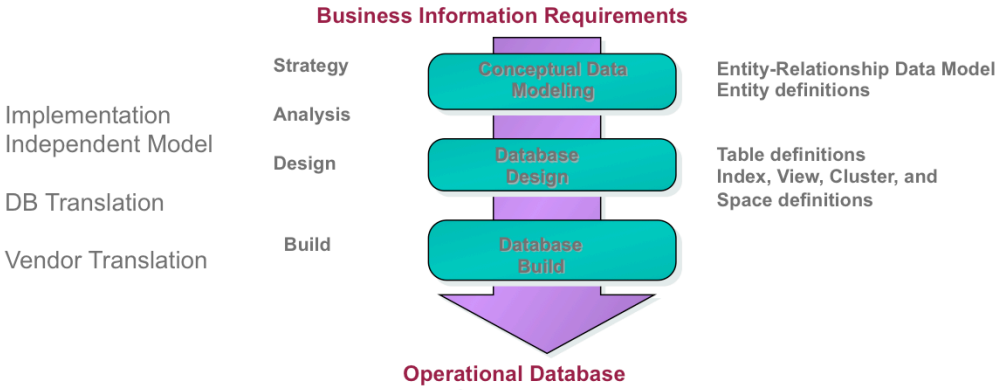
Data Modeling and Database Development

Entities, Attributes and Relationships

Database Development Process

Database development is a top-down, systematic approach that transforms business information requirements into an operational database



Where and How will rules be implemented?
-Declaratively: DDL, Check Constraint
-Programmatically: Trigger, P/LSQL &App code & What tier: DB, App server/ Web Service, client code

Conceptual and Logical Data Models

Conceptual

- High –level model
- Most important, obvious, primary entities and relationships
- No or few attributes
- No heavy hunting, or analysis
- Go broad before we go deep
- 7 – 18 +/-
- Scope and boundary validation and cross-check
- Sanity check – in-line with business requirements
- Completeness check - not missing anything obvious (no Doh! moments)

Logical

- All entities, attributes and relationships
- 3NF
- M:M resolved
- Identify UIDs
- Data Types
- Domains

Defining Entities

Name (don't get stuck on naming)

Examples of data (instance examples)

Definition

2 naturally occurring attributes and / or relationships

For each attribute:

- Definition, example value, required? Unique? Multiple values:
  - Multiple values can be OK – if "modeling the natural form of the data" and you plan to resolve to 1NF later. But don't get pedantic about it – pick your battles.

Business Information Requirements

Here is a set of information requirements:

"I manage the Human Resources Department for a large company. We need to keep information about each of our company's employees. We need to track each employee's first name, last name, job or position, hire date, and salary. For any employees on commission, we also need to track their potential commission. Each employee is assigned a unique employee number.

Our company is divided into departments. Each employee is assigned to a department -- for example, accounting, sales, or development. We need to know the department responsible for each employee and the department location. Each department has a unique number -- for example, accounting is *10* and sales is *30*.

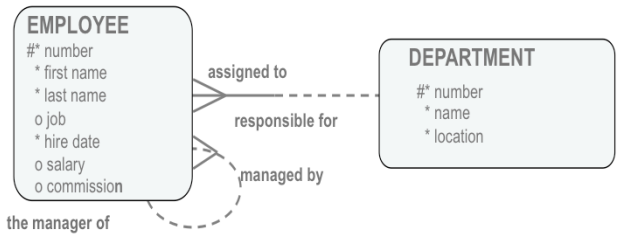
Some of the employees are managers. We need to know each employee's manager, and the employees each manager manages."

Data Modeling

In modeling, define and model the things of significance about which the business needs to know or hold information, and the relationships between them.

Example

The following entity-relationship model represents the information requirements of the Human Resources Department.



An Entity Relationship data model should accurately model the organization's information needs and support the functions of the business.

Database Design Overview

Table Name: EMPLOYEES

Column Name		FNAME	LNAME	JOB	HIREDT	SAL	COMM	MGR	DEPTNO
Key Type	PK							FK1	FK2
Nulls/Unique	NN, U	NN	NN		NN				NN
Datatype/ Size	NUM(9)	VC2(15)	VC2(20)	VC2(15)	DATE	NUM(9,2)	NUM(9,2)	NUM(9)	NUM(5)
Sample Data	112345	LAMB	CHOP	INSTR	01-MAR -53			8761123	10

Table Name: DEPARTMENTS

Column Name	DEPTNO	NAME	LOC
Key Type	PK		
Nulls/ Unique	NN, U	NN	NN
Datatype/ Size	NUM(5)	VC2(25)	VC2(20)
Sample Data	10	EDUC	SF

In database design, map the information requirements reflected in the entity-relationship model into a relational database design using a **Table Instance Chart.**

Database Build Overview

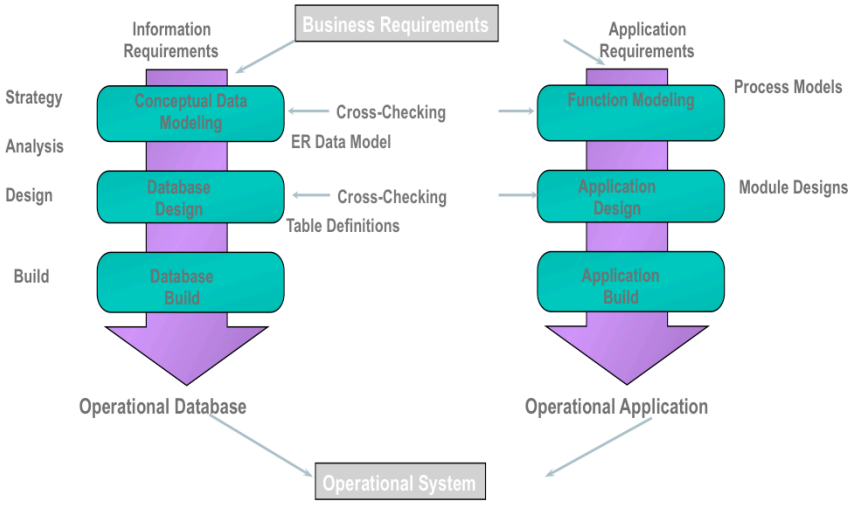
In database build, create physical relational database tables to implement the database design.

```
SQL>  CREATE TABLE DEPARTMENT
2     (DEPTNO     NUMBER(5)      NOT NULL   PRIMARY KEY,
3     NAME        VARCHAR2(25)   NOT NULL,
4     LOC         VARCHAR2(30)   NOT NULL);
```

```
SQL>  CREATE TABLE EMPLOYEES
2     (EMPNO      NUMBER(9)      NOT NULL   PRIMARY KEY,
3     FNAME       VARCHAR2(15)   NOT NULL,
4     LNAME       VARCHAR2(20)   NOT NULL,
5     JOB         VARCHAR2(15),
6     HIREDT      DATE           NOT NULL,
7     SAL         NUMBER(9,2),
8     COMM        NUMBER(9,2),
9     MGR         NUMBER(9)   REFERENCES EMPLOYEES,
10    DEPTNO      NUMBER(5)   REFERENCES DEPARTMENT );
```

Structured Query Language (SQL) is used to create and manipulate relational databases.

Database and Application Development

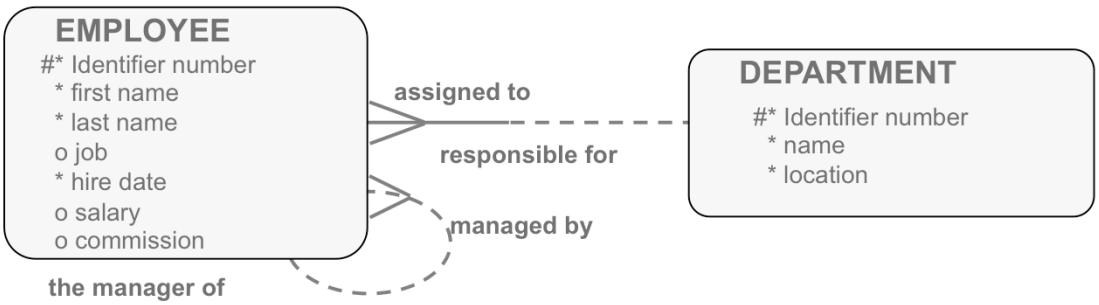


Data Modeling

The goal of Data Modeling is to develop an entity-relationship model that represents the information requirements of the business.

Example

The following entity-relationship model represents the information requirements of the Human Resources Department.



Entity - Relationship Model Components:

- Entities - the *things of significance* about which information need to be held
- Relationships - how the things of significance are related
- Attributes - the specific information which needs to be held

Conceptual Data Modeling (cont.)

An Entity-Relationship model is an effective means for collecting and documenting an organization's information requirements.

Robust Syntax
- An E-R Model documents an organization's information requirements in a clear, precise format.

User Communication
- Users can easily understand the pictorial form of an E-R Model.

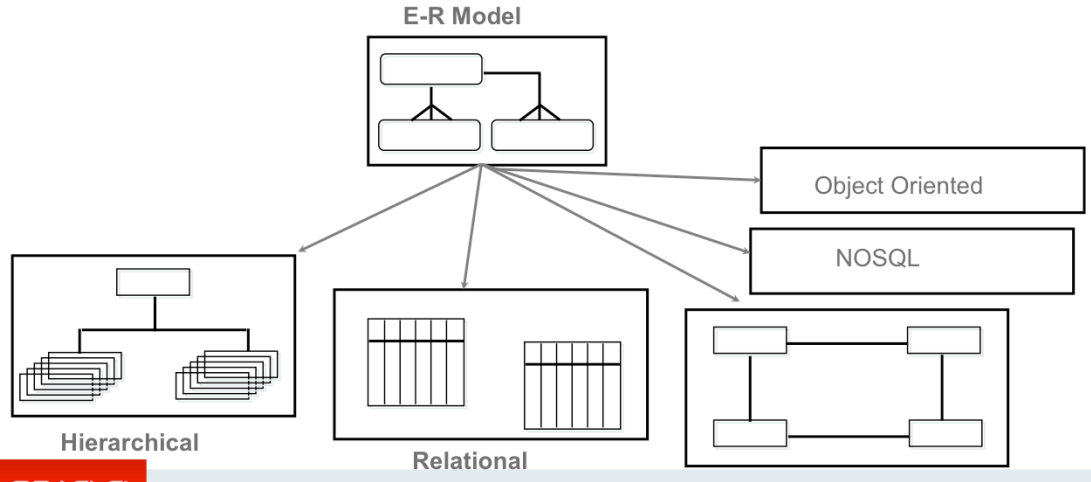
Ease of Development
- An E-R Model can be easily developed and refined, compared with building physical DB structures like tables, etc.

Definition of Scope
- An E-R Model provides a clear picture of the scope of an organization's information requirements.

Integration of Multiple Applications
- An E-R Model provides an effective framework for integrating multiple applications, development projects, and / or purchased application packages.

Data Modeling (cont.)

Data Modeling is independent of the hardware or software to be used for implementations. An E-R Model can be mapped to any database implementation. When implementing in the Cloud, you may not know what type of DB is offered, and can change. An Implementation Independent model is a good starting point, regardless of ultimate implementation.



Entities

An Entity is a thing of significance about which information needs to be known or held.

Alternate Entity Definitions

- An object of interest to the business
- An Entity is a class or category of thing.
- An Entity is a named thing.

Examples

The following might be things of significance about which a business needs to hold information:

Employee
Department
Project

Entities (cont.)

Attributes describe entities and are the specific pieces of information which need to be known.

Examples
Possible attributes for the entity EMPLOYEE are:

badge number, first name, last name, date of birth, and salary

Possible attributes for the entity DEPARTMENT are:

name, identifier number and location

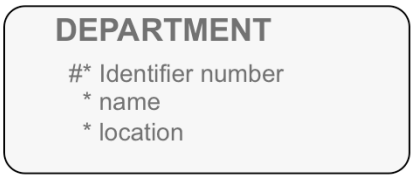
An entity must have attributes that need to be known from the business's viewpoint or it is not an entity within the scope of the business's requirements.

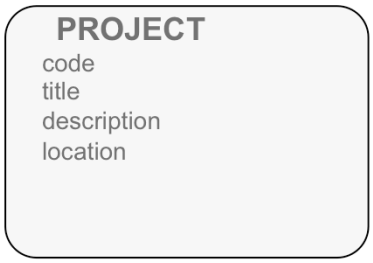
Entities (cont.)

Entity Diagramming Conventions

- **Soft box with any dimensions**
- **Singular, unique entity name**
- **Entity name in upper case**
- **Optional synonym name (in parentheses)**
- **Attribute names in all lower case**

Examples





- A synonym is an alternate name for an entity.
- Synonyms are useful when two groups of user have different names for the same thing of significance.

Entities (cont.)

Each Entity must have multiple occurrences of instances.

Examples

The entity EMPLOYEE has one occurrence for each employee in the business:
Jim Brown, Mary Jones, Juan Gomez, and Jill Judge are all occurrences of the entity EMPLOYEE.

The entity DEPARTMENT has one occurrence for each department in the company:
Finance, Sales, Engineering, and Information Systems are all occurrences of the entity DEPARTMENT.

Each Entity instance has specific values for the Entity's attributes.

Examples

The entity EMPLOYEE has attributes of first name, last name, date of birth, badge number, and salary.
The instance Jim Brown has the following values: first name Jim, last name Brown, badge number 1322, date of birth 15-MAR-50, an salary $55,000.

- Instances are sometimes mistaken for entities.
- An entity is a class or category of thing - e.g. EMPLOYEE.
- An instance is a specific thing - e.g. the employee Jim Brown.

Entities (cont.)

Each instance must be uniquely identifiable from other instances of the same Entity. An attribute or set of attributes that uniquely identify an Entity is called a Unique Identifier (UID).

Example

Each employee has a unique badge number. Badge number is a candidate for the entity EMPLOYEE's UID.

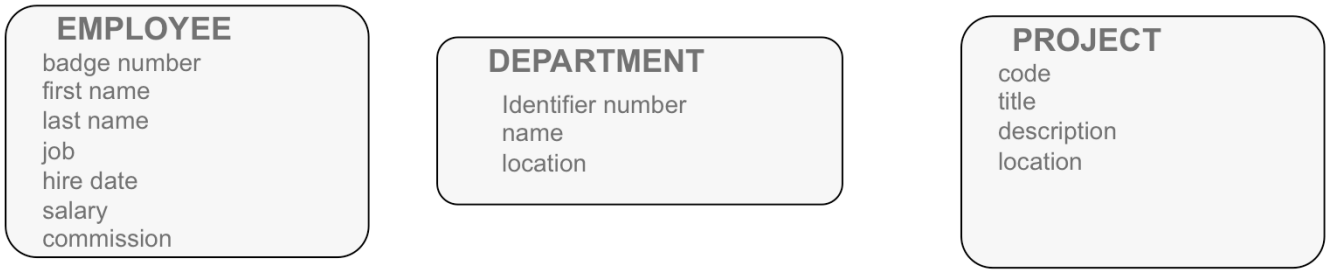
If an Entity cannot be uniquely identified, it may not be an Entity.
- Attributes which uniquely identify an Entity and are part of the Entity's UID are tagged with #.

Entities (cont.)

Look for attributes that uniquely identify an Entity.

Example

What attributes might uniquely identify the following Entities?



Identify and Model Entities

Follow the steps below and identify and model Entities from a set of interview notes.

1 Examine the nouns. Are they things of significance?

2 Name each Entity.

3 Is there information of interest about the Entity that the business needs to hold?

4 Is each instance of the Entity uniquely identifiable? Which attribute or attributes could serve as its UID?

5 Write a description of it. "An EMPLOYEE has significance as a paid worker at the company. For example, Jon Brown and Mary Smith are Employee's.

6 Diagram each Entity and a few of its attributes.

Do not disqualify a candidate Entity too soon. Additional attributes of interest to the business may be uncovered later.

Identify and Model Entities (cont.)

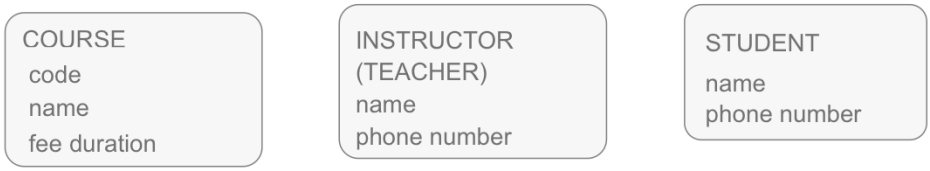
Identify and model the entities in the following set of information requirements.

"I'm the manager of a training company that provides instructor-led courses in management techniques. We teach many courses, each of which has a code, a name, and a fee. *Introduction to UNIX* and *C Programming* are two of our more popular courses. Courses vary in length from one day to four days. An instructor can teach several courses. Paul Rogers and Maria Gonzales are two of our best teachers. We track each instructor's name and phone number. Each course is taught by only one instructor. We create a course and then line up an instructor. The students can take several courses over time, and many of them do this. Jamie Brown from AT&T took every course we offer! We track each students name and phone number. Some of our students and instructors do not give us their phone numbers."

Identify and Model Entities (cont.)

Solution

The following entities model the Training Company's information requirements.



Entity Descriptions

- **A COURSE has significance as a training event offered by the Training Company. For example, *Introduction to UNIX* and *C Programming*.**
- **A STUDENT has significance as a participant in one or more COURSEs. For example, Jamie Brown.**
- **An INSTRUCTOR has significance as a teacher of one or more COURSES. For example, Paul Rogers and Maria Gonzales.**

Exercise

Identify and model the entities in the following set of information requirements.
Write a brief description of each entity. Show at least two attributes for each entity.

"I'm the owner of a small video store. We have over 3,000 video tapes that we need to keep track of.

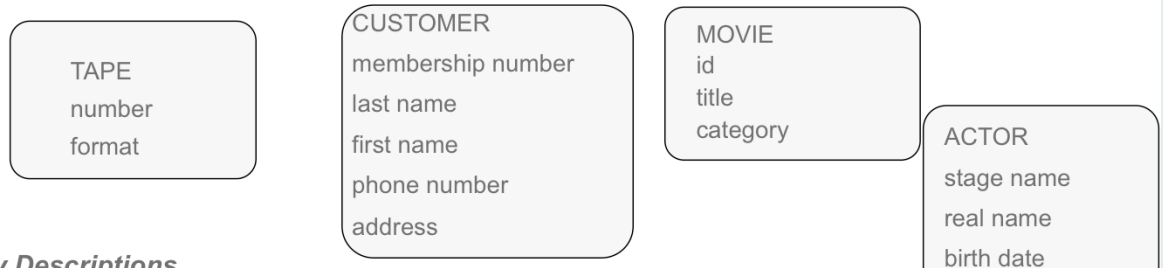
Each of our video tapes has a tape number. For each movie, we need to know its title and category (e.g. comedy, suspense, drama, action, war, or sci-fi). Yes, we do have multiple copies of many of our movies. We give each movie a specific id, and then track which movie a tape contains. A tape may be either Beta or VHS format. We always have at least one tape for each movie we track, and each tape is always a copy of a single, specific movie. Our tapes are very long, we don't have any movies which require multiple tapes.

We are frequently asked for movies starring specific actors. John Wayne and Katherine Hepburn are always popular. So we'd like to keep track of the star actors appearing in each movie. Not all of our movies have star actors. Customers like to know each actor's "real" birth name and date of birth. We track only actors who appear in the movies in our inventory.

We have lots of customers. We only rent videos to people who have joined our "video club." To belong to our club, they must have good credit. For each club member, we'd like to keep their first and last name, current phone number, and current address. And, of course each club member has a membership number.

Then we need to keep track of what video tapes each customer currently has checked out. A customer may check out multiple video tapes at any given time. We just track current rentals. We don't keep track of any rental histories."

EXERCISE *Solution*



Entity Descriptions

A MOVIE has significance as a movie title available at the Video Store. For example, *Gone with the Wind, Citizen Kane,* and *Star Wars*.

A TAPE has significance as an individual video tape available to rent. For example, a tape of the movie *Citizen Kane*.

An ACTOR has significance as a well-known actor who is featured in a movie carried by the Video Store. For example, Kathryn Hepburn and John Wayne.

A CUSTOMER has significance as a member of the Video Store's club. Customers may rent video tapes from the store.

Attributes

Attributes are information about an entity that needs to be known or held. Attributes describe an entity by qualifying, identifying, classifying, quantifying, or expressing the state of the entity.

Example

What are some attributes of the entity EMPLOYEE?

badge number or payroll number identify an EMPLOYEE.

first name and last name qualify an EMPLOYEE.

payroll category (e.g. weekly or salary) classify an EMPLOYEE.

age quantifies an EMPLOYEE.

employment status (e.g. active, on leave, terminated) express the status of an EMPLOYEE.

Attributes (cont.)

Attributes represent a type of description or detail, not an instance.

Example

77506 **and** ***76311*** **are values of the attribute badge number.**

John **is a value of the attribute first name of EMPLOYEE**

Quick Notes

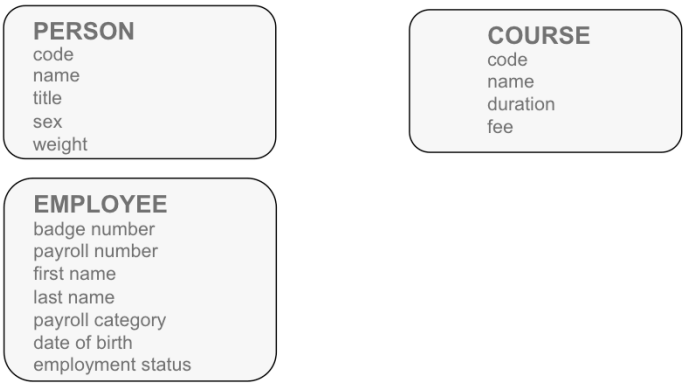
- **Attribute names should be clear to the user, not codified for the developer**
- **The entity's name is always a qualifier of the attribute name - e.g.,** ***code of COURSE*****. Therefore, an attribute's name should not include its entity's name.**
- **Attribute names should be specific - e.g., is it** ***quantity*****,** ***quantity returned*****, or** ***quantity purchased*****?**
- **Always clarify a date attribute with a descriptor or verb phrase, e.g.,** ***date of contact, date ordered.***
- **An attribute should only be assigned to a single entity.**

Attributes (cont.)

Diagramming Conventions

- **Attribute names are singular and shown in lower case.**
- **List attribute names in their entity's soft box.**

Example



Attributes (cont.)

Always break attributes down into their lowest meaningful components.

Examples

The name of a PERSON can be broken down into last name and first name.

The number of an ITEM consists of type, vendor, and item number.

Break down aggregate attributes and embedded code fields into simple attributes.

Attributes (cont.)

Verify that an attribute is not derived or calculated from the existing values of other attributes.

Common Derived Data

- **Counts (e.g. the number of salesmen in a region)**
- **Totals (e.g. the total number of each salesman's monthly sales)**
- **Max/Min/Average (e.g. statistics on the sales of a group of salesmen)**
- **Other calculations (e.g. a salesman's commission calculated at 10% of sales)**

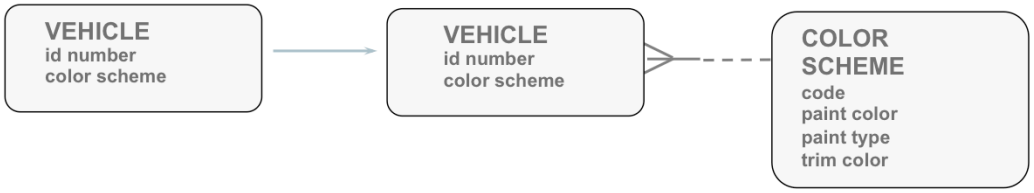
Do not include derived attributes in an E-R Model.

Distinguish Attributes and Entities

If an attribute has attributes of its own, then it is really an entity.

Example

Determine if all the attributes of VEHICLE are really attributes.



Initially, the user identified color scheme as an attribute of VEHICLE. Later, the user identified the requirement to track the paint color, paint type, and trim color for each color scheme. Color scheme then had attributes of its own, and became an entity with a relationship to VEHICLE.

Distinguish Attributes and Entities

Example

Determine if all the attributes of EMPLOYEE are really attributes.

Number of dependents is an attribute of EMPLOYEE, but if it is necessary to keep each dependent's name and age, then DEPENDENT becomes an entity. Number of dependents can now be derived. (Use "count" not "number" to avoid confusion).

Quick Notes

- Entities have attributes
- Attributes have no attributes of their own.

Distinguish Attributes and Entities

Entity Characteristics	Attribute Characteristics
Anything about which information must be held	Qualifies an entity
Possesses one or more attributes	Does not possess attributes of its own
If an entity has no attributes, it may be only an attribute	If an attribute has an attribute, then it is an entity or has no significance
May have multiple occurrences associated with another entity via a relationship	Caveat: Has a single value for each entity occurrence (no repeating groups)

All entities are nouns, but not all nouns are entities.

Attribute Optionality

Identify each attribute's optionality using an attribute tag.

Mandatory Attributes

- A value must be known for each entity occurrence.
- Tagged with *.

Optional Attributes

- A value may be known for each entity occurrence.
- Tagged with o.

Example

Identify the attributes for the PERSON entity. Determine their optionality.

The title and weight attributes are optional.
The remaining attributes are mandatory.

Attribute Optionality (cont.)

Use sample attribute instance data to validate attribute optionality.

Example

Are the mandatory and optional attribute tags for the PERSON entity correct? Use an Entity Instance Chart to validate that the mandatory and optional attribute tags for the PERSON entity are correct.

Attribute Optionality (cont.)

Entity Name: PERSON

Attribute Name	code	name	title	sex	weight
Tags	*	*	o	*	o
Sample Data	110	Garcia	President	F	——
	301	Slick	Treasurer	M	210
	134	Freiberg	——	F	110
	340	Dylan	Secretary	M	——
	589	Lambchop	——	F	195

Quick Note

- An Entity Instance Chart is useful for logging sample attribute data.

Conditional Optionality

Hard (or impossible) to model declaratively:

- Attribute value or optionality depends on other values or conditions
- Relationship Cardinality depends on varying conditions

Add note to model and make sure developers/designers see it.

Identify Attributes

Identify attributes by examining interview notes and by asking the user questions.

Attributes may appear in interview notes as :

- descriptive words and phrases.
- nouns.
- prepositional phrases (e.g. Salary amount for each employee).
- possessive nouns and pronouns (e.g. Employee's name).

Questions to Ask the User :

- What information do you need to know or hold about entity x?
- What information would you like displayed or printed about entity x?

Identify Attributes (cont.)

Examine documentation on existing manual procedures or automated systems to discover additional attributes and omissions.

Paper Forms	Computer Reports	Computer Files
Headings	Fields	Record layouts
Prompts	Headings	File Dumps
	Sort Orders	

Question to Ask the User :

- **Is this attribute really needed?**

Exercise

Develop an E-R Diagram for the following situation. Be sure to tag each attribute with its optionality.

Our regional Oracle User's Group has grown to include over 200 members. We're an all volunteer organization, and our records are a mess. We need an information system to help us keep track of all our affairs.

We definitely need to automate our membership records. for each member, we need to keep the member's name, title, mailing address, office phone number, type of membership (individual or corporate), and whether or not the member is current on dues. We collect dues on a yearly basis, and everyone's dues are due in January. We also like to know which company a member works for, but keeping this information current is a real chore because our members are always changing companies. We only try to track a single current employer for each member. Our members come from many different companies including Coors, EG&G, and Storage Tech. A few of our members are unemployed or self-employed. For each company, we keep the company name, address, and type of business. We have a standard set of type of business codes. We only keep the main company address for each company.

Exercise (cont.)

We also like to know which company a member works for, but keeping this information current is a real chore because our members are always changing companies. We only try to track a single current employer for each member. Our members come from many different companies including Coors, EG&G, and Storage Tech. A few of our members are unemployed or self-employed. For each company, we keep the company name, address, and type of business. We have a standard set of type of business codes. We only keep the main company address for each company.

We hold various events during the year, and we'd like to track information about each event. Some of our annual events include the September Meeting, the November Meeting, the annual Training Day in January, and our April Meeting. We also hold special events each year. For example, we held a special CASE day last May, and Alex Gersznowicz from ODK Inc. came and spoke. We hold our events at several different locations around town including AT&T, Redrocks Community College, and D.U. We'd like to track each event's date, and optional description of the event, number of attendees, where it was held, how much money we spent on it, and any comments on the event.

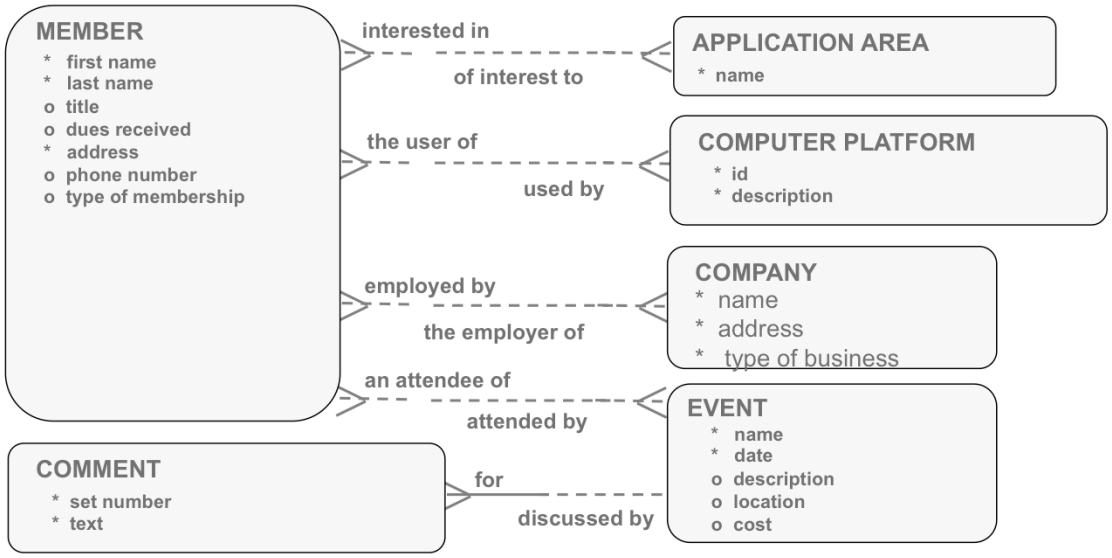
Exercise (cont.)

We also track which members attend which events. Some of our members are really active, and others attend very infrequently or just enjoy receiving our newsletter.

We also need to track what type of computer platforms our members are using. We have a unique, three-digit identification tag for each type of platform. For example, *001* is for IBM/MVS; *002* is for IBM/VM; *003* is for VAX/VMS; *020* is for OS/2; *030* is for PC/DOS; *050* is for Sun Unix; and *080* is for other Unix platforms.

We also like to track which application areas each member is interested in. For example, accounting, human resources, oil and gas, pharmaceuticals, and health systems. The applications should be portable, so we don't need to know which platforms they run on.

Exercise Solution



Relationships

A relationship is a two-directional, significant association between two entities, or between an entity and itself.

Relationship Syntax

Each *entity1* { must be / or / may be } *relationship name* { one or more / or / one and only one } *entity2*

Example

The relationship between INSTRUCTOR and COURSE is:

Each COURSE may be taught by one and only one INSTRUCTOR.

Each INSTRUCTOR may be assigned to one or more COURSEs.

Relationships (cont.)

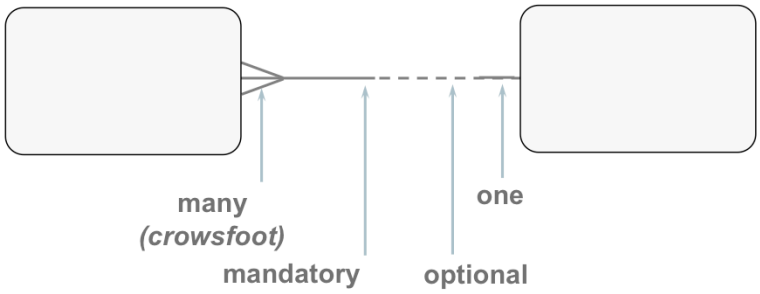
Each direction of a relationship has:

- a name - e.g., taught *by* or assigned *to.*
- an optionality - either *must be* or *may be.*
- a degree - either *one and only one* or *one or more.*

Diagramming Conventions

- A line between two entities
- Lower case relationship names
- Optionality
  - Optional *(may be)*
  - Mandatory *(must be)*
- Degree
  - one or more
  - one and only one

Relationships (cont.)



Relationships (cont.)

First read a relationship in one direction, and then read the relationship in the other direction.

Example

Read the relationship between EMPLOYEE and DEPARTMENT.

Relationships (cont.)

Example

Read the relationship first from left to right, and then from right to left.

Each EMPLOYEE must be assigned to one and only one DEPARTMENT.

Each DEPARTMENT may be responsible for one or more EMPLOYEEs.

Relationships (cont.)

Read the relationship between STUDENT and COURSE.

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

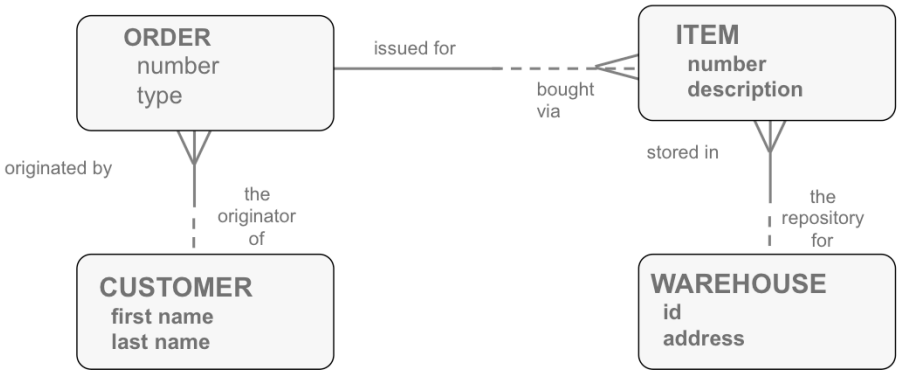
Read the relationship between PAYCHECK and EMPLOYEE.

Each PAYCHECK must be for one and only one EMPLOYEE.
Each EMPLOYEE may be the receiver of one or more Paychecks.

Exercise

Read the relationships.

Write the relationship sentence for this E-R diagram.



Exercise *Solution*

Each ORDER must be issued for one or more ITEMs.

Each ITEM may be bought via one or more ORDERs.

Each ORDER must be originated by one and only one CUSTOMER.

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

Each ITEM must be stored in one and only one WAREHOUSE.

Each WAREHOUSE may be the repository for one or more ITEMs.

Exercise

Draw an Entity-Relationship diagram to represent the following:

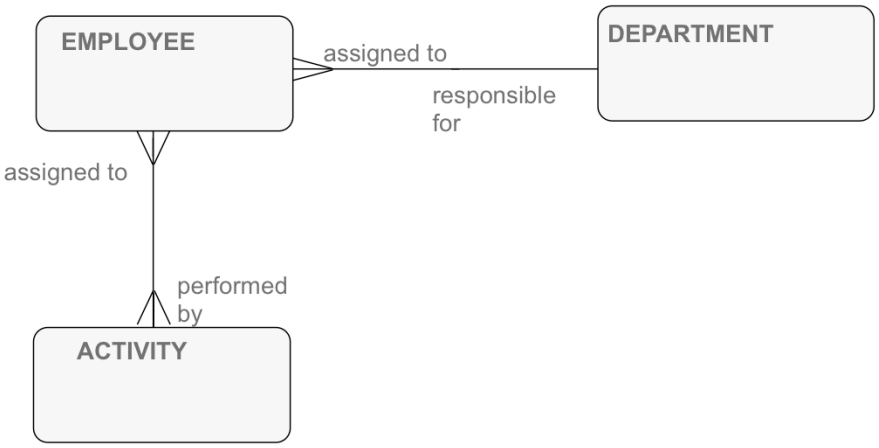
a Each EMPLOYEE must be assigned to one and only DEPARTMENT.

b Each DEPARTMENT may be responsible for one or more EMPLOYEEs.

c Each EMPLOYEE may be assigned to one or more ACTIVITIES.

d Each ACTIVITY may be performed by one or more EMPLOYEEs.

Exercise *Solution*

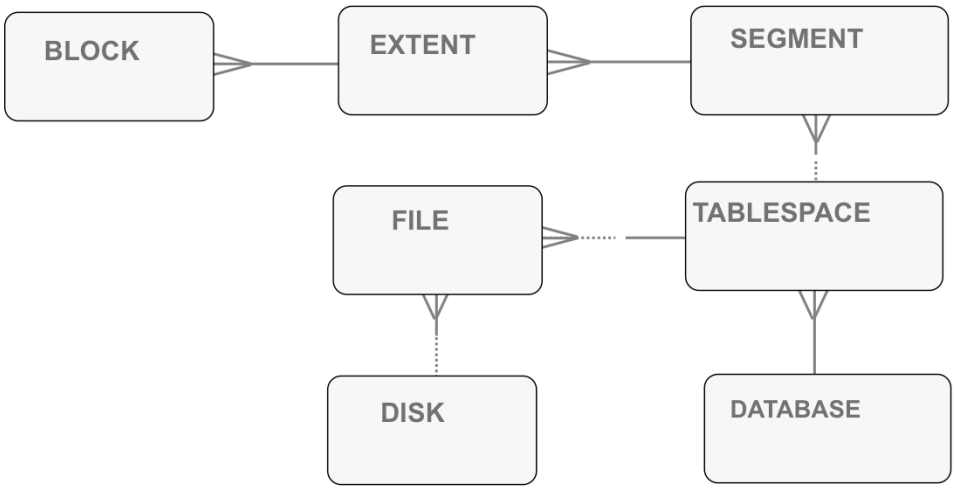


Exercise

Draw an Entity-Relationship diagram to represent the following:

a Each ORACLE DATABASE must be made up of one or more TABLESPACEs.
b Each TABLESPACE must be part of one and only one ORACLE DATABASE.
c Each TABLESPACE must be made up of one or more FILEs.
d Each FILE may be part of one and only one TABLESPACE.
e Each TABLESPACE may be divided into one or more SEGMENTs.
f Each SEGMENT must be included in one and only one TABLESPACE.
g Each SEGMENT must be inclusive of one or more EXTENTs.
h Each EXTENT must be included in one and only one SEGMENT.
I Each EXTENT must be composed on one or more BLOCKs.
j Each BLOCK must be part of one and only one EXTENT.
k Each FILE must reside on one and only one DISK.
l Each DISK may be the host for one or more FILEs.

Exercise *Solution*



Relationship Types

There are three types of relationships.

Relationship Types

- Many to One Relationship
- Many to Many Relationship
- One to One Relationship

All relationships should represent the information requirements and rules of the business.

Relationship Types (cont.)

A Many to One Relationship (M to 1 or M:1) has a degree of *one or more* in one direction and a degree of *one and only one* in the other direction.

Example

There is a M:1 relationship between CUSTOMER and SALES REPRESENTATIVE

Each CUSTOMER must be visited by one and only one SALES REPRESENTATIVE.

Each SALES REPRESENTATIVE may be assigned to visit one or more CUSTOMERs.

Relationship Types (cont.)

A Many to Many Relationship (M to M or M:M) has a degree of *one or more* in both directions

There is a M:M relationship between STUDENT and COURSE.

Each STUDENT may be enrolled in one or more COURSEs.

Each COURSE may be taken by one or more STUDENTs.

Relationship Types (cont.)

There is a M:M relationship between EMPLOYEE and JOB.

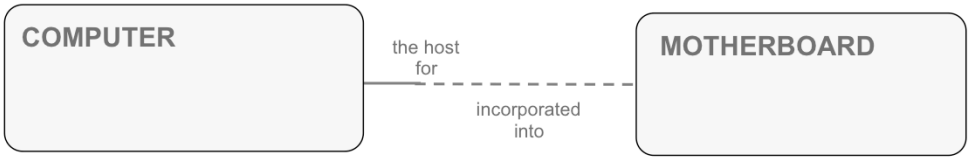
Each EMPLOYEE may be assigned to one or more JOBs.

Each JOB may be carried out by one or more EMPLOYEEs.

Relationship Types (cont.)

A One to One Relationship (1 to 1 or 1:1) has a degree of *one and only one* in both directions.

There is a 1:1 relationship between COMPUTER and MOTHERBOARD.



Each COMPUTER may be the host for one and only one MOTHERBOARD.

Each MOTHERBOARD may be incorporated into one and only one COMPUTER.

Analyze and Model Relationships

Follow a series of five steps to analyze and model relationships.

Steps

1. **Determine the existence of a relationship.**
2. **Name each direction of the relationship.**
3. **Determine the optionality of each direction of the relationship.**
4. **Determine the degree of each direction of the relationship.**
5. **Read the relationship aloud to validate it.**

Determine a Relationship's Existence

- Existence
- Name
- Optionality
- Degree
- Validate

Determine the existence of a relationship.
Examine each pair of entities to determine if a relationship exists.

Ask About a Relationship's Existence

- **Does a significant relationship exist between ENTITY A and ENTITY B?**

Consider the entities DEPARTMENT and EMPLOYEE.

Is there a significant relationship between DEPARTMENT and EMPLOYEE?

Yes, there is a significant relationship between DEPARTMENT and EMPLOYEE?

Name the Relationship (cont.)

- Existence
- → Name
- Optionality
- Degree
- Validate

Use a list of relationship name pairs to assist in naming relationships.

Useful Relationship Name Pairs

based on	the basis for
bought from	the supplier of
description of	for
operated by	the operator for
represented by	the representation of
responsible for	the responsibility of

Do not use *related to* or *associated with* as relationship names.

Determine the Relationship's Optionality

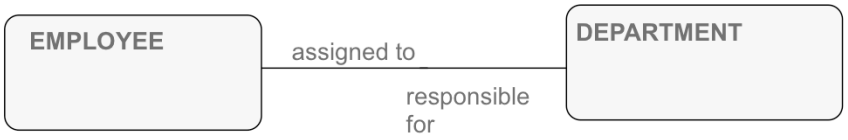
- Existence
- Name
- ➡ Optionality
- Degree
- Validate

Determine the optionality of each direction of the relationship.

Consider the relationship between DEPARTMENT and EMPLOYEE.

Must an EMPLOYEE be assigned to a DEPARTMENT? Always?
Is there any situation in which and EMPLOYEE will not be assigned to a DEPARTMENT?
No, an EMPLOYEE must always be assigned to a DEPARTMENT.

Must a DEPARTMENT be responsible for an EMPLOYEE?
No, a DEPARTMENT does not have to be responsible for an EMPLOYEE.



Draw the relationship lines, with the relationship names.

Determine the Relationship's Degree

- Existence
- Name
- Optionality
- ➡ Degree
- Validate

Determine the degree of the relationship in both directions.

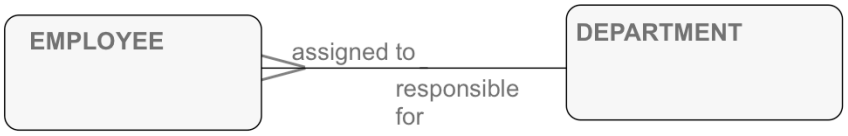
Consider the relationship between DEPARTMENT and EMPLOYEE.

May an EMPLOYEE be assigned to more than one DEPARTMENT?

No, an EMPLOYEE must be assigned to only one DEPARTMENT.

May a DEPARTMENT be responsible for more than one EMPLOYEE?

Yes, a DEPARTMENT may be responsible for one or more EMPLOYEEs.



Add the relationship degrees to the E-R Diagram.

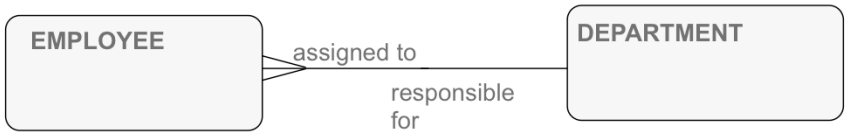
Validate the Relationship

- Existence
- Name
- Optionality
- Degree
- ➡ Validate

Re-examine the E-R Model and validate therelationship.

Read the Relationship Aloud

- **Relationships must be readable and make business sense.**



Each EMPLOYEE must be assigned to one and only one DEPARTMENT.

Each DEPARTMENT may be responsible for one or more EMPLOYEEs.

Exercise

Analyze and model the relationship in the following set of information requirements.

"I'm the manager of a training company that provides instructor-led courses in management techniques. We teach many courses, each of which has a code, a name, and a fee. *Introduction to UNIX* and *C Programming* are two of our more popular courses. Courses vary in length from one day to four days. An instructor can teach several courses. Paul Rogers and Maria Gonzales are two of our best teachers. We track each instructor's name and phone number. Each course is taught by only one instructor. We create a course and then line up an instructor. The students can take several courses over time, and many of them do this. Jamie Brown from AT&T took every course we offer! We track each student's name and phone number. Some of our students and instructors do not give us their phone numbers."

Exercise

Analyze and model the relationship in the following set of information requirements.

"I'm the owner of a small video store. We have over 3,000 video tapes that we need to keep track of.

Each of our video tapes has a tape number. For each movie, we need to know its title and category (e.g. comedy, suspense, drama, action, war, or sci-fi). Yes, we do have multiple copies of many of our movies. We give each movie a specific id, and then track which movie a tape contains. A tape may be either Beta or VHS format. We always have at least one tape for each movie we track, and each tape is always a copy of a single, specific movie. Our tapes are very long, we don't have any movies which require multiple tapes.

We are frequently asked for movies starring specific actors. John Wayne and Katherine Hepburn are always popular. So we'd like to keep track of the star actors appearing in each movie. Not all of our movies have star actors. Customers like to know each actor's "real" birth name and date of birth. We track only actors who appear in the movies in our inventory.

We have lots of customers. We only rent videos to people who have joined our "video club." To belong to our club, they must have good credit. For each club member, we'd like to keep their first and last name, current phone number, and current address. And, of course each club member has a membership number.

Then we need to keep track of what video tapes each customer currently has checked out. A customer may check out multiple video tapes at any given time. We just track current rentals. We don't keep track of any rental histories."

Exercise *Solution*