationships does not account for the modality, there are two different ways that a One-to-One relationship can occur based on different combinations of modality. The two varieties of a One-to-One relationship are shown on the slide.



ERD Example

A doctor can be scheduled for many appointments, but may not have any scheduled at all. Each appointment is scheduled with exactly 1 doctor. A patient can schedule 1 or more appointments. One appointment is scheduled with exactly 1 patient. An appointment must generate exactly 1 bill, a bill is generated by only 1 appointment. One payment is applied to exactly 1 bill, and 1 bill can be paid off over time by several payments. A bill can be outstanding, having nothing yet paid on it at all. One patient can make many payments, but a single payment is made by only 1 patient. Some patients are insured by an insurance company. If they are insured, they can only carry insurance with one company. An insurance company can have many patients carry their policies. For patients that carry insurance, the insurance company will make payments, each single payment is made by exactly 1 insurance company.

Paragraph and diagram reprinted from :
http://www2.cs.uregina.ca/~bernatja/crowsfoot.html

Component 10/Unit 3f

Health IT Workforce Curriculum
Version 1.0/Fal 2010

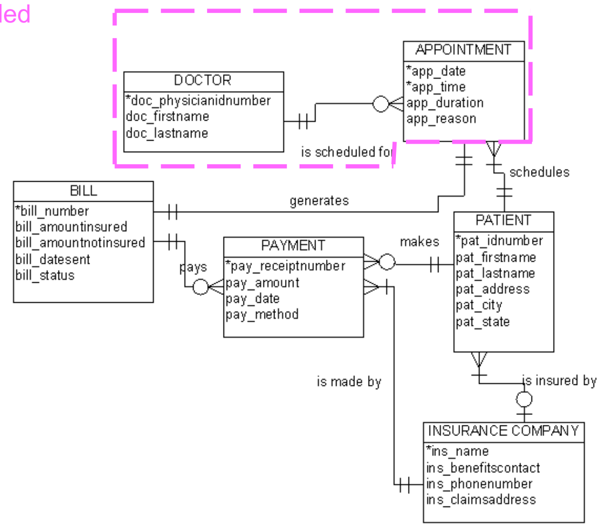
Slide 21

Read the paragraph, we will go through sentence by sentence on the next several slides and trace through the associated ERD.

A data analyst will met with clients (who are subject matter experts) and work with them to elicit the information needed to model their data needs. This paragraph is a very simplified example because each data need is clear and completely stated. Thus it does not represent the information that an analyst encounters in a real-world setting.

ERD Example

A doctor can be scheduled
for many appointments,
may not have any
scheduled at all. Each
appointment is schedul
with exactly 1 doctor.



Component 10/Unit 3f

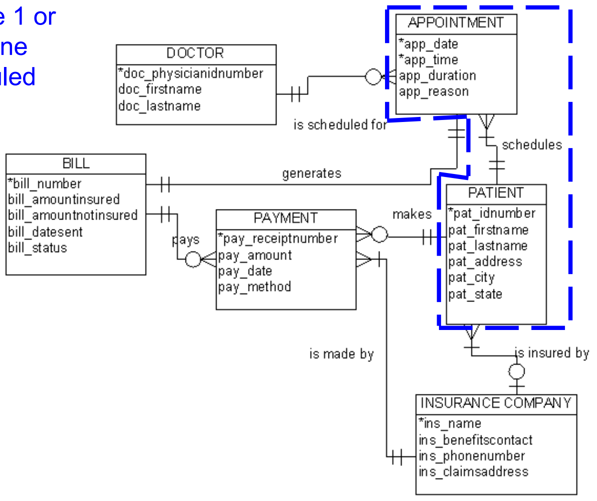
Health IT Workforce Curriculum
Version 1.0/Fal 2010

Slide 22

A doctor can be scheduled for many appointments, but may not have any scheduled at all. (i.e., a doctor can have zero to many appointments scheduled) Each appointment is scheduled with exactly 1 doctor. (i.e., an appointment can only have one doctor associated with it).

ERD Example

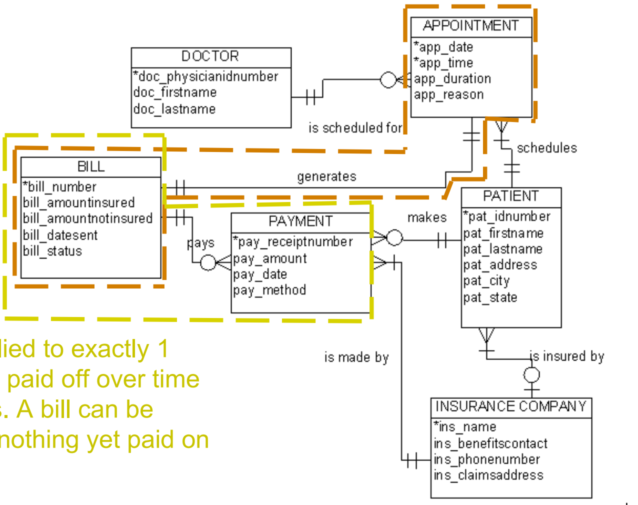
A patient can schedule 1 or more appointments. One appointment is scheduled with exactly 1 patient.



A patient can schedule 1 or more appointments. One appointment is scheduled with exactly 1 patient.

ERD Example

An appointment must generate exactly 1 bill, a bill is generated by only 1 appointment.

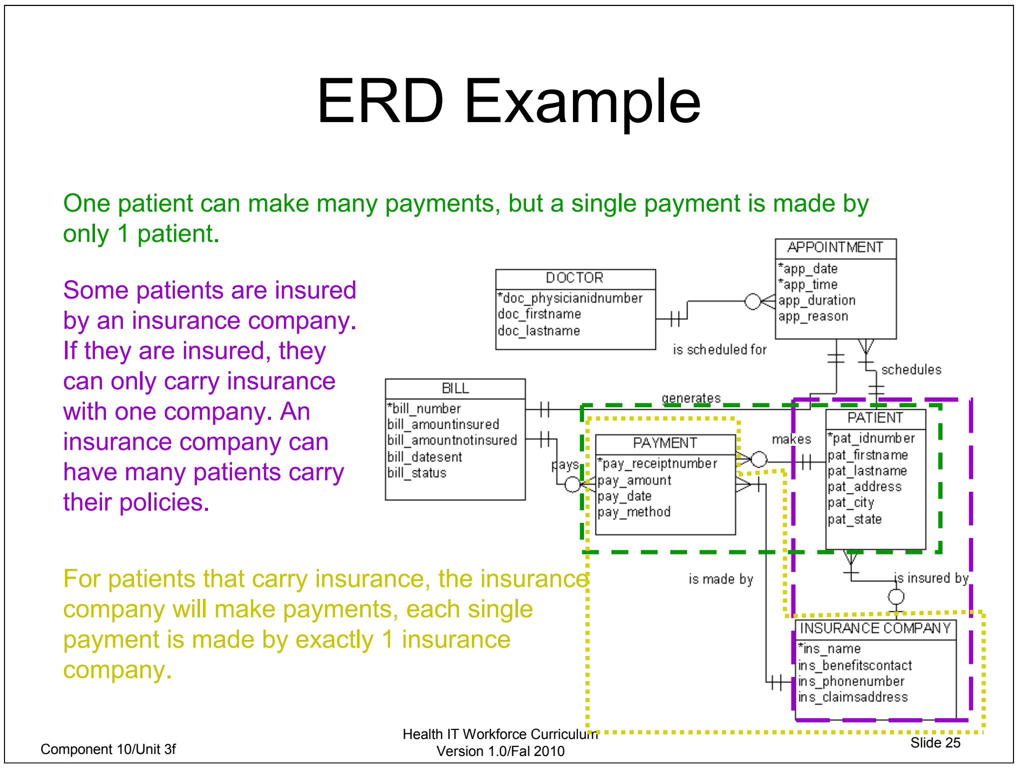


One payment is applied to exactly 1 bill, and 1 bill can be paid off over time by several payments. A bill can be outstanding, having nothing yet paid on it at all.

Component 10/Unit 3f Version 1.0/Fal 2010 Slide 24

In orange, An appointment must generate exactly 1 bill, a bill is generated by only 1 appointment.

In yellow, One payment is applied to exactly 1 bill, and 1 bill can be paid off over time by several payments. A bill can be outstanding, having nothing yet paid on it at all.



ERD Example

One patient can make many payments, but a single payment is made by only 1 patient.

Some patients are insured by an insurance company. If they are insured, they can only carry insurance with one company. An insurance company can have many patients carry their policies.

For patients that carry insurance, the insurance company will make payments, each single payment is made by exactly 1 insurance company.

Component 10/Unit 3f

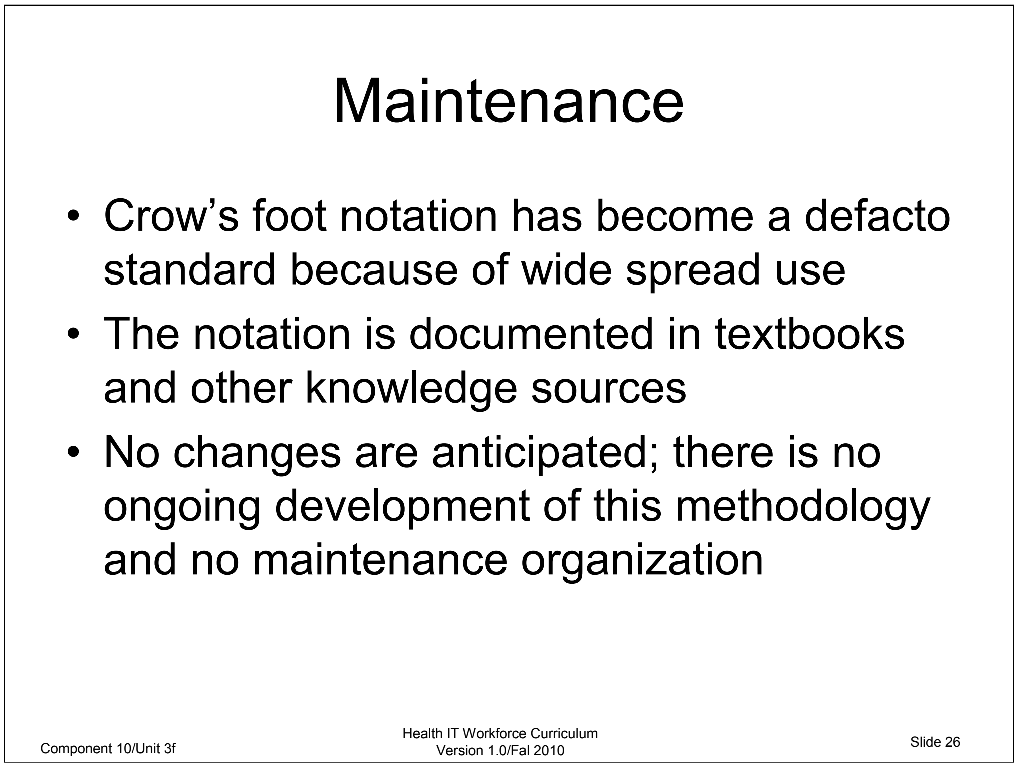
Health IT Workforce Curriculum
Version 1.0/Fal 2010

Slide 25

In green, one patient can make many payments, but a single payment is made by only 1 patient.

In purple, some patients are insured by an insurance company. If they are insured, they can only carry insurance with one company. An insurance company can have many patients carry their policies.

In yellow, for patients that carry insurance, the insurance company will make payments, each single payment is made by exactly 1 insurance company.

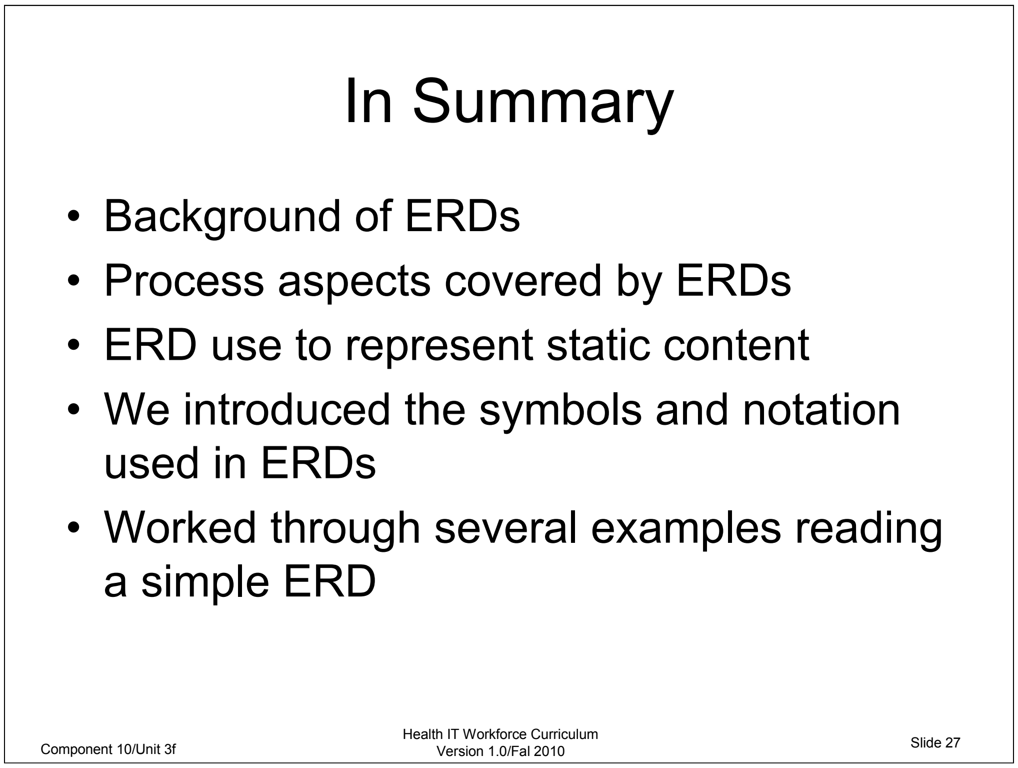


Maintenance

- Crow's foot notation has become a defacto standard because of wide spread use
- The notation is documented in textbooks and other knowledge sources
- No changes are anticipated; there is no ongoing development of this methodology and no maintenance organization

Component 10/Unit 3f
Health IT Workforce Curriculum
Version 1.0/Fal 2010
Slide 26

Crow's foot notation is one of several ways to represent static data content. Because of widespread use, it has become a defacto standard. Importantly, the notation encompases key information about static content that are needed to specify a database. Most likely for this reason, the concepts represented in this notation have been carried forward and built upon in UML class diagrams.



In Summary

- Background of ERDs
- Process aspects covered by ERDs
- ERD use to represent static content
- We introduced the symbols and notation used in ERDs
- Worked through several examples reading a simple ERD

Component 10/Unit 3f

Health IT Workforce Curriculum
Version 1.0/Fal 2010

Slide 27

As a workflow process analyst and re-design specialist, you may encounter ERDs and need to get information from them about data that are used or generated in a process. Thus, being able to recognize an ERD and read pertinent information from one is important.

In summary, we covered the background of ERDs and process aspects covered by them (i.e., static content).

We introduced the symbols and notation used in ERDs, and worked through several examples reading a simple ERD. You should now be able to recognize an ERD and be able to read and interpret the diagram. For more exposure to ERDs and the relational data model, we suggest a course in relational databases.

References

1.) Chen, P. The Entity-Relationship Model: Toward a Unified View of Data (1976), ACM Transactions on Database Systems, 1:9-36.

2.) Bachman, C. W. (1969) Data Structure Diagrams. DATA BASE 1(2): 4-10

3.) Entity-relationship model Wikipedia topic, http://en.wikipedia.org/wiki/Entity_relationship_diagram#cite_note-0

Other reading

Codd, E.F., A relational model of data for large shared databanks. Communications of the ACM, vol 13 no 6. 1970. http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf