Lesson 7

Entity Relationship Diagrams

OVERVIEW

This data modeling technique provides a precise method for detailing and illuminating the interrelationships of the data used by a system. You can depict the "entities" (see definition below) in the data you are modeling and the relationships between them by drawing them onto an entity relationship diagram (ERD). The data model (ERD) shows the major data objects of an application and how they fit together using the relationships. You can define the primary keys for the data entities and the composition of the data attributes of the entities in the Visible Analyst repository. (Defining primary keys and adding data attributes are explained in Lesson 16, Working with the Repository Functions.) The defined components can then be displayed on your ERD diagram by selecting these options from the View menu.

A diagram containing a picture of all or a subset of your data is called a "view." Each view can show an arbitrarily large or small part of your data model. You can show multiple views of your data model by including different combinations of entities and relationships on various diagrams. However, the entire data model, including the data elements composing each entity, is retained in the repository and can be accessed by creating a global view of the data model. This feature is explained in this lesson.

Definitions

The important diagram constructs in entity relationship data modeling include:

Entity — The entity (or, more properly, the entity type) is nothing more than a real-world object that you want to describe. The most generic type of entity is really a fundamental or independent entity, but is usually simply called an entity. It is composed of data elements (also called attributes), and you can describe these in the entity's repository composition field. A fundamental entity is an object or event. It is represented on an entity relationship diagram as a rectangle and is accessed by the first symbol button on the control bar.

Associative Entity

Another type is the associative entity (sometimes called a junction, intersection or concatenated entity, a gerund or a correlation table). This is basically a relationship (see below) about which you want to store information. It can only exist between two other entities. For example, the relationship between a customer and a product produces as a by-product the associative entity purchase order. A purchase order entity would not exist without the relationship between the other two entities. An associative entity is represented as a rectangle with straight diagonal lines across each corner. It is accessed by the second symbol button on the control bar.

Attributive Entity

The third entity type is the attributive or dependent entity. This is used to show data that is wholly dependent upon the existence of a fundamental entity. It is also used to show repeating subgroups of data. For example, the associative entity purchase order may have a dependent attributive entity named shipment showing the full or partial shipments that fulfill the purchase order. It is represented as a rectangle with rounded lines across each corner and is accessed by the last symbol button on the control bar.

Relationship

A relationship shows how one entity interacts with or can be affiliated with another entity. It appears on a diagram as a line drawn between two entities. Relationship lines ordinarily have two labels, one for each direction. The relationship lines can have terminators that show that the entities relate to each other on a one-to-one, one-to-many, or many-to-many basis (the relationship's cardinality), and whether the relationship is optional or mandatory. There are four line buttons on the control bar. Line types may be changed after they are drawn on the diagram.

Supertype/Subtypes

Specialized subtype entities can be created that are based on a generalized supertype entity and share common attributes. Only the attributes unique to the specialized entity need to be listed in the subtype entity. This is closely related to the object class inheritance concept. Visible Analyst also provides a detail field for specifying the exact number of relationships, if known. The supertype/subtype button is the fifth line button on the control bar.

Cluster

A cluster is a collection of entities and the relationships between them. It is not truly a part of your data model because it carries no new information. However, it can be very useful when you want to show very large data models on a single diagram and still have it

comprehensible. You have the ability to cluster groups of entities and show these clusters and the relationships between them in summary fashion on a diagram. This limits the amount of detail on the diagram so that the larger outlines of what is contained in your data model are more visible.

A cluster is created in the repository and entities are added to its composition field. A cluster view can then be created by Visible Analyst to display the *pseudo*-relationships between clusters rather than real relationships between specific entities. The diagram Visible Analyst generates is an *unstructured* diagram, but the information contained in the diagram pertains to your entity relationship diagrams. For more information on Clusters, see the *Operation Manual* or the online help system.

View Object

A view object can be thought of as a derived or virtual table. It is composed of two components: a list of column names and a select statement used to filter information from the tables in the view. For each view, there is one primary select clause and any number of sub-select and union select clauses. Using the Define View dialog box, you select the tables and columns and define the join relationships, clauses and flags to be used by the view. For more information on view objects, see the *Operation Manual* or the online help. (View objects are not available in the Education Editions of Visible Analyst.)

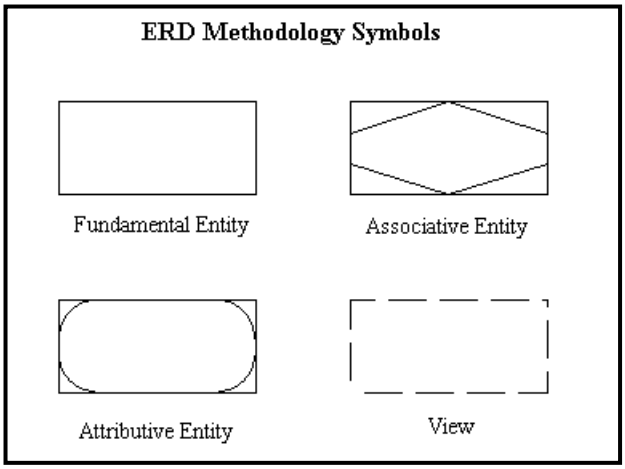

Figure 7-1 Entity Relationship Diagramming Symbols

Relationship Cardinality

Visible Analyst supports four different relationship cardinality notations: IDEF1X, Crowsfoot, Arrow, and Bachman. The type of notation you use is up to you, and you select it when a new project is created. The number of names per relationship line is also your choice. You can indicate one or two names per relationship. For this lesson, we use the standard Crowsfoot notation with two names per relationship.

If you select IDEF1X as the relationship cardinality when creating the project, the default notation is IDEF1X. You would then select Crowsfoot, Arrow or Bachman as an alternate cardinality notation.

DEVELOPING YOUR DATA MODEL

Each entity relationship diagram is complete in and of itself and shows one view of the data model of your project. (Remember that a view is a portion or subset of your entire data model represented on a single diagram.) When beginning your data model, you must manually add new entities and relationships to a view diagram. After this has been done, you can create additional views by using the File menu View function to select existing entities and relationships from the repository. Visible Analyst automatically draws the views for you. Then you can add to or subtract from each view and rearrange it as you wish. Thus you avoid having to draw portions of your data model repeatedly on different views.

Adding Entities to a View

Since the basic building block of the data model is the entity type (or simply, the entity) and since relationships cannot exist except to relate already existing entities, you begin by adding entities to a view.

Set the Zoom Level:	1	From the View menu, select 66% zoom so that you can see all of your needed workspace.
Create a New Diagram:	2	From the File menu, select New Diagram.
	3	Select the diagram type to be Entity Relationship with standard drawing method.
	4	Select the Page Size to be Standard.
	5	Click OK.
Add Entities:	6	Click the first symbol icon, the rectangle. This is a fundamental entity.

7 Place the cursor in the middle of the diagram workspace and click the *left* mouse button. An entity is drawn.

8 Name the entity "Student Driver" and click OK.

9 Add another fundamental entity below the first, and name it "Driving School."

10 Add another fundamental entity below Driving School, and name it "Driving Lessons."

Figure 7-2 New Entities

Save the Diagram:

11 From the File menu, choose Save and name the diagram "Driving School View."

Changing a Symbol Type

In the diagram we have created, the entity Driving Lessons is actually an attributive entity because the entity exists solely because it is an attribute of the fundamental entity Driving School. Since we placed it on the diagram as a fundamental entity, it is necessary to change the symbol type.

Select Symbol to Change:	1	Put the cursor in selection mode by clicking the ↖ button on the control bar.
	2	Click the symbol labeled Driving Lessons with the *right* mouse button so that its Object menu appears.
Change the Entity Type:	3	Select Change Item. The Scope must be set to Global change in the Change Object dialog box. This option is important when you change an object's type or label. Selecting Global causes the change to be made on every diagram where that object occurs. If you select Individual, the change is only made to the selected object. A Local change would modify all occurrences on the current diagram. All changes to a symbol type must be Global.
	4	Select Change Type.
	5	Select Attributive Entity and click OK.
	6	Click OK on the Change Object dialog box. The symbol is changed on the diagram.

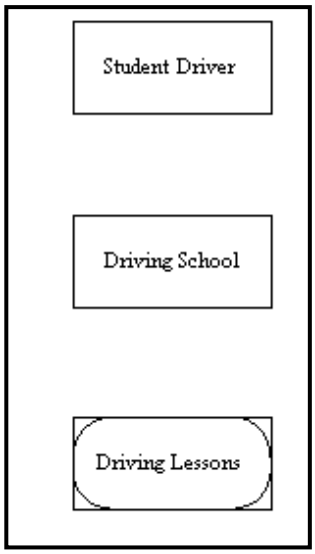

Figure 7-3 Changed Entity Type

Adding Relationship Lines

We need to establish the relationships between the entities on the current diagram.

Draw the Relationship:

1 Click the first line button on the control bar.

2 Draw a line from Driving School to Student Driver. The procedure is the same as that used to draw a line in Lesson 5 - Diagramming. Click and hold the *left* mouse button where you want the line to begin, drag the line to where you want it to end. If you release the button within the symbol, the line is connected automatically. If not, you must double-click the left mouse button to end the line.

Note

- When you use an elbow line and the elbow in the line does not bend in the direction that you want it to, click the *right* mouse button while you are still holding the *left* one, and the elbow inverts.

Label the Relationship:

3 Enter "Instructs" for the label of the first relationship. To set the relationship cardinality, click One for the Minimum, and click Many for the Maximum. This means that "Driving School instructs one or many Student Drivers." If you know the exact maximum number of relationships, you can enter it in the detail box. (See Figure 7-4.)

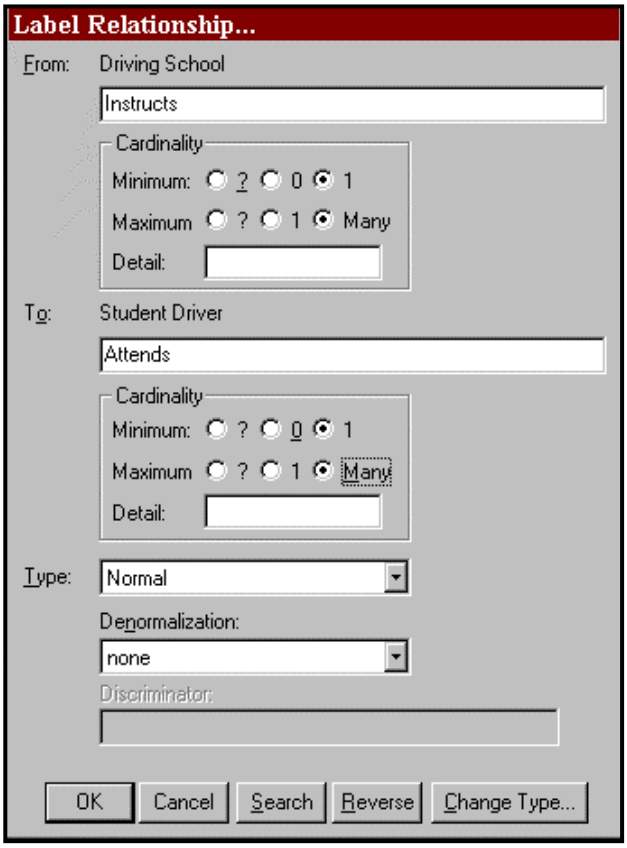

Figure 7-4 Label Relationship Dialog Box

4 Press the TAB key to move the cursor to the next field or click the mouse in the other label field.

5 Enter “Attends” for the reverse relationship name. For the Minimum click One, and for the Maximum click Many. (This deliberate error is added to demonstrate the capabilities of the Analyze function.) It means a “Student Driver attends one to many Driving School.” Both of these relationships are considered mandatory because it is necessary to attend driving school to be a student driver, and it is necessary to have students to be a driving school. Ensure that Type is set to Normal, and click OK.

Draw Another Relationship:	6	Draw a line from Driving School to Driving Lessons. For the first label, type "Offers," and set Minimum to Zero and Maximum to Many. For the second label, type "Are Provided By." Because this is an Identifying relationship, the cardinality is automatically set to 1:1. Click OK.
Save:	7	Press CTRL+S to save the diagram.

Analyzing the Diagram

The Analyze function checks to ensure that the diagram is syntactically correct, meaning that all relationship lines and symbols are labeled. You can also use the Analyze function to check for certain normalization errors.

Start Analyze:	1	Select Analyze from the Diagram menu.
	2	Choose Current Diagram and Syntax Check. Click OK. It tells you that the current diagram is correct.
Insert an Error:	3	Add a symbol to the diagram without naming it.
Analyze Again:	4	Run Analyze again. You see an error message indicating that there is one unnamed entity. Click Cancel to return to the diagram. The unnamed entity can be deleted from the diagram by highlighting it with the cursor in selection mode and pressing Delete.
Analyze Still Again:	5	Run Analyze again, but this time choose Normalization. You see the error message that the relationship "Driving School [Instructs] Student Driver" is not normalized. This is true. The error indicates that the cardinality is 0:many or many:many in both directions. It is flagged as an error because optional:optional and many:many relationships can be difficult to implement. Click Cancel to close the box.
Correct Cardinality Error:	6	To change the cardinality of the relationship Attends, click the relationship line with the *right* button.
	7	Select Change Item. Change the cardinality for Attends from a maximum of Many to a maximum of One.

8 Click OK.

Analyze Once More: 9 Select Analyze from the Diagram menu. Choose Normalization and click OK. The diagram is now correct.

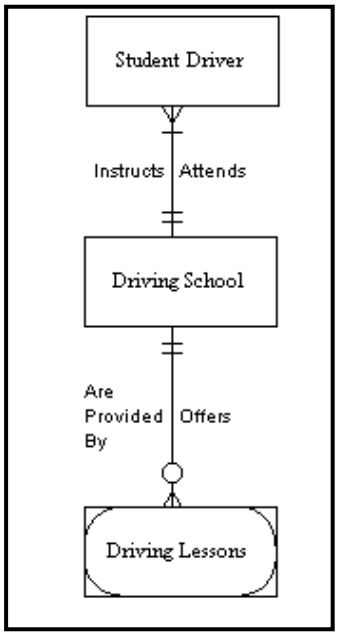

Figure 7-5 Normalized Diagram

Automatically Generating a View of Your Data Model

Another very useful feature of Visible Analyst is the ability to generate new data model views automatically. Since a data model can become very large and sometimes very difficult to decipher with many relationship lines and symbols, generating a specific view of the data model allows you to focus on one portion of your data model without having to redraw all of the symbols and connections that you want to have on the diagram. The function for generating a view is found on the View of Data Model submenu from the File menu.

There are three different options for generating a view from this function.

- There is an option to generate a global view of your data model. All of the entities and relationships that are in the repository are placed on one diagram. This feature is important when additions are made to one portion of the data model and you would

like to see how those changes have affected the entire model. Another use for this feature is to generate an entity relationship diagram for imported entity information.

- You can generate a new view, allowing you to choose from the entities you have already created on a diagram or in the repository those entities and attached relationships you would like displayed on a new diagram. This allows you to make additions or changes to your entire data model while concentrating on only one portion.
- The other view option from the View of Data Model option is Process. A process view is an entity relationship diagram that represents a subset of your data model and is based upon a process existing on a data flow diagram or in the repository. Data elements that enter or leave the selected process in data flows and that are also contained in the composition of entities cause those entities to appear in the process view, along with the relationships existing between pairs of entities. A process view allows you to concentrate on the specific portion of your data model that is involved with the selected process. This is the type of view that you now create. The composition information for the entities that appeared, as well as the attribute information of the particular process, has already been entered for you in the sample diagrams we supplied. This is so that you do not have to enter the information necessary to demonstrate this feature of Visible Analyst.

To create the process view:

Start View Generation: 1 Select View of Data Model from the File menu, then choose Process. The Select Process for Views dialog box appears.

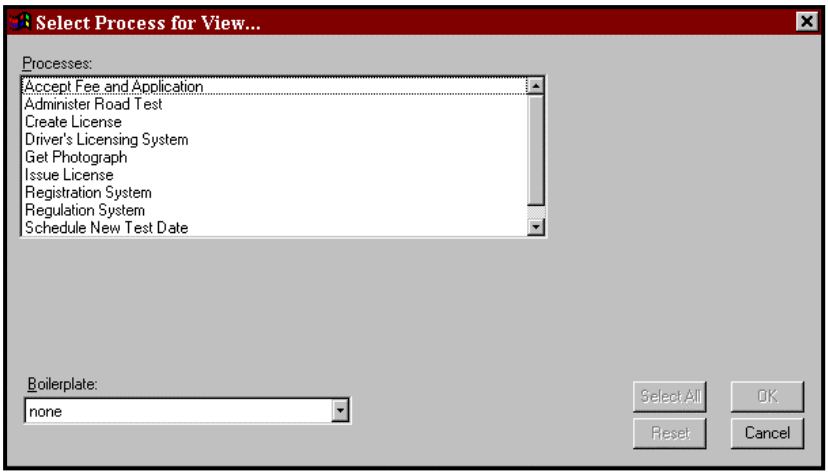

Figure 7-6 Process View Dialog Box

Select the Process:	2	Click the process Issue License and click OK. Visible Analyst searches the repository for entities that contain data elements in common with the data flows that are attached to Issue License and creates a "View" of the data model.
Save the New View:	3	Select Save from the File menu.
	4	Title the diagram "Process View: Issue License." This diagram is a subset of your entire data model.
	5	Click OK.

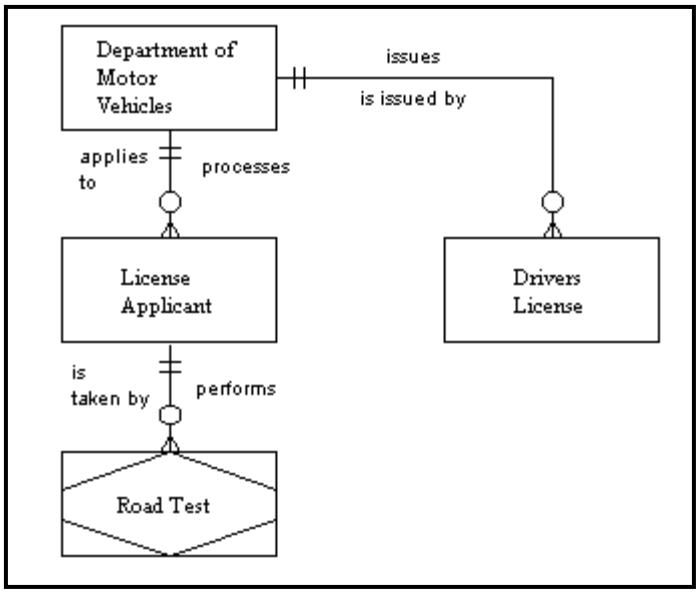

Figure 7-7 The Generated Process View