How sometimes optional attributes imply that there are different types of entities. (for example ,we may want to consider student entity type.)an attribute without a value is said to be null. Thus, each entity has an identifying attributes. If you try to create an entity that has only an identifier, that entity is likely not legitimate .Such a data structure may simply hold a list of legal values for some attribute, which is better kept outside the database.

**Simple versus composite attributes** Some attributes can be broken down into meaningful component part (detailed attributes). A common example is Name, which we saw in Figure 2-5; another is Address, which can usually be broken down into the following component attributes: Street Address, City, State, and Postal Code. A composite attribute is an attribute, such as Address, that has meaningful component parts, which are more detailed attributes. Figure 2-7 shows the notation that we use for composite attributes applied to this example. Most drawing tools do not have a notation for composite attributes, so you simply list all the component parts.

Composite attributes provide considerable flexibility to users, who can either refer to the composite attribute as a single unit or else refer to individual components of that attribute. Thus, for example, a user can either refer to Address or refer to one of its components, such as Stress Address. The decision about whether to subdivide an attribute into its components parts depends in whether users will need to refer to those individual components, and hence, they have organizational meaning. Of course, the designer must always attempt to anticipate possible future usage patterns for the database.

A simple (or atomic) attribute is an attribute is an attribute that cannot be broken down into smaller components that are meaningful for the organization. For example, all the attributes associated with AUTOM Identifier for the Student entity type introduced earlier is Student ID,whereas the identifier for AUTOMOBILE is vehicle ID.Notice that an attribute such as student name is not a candidate identifier,because many students may potentially have the same name,and students,like all people,can change their names.to be a candidate identifier,each entity instance must have a single value for the tattribute and the er diagram,as shown in the student entity type example in figure 2~9a.to be an identifier,the attribute is also required(so the distinguishing value must exist),so an identifier is also in bold.some er drawing software will place a symbol,called a stereotype,in front of the identifier(eg,[id]or[pk])

For some entity types,there is no single(or atomic)attribute that can serve as the identifier(ie,that will ensure uniqueness).howere,two(or more)attributes used in combination may serve as the identifier.a composite identifier is an identifier that consists of a combination is required to identify uniquely individual occurrences of flight.we use the convention that the composite are not underlined. Some data modelers think of a composite identifier as “breaking a tie”created by a simple indentifier. Even with Flight ID,a data modeler would ask a question,such as “Can two flights with the same number occur on the same data ?”If so,yet another attribute is needed to form the composite indentifier and to break the tie.

Some entities may have more than one candidate indentifier.if zhe is more than one candidate identifier ,the designer must choose one of them as the idenyifier.bruce(1992)suggests the follow criteria for select identifiers:

1. choose an identifier that will not chang its value over the life of each instance of the entity type .for example ,the combination of employeename and payoll address (ever if unique ) would be a poor choice as an identifier for employee because the values of employee name and payrill address could easily chang during an employee’s term of employment
2. choose an identifier such that for each instance of the entity,the attribute is guaranteed to have valid values and not be null (or unknown),if the identifier is a composite attribute,suce as flight id in figure2~9,make sure that all parts of the identifier will have valid values.
3. avoid the use of so~called intelligent identifiers (or keys),whose structure indicates classifications,locations,and so on.for example,the first two digits of an identifier value may indicate the warehouse location.such codes are often changed as conditions chang,which renders the identifier values invalid.

OBILE are simple: Vehicle ID, Color,

SINGLE-VALUEDVERSUS MULTIVALUED attributes figure 2‑‑‑6 shows two entity instances with their respective attribute value. For each entity instance ,each of the attributes in the figure has one value .it frequently happens that there is an attribute that may have more one value for a given instance . for example ,the employee entity type in figure 2—8 has an attribute named skill, whose values record the skill (or skills) for that employee. Of course some employees may have more than one skill, such as php programmer and C++ programmer .a MULTIVALUED attribute is an attribute that may take on more than one value for a given entity (or relationship )instance. In this text, we indicate a MULTIVALUED attribute with curly brackets around the attribute name, as shown for the skill attribute in the employee example in figure 2—8 .in Microsoft Visio, once an attribute is placed in an entity ,you can edit that attribute (column),select the collection tab ,and choose one of the options. (typically, MULTIVALUED will be your choice , but one of the other options may be more appropriate for a given situation.) other e—e diagramming tools may use an asterisk （\*）after the attribute name , or you may have to use supplemental documentation to specify a MULTIVALUED attribute .

MULTIVALUED and composite are different concepts, although beginner data modelers often confuse these terms skill, a MULTIVALUED attribute, may occur multiple times for each employee; Employee Name and Payroll Address are both likely composite attributes, each of which occurs once for each employee, but which have component, more atomic attributes, which are not shown in Figure 2-8 for simplicity. See problem and exercise 16 to review the concepts of composite and MULTIVALUED attributes.

STORED VERSUS DERIVED ATTIBUTES Some attribute values that are of interest to users can be calculated or derived from other related attribute values that are stored in the database. For example , suppose that for ab organization , the EMPLOYEE entity type has a Date Employee attribute . If users need to know how many years a person has been employed, that value can be calculated using Date Employed and today’s date. A derives attribute is an attribute whose values can be calculated from related attribute values(plus possibly data not in the database, such as today’s date, the current time , or a security code provided by a system user). We indicate a derived attribute in an E-R diagram by using square brackets around the attribute name, as shown in Figure 2-8 fir the Years Employed attribute. Some E-R diagramming tools use a notation of a forward slash(/)in front of the attribute name to indicate that it is derived.(This notation is borrowed from UML for a virtual attribute.)

In some situations, the value of an attribute can be derived from attributes in related entities. For example , consider an invoice created for each customer at Pine Valley Furniture Company. Order Total would be an attribute of the INVOICE entity, which indicates the total dollar amount that is billed to the customer . The value of Order Total can be computed by summing the Extended Price values(unit price times quantity sold) for the various line items that are billed on the invoice. Formulas for computing values such as this are one type of business rule.

IDENTIFIER ATTRIBUTE An identifier is an attribute (or combination of attributes )whose value distinguishes individual instances of an entity type . That is, no two instances of the entity type may have the same value for the identifier attribute . The identifier for the STUDENT entity type introduced earlier is Student ID, whereas the identifier for AUTOMOBILE is Vehicle ID. Notice that an attribute such as Student Name is not a candidate identifier , because many students may potentially have the same name , and students, like all people , can change their names. To be a candidate identifier , each entity instance must have a single value for the attribute and the attribute must be associated with the entity.We underline identifier names on the E-R diagram, as shown in the STUDENT entity type example in Figure 2-9a.To be an identifier, the attribute is also required(so the distinguishing value must exist), so an identifier is also in bold .Some E-R drawing software will place a symbol , called a stereotype, in front of the identifier(e.g. ID or PK).

For some entity types, there is no single (or atomic) attribute that can serve as the identifier(i.e. That will ensure uniqueness).However,two (or more) attributes used in combination may serve as the identifier . A composite identifier is an identifier that consists of a composite attribute . Figure 2-9b shows the entity FLIGHT with the composite identifier Flight ID.Flight ID in turn has component attributes Flight Number and Date. This combination is required to identify uniquely individual occurrences of FLIGHT.We use the convention that the composite attribute (Flight ID)is underlined.Some data modelers think of a composite identifier as “breaking a tie” crated by a simple identifier. Even with Flight ID, a data modeler would ask a question, such as “Can two flights with the same number occur on the same date?” If so yet another attribute is needed to form the composite identifier and to break the tie.

Some entities may have more than one candidate identifier .If there is more than one candidate identifier , the designer must choose one of them as the identifier . Bruce (1992)suggests the following criteria for selecting identifiers:

1. Choose an identifier that will not change its value over the life of each instance of the entity type .For example , the combination of Employee Name and Payroll Address (even if unique) would be a poor choice as an identifier for EMPLOYEE because the values of Employee Name and Payroll Address could easily change during an employee’s term of employment .
2. Choose an identifier such that for each instance of the entity , the attribute is guaranteed to have valid values and not be null(or unknown) .If the identifier is a composite attribute , such as Flight ID in Figure 2-9b, make sure that all parts of the identifier will have valid values.
3. Avoid the use of so-called intelligent identifiers(or keys ) , whose structure indicates classifications ,locations, and so on . For example , the first two digits of an identifier value may indicate the warehouse location .Such codes are often changed as conditions change, which renders the identifier values in valid.