Data Modeling Using Oracle (Barker Notations)

cetinerg@itu.edu.tr
Assoc.Prof.Dr.B.Gültekin Çetiner

DATA MODELING

- ***Entity Definition and Relations***
- ***Defining Attributes and Unique Identifiers***
- ***Normalizing Data Model***
- ***Understanding Advanced Relations***
- ***Transform Data Model to Database (Designing Database from Data Model)***

- *Defining Entities*
- *Understanding Relations*

cetinerg@itu.edu.tr

Defining Entities

- *Defining Entities*

cetinerg@itu.edu.tr

Assoc.Prof.Dr.B.Gültekin Çetiner

Defining Entities

- **Starting point in developing a good data model is a good set of data requirements.**

These requirements might be in forms of
**** interview notes,***
**** verbal conversations with the user and,***
**** formal requirements specification document (RSD).***

Defining Entities

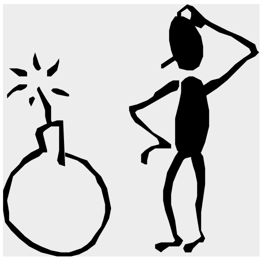
- These requirements may include information about just data, or about the data and the business functions * which use this data.

* Functions required to be implemented in a business without regarding the knowledge of how to do it.

Defining Entities

- **First step is to digest information and then to find the entities* about which the business needs to store data.**

*** Entity: A thing of significance, whether concrete or abstract, about which information needs to be known or held.**

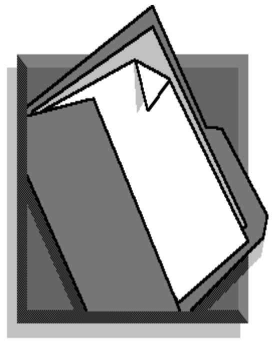


Defining Entities

- **What is an entity? Some definitions are as follows:-**

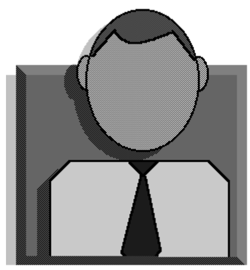
Defining Entities

- **An entity is a thing of significance about which information needs to be stored. In other words, an entity is something important enough to your organization so the organization is willing to spend money to keep records about it: CUSTOMERs, EMPLOYEEs etc.**



Defining Entities

- **An entity is a class or category of thing. A single employee is not an entity, but the general category EMPLOYEE is an entity.**



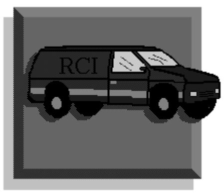
- **An entity is a named thing. It may be a tangible thing, such as a TRUCK, or a concept, such as a COST CENTER.**

Entity Examples

We have already seen some examples of entities,

Some more examples are as follows:-

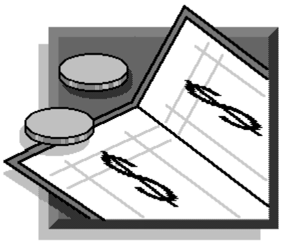
TRUCK



Entity Examples

We have already seen some examples of entities,
Some more examples are as follows:-

BANK ACCOUNT

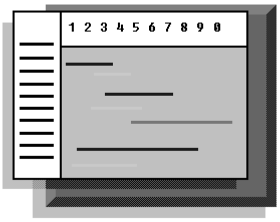


Defining Entities

Entity Examples

We have already seen some examples of entities,
Some more examples are as follows:-

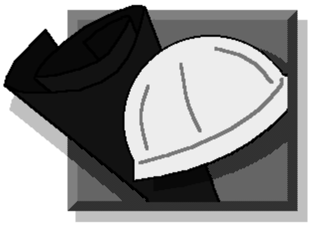
PROJECT



Entity Examples

We have already seen some examples of entities,
Some more examples are as follows:-

CONTRACTOR

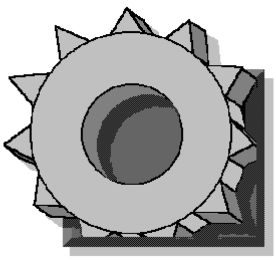


cetinerg@itu.edu.tr
Assoc.Prof.Dr.B.Gültekin Çetiner

Entity Examples

We have already seen some examples of entities,
Some more examples are as follows:-

PART



cetinerg@itu.edu.tr
Assoc.Prof.Dr.B.Gültekin Çetiner

Entity Examples

We have already seen some examples of entities,
Some more examples are as follows:-

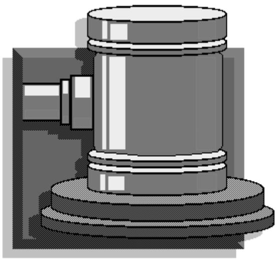
INSURANCE POLICY

Defining Entities

Entity Examples

We have already seen some examples of entities,
Some more examples are as follows:-

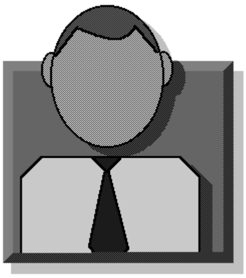
DEFENDANT



Entity Examples

We have already seen some examples of entities,
Some more examples are as follows:-

CUSTOMER



Diagramming Entities

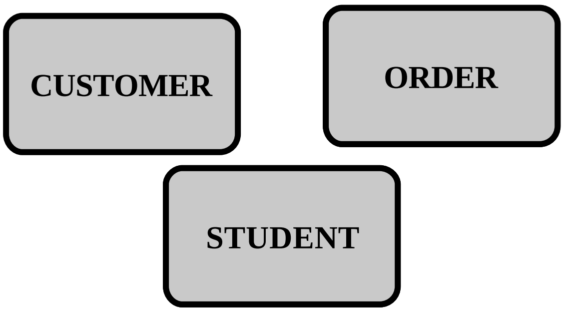
Entities are drawn in diagram as soon as they are identified.

Defining Entities

For now, we just draw boxes to represent entities. You can choose some conventions for drawing entities on E-R Diagrams.

*** Boxes are soft, i.e. with rounded corners.**

For example;



Understanding Relations

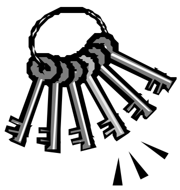
- **Beginning**
- **Identify Relationships**
- **Relationship Sentences**
- **Relationship Names**
- **Optionality in Relationships**
- **Relationship Degree**
- **Diagramming Relationships**
- **Relationship Types**
- **Validating Relationships**

Relationship Sentences

Even though the E-R Diagram consists of boxes and lines just a few words, you as the analyst should be able to "read" it to either a technical Database Administrator or a non-technical business person.

E-R Diagrams are essential for communicating data requirements in business.

The key to make your E-R Diagram "readable" is the Relationship Sentence. It is a complete sentence and can be constructed in any language.



Suppose you have two entities: CUSTOMER and ORDER. We can describe how customers and orders are related with two sentences:

"Each CUSTOMER may be the originator of one or more ORDERs."

"Each ORDER must be placed by one and only one CUSTOMER."

Let's divide one of these sentences.

"Each ORDER must be placed by one and only one CUSTOMER."

The formal syntax of the sentence is:

Each	**ENTITY1**	**{must be}** **or** **{may be}**	**name**	**{one or more}** **or** **{one and only one}**	**ENTITY2**

So, ORDER and CUSTOMER are the entities, and "placed by" is the name of the relationship.

The statement “*must be*” or “*may be*” describes whether the relationship is mandatory or optional.

The statement “*one or more*” or “*one and only one*” describes the cardinality (degree) of relationship.

		{must be}		**{one or more}**	
Each	**ENTITY1**	**or**	**name**	**or**	**ENTITY2**
		{may be}		**{one and only one}**	

We will look at each of these parts of the sentence in detail.

Let’s look at the following sentence:

“Each ORDER must be made by one and only one CUSTOMER”

The choice of term “must be” indicates that an order cannot exist without a customer to place it. This relationship is <u>mandatory</u>.

If there is ORDER then there must be CUSTOMER who made.

Let's look at the other half of the relationship sentence.

"Each CUSTOMER may be the originator of one or more ORDERs"

The choice of the term "may be" indicates that a customer may exist in our database without ever placing an order.

Think of a CUSTOMER who has been sent a CATALOG but never bought anything yet.

Is the relationship mandatory or optional?

"Each EMPLOYEE ____ assigned to one and only one DEPARTMENT"

A. must be

B. may be

Understanding Relations

Is the relationship mandatory or optional?
Choose "must be" or "may be"

"Each PROJECT _______ carried out by one or more EMPLOYEEs"

A. must be
B. may be

cetinerg@itu.edu.tr
Assoc.Prof.Dr.B.Gültekin Çetiner

Understanding Relations

Is the relationship mandatory or optional?
Choose "must be" or "may be"

"Each RESERVATION ________ made by one and only one EMPLOYEE"

A. must be
B. may be.

cetinerg@itu.edu.tr
Assoc.Prof.Dr.B.Gültekin Çetiner

Relationship Degree

Now let's look at the other end of the Relationship Sentences;

"Each ORDER must be made by <u>one and only one</u> CUSTOMER."

"Each CUSTOMER may be the originator of <u>one or more</u> ORDERs."

Understanding Relations

The degree of relationship is stated as either "*<u>one and only one</u>*" or "*<u>one or more</u>*". The "one and only one" is known as a "single-valued" relationship.

"Each ORDER must be made by <u>one and only one</u> CUSTOMER."

“One or more” means “one, or any number”. “One or more” usually used in optional relations. “One or more” allows the CUSTOMER to place one ORDER, a hundred or a thousand. The CUSTOMER may also be in database with zero orders.

This relation is called “many-valued relationship”.

“Each CUSTOMER may be the originator of <u>one or more</u> ORDERs.”

Choose the relationship degree using one of the options.

“Each ORDER must be made up of _______ ORDER LINE ITEMs”

A. “one and only one”
B. “between one and twelve”
C. “one or more”
D. “any number except zero”

Choose the relationship degree using one of the options.

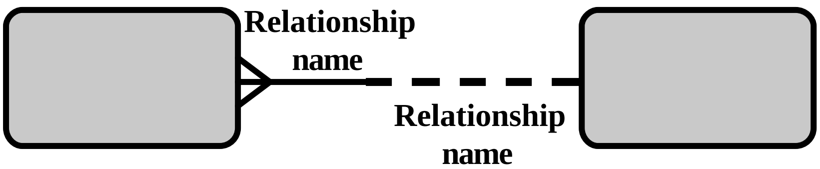
“Each ORDER LINE ITEM must be contained in ________ ORDER”

A. “one and only one”
B. “one or more”
C. “zero, or one or more”
D. “none of the above”

Diagramming Relationships

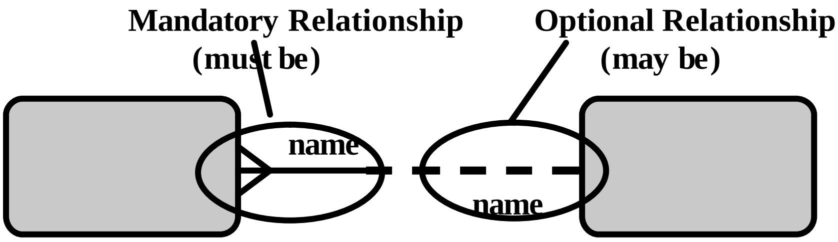
Now it is time to represent all these in an E-R Diagram.

This diagram indicates how relationship names, optionality and degree are indicated.



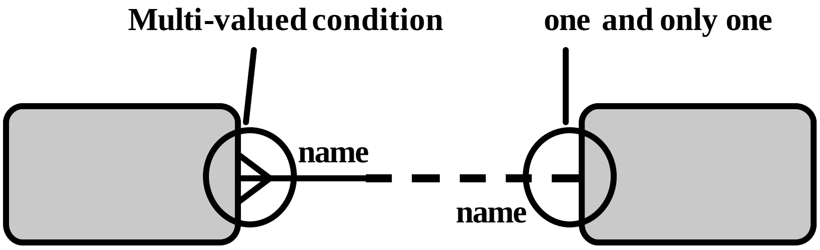
Understanding Relations

Each half of the line is either dashed ( ) to indicate an optional relationship, or solid ( ) to indicate a mandatory one.



Understanding Relations

The "crow's foot" indicates "one or more" or "multi-valued" condition. If there is not a crow's foot then the relationship degree is "one and only one".

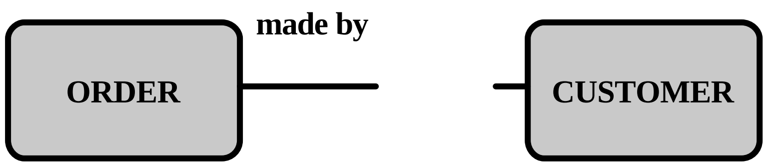


Understanding Relations

"Each ORDER must be made by one and only one CUSTOMER"

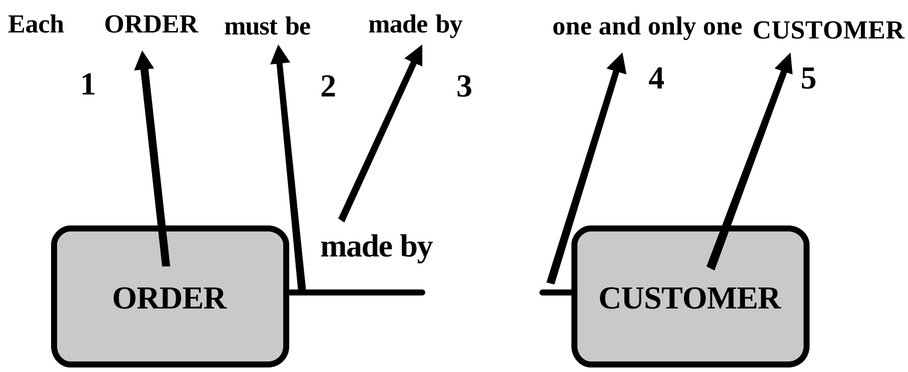
sentence is mandatory and single-valued. Therefore, we draw a solid line from ORDER to CUSTOMER and write the relation name as *made by*.

Absence of crow-foot shows single-valued relation.



Understanding Relations

Read the sentence in the following order;

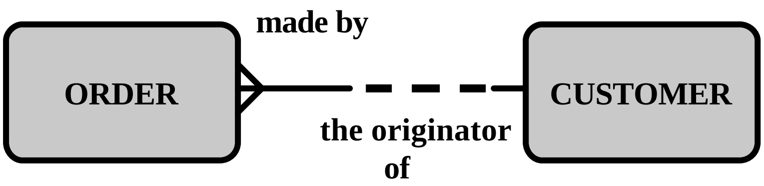


Understanding Relations

Other sentence,

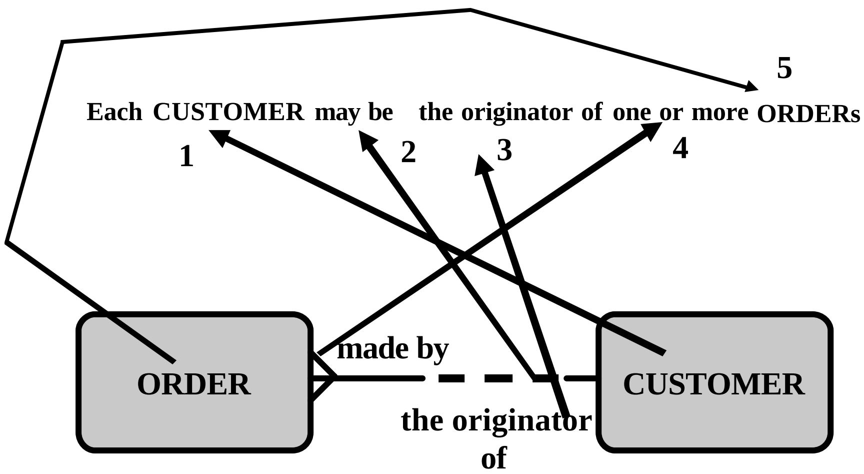
"Each CUSTOMER may be the originator of one or more ORDERs"

Optional and multi-valued. Therefore, a dashed line is drawn from CUSTOMER to ORDER and the relation sentence '*originator of*' is written below this line. A crow-foot is drawn next to ORDER entity to show the multi-valued relation.



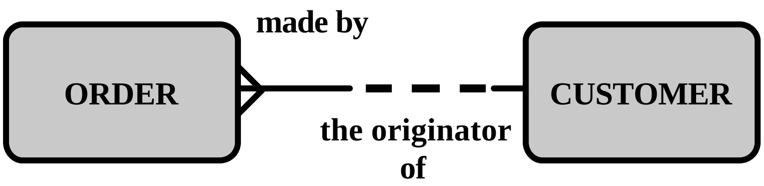
Understanding Relations

You can read this part in the following order;



Understanding Relations

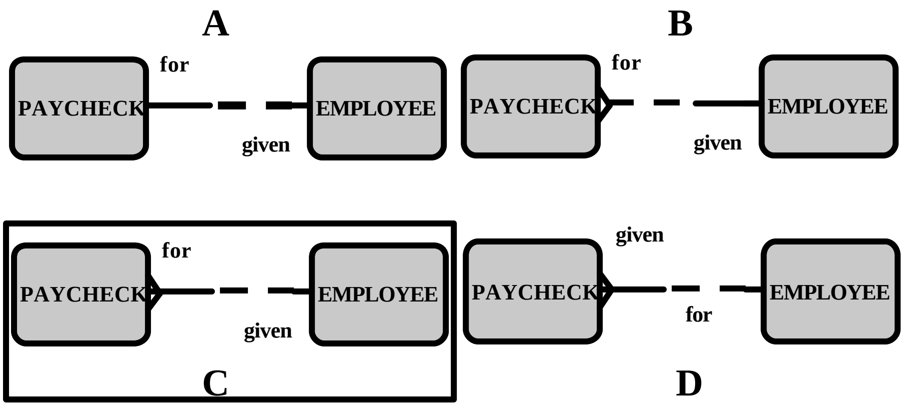
We construct 2 sentences for each pair of entities. The following type of relation is called one-to-many (or many-to-one).



Understanding Relations

Find the correct E-R Diagram for the following sentences.

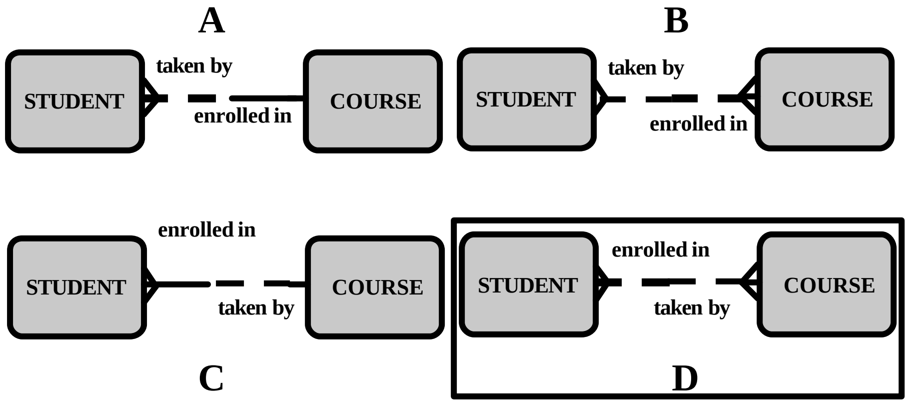
"Each PAYCHECK must be for one and only one EMPLOYEE."
"Each EMPLOYEE may be given one or more PAYCHECKs."



Understanding Relations

Show the diagram for STUDENT and COURSE entities according to following sentences.

"Each STUDENT may be enrolled to one or more COURSEs."
"Each COURSE may be taken by one or more STUDENTs."



cetinerg@itu.edu.tr

Assoc.Prof.Dr.B.Gültekin Çetiner

Understanding Relations

Types of relations

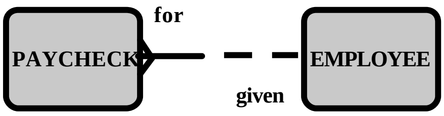
There are 3 types of relations.

cetinerg@itu.edu.tr

Assoc.Prof.Dr.B.Gültekin Çetiner

Understanding Relations

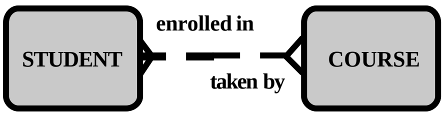
The relation between PAYCHECK and EMPLOYEE is a one-to-many relation. It is also called many-to-one relation.



When finished ER Diagram usually consists of this type of relations.

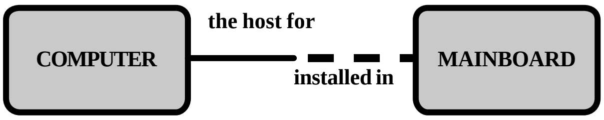
Understanding Relations

The relation between STUDENT and COURSE is many-valued in both sides. This kind of relation is called many-to-many relation.



Understanding Relations

3rd type of relation is single -valued in both sides. For example, COMPUTER and MAINBOARD entities might have this type of relation.



"Each COMPUTER must be the host for one-and-only one MAINBOARD."

"Each MOTHERBOARD may be installed in one and only one COMPUTER."

Understanding Relations

This relationship is a fairly unusual relationship. They are not seen during analysis and design stages. But may appear in a finished database.

Identify Attributes

Diagram Attributes

An attribute is a fact about an entity, for example:

***name* and *address* are facts about EMPLOYEE**

***altitude* and *mean January temperature* are facts about CITY**

***name* and *first enrollment date* are facts about STUDENT**

EMPLOYEE
name
address

CITY
altitude
mean January temperature

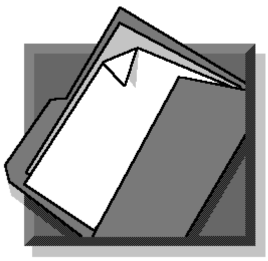
STUDENT
name
first enrollment date

cetinerg@itu.edu.tr Assoc.Prof.Dr.B.Gültekin Çetiner

Identify Attributes

Some possible sources are:

Headings from existing printed reports

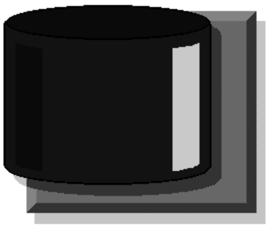


cetinerg@itu.edu.tr Assoc.Prof.Dr.B.Gültekin Çetiner

Identify Attributes

Some possible sources are:

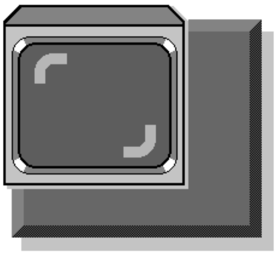
Headings from existing printed reports
Fields stored in existing files and databases



Identify Attributes

Some possible sources are:

Headings from existing printed reports
Fields stored in existing files and databases
Captions from screens



Identify Attributes

Some possible sources are:

Headings from existing printed reports
Fields stored in existing files and databases
Captions from screens
Nouns that business people use in everyday conversations

Identify Attributes

Some possible sources are:

Headings from existing printed reports
Fields stored in existing files and databases
Captions from screens
Nouns that business people use in everyday conversations
Values used for sorting reports

Identify Attributes

Some possible sources are :

Headings from existing printed reports
Fields stored in existing files and databases
Captions from screens
Nouns that business people use in everyday conversations
Values used for sorting reports

When collecting requirements from existing procedures or software, BEWARE OF DERIVED DATA.

Derived data are not attributes

cetinerg@itu.edu.tr
Assoc.Prof.Dr.B.Gültekin Çetiner

Identify Attributes

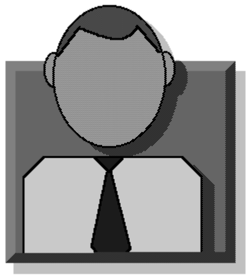
Name Attributes

Names of attributes should be the names that people in the business use every day when they discuss this item of information.

cetinerg@itu.edu.tr
Assoc.Prof.Dr.B.Gültekin Çetiner

Identify Attributes

For example, if you think of an attribute called *Record number* for an entity called EMPLOYEE, but the people in business call it *Employee ID* then you should use *Employee ID* for the attribute name.



EMPLOYEE
Employee ID

Identify Attributes

Name Attributes: Uniqueness

Attribute names must, at least, be unique within a single entity.

If possible, they should be unique across the entire E-R Diagram. That is hard to do, but it is a good goal to aim for.

Identify Attributes

For example ;

EMPLOYEE
*** id**
*** name**
*** last name**
o title
*** sex**
o weight

The "asterisks" indicate that an EMPLOYEE must include values for *id, last name, first name* and *sex.*

The "circles" show that there may or may not be a *title* or *weight* known for each EMPLOYEE.

Identify Attributes

For the entity SHIP, the attributes *name, country of origin, registration number, length, width,* and *class* have been identified. *Width* and *class* are often unknown, but the other attributes are required. Which of the following diagrams correctly models SHIP and its attributes?

A

SHIP
o name
o country of origin
o registration number
*** length**
o width
*** class**

B

SHIP
*** name**
*** country of origin**
*** registration number**
*** length**
 # width
 # class

C

SHIP
*** name**
*** country of origin**
*** registration number**
*** length**
o width
o class

D

SHIP
*** name**
*** country of origin**
*** registration number**
o length
o width
o class

Identify Attributes

Distinguishing Attributes and Entities

Sometimes it is not exactly clear whether a piece of information is an attribute of entity, or an entity itself. A noun might be en entity or an attribute, depending on the business requirements.

cetinerg@itu.edu.tr

Assoc.Prof.Dr.B.Gültekin Çetiner

Identify Attributes

If you are in doubt, ask this question about the thing:

“Do we need to store any facts about this thing?”

If the answer is YES, then it is an ENTITY.

If the answer is NO, then it is an ATTRIBUTE of an entity.

cetinerg@itu.edu.tr

Assoc.Prof.Dr.B.Gültekin Çetiner

Unique Identifiers (UID) and Primary Keys

Identifying UIDs

Diagram UIDs in E-R Diagram

UIDs via Relationships

Composite UIDs

Artificial UIDs

Candidate UIDs

Assigning Unique Identifiers

Assigning Unique Identifiers

UIDs and Primary Keys

If you have worked with almost any kind of files or databases, you are probably familiar with Primary Keys. UIDs and Primary Keys are not exactly the same thing.

When our logical data model is converted into a physical database design, the UIDs will become the Primary Keys of the files, segments, or tables.

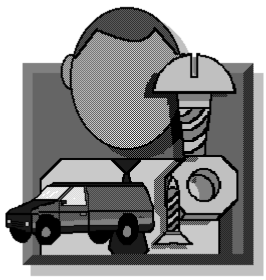
Unique Identifier is the term for Data Model
Primary Key is the term for Physical Database

Assigning Unique Identifiers

Identify UIDs

Usually, all you have to do to find a UID is to ask a business person,

"How do you uniquely identify/differentiate a customer/part/course/student/employee/truck?"



Assigning Unique Identifiers

Diagram UIDs

Now let’s consider the EMPLOYEE entity. Each employee has a unique badge number. So the attribute “badge number” should be the UID for the EMPLOYEE entity.

EMPLOYEE
*** badge number**
*** first name**
*** last name**
o job title

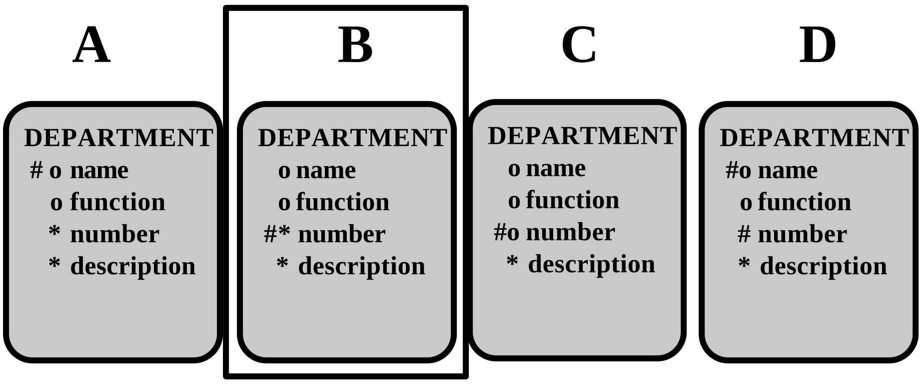
Assigning Unique Identifiers

You indicate this with a # (pound sign) before the attribute name and optionality tag. Each instance of EMPLOYEE is uniquely identified by the attribute “*badge number.*”

EMPLOYEE
#* badge number
*** first name**
*** last name**
o job title

Assigning Unique Identifiers

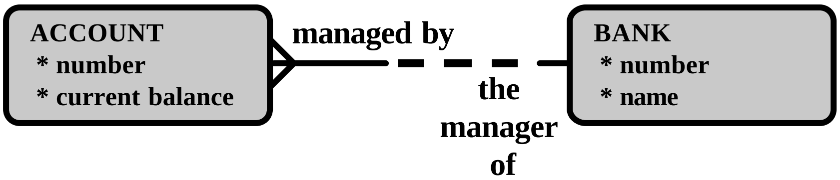
How do you model the UID for the entity DEPARTMENT.



Assigning Unique Identifiers

UIDs via Relationships

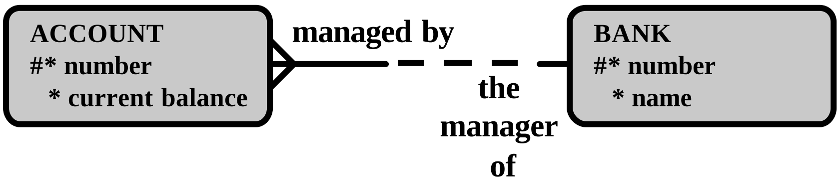
Suppose we have two related entities, BANK and ACCOUNT. Let's determine the UID for each entity. Banks have unique numbers assigned by some government body; let's call this attribute "number" and let it be the UID of BANK.



Assigning Unique Identifiers

So we tag the BANK's attribute "*number*" with a "*#.*" Now what is the UID of ACCOUNT? The problem is that accounts do not necessarily have unique numbers across all banks in the world or even in the country. The account number is usually just a number that is unique within a single bank.

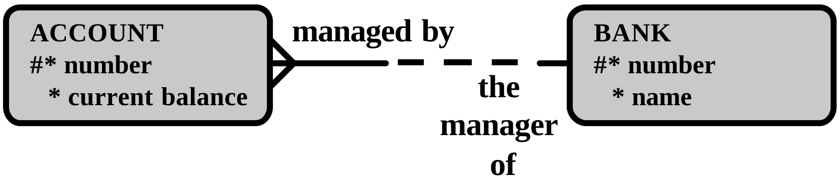
In other words, different BANKs may use the same ACCOUNT numbers. Therefore, "number" in ACCOUNT cannot uniquely identify all instances of ACCOUNT.



Assigning Unique Identifiers

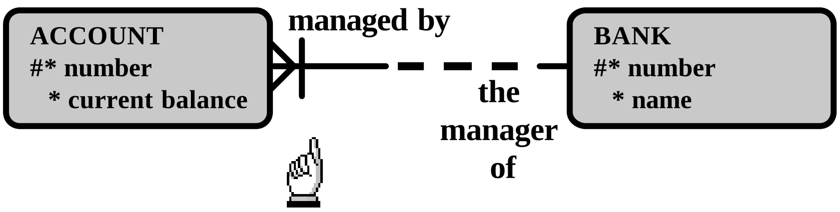
In data modeling terms, an account is uniquely identified by its number, plus its relationship to BANK.

The UID for ACCOUNT is composed of both the attribute "number" and the BANK that manages the ACCOUNT.



Assigning Unique Identifiers

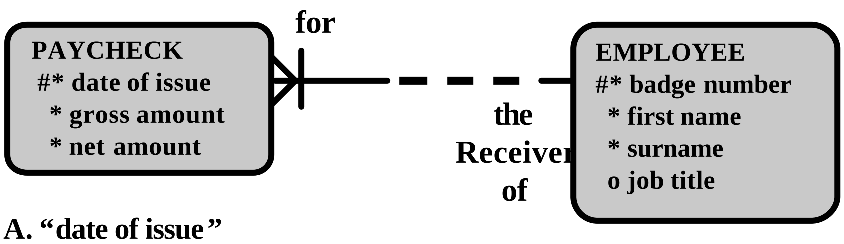
In an E-R Diagram, a relationship which is part of a UID is indicated by a bar across the relationship line, like this:



The bar across the relationship from ACCOUNT to BANK indicates that the relationship is part of the UID of ACCOUNT.

Assigning Unique Identifiers

What is the UID of the entity PAYCHECK?



A. “date of issue”

B. the EMPLOYEE that the PAYCHECK is for

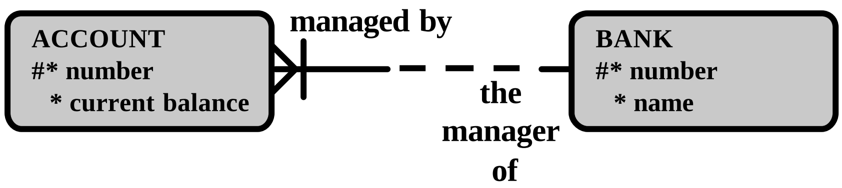
C. “badge number”

D. “date of issue” and the EMPLOYEE that the PAYCHECK is for

Assigning Unique Identifiers

Composite UIDs

In the previous example, a bank account is identified by its own number, plus the number of the bank the ACCOUNT is managed by.



Assigning Unique Identifiers

This illustrates a very common situation: the UID of an entity may consist of any number of attributes and/or relationships. These are called *"compound"* UIDs, or *"composite"* UIDs.

Resolving Many-to-many Relations

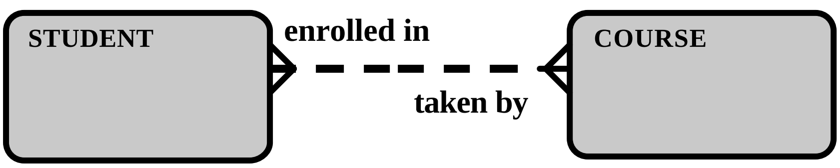
- **Resolving Many-to-many Relations**

Resolving Many-to-many Relations

In a many-to-many relationship, relationship is multi-valued at both ends.

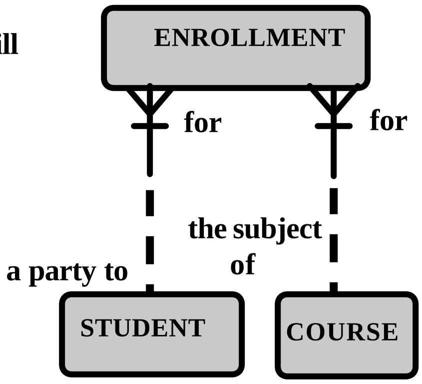
For example ;

“Each STUDENT may be enrolled in one or more COURSEs”
“Each COURSE may be taken by one or more STUDENTs”



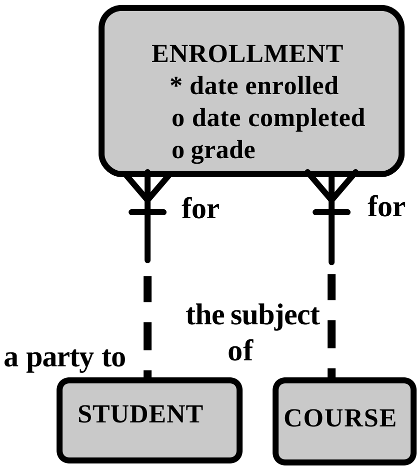
Resolving Many-to-many Relationships

To accommodate these facts about the relationship we "resolve" a many-to-many relationship by adding a third entity, ENROLLMENT, which will associate the two entities.



Assoc.Prof.Dr.B.Gültekin Çetiner

The facts about the relationship between STUDENT and COURSE now become attributes of the entity ENROLLMENT.

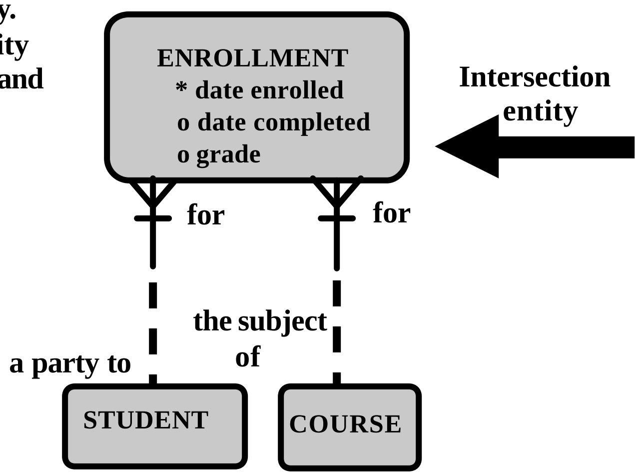


Assoc.Prof.Dr.B.Gültekin Çetiner

Resolving Many-to-many Relations

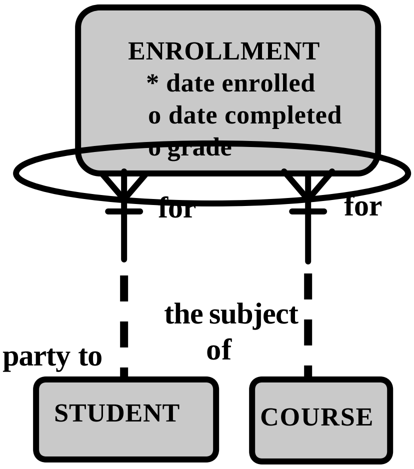
Intersection Entities

A new entity added to resolve a many-to-many relationship is called an intersection entity.
We also call this entity "association entity" and "junction entity".



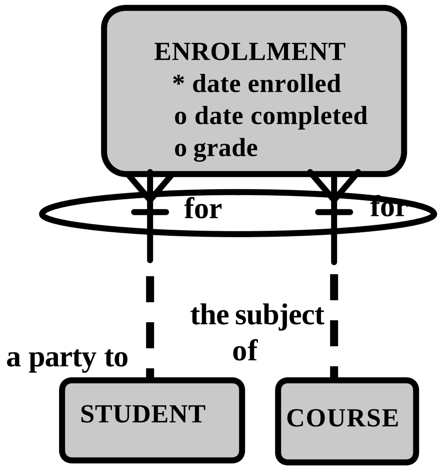
Resolving Many-to-many Relations

Note that any intersection entity will be on the many-valued end of the relationships to the two original entities.



Resolving Many-to-many Relations

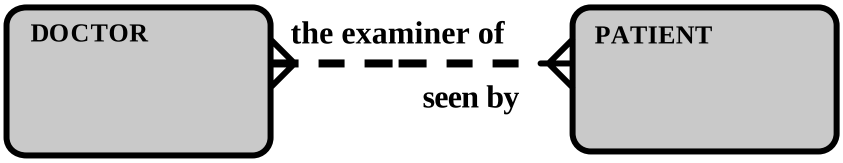
And the Unique Identifier of an intersection entity will usually be its <u>relationships</u> to the two original entities, not any of its attributes.



cetinerg@itu.edu.tr Assoc.Prof.Dr.B.Gültekin Çetiner

Resolving Many-to-many Relations

If one DOCTOR may be the examiner of many PATIENTs, and a PATIENT may e seen by many DOCTORs over time, what is the most likely intersection entity to connect them?



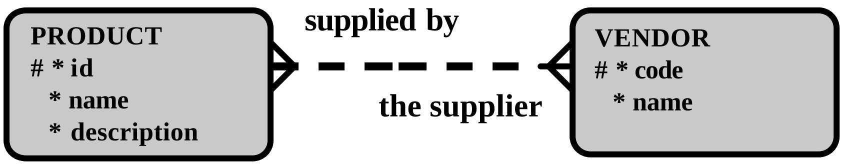
A. patient number **B. clinic address**

C. VISIT **D. DOCTOR NUMBER**

cetinerg@itu.edu.tr Assoc.Prof.Dr.B.Gültekin Çetiner

Resolving Many-to-many Relations

What is the likely intersection entity between PRODUCT and VENDOR if there is many-to-many relationship?



A. CATALOG **B. CUSTOMER**

C. PRICE **D. DATE OF DELIVERY**

cetinerg@itu.edu.tr Assoc.Prof.Dr.B.Gültekin Çetiner