The Database Development Process

We will be using the traditional SDLC (Systems Development Life Cycle) to analyse and design our database.

# Data Analysis and Database Design

**Analysis Phase**

1. Review existing forms, reports, documentation and files. Interview users. Conduct JRP sessions.

**Design Phase**

1. **Identify the principle entities. Name and define each entity**.
2. Identify the relationships and cardinalities.
3. **Draw the Conceptual Data Model.**
4. Resolve non-specific (many-to-many) relationships.
5. Determine candidate, primary and alternate key attributes.
6. **Draw the preliminary Logical Data Model.**
7. Identify and associate attributes with entities or relationships.
8. Include generalization hierarchies
9. Refine the attribute definitions.
10. Create the domains.
11. **Add the attributes to the Logical Data Model.**
12. Normalize the data model.
13. Refine the relationship definitions.
14. **Add integrity constraints to the Logical Data Model.**

**Implementation Phase**

1. Convert Logical Data Model to Physical Data Model (Relational Data Model).
2. Generate DDL statements from Physical Data Model.
3. Create database.

### 

## Data Modeling

*What is a model?* Think of an airplane model, a model train, a fashion model

Webster's Dictionary defines a model as a "description or analogy used to visualize something that cannot be directly observed."

**Data Modeling** is

* the process of creating a representation of the users' view of the data

A **data model**

* represents data structures and their characteristics, relations, constraints and transformations.
* whose main function is to help us understand the complexities of the real-world environment.
* is used as communication tools between systems designer, system developers and business professionals.
* is the basis for all subsequent work in the development of databases and their applications

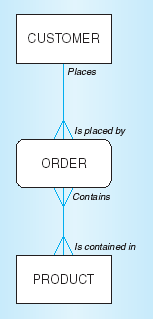
If the data model incorrectly represents the users' view of the data, the users will find the applications difficult to use, incomplete and very frustrating.

### Essential Elements for a good data model

For a data model to be valid, it must be

* an accurate reflection of the real world that it seeks to model. Errors within the data model will be automated when the resulting system is implemented
* implementation independent. No consideration should be given to automation or technology issues, at least not initially.
* sufficiently broad in scope. It should be based on an enterprise perspective, not an application perspective. Remember that data is a corporate resource, not an application resource.
* flexible to account for changes in the business environment, technology or the scope of the application or database.

### Entity Relationship Diagram (ERD)

* a data model utilizing several notations to depict data in terms of the entities and relationships described by that data.
* a data modeling tool that depicts the associations among different categories of data within a business or information system
* does not imply how data is implemented, created, modified, used, or deleted.
* first described in 1976 by Peter Chen in a landmark paper entitled, "The Entity-Relationship Model: Toward a unified View of Data". Since then several notations for the model have been developed (Chen, Martin, Bachman, Merise, IDEF1X).

#### CASE Tools

* Computer-aided software engineering tool
* Automated tools for data modeling
* **Common CASE tools:**
* Oracle Data Modeler
* ErWin – best with PowerBuilder and SQL Server
* Microsoft Visio
* Visual Paradigm
* SDesigner
* Chen Modeler
* **Capabilities**
* Create and revise data models
* check your model for mechanical errors, completeness and consistency
* Forward generate a physical database design based upon the logical data model
* Reverse generate a logical data model from an existing physical database

### Data Model Components: Entities, Relationships and Attributes

Attributes – details about an entity which seek to describe, quantify, identify or otherwise explain an entity

* Synonyms: *element*, *property*, and *field*.

## Database Design – Step 1: Identify Main Entities and Relationships

Entity – a class of persons, places, objects, events or concepts of significance which exists within the enterprise about which we need to capture and store data

***Entity Instance***

* a single occurrence of an entity
* An entity is a class of objects; an entity instance is a particular value of one these objects.
* in modeling, we are not concerned with the individual instances of an entity

***Examples*:** Customer 12345 is an instance of the entity Customer.

Professor is an entity; the data about Professor Smith is an entity instance.

***What Should an Entity Be?***

SHOULD BE:

* An object that we want to store information about
* An object that will have **many** instances in the database
* An object that will be composed of multiple attributes
* An object that we are trying to model

SHOULD NOT BE:

* A user of the database system
* An output of the database system (e.g., a report)

## Discover Entities

**Questions to ask**

*What are the subjects of the business? In other words, what types of persons, organizations, organizational units, places, things, materials, or events are used in, or interact with this system, about which data must be captured or maintained?*

*How many instances of each subject exist?*

***Where to look***

* business rules
* interviews with users
* existing forms and reports
* existing documentation
* questionnaires

***Techniques***

* Study existing forms and files.
* Reverse engineer existing files and databases into physical data models.
* Conduct interviews or JRP sessions with system owners and users. Pay attention to key words in their discussion and specifically ask them to identify things about which they would like to capture, store, and produce information

***Finding the Entities***

* look for useful nouns that describe significant objects that we want to store information about
* can include persons, places, objects, events or concepts

***Naming Entities***

* the entity name should be:
  + a noun or a noun phrase - the person, event, place, or tangible thing about which we want to store data
  + singular so as to distinguish the logical concept of the entity from the actual instances of the entity
  + specific to the organization – e.g. one organization might use CUSTOMER, where another organizations uses the term CLIENT
  + distinct from all other entity names – e.g. use CUSTOMER ORDER and PURCHASE ORDER, not ORDER
  + concise – use as few words as possible
  + should not include abbreviations or acronyms
  + in the case of event entities, should be named for the result of the event – e.g. ASSIGNMENT
* there must be a way to uniquely distinguish one entity instance from another
* examples:

Persons: agency, contractor, customer, department, division, employee, instructor, student, supplier.

Places: sales region, building, room, branch office, campus.

Objects: book, machine, part, product, raw material, software license, software package, tool, vehicle model, vehicle.

Events: application, award, cancellation, class, flight, invoice, order, registration, renewal, requisition, reservation, sale, trip.

Concepts: account, block of time, bond, course, fund, qualification, stock.

#### Entity Definitions

* usually starts with *"An X is …"*
* should be expressed in business terms.
* should not be defined in technical terms – e.g. don’t define it as ‘data about …’.
* should include a statement of what the **unique characteristic** for each instance of the entity – could include identifier – e.g. "An expense is a payment of the purchase of some good or service. An expense is identified by a journal entry number."
* often include a description of when an instance of the entity is created and deleted – e.g. A customer instance is implicitly created when the person or organization places its first order and a customer ceases to be a customer if it has not placed an order for more than three years
* sometimes include a description of when an instance might change into an instance of another entity type – e.g. "A bid is a legal offer by our organization to do work for a customer. A bid is created when an officer of our company signs the bid document; a bid becomes an instance of contract when we receive a copy of the bid signed by an officer of the customer"
* sometimes include what history is to be kept about instance of the entity. E.g. "when an item price changes the date of the change is recorded and the previous price is recorded in the item history entity"

**Entity Definition Examples**

| **Entity Name** | **Business Definition** | **Synonyms** |
| --- | --- | --- |
| agreement | A contract whereby a member agrees to purchase a certain number of products within a certain time. After fulfilling that agreement, the member becomes eligible for bonus credits that are redeemable for free or discounted products. | contract |
| member | An active member of the clubs.  Note: A target system objective is to re-enrol inactive members as opposed to deleting them. | customer |
| member order | An order generated for a member as part of a monthly promotion, or an order initiated by a member.  Note: The current system only supports orders generated from promotions; however, customer initiated orders have been given a high priority as an added option in the proposed system. | order |
| transaction | A business event to which the Member Services System must respond. |  |
| product | An inventoried product available for promotion and sale to members.  Note: System improvement objectives include (1) compatibility with new bar code system being developed for the warehouse, and (2) adaptability to a rapidly changing mix of products. |  |
| promotion | A monthly or quarterly event whereby special product offerings are made available to member. |  |

***Quantifying the Entities***

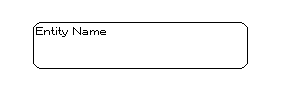
* identify the initial number or current number of entity instances
* identify the maximum number of instances over the life of the model
* identify the minimum number of instances
* identify the estimated growth rate of instances

This will help identify the scale of the model and to ensure that the entities are correct.

*Example of importance of scale:* Suppose that a company has a customer base of several thousand companies. Allowing for future growth, we could arrive at a maximum number of Customer instances. However, if future plans include direct business-to-consumer sales over the web, the maximum number of Customer instances could be millions of retail customers.

### Drawing an Entity on a Data Model

***The Entity Symbol:***



## Database Design – Step 2: Identify the relationships and cardinalities.

## Relationship

* a natural business association existing between one or more entities
* may represent an event that links the entities or merely a logical affinity that exists between the entities
* relates to an event or a definition
* reflexive or recursive relationships represent a situation where both ends of the relationship are served by instances of the same entity

***Example:***

* **a customer *deposits* a cheque
* a cheque *is deposited by* a customer

## Identify Relationships

**Questions to ask**

*What events occur that imply associations between subjects? What business activities or transactions require involve handling or changing data about several different subjects of the same or a different type?*

***Where to look***

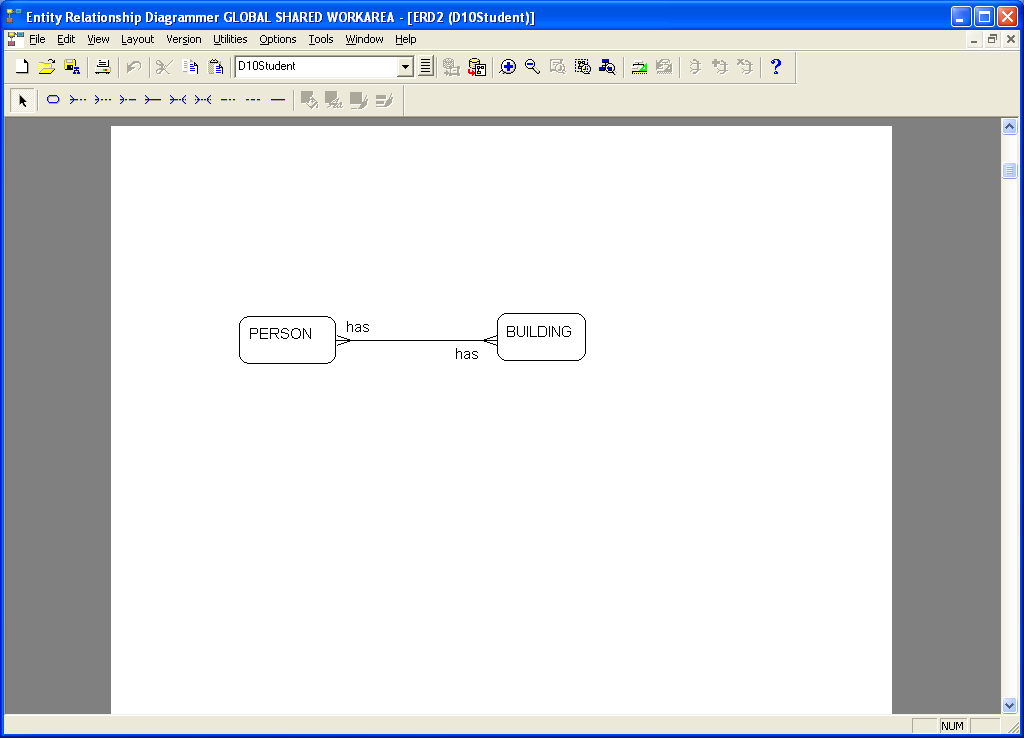
* business rules
* interviews with users
* existing forms and reports
* existing documentation
* questionnaires

#### Relationship Names

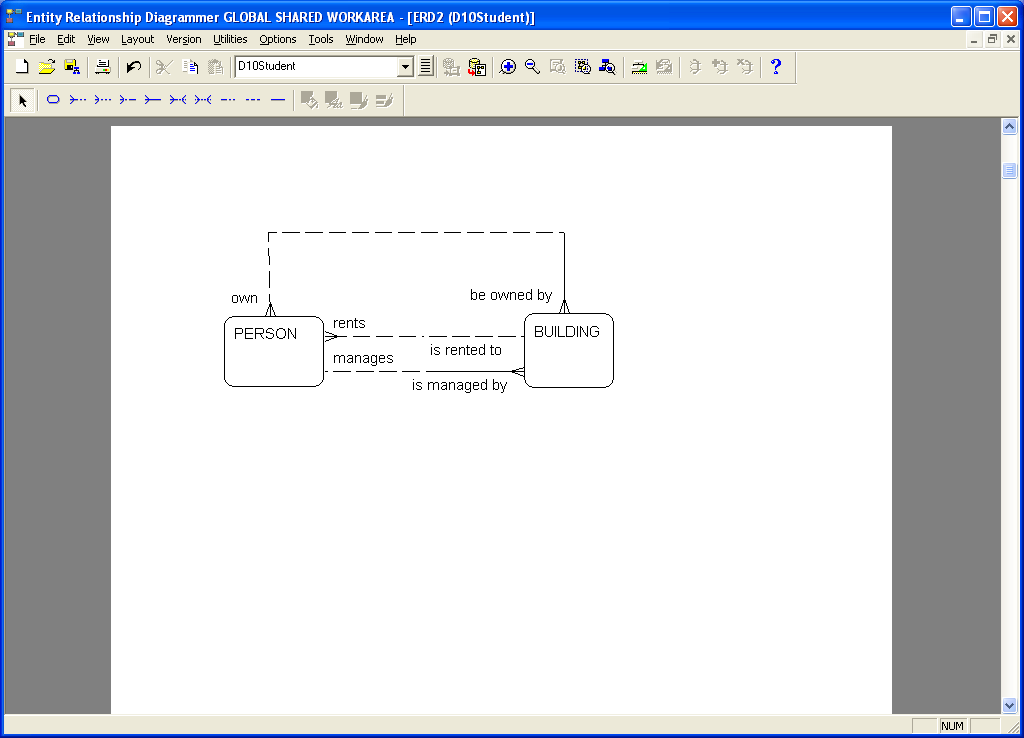
* should be *a verb phrase* that represent actions being taken, usually in the present tense
* should *use action verbs* found in the definition of the relationship
* should be as specific as possible so that its meaning is clear (i.e. not vague like *has* or *is related to*)

***Examples:***

* 1. Consider the relationship between DEPARTMENT and EMPLOYEE entities:
* A department has members that are employees
* An employee belongs to a department.
  1. ***Relationships Must Be Specific***: Consider the relationship between a PERSON and a BUILDING.



becomes



***Capturing Relationship Details***

**Questions to ask**

*Is each business activity or event handled the same way or are there special circumstances? Can an event occur with only some of the associated subjects, or must all the subjects be involved?*

For each relationship, you must specify the following characteristics of the relationship

* cardinality
* optionality
* relationship names

**Cardinality**

* the maximum number of times an occurrence of an entity occurs in relationship to another entity
* all relationships are bidirectional, so cardinality must be defined in both directions for every relationship
* Valid values are:
* 1:1 (one to one)
* 1:N (one to many)
* M:N (many to many)

***Examples:***

1:1 A person is married to one spouse. A spouse is married to one person.

1:N A customer deposits many cheques. A cheque is deposited by only one customer.

M:N A student takes many courses. A course is taken by many students

**Optionality**

* must each and every instance on the entity participate in the relationship?
* Aka participation
* For example, consider a relationship between ENTITY A and ENTITY B. Depending upon the optimality specified, one may interpret the relationship as follows:
* Entity A Participation
  + Each and every instance of ENTITY A must participate
  + A given instance of ENTITY A may participate
* Entity B Participation
  + Each and every instance of ENTITY B must participate
  + A given instance of ENTITY B may participate

***Drawing a Relationship on a Data Model***

***Symbol:***

* there are many symbolic ways of representing cardinality
* in most notations cardinality and optionality are combined in a single symbol. Optionality is considered to be **minimum** cardinality
* Oracle Data Modeler can useBarker or Bachman notation
* the Bachman method uses arrows and IDE1FIX uses dots

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Cardinality**  **Interpretation** | **Minimum**  **Instances** | **Maximum**  **Instances** | **Barker** | **Merise** | **Crow's feet**  **Notation** | **IDE1FIX** |
| Exactly one | 1 | 1 |  | 1,1 |  |  |
| Zero or more | 0 | 1 |  | 0,1 |  |  |
| One or more | 1 | >1 |  | 1,n |  |  |
| Zero, one or more | 0 | >1 |  | 0,n |  |  |
| More than one | >1 | >1 |  | 1,n |  |  |

***Barker Relationship Optionality Symbols***

Optional

Mandatory

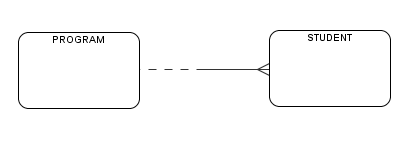
***Barker Relationship Cardinality Symbols***

1:1 (mandatory to optional)

1:N (mandatory to mandatory)

M:N (mandatory to optional)

**Example:**



The cardinality is **1:N**.

Read as:

* A PROGRAM *has enrolled* **one or more** STUDENTs.
* A STUDENT *is enrolled in* **only one** PROGRAM.

**Participation of PROGRAM:**

A given PROGRAM may participate in the relationship. (i.e. A given PROGRAM **may** have many STUDENTs enrolled.)

**Participation of STUDENT:**

A given STUDENT must participate in the relationship. (i.e. A given STUDENT **must** be enrolled in only one PROGRAM.)

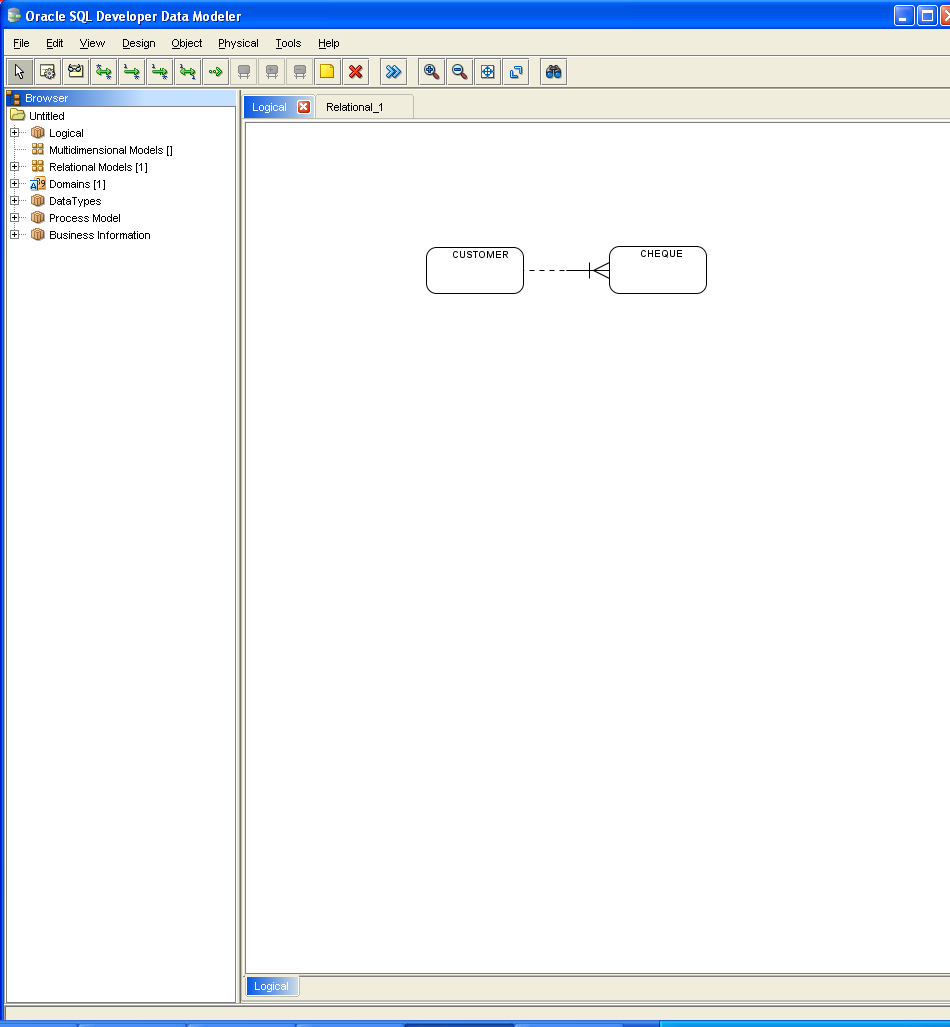
#### Relationship Characteristics

**Identifying**: Relationships are identifying if one entity (a weak entity) depends on another entity (a strong entity) for its unique identity. (e.g. a student’s spouse depends on the student for its unique identity)

***Relationship Characteristics Symbols***

Identifying

**Example:**

**

The cardinality is **1:N**.

Read as:

* A customer *may* *deposit* **one or more** cheques.
* A cheque *must be deposited by* **exactly one** customer.
* A cheque *cannot exist* unless it is deposited by a customer

**Relationship Name**

* Each end of a relationship should be named with a verb or verb phrase.
* For example, consider the relationship between DEPARMENT and EMPLOYEE entities:
* A department has members that are employees
* An employee belongs to a department.

**Interpreting Relationship Details**

Consider the sentences below that describe the relationship between DEPARTMENT and EMPLOYEE. Identify the words/phrases that express the:

1 Cardinality

2 Optionality

3 Relationship name

One and only one DEPARTMENT may participate in having members who are employees.

1

2

3

#### Relationship Definitions

* should explain what action is being taken and possibly why it is important. It may be important to state who or what does the action, but it is not important to explain how the action is taken.
* may give examples to clarify the action – e.g. for the *is registered for* relationship between student and course, it may be useful to explain that this covers both on-site and on-line registration and includes registrations made during the drop/add period.
* should explain any optional participation – explain what conditions lead to zero associated instances, whether this can happen only when an entity instance is first created, or whether this can happen at any time. e.g. the *is registered for* relationship links a course with the students who have signed up to take the course and the courses a student has signed up to take. A course will have not students registered for it before the registration period begins and may never have any registered students. A student will not be registered for any courses before the registration period begins and may not register for any classes (or may register for classes and then drop any or all classes.)
* should explain the reason for any explicit maximum cardinality other than many. E.g. “is assigned to links an employee with the projects to which that employee is assigned and the employees assigned to a project. Due to our labour union agreement, an employee may not be assigned to more than four projects at a given time.”
* should explain any mutually exclusive relationships – those for which an entity instance can participate in only one of several alternative relationships. e.g. “*plays on* links an intercollegiate sports team with its student players and indicates on which teams a student plays. Students who play on intercollegiate sorts teams cannot also work in a campus job.”
* should explain any restrictions on participation in the relationship e.g. “*is supervised by* links an employee with the other employees he or she supervises. An employee cannot supervise him- or herself and an employee cannot supervise other employees if his or her job classification level is below 4.”
* should explain the extent of history that is kept in the relationship e.g. “*places* links a customer with the orders they have placed with our company. Only two years of orders are maintained in the database, so not all orders can participate in this relationship.
* should explain whether an entity instance involved in a relationship instance can transfer participation to another relationship instance. E.g. “*places* links a customer with the orders they have placed with our company. An order is not transferable to another customer.”

## Database Design – Step 3: Conceptual Data Model

### The Conceptual Data Model

A **Conceptual Data Model** is an **entity relationship diagram** that includes the fundamental or independent **entities** and the ***relationships*** between them.

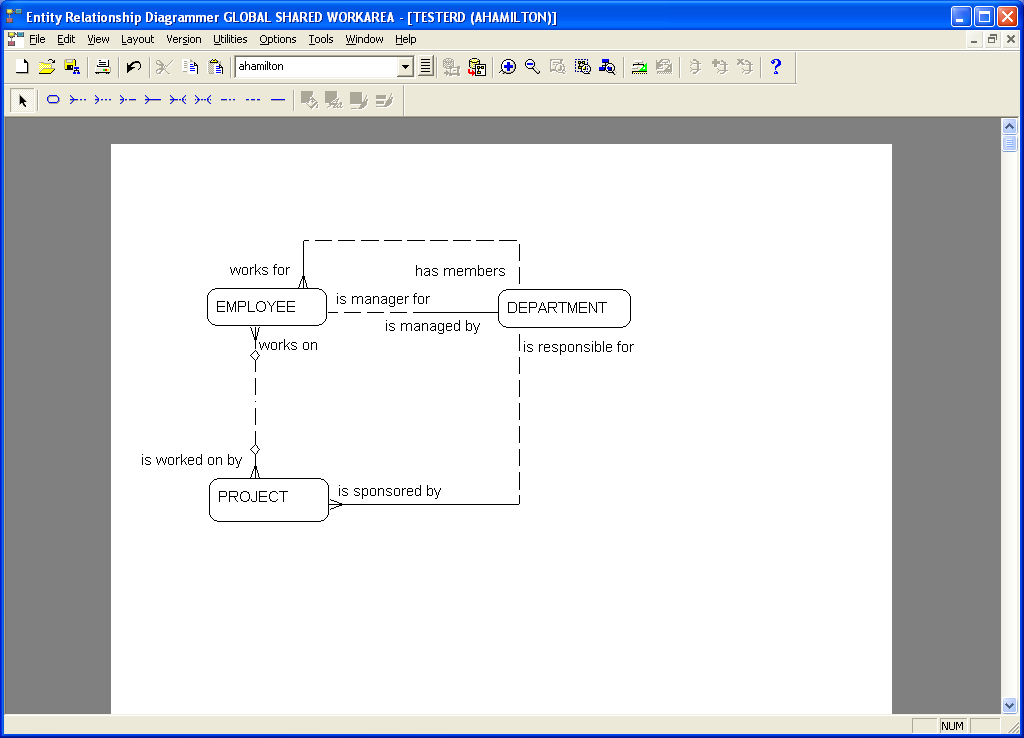
Once the entities and relationships have been discovered, a conceptual model can be drawn to provide a graphical representation of the most important data elements of the business.

It is often used in the initial design phases of a project and does not usually include details such as attributes and keys. It is the first step in constructing a clear and accurate visual representation of the business.

***A Simple Conceptual Data Model***

A small company has been set up to develop software for small businesses. There are five departments in the company: Sales, Research and Development, Customer Support, Human Resources and Accounting. Each department has a number of employees and each employee works in only one department. One of the department employees is designated as the department manager. Employees within the various departments work on projects. Departments are responsible for the projects.

Identify the Entities and Relationships then **draw the Conceptual Data Model** for this business case.



1. A department may have 1 or more members who are employees

An employee must belong to only one department.

1. An employee may be manager for only one department

A department must be managed by only one manager.

1. An employee may work on many projects and is transferable

A project may be worked on by many employees and is transferable

1. A department may be responsible for many projects

A project must be sponsored by only one department.

***Exercise***:

Identify the main entities and relationships and **draw an entity relationship diagram representing a conceptual data model** for the situation described in the following scenario.

The Daddy Big Bucks bank holding company manages a number of major banks. Each bank has several branches. In the event of a bank closure, all branches of the bank would close as well.

When a new branch is opened for a bank, it remains with the bank for the duration of its existence. A branch is never reassigned to another bank.

Each bank has many customer accounts. The accounts are held and managed by a specific bank branch. The accounts may be either checking or savings.

The bank also issues and manages many loans for customers. The loans, like the accounts, are managed at the branch level. The types of loans available are auto, personal, home mortgage and college tuition.

1. Identify the entities:

BANK, BRANCH, ACCOUNT, CUSTOMER, LOAN

2. Identify the relationships:

A BANK may open many branches.

A BRANCH must be associated with only one bank.

A BRANCH may hold many accounts.

An ACCOUNT must be held by only one branch.

A CUSTOMER may own many accounts.

An ACCOUNT must belong to only one customer.

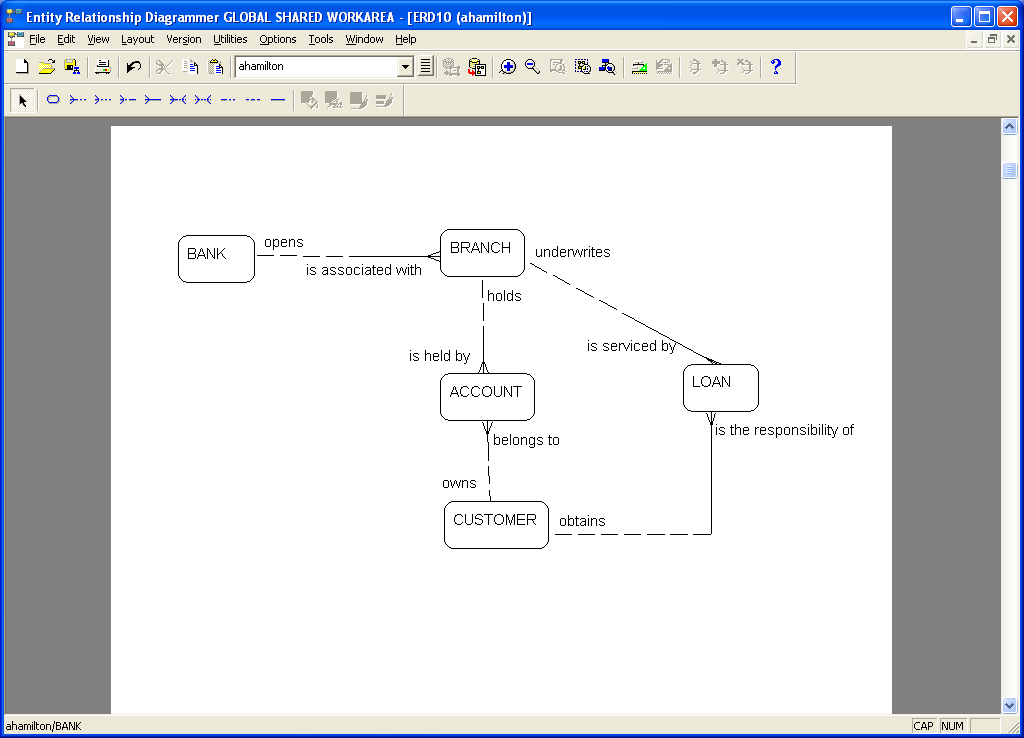
A CUSTOMER may obtain many loans.

A LOAN is the responsibility of only one customer.

A BRANCH may undertake many loans.

A LOAN must be serviced by only one BRANCH.

3. Draw the entity relationship diagram:



***Exercise***

The following ERD was drawn to represent the database needed for a college club’s expense system. The club treasurer manages accounts, receives expense reports, and records expense transactions against each account. Do you see any problems with the representation shown here?



***Problems***:

1. The treasurer is the person entering data about accounts and expenses and receiving expense reports. She is the user of the system, not part of the system. The system is not collecting data about the treasurer.
2. The expense report is computed from expense transactions and account balances. It is the result of extracting data from the database. No data that is only on the expense report is required to be kept.

***Corrected ERD:***

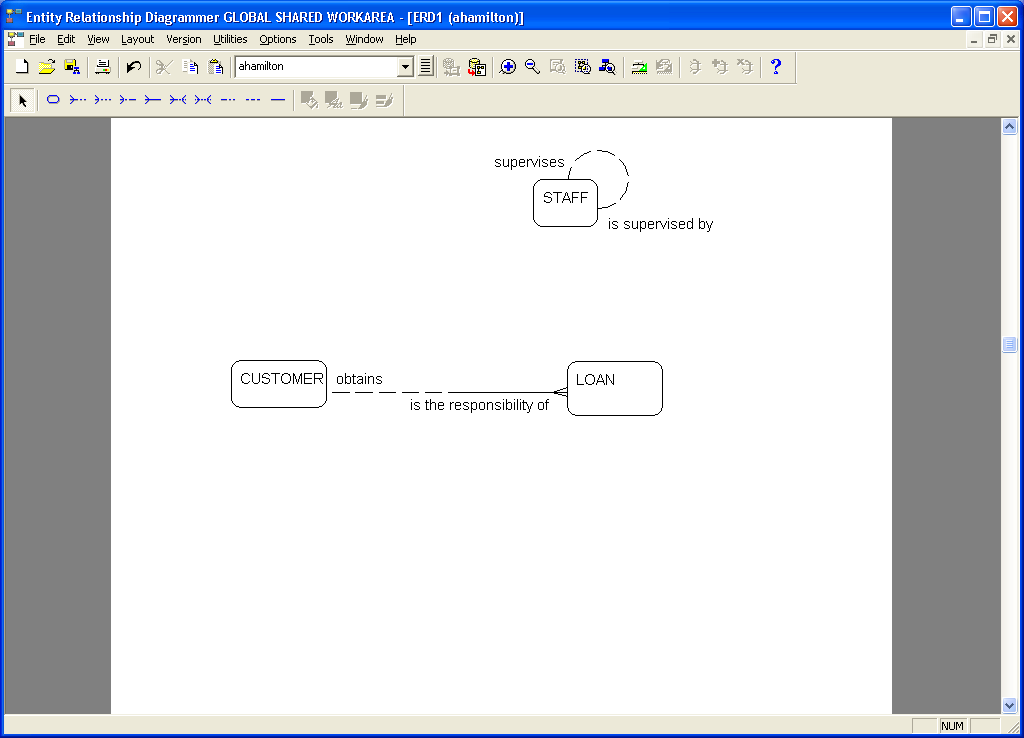


## Degree of Relationship

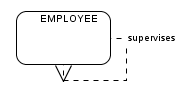
* the number of participating entities in a relationship.
* Most relationships are **binary** – involving two entities
* **Recursive** relationship aka unary aka reflexive relationship – the same entity participates more than once in different roles
* **Ternary** relationships – involve three entities

***Examples:***

### Binary Relationship

* a **CUSTOMER** may obtain many **LOAN**s**. A LOAN** is the responsibility of exactly one **CUSTOMER**

### Recursive Relationship

* A **PERSON** may be married to one **PERSON**.
* An **EMPLOYEE** may supervise many **EMPLOYEE**s. An **EMPLOYEE** is supervised by one **EMPLOYEE**.

### Ternary Relationship

Consider the following diagram with two binary relationships:



and the following data in the relationships:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **“teaches”** | |  | **“is offered in”** | |
| **Professor No** | **Course No** |  | **Course No** | **Session No** |
| 01261  01261  01261  01424  01591  01591 | MAT1139  MAT1140  MAT1142  MAT1140  ENG1001  ENG1002 |  | MAT1139  MAT1139  MAT1140  MAT1140  MAT1140  ENG1001  ENG1002 | AUT  WIN  AUT  WIN  SUM  AUT  AUT |

A professor teaches one or more courses. A course is taught by one or more professors. A course is offered in zero or more sessions. A session offers one or more courses. From this diagram can you answer the question:

**Who teaches Mat1140 during the Winter Session?**

Professors 01261 and 01424 taught Mat1140 and Mat1140 was offered during the AUT and WIN sessions. But we don’t know in which session each teacher taught Mat1140.

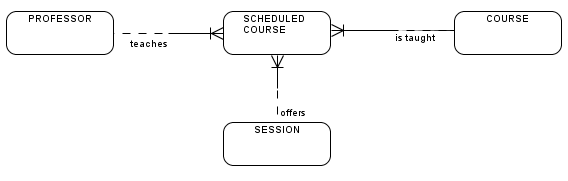
To answer this question we need to add a new type of entity - an associative entity.

### An Associative Entity

* represents a relationship containing attributes or the junction in a ternary relationship
* It inherits its primary key from more than one other entity (parents).
* Each part of that concatenated key points to one and only one instance of each of the connecting entities.
* Oracle Designer does not have a special symbol for an associative entity
* the conversion of a relationship to an associative entity causes the relationship notation to move. That is, the “many” cardinality now terminates at the associative entity, rather than at each participating entity.

To determine who taught which course when, we add an associative entity that connects all three entities. This creates a three-way relationship between PROFESSOR, COURSE and SESSION – a ternary relationship.

**A Ternary Relationship**



We have added the associative entity **SCHEDULED COURSE** and we can now answer the question "**Who teaches Mat1140 during the Winter Session?"**

**Occurrences of the SCHEDULED COURSE Associative Entity**

|  |  |  |
| --- | --- | --- |
| **Professor No** | **Course No** | **Session Code** |
| 01261  01261  01261  01261  01424  01424  01424  01591  01591 | MAT1139  MAT1139  MAT1140  MAT1142  MAT1140  MAT1140  MAT1140  ENG1001  ENG1002 | AUT  WIN  AUT  SUM  AUT  WIN  SUM  AUT  AUT |

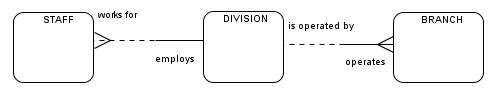
## Connection Traps

* occur due to a misinterpretation of the meaning of certain relationships.
* Two main types: fan traps and chasm traps

***Fan Traps***

* where a model represents a relationship between entities, but the pathway between certain entity occurrences is ambiguous
* occur where two or more 1:M relationships fan out from the same entity.

***Example***

Consider the following data model:

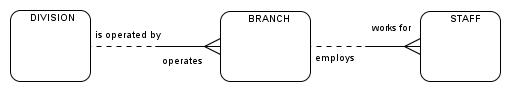
A problem arises when we want to know which members of staff work at a particular branch. Consider the following set of sample data:

* Division D1 employs staff numbers SG37 and SA9
* Division D2 employs staff number SL21
* Division D1 operates Branch B003 and B007
* Division D2 operates branch B005

Question: "At which branch does staff number SG37 work?"

* we are unable to give a specific answer – only say that he/she works at B003 or B007

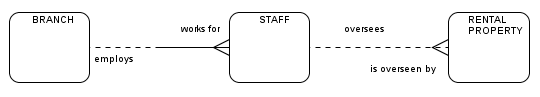
This results from a misinterpretation of the relationships between staff, branch and division.

To resolve the fan trap: restructure the ER model to represent the correct association between these entities:

***Chasm Traps***

* where a model suggest the existence of a relationship between entities, but the pathway does not exist between certain entity occurrences.
* May occur where there are one or more relationships with optional participation forming part of the pathway between related entities.

***Example***

Consider the following data model:

Note that not all properties are overseen by a member of staff and not all staff oversee property. The problem arises when we want to know which properties are available at each branch.

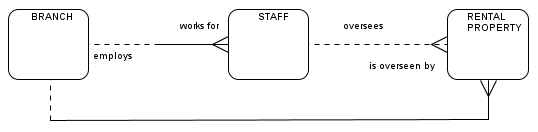
Consider the following set of sample data:

* Branch B003 employs staff number SG37
* Branch B007 employs staff number SA9
* Branch B005 employs staff number SL21
* Staff Number SG37 oversees rental property PG36
* Staff Number SL21 oversees rental property PL94
* Rental property PA14 is not overseen by a member of staff

Question: "At which branch Is property PA14 available?"

* The inability to answer this question is considered to be a loss of information

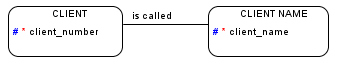
The optionality of both the Staff and Rental Property entities in the “oversees” relationship means that some properties cannot be associated with a branch through a member of staff.

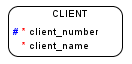
To resolve the chasm trap: add the missing relationship and redraw the ER model to represent the complete set of associations between these entities:

***Relationship Guidelines***

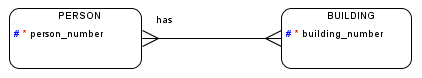
1. Two entities associated by a relationship of ***"1,1", "1,1"*** represent one and the same entity.

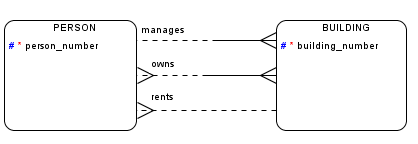
***Example:***

**Relationships “1:1”, “1:1”**

**becomes**

1. Relationships must be specific.

**Relationships Must Be Specific**

**becomes**

### http://image.slidesharecdn.com/datamodelingbusinessedbinov2014-141112083858-conversion-gate02/95/data-modeling-for-the-business-29-638.jpg?cb=1454091202