Section 2: Creation of a Database Schema

In this section you will learn how to create a database schema based on the E-R diagram you created in Section 1. As you proceed through this section, you may either use the E-R diagram you created in Section 1, or you may download my section 1 E-R diagram.

The section has two main parts. First, we ensure that all fundamental (strong) entities have a primary key. Second, we define the domain for each attribute. Third, we let ER*win* do the work of converting the E-R diagram into a relational schema.

Assigning primary keys

Fundamental entities are those that do not depend on any other entity for their primary key. In particular, fundamental entities *are not*:

- subtypes
- associative (composite) entities
- weak (attributive) entities.
- entities at the 'many' end of an identifying 1-to-many relationship. Actually, associative and weak entities are examples of this kind of entity.

In our current example, our E-R diagram appears as in figure 2.1

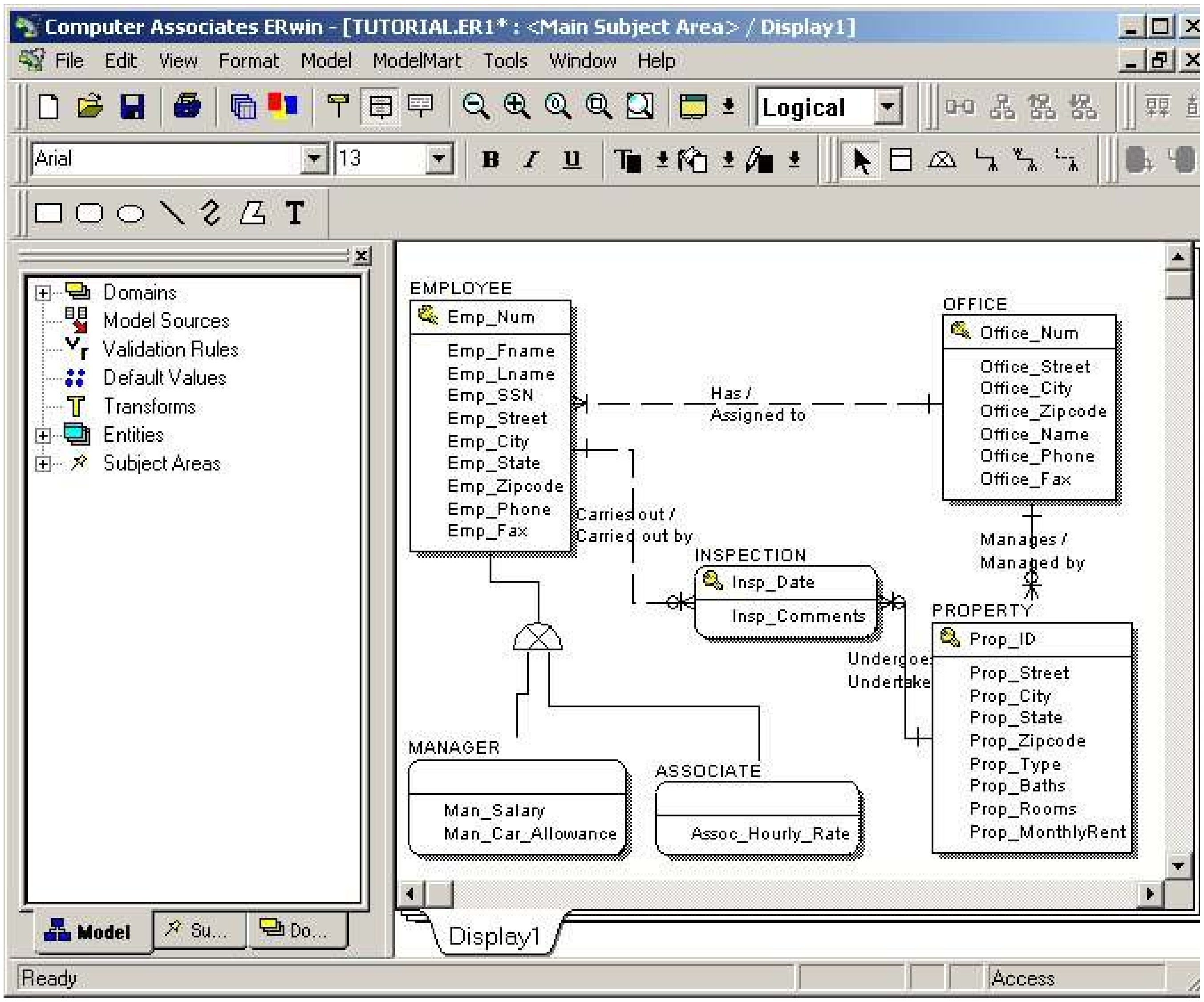


Figure 2.1 E-R diagram

Notice that MANAGER and ASSOCIATE do not have primary keys (they are subtypes). The primary key of INSPECTION includes Insp_Date, but will also include Prop_ID, because the relationship INSPECTION undertaken at PROPERTY is an identifying relationship. Our fundamental entities are: EMPLOYEE, OFFICE, PROPERTY. Each of these three has a fully defined primary key.

Defining Domains

The *domain* of an attribute is the set of values that attribute is permitted to have. For example, the domain of an attribute Day_of_Week would consist of seven values: {Monday, Tuesday, ..., Sunday}. To define a domain we need to define:

1. a datatype (mandatory). Data types can be used to specify character string data, numbers (several flavors), date and time data, etc. Every attribute must at least have a data type. For example, Assoc_Hourly_Rate must be a decimal number with two

decimal places. Prop_Rooms must be an integer. Emp_Fname must be a variable-length character string of up no more than 50 characters.

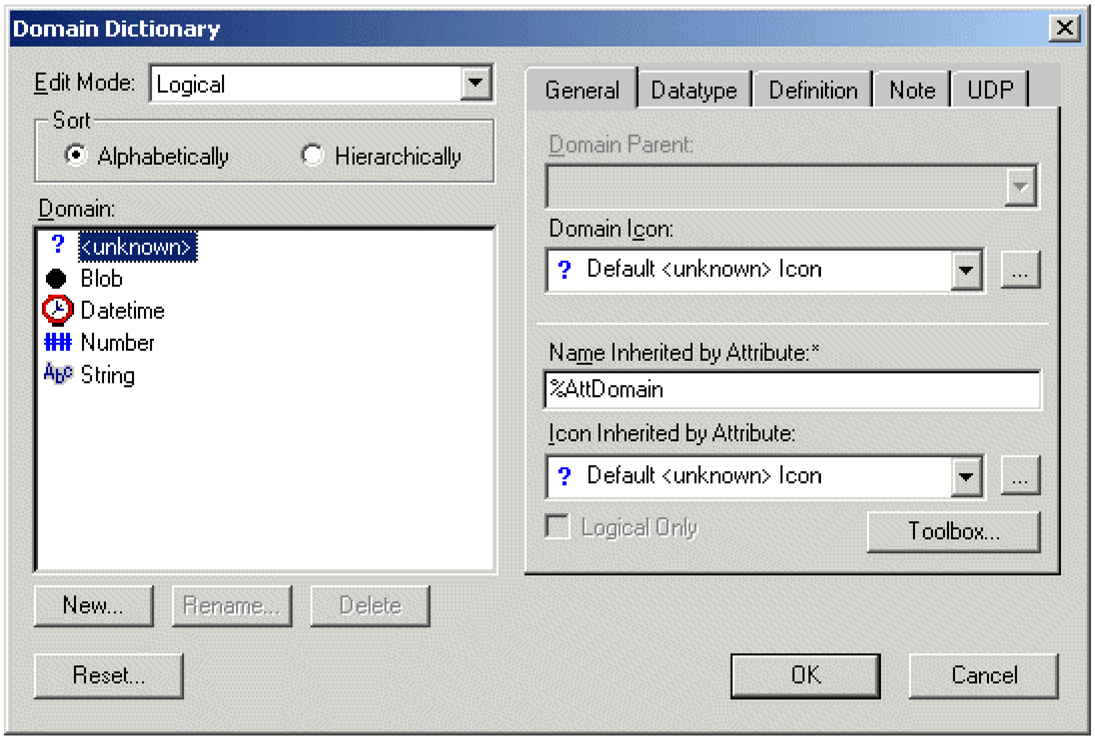
2. constraints (optional). For some attributes, the set of permitted values may be narrower than the set defined by the data type alone. The attribute Day_of_Week is one example. In our E-R diagram, several attributes have more narrowly defined domains. For example, the value of Prop_Type must be in {Home, Apartment, Condo, Commercial}. The value of Prop_MonthlyRent must be greater than $0.

One of the attractive features of domains, aside from the enhanced data integrity they provide, is that domains can be shared by more than one attribute. For example, the domains Emp_State and Prop_State are identical: the list of 50 possible state codes. In an E-R diagram, we can define the domain once, and apply it to multiple attributes. The benefit of this is consistence across attributes and ease of maintenance. If a new state were added to the Union, for example, we could modify Emp_State and Prop_State to accept the new state code ('PR' for Puerto Rico?) simply by making a single change to the domain definition itself.

We will define all the domains we'll need in our E-R diagam first, then apply them to our attributes.

Creating domains

From the Model Menu option, choose Domain Dictionary. You will see a window that looks like Figure 2.2


Domain Dictionary

Edit Mode: Logical

Sort: [✓ Alphabetically] Hierarchically

Domain:
- <unknown>
- Blob
- Datetime
- Number
- String

Domain Parent:

Domain Icon: Default <unknown> Icon

Name Inherited by Attribute:* %AttDomain

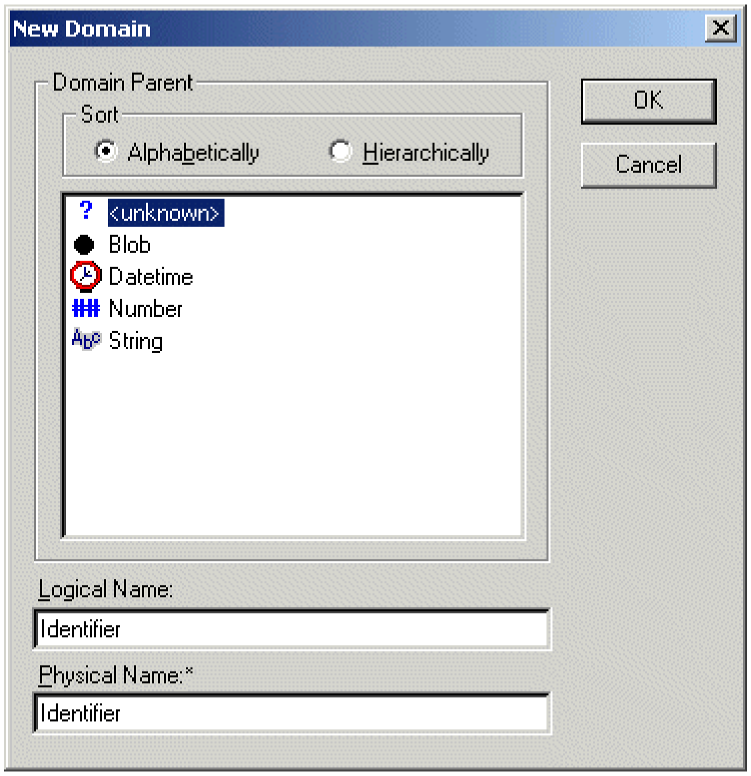
Icon Inherited by Attribute: Default <unknown> Icon

Logical Only

Figure 2.2 Domain Dictionary

We'll first create a new domain that can be applied to numeric identifiers we'll use, such as Emp_Num, Prop_ID, and Office_Num. Click on *New...* to create a new domain. Type in

Identifier as the name of the domain in the Logical Name column as shown in Figure 2.3.



Sort: [✓ Alphabetically] Hierarchically

- <unknown>
- Blob
- Datetime
- Number
- String

Logical Name: Identifier

Physical Name:* Identifier

Figure 2.3 New Domain

Click *OK* to proceed. You will return to the Domain Dictionary window and see that a new domain, *Identifier,* has been created, as shown in Fiure 2.4. Click on the *Data Type* tab, and choose the INTEGER datatype for this domain.

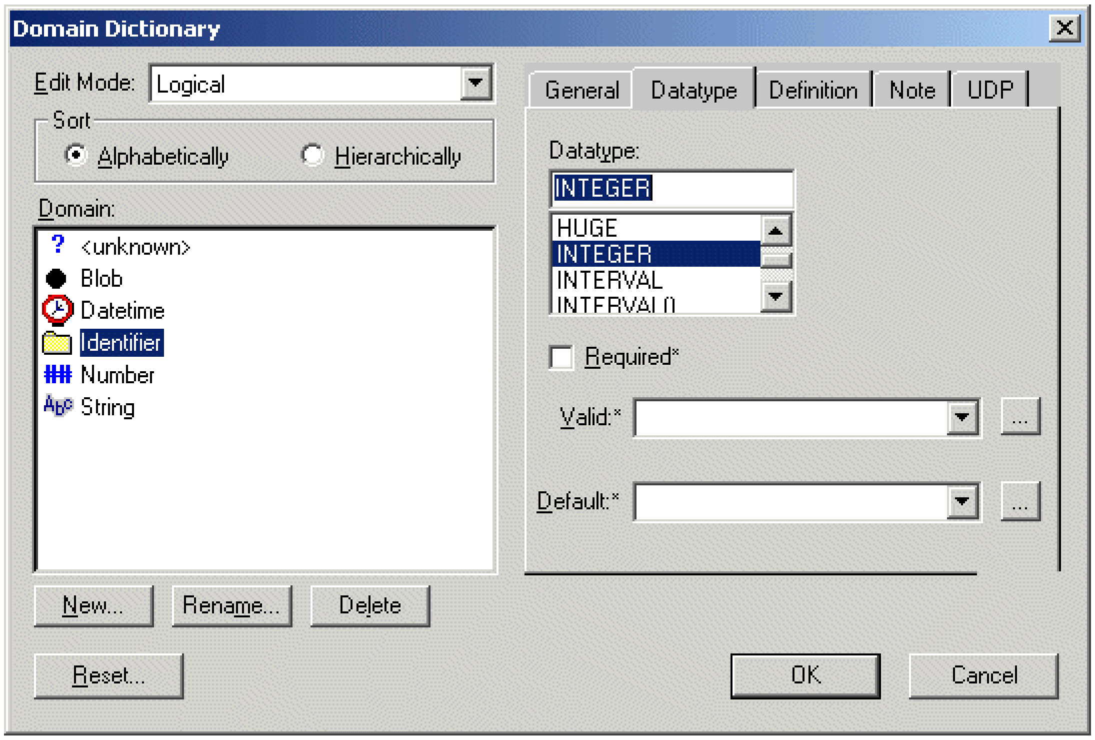


Figure 2.4 Assigning a Data Type to a New Domain

We'll now create another domain, State, to be used for Emp_State, Prop_State, and Office_State. Do NOT click *OK* in the Domain Dictionary window. Rather, click on *New...* again. As shown in Figure 2.5, name this new domain *State*

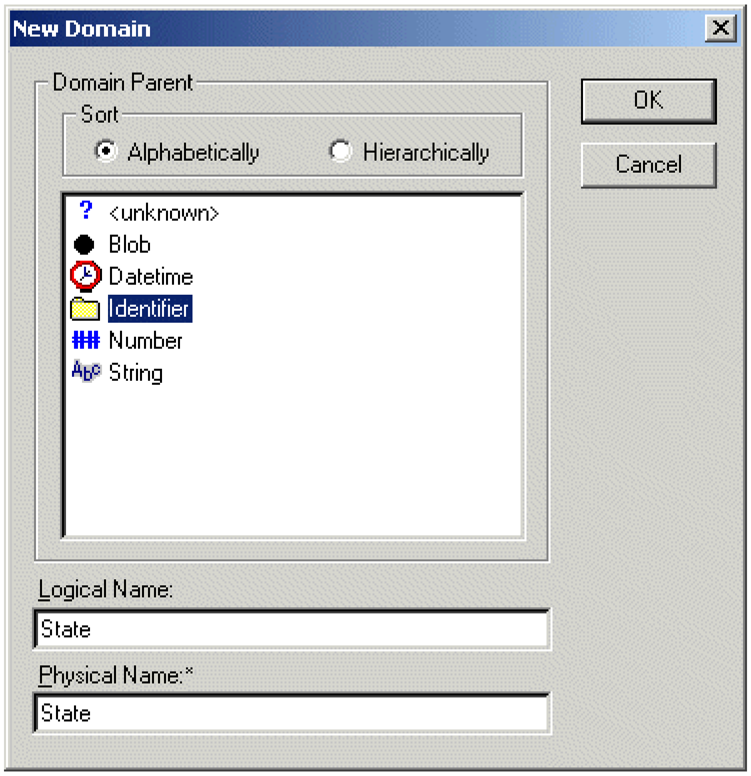


Figure 2.5 The State Domain

Click *OK*. Choose the VARCHAR() datatype from the Data Type tab, and type in the number 2 inside the parentheses, as shown in Figure 2.6. This indicates that the data type is a two-character string. (Actually, CHAR(2) would be the most appropriate, **but when the target database is Microsoft Access, ER*win* doesn't properly associate CHAR() with the Microsoft TEXT datatype**.

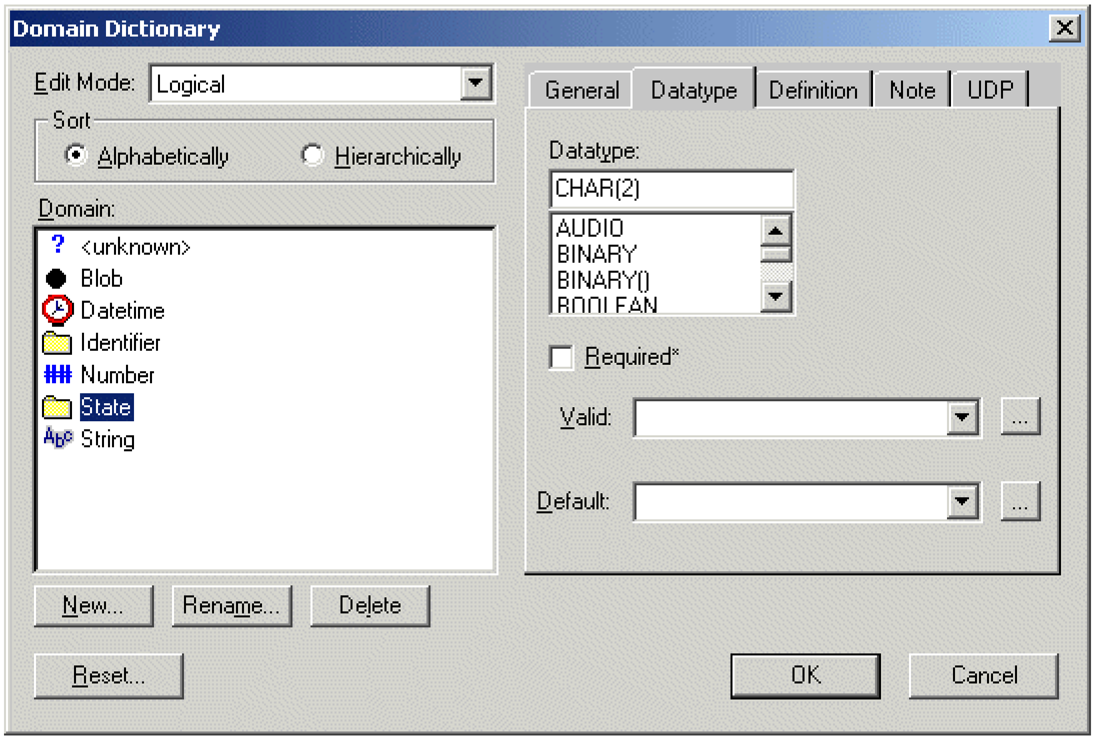


Figure 2.6 State Datatype

We now need to specify the state codes that constitute valid values for this domain. Click on the ellipsis (...) to the right of the combo box field labeled *Valid:* You will see a window in which you can create validation rules, as shown in Figure 2.7.

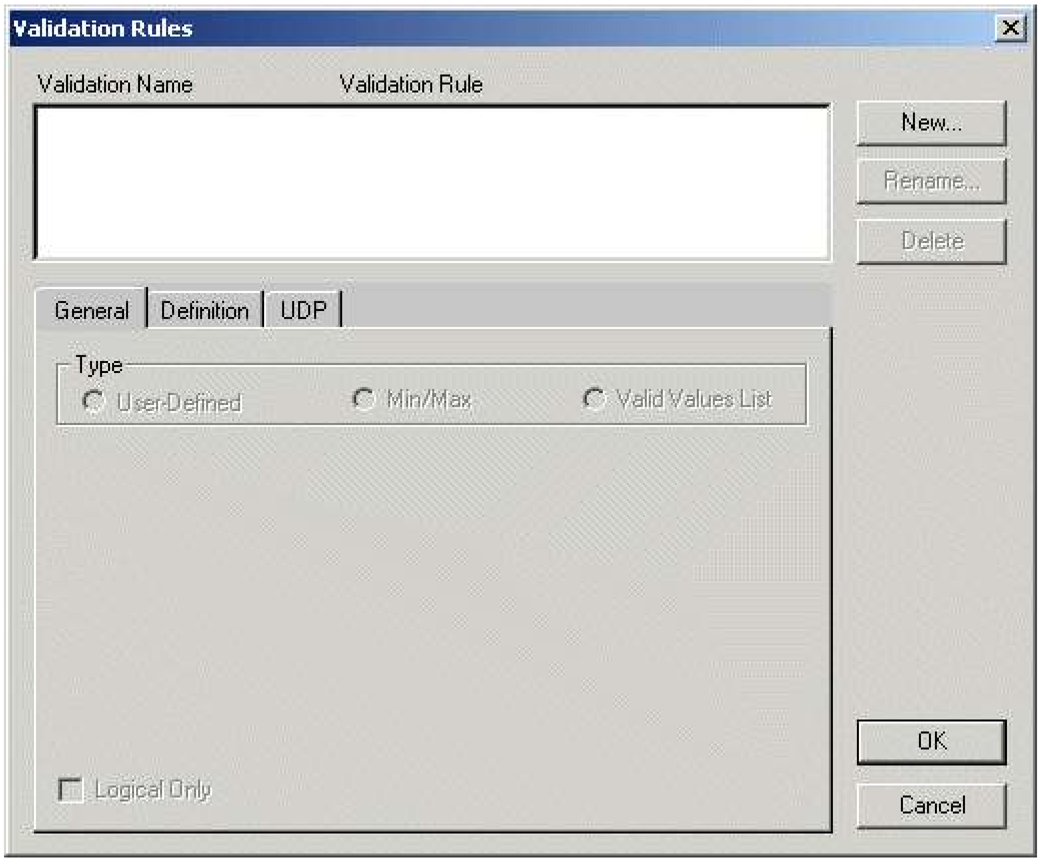
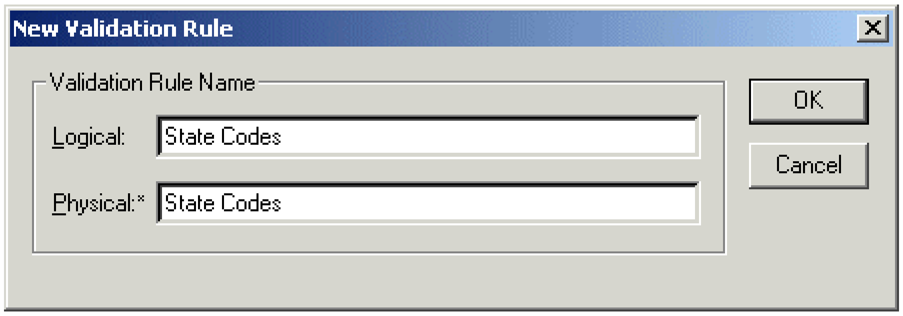


Figure 2.7 Validation Rules

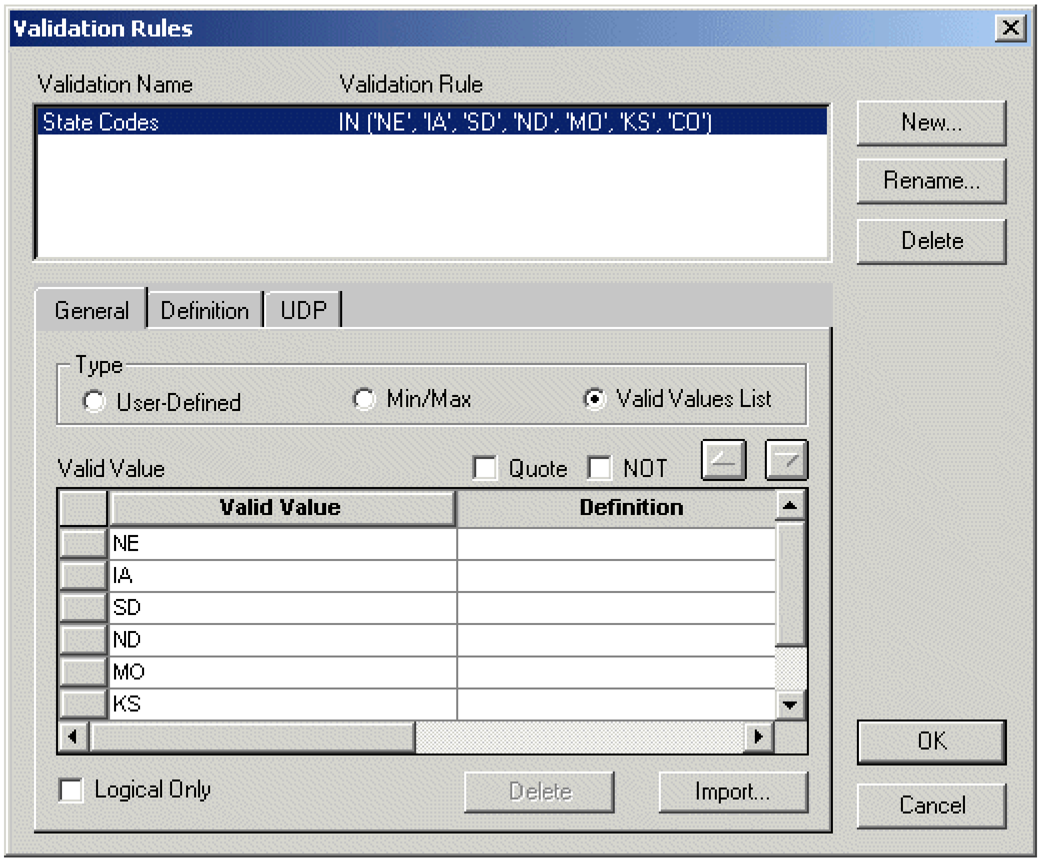
We will create a new validation rule for the State domain. Click on *New...* . Name the validation rule **State Codes** as shown in Figure 2.8.



Logical: State Codes
Physical:* State Codes

Figure 2.8 State Codes

Click on *OK.* You will come back to the Validation Rules window. Here, click the radio button labeled *Valid Values List* and begin typing in valid state codes in the *Valid Value* column, as shown in Figure 2.9.



Validation Name	Validation Rule
State Codes	IN ('NE', 'IA', 'SD', 'ND', 'MO', 'KS', 'CO')

Type: User-Defined, Min/Max, [✓ Valid Values List]

Valid Value ☐ Quote ☐ NOT

Valid Value	Definition
NE	
IA	
SD	
ND	
MO	
KS	

☐ Logical Only

Figure 2.9 Entering Valid Domain Values

You will notice that ER*win* builds a SQL IN clause listing the values you have specified. When you have completed, click *OK.*

Use a similar set of steps to create the domains with the data types and validation rules indicated in Table 2.1. Be sure that the validation rule for the domain is either empty, or is the validation rule you want. Sometimes ER*win* will allow a validation rule created for one domain to be applied to the definition of a second domain without the user's involvement.

Domain name	**Datatype**	**Validation rule**
Name	VARCHAR (50)	
Street	VARCHAR (100)	
City	VARCHAR (50)	
State	VARCHAR (2)	Note: Although State should be CHAR(2) for fixed-length strings, I think there is a bug in the software in which such data types will eventually get converted into the Byte datatype in Access. This might be because Access doesn't have a proper fixed-length string data type. Not quite what we want...
	VARCHAR	

Zipcode	(10)	
Phone	VARCHAR (15)	
Currency	DECIMAL (9,2)	Create a new validation rule called *Currency* with a Minimum value = 0
SSN	VARCHAR (11)	Note: This should be CHAR(11), but see the note above to understand why we are using VARCHAR(11) instead.
Comment	VARCHAR (255)	
Property Type	VARCHAR (25)	Create a new validation rule called *Property Type* with a list of valid values consisting of: {Home, Apartment, Condo, Commercial}
Date	DATE	
Non negative Integer	INTEGER	Create a new validate rule called *non negative integer* with a Minimum value = 0

Table 2.1 Domain creation

When you are finished, the Domain Dictionary should show the domains you created, as shown in Figure 2.10

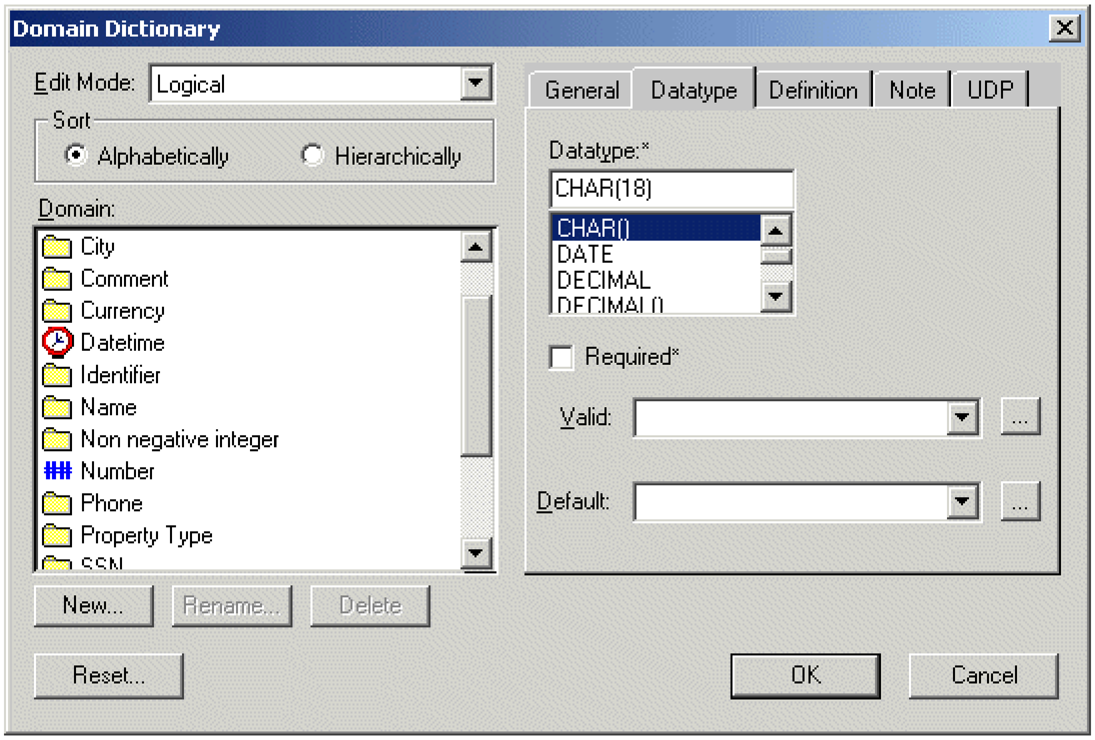
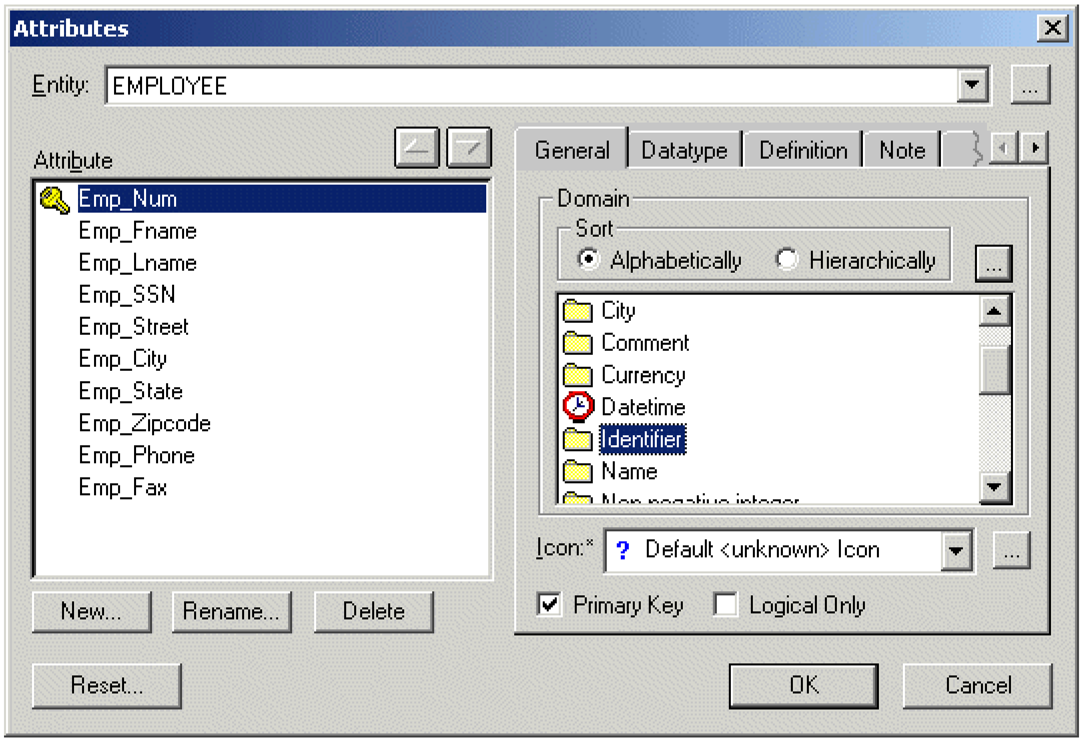


Figure 2.10 Domain List

Click *OK* when you have completed.

Applying Domains to Attributes

The final step of working with domains is to associate a domain with each attribute. To do this, double-click on one of the entities in your E-R diagram, such as EMPLOYEE. You will see a list of attributes in the left-hand portion of the window, and a list of the domains on the right-hand side. To associated a domain with an attribute, click first on the attribute, then click on the associated domain. You will not receive any additional informational message that this association has been made. For example, Figure 2.11 shows that the Emp_Num attribute has been associated with the Identifier domain.


Attributes

Entity: EMPLOYEE

Attribute:
- Emp_Num
- Emp_Fname
- Emp_Lname
- Emp_SSN
- Emp_Street
- Emp_City
- Emp_State
- Emp_Zipcode
- Emp_Phone
- Emp_Fax

Domain
Sort: [✓ Alphabetically] Hierarchically
- City
- Comment
- Currency
- Datetime
- Identifier
- Name
- Non-negative integer

Icon: Default <unknown> Icon

[✓] Primary Key [ ] Logical Only

Figure 2.11 Associated a Domain with an Attribute

Associate each of the attributes in your E-R diagram with a domain, as instructed in Table 2.2. Notice that the same domain may be used for more than one attribute.

Attribute	Domain
Emp_Num	Identifier
Emp_Fname	Name
Emp_Lname	Name
Emp_Street	Street
Emp_City	City
Emp_State	State
Emp_Zipcode	Zipcode
Emp_SSN	SSN
Emp_Phone	Phone
Emp_Fax	Phone
Office_Num	Identifier

Office_Street	Street
Office_Zipcode	Zipcode
Office_Name	Name
Office_Phone	Phone
Office_Fax	Fax
Insp_Date	Date
Insp_Comments	Comment
Prop_ID	Identifier
Prop_Street	Street
Prop_City	City
Prop_State	State
Prop_Zipcode	Zipcode
Prop_Type	Property Type
Prop_Baths	Non negative integer
Prop_Rooms	Non negative integer
Prop_MonthlyRent	Currency
Man_Salary	Currency
Man_Car_Allowance	Currency
Assoc_Hourly_Rate	Currency

Table 2.2 Attribute - Domain Association

Converting an E-R Diagram into a Relational Schema

The first stop in converting an E-R Diagram into a relational schema is to transform each entity and each relationship into its counterpart in a relational schema consisting of relational tables. We have covered in class the rules for such conversions. ER*win* is easily able to manage this transformation based on those rules.

From the Format->Entity Display menu option, make sure that the following three options are checked (and choose them if they are not):

1. Primary Key Designator (should already be checked)
2. Foreign Key Designator (FK)
3. Show Migrated Attributes

When these three are checked, the E-R Diagram should appear as in Figure 2.12

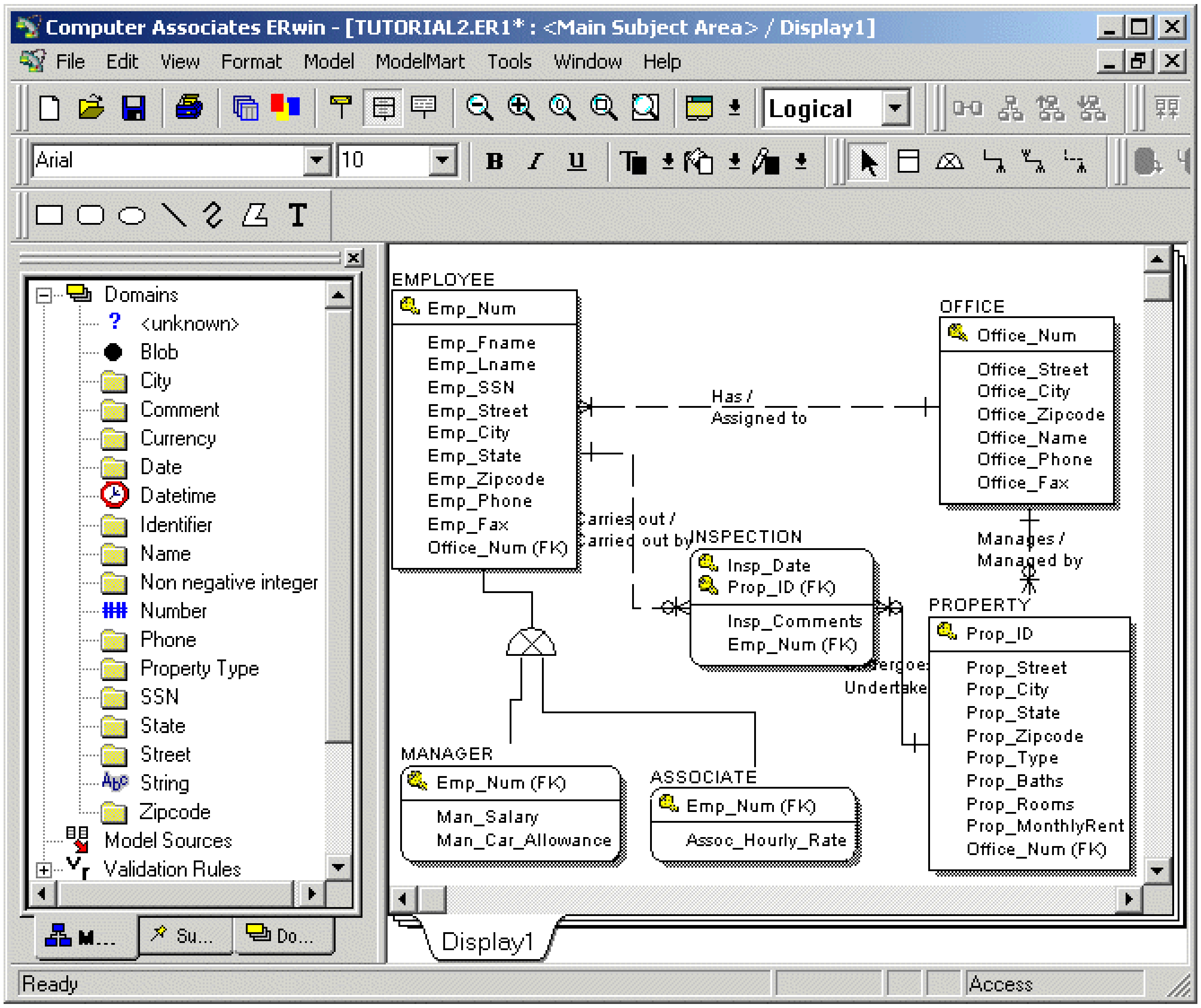


Figure 2.12 E-R Diagram Showing Migrated Attributes and Foreign Keys

Notice that ER*win* has automatically done the following:

1. Foreign keys are used to reflect the one-to-many relationships. Because ER*win* automatically provides foreign keys when such relationships exist, any foreign keys created by the data modeler would not be redundant and would have to be deleted.
2. Foreign keys that result from identifying relationships automatically become part of the primary key (see INSPECTION).
3. Subtypes inherit the primary key of the supertype, and those primary keys also serve in a foreign key capacity.

Finally, to transform the logical E-R diagram into a physical Relational schema, simply choose *Physical* instead of *Logical* in the combo box (Logical)at the top of the window shown in Figure 2.12. The result is shown in Figure 2.13.

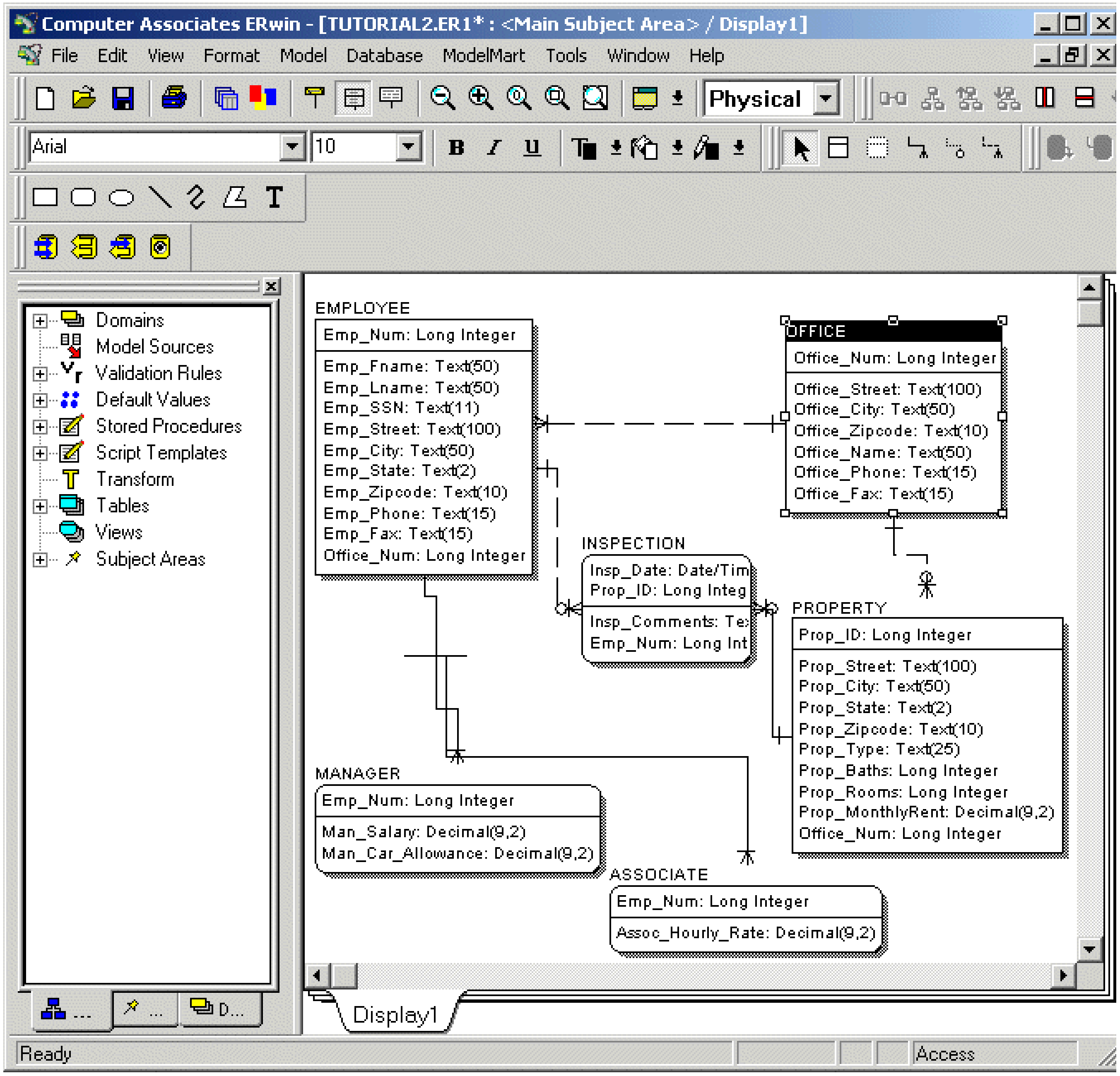


Figure 2.13 The Physical Relational Schema Model

Notice that the datatypes, as defined in the domain definitions, are all clearly represented.

In our final section, Section 3, we will examine how we can use ER*win* to create a database in Microsoft Access (or almost any other DBMS) directly, based on our design.