Database analysis

An entity type name should be specific to the organization.Thus,one organization may use the entity type name customer ,and another organization may use the entity type name client(this is one task ,for example, done to customize a purchase data model).the name should be descriptive for everyone in the organization and distinct form all other entity type names within that organization. For example ,a purchase order for orders placed with supplies is distinct from customer order for orders placed with a company by its customers. both of these entity types cannot be named order.

An entity type name should be concise , using as few word as possible .for example , in a university database ,an entity type Registration for the event of a student registering for a class is probably a sufficient name for this entity type ；student registration for class ,although precise ,is probably too wordy because the reader will understand registration from its use with other entity types .

An abbreviation ,or a short name ,should be specified for each entity type name ,and the abbreviation may be sufficient to use in the E-R diagram ,abbreviations must follow all of the name rules as do the full entity names .

Event entity types should be named for the result of the event ,not the activity or process of the event .for example ,the event of the project manager assigning an employee of the work on a project results in an assignment ,and the event of a student contacting his or her faculty adviser seeking some information is a contact .

The name used for the same entity type should be the same on all E-R diagrams on which the entity type appears .thus ,as well as being specific to the organization ,the name used for an entity types should be a standard ,adopt by the organization for all references to the same kind of date .however ,some entity types will have aliases ,or alternative names ,which are synonyms used in different parts of the organization .for example ,the entity item may have aliases of material (for production)and drawing(for engineering）。Aliases are specified in documentation about the database ,such as the repository of a case tool.

There are also some specific guidelines for defining entity types,which follow:

An entity type definition usually starts with “an x is …..”this is the most most direct and and clear way to state the meaning of an entity type.

An entity type definition should include a statement of what the unique characteristic is for each instance of the entity type.in many cases,stating the identifier for an entity type helps convey the meaning of the entity. An example for figure 2-4b is”an expense is a payment for the purchase of some good or service.An expense is identified by a journal entry number.”

An entity type definition should make it clear what entity instances are included and not included in the entity type;often,it is necessary to list the kids of entities that are excluded.for example,”A customer is a person or organization that has placed an order for a product form form us or one that we have contacted to advertise or promote our products.a customer does not include persons or organizations that buy our products only through our customers,distributors,or agents.”

An entity type definition often includes a description of when an instance of the entity type is created and deleted.for example ,in the previous bullet point ,a customer instance is implicitly created when the person or organization places its first order ;because this definition does not specify otherwise,implicitly a customer instance is never deleted,or it is deleted based on general rules that are specified about the purging of data from the database.a statement about when to deleted an entity instance is sometimes referred to as the retention of the entity type.a possible deletion statement for a customer entity type definition might be’ “A customer ceases to be a customer if it has not place an order for more than three years .”

For some entity types ,the definition must specify when an instance might change into an instance of another entity type .For example ,consider the situation of a construction company for which bids accepted by potential customers become contra.In this case, a bid might be defined by “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 copy of the bid signed by an officer of the customer .”This definition is also a good example to note how one definition can use other entity type names(in this case , the definition of bid uses the entity type name CUSTOMER).

For some entity types, the definition muset specify what history is to be kept about instance of the entity type. For example, the characteristics of an ITEM in Figure 2-1may change over time,and we may need to keep a complete history of the individual values and when they were in effect.As we will see in some examples later, such statements about keeping history may have ramifications about how we represent the entity type on an E-R diagram and eventually how we store data for the entity instances.

Attributes

Each entity type has a set of attributes associated with it.An attribute is a property or characteristic of an entity type that is of interest to the organization.(Later, we will see that some types of relationships may also have attributes.)Thus,an attributes has a noun name. Following are some typical entity types and their associated attributes:

In naming attributes, we use an initial capital letter followed by lowercase letters. If an attribute name consists of more than one word,we use a space between the words and we start each word with a capital letter,for example, Employee Name or Student Home Address.In E-R diagrams,we represent an attribute by placing its name in the entity it describes.Attributes may also be associated with relationships,as described later, Note that an attribute is associated with exactly one entity or relationship.

Notice in figure 2-5 that all of the attributes of DEPENDENT are characteristics only of an employee’s dependent , not characteristics of an employee.In traditional E-R notation, an entity type(not just weak entities but any entity) does not include attributes of entities to which it is related (what might be called foreign attributes) . For example, DEPENDENT does not include any attributes that indicates to which employee this dependent is associated. This nonredundant feature of the E-R data model is consistent with the shared data property of database. because of relationships,which we discuss shortly,someone accessing data from a database will be able to associate attributes from related entities(e.g. Show on a display screen a Dependent Name and the associated Employee Name).

REQUIRED VERSUS OPTIONAL ATTRIBUTES Each entity(or instance of an entity type)potentially has a value associated with each of the attributes of that entity type. An attribute that must be present for each entity instance is called a required attributes , whereas an attribute that may not have a value is called an optional attribute. For example ,Figure 2-6 shows two STUDENT entities (instance) with their respective attribute values. The only optional attribute for STUDENT is Major.(Some students, specifically Melissa Kraft in this example, have not chosen a major yet, MIS would , of course, be a great career choice!) However, every student must , by the rules of the organization , have values for all the other attributes;that is , we cannot store any data about a student in a STUDENT entity instance unless there are values for all the required attributes. In various E-R diagramming notations, a symbol might appear in front of each attribute to indicate whether it is required(e.g.\*) or optional (e.g. o),or required attributes will be in boldface, whereas optional attributes will be in normal font (the format we use in this text );in many cases, required or optional is indicated within supplemental documentation. In Chapter 3, when we consider entity super types and subtypes, we will see