CHAPTER 4

MASTER-DETAIL FORMS

CHAPTER OBJECTIVES

In this Chapter, you will learn about:

✔ Master-Detail Forms Page 108

In the previous Chapters, you created and worked with forms that had only one base-table block. In this Chapter, you will create a form with multiple base-table blocks and establish a relationship between them. The relationship will, among other things, allow your users to issue a query in the master block, which will cause the form to issue a corresponding query in the detail block.

You will create the master and detail blocks using wizards. This will automatically create an object called a relation. You will adjust the properties of the relation to change the behavior of the form.

go to contents

LAB 4.1

MASTER-DETAIL FORMS

LAB OBJECTIVES

After this Lab, you will be able to:

- Create a Master-Detail Form
- Work with Master-Detail Forms and Relations

In the STUDENT schema, as in almost all schemas in relational databases, there are tables that are related. If the relationship is a primary-foreign key relationship, then one table can be considered the parent and the other can be considered the child.

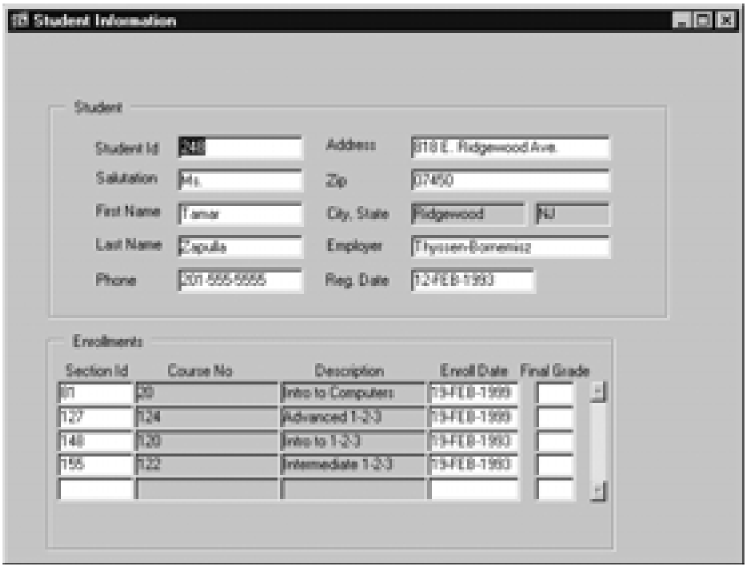
■ FOR EXAMPLE:

In the STUDENT database, there is a table containing data about students and a table containing data about enrollments. There is a primary-foreign key relationship between the STUDENT and ENROLLMENT tables that tells you that for each student record there may be zero or many enrollment records.

In Forms, you can work with this relationship using a master-detail form like the one in Figure 4.1.

A master-detail form is powerful in that it allows you to relate two blocks in the same way that two tables are related in a database. Thus, the users can see the data from both tables in a meaningful way. For instance, when they query on a certain student record, they will see only that student's corresponding enrollment records. Not only will they be allowed to see the records, but they will be able to insert, update, and delete records in these blocks as well.

go to contents



Student

Student Id: 248
Salutation: Ms.
First Name: Tamar
Last Name: Zapula
Phone: 201-555-5555
Address: 818 E. Ridgewood Ave.
Zip: 07450
City, State: Ridgewood NJ
Employer: Thyssen-Bornemisz
Reg. Date: 12-FEB-1993

Enrollments

Section Id	Course No	Description	Enroll Date	Final Grade
81	20	Intro to Computers	19-FEB-1999	
127	124	Advanced 1-2-3	19-FEB-1999	
148	120	Intro to 1-2-3	19-FEB-1993	
155	122	Intermediate 1-2-3	19-FEB-1993	

Figure 4.1 ■ A master-detail form showing a `Student` and her associated `Enrollments`.

The form in Figure 4.1 was created using wizards and then manually edited in the Layout Editor.

Note that the STUDENT items that belong to the master block are laid out in Form style with only one record displayed. The ENROLLMENT items that belong to the detail block are laid out in Tabular style with five records displayed. This single-record block to multi-record block layout style is typical for a master-detail form.

The wizards can help tremendously in creating master and detail blocks. They create blocks as they normally do, but also help you create and configure the objects that will coordinate the master and detail block.

There are a number of objects that work together to coordinate the processing of master and detail blocks. The main object is called a relation. Its job is to hold the join condition that relates the blocks.

The relation object has a number of properties that control how the master and detail blocks are coordinated. The creation of the relation object also initiates the creation of some triggers and program units. These triggers are written by the Form Builder for you.

LAB 4.1 EXERCISES

4.1.1 CREATE A MASTER-DETAIL FORM

In the following Exercises, you will create a master-detail form based on the `INSTRUCTOR` and `SECTION` tables.

a) Which table will the master block be based on? the detail? Why?

____

____

Create the master block and its canvas using the wizards. Include all of the columns as items in the blocks, but do **not** display the audit columns on the canvas. Leave `Enforce data integrity` **unchecked.** Lay the items out in `Form` style.

b) Did you have to do anything to indicate that this was to be a master block?

____

____

Start to create the detail block using the wizards. Leave `Enforce data integrity` **unchecked.**

c) Have you encountered a new wizard page? What will it help you do?

____

____

`Auto-join data blocks` should be checked. Click the `Create Relationship` button.

d) What is this `List of Values` dialog showing you? What happens after you click the `OK` button?

____

____

go to contents

e) Because the wizard has already written the join condition for you, do you need to change the values in the `Detail Item` and `Master Item` list items?

Click the `Next` button and continue on to the Layout Wizard. The `SECTION` block should appear directly below the items in the `INSTRUCTOR` block.

f) Which canvas should you choose for the `SECTION` block? Is it necessary to create a new one?

Continue through the Layout Wizard until you get to the items page.

g) Which of these items already appears in the `INSTRUCTOR` frame? Is it necessary to display it again in the `SECTION` frame?

Continue through the Layout Wizard. Select `Tabular` as the display style and display five records. Click the `Finish` button when you are done.

h) If the layout of the `SECTION` items is not so pleasing, how can you make quick changes?

Run the form and issue a query.

i) Have the items in both frames been populated?

go to contents

j) What happens to the records in the SECTION frame when you scroll to the next record in the INSTRUCTOR frame?

Save the form you have just created as R_INS_SEC.fmb.

4.1.2 WORK WITH MASTER-DETAIL FORMS AND RELATIONS

Use R_INS_SEC.fmb for all of the Exercises in this section.

a) Which block is listed first in the Object Navigator? What would be the problem if SECTION were listed first?

b) Has a relation object been created? What is it called? Which of the blocks owns the relation?

c) What is the value of the SECTION.INSTRUCTOR_ID item's Copy Value from Item property? Why has the Form Builder done this?

d) What form-level triggers have been created? What block-level triggers have been created?

go to contents

e) Judging from their names, what do you think these triggers do?

Expand the Object Navigator so that you can see the triggers under the INSTRUCTOR block. Change the INSTRUCTOR_SECTION relation's Delete Record Behavior property to Cascading.

f) What has happened to the triggers under the INSTRUCTOR block?

Double-click the new trigger that has been created and view its code.

g) Judging from the code you see and what you know about the term *cascading* from SQL, what will happen if you run the form and delete a master record?

Run the form. Navigate to SECTION_ID in the SECTION frame and issue a query.

h) Were the SECTION records returned to the form? How about the INSTRUCTOR records?

Exit the form and return to the Form Builder. Change the INSTRUCTOR_SECTION relation's Prevent Masterless Operations property to Yes. Run the form. Navigate to SECTION_ID in the SECTION frame and issue a query.

go to contents

i) Were the SECTION records returned to the form this time? What does the Prevent Masterless Operations property do?

Exit the form and return to the Form Builder. Change the INSTRUCTOR_SECTION relation's Deferred property to Yes and the Automatic Query property to No. Run the form and issue a query.

j) Were the items in the SECTION frame populated?

k) What then does the Deferred property defer?

Navigate to SECTION_ID in the SECTION frame and click the Execute Query button on the toolbar.

l) What happened after you issued the second query?

m) How could this be useful?

Exit the form and return to the Form Builder. Change the INSTRUCTOR_SECTION relation's Automatic Query property to Yes. Run the form and issue a query.

n) Will you have to issue a second query this time? If not, what should you do to populate the items in the SECTION frame?

go to contents

LAB 4.1

o) How could this be useful?

LAB 4.1 EXERCISE ANSWERS

4.1.1 ANSWERS

a) Which table will the master block be based on? the detail? Why?

Answer: The master block will be based on the `INSTRUCTOR` *table and the detail block will be based on the* `SECTION` *table.*

There is a one-to-many relationship between the `INSTRUCTOR` and `SECTION` tables. `INSTRUCTOR_ID` is the primary key in the `INSTRUCTOR` table and the foreign key in the `SECTION` table.

A master-detail form is used to establish and display a one-to-many (primary key-to-foreign key) relationship between blocks. The primary-key items are always in the master block and the foreign-key items are always in the detail block.

In this Exercise, you built a form that displays one instructor and one or many of the sections that this instructor teaches. Therefore, the master block will be based on the `INSTRUCTOR` table because it is on the "one" side of the relationship. The detail block will be based on the `SECTION` table because it is on the "many" side of the relationship.

Master-detail relationships in forms can also be based on REF columns, which are a type of object column that can be used if your database contains object tables. Object tables are not within the scope of this book, so you will not learn about REF columns here. Refer to the Oracle Forms Reference Manual for more details.

Create the master block and its canvas using the wizards.

b) Did you have to do anything to indicate that this was to be a master block?

Answer: No.

go to contents

When you are creating the first block in a form, the wizard does not know whether or not you intend to create a master-detail form. The wizard will only prompt you with master-detail pages and questions if there is already another block in the form.

When you create the detail block, the wizard will ask if you want to create a relation object that will establish the join condition between the two blocks and manage the coordination of their records.

c) Have you encountered a new wizard page? What will it help you do?

Answer: Yes. It will help to create the relation object between the master and detail blocks.

d) What is this `List of Values` dialog showing you? What happens after you click the `OK` button?

Answer: The `Data Blocks` *dialog that displays a list of base table blocks along with their foreign-key names will appear. After you click OK, the wizard writes a join condition for the two blocks and displays it in the* `Join Condition` *field.*

The relation will be based on the join condition you see here. The relation will also initiate the creation of other objects that will assist in the coordination of the two blocks.

Before the join condition was created, you were presented with the `Data Blocks` dialog box, which displayed a list of blocks that you could choose from for establishing the relationship.

e) Because the wizard has already written the join condition for you, do you need to change the values in the `Detail Item` and `Master Item` list items?

Answer: No.

`Auto-join data blocks` was checked, so the wizard wrote the join condition for you. If you had not checked `Auto-join data blocks`, then you would have had to use the `Detail Item` and `Master Item` poplists to create the join condition yourself.

In the `STUDENT` schema, all of the related tables have explicit primary-foreign key constraints declared at the database level. However, you do not have to have such constraints declared to create a master-detail form. You can build a master-detail form between two blocks where there exists only a logical primary-foreign key relationship. In this case, you would not check `Auto-join data blocks` and create the join condition yourself. You would select the items that make up the logical join and the wizard would use those items to write the relation object's join condition.

f) Which canvas should you choose for the SECTION block? Is it necessary to create a new one?

Answer: You would choose INSTRUCTOR *if you had already renamed it, or* CANVAS# *if it still had the default name. And you wouldn't have to create a new canvas because you want the items in both blocks to appear on the same canvas.*

g) Which of these items already appears in the INSTRUCTOR frame? Is it necessary to display it again in the SECTION frame?

Answer: INSTRUCTOR_ID *already appears in the* INSTRUCTOR *frame. No, it is not necessary to display it again.*

In most master-detail forms, it would be redundant to display the join item, in this case INSTRUCTOR_ID, in both frames.

h) If the layout of the SECTION items is not so pleasing, how can you make quick changes?

Answer: Reenter the Layout Wizard and adjust some of the column widths.

Run the form and issue a query.

i) Have the items in both frames been populated?

Answer: Yes they have.

j) What happens to the records in the SECTION frame when you scroll to the next record in the INSTRUCTOR frame?

Answer: They are coordinated. If you go to the next record in the INSTRUCTOR *frame, you see its corresponding records appear in the* SECTION *frame.*

4.1.2 ANSWERS

Use R_INS_SEC.fmb for all of the Exercises in this section.

a) Which block is listed first in the Object Navigator? What would be the problem if SECTION were listed first?

Answer: It depends on what object or node you had selected in the Object Navigator when you clicked the Create *button. If* SECTION *were listed first, the default navigation would be wrong.*

go to contents

This is a bit of a review from the last Chapter. Remember, when you create an object using the Object Navigator, it positions the new object directly below the object you have selected. The object's position in the Navigator will affect default navigation. So, if you had the `Data Blocks` node selected when you created the `SECTION` block, it would have been placed first, ahead of the `INSTRUCTOR` block. When you ran the form for Question h of Exercise 4.1.1, the cursor would have been placed in the `SECTION_ID` item instead of `INSTRUCTOR_ID`. It would have been best to select the `INSTRUCTOR` block, then click the `Create` button so that the `SECTION` block would have been positioned properly in the Navigator and default navigation would have been smoother.

If your blocks are positioned incorrectly, simply drag `SECTION` below `INSTRUCTOR` to fix the problem. This is a rather small point, but it can be rather annoying if not done correctly.

b) Has a relation object been created? What is it called? Which of the blocks owns the relation?

Answer: Yes, the relation object has been created. It is called `INSTRUCTOR_SECTION`*, and it belongs to the* `INSTRUCTOR` *block.*

As described in Exercise 4.1.1, the relation object is what holds the join condition and manages the coordination between the master and detail blocks.

The master block always contains the relation object. Don't be confused by the fact that the wizards create the relation object during the creation of the detail block.

c) What is the value of the `SECTION.INSTRUCTOR_ID` item's `Copy Value from Item` property?

Answer: The value is `INSTRUCTOR.INSTRUCTOR_ID`.

The `Copy Value from Item` property is set whenever a relation gets created. The Form Builder analyzes the join condition and sets the value of the detail block's foreign-key item's `Copy Value from Item` property to the name of the master block's primary-key item.

In this case, the foreign-key item in the detail block is `SECTION.INSTRUCTOR_ID`. The primary-key item in the master block is `INSTRUCTOR.INSTRUCTOR_ID`. So, the value of the `Copy Value from Item` property of `SECTION.INSTRUCTOR_ID` is set to `INSTRUCTOR.INSTRUCTOR_ID`.

go to contents

Why has the Form Builder done this?

Answer: The `Copy Value from Item` *property is what coordinates the population of the detail block.*

When the Form Builder issues a query in the master block, it needs to return corresponding rows to the detail block. This ensures that when a record is queried in the master block, the corresponding records will be brought back to the detail block.

d) What form-level triggers have been created?

Answer: An `ON-CLEAR-DETAILS` *trigger has been created.*

A description of this trigger will follow Question e.

What block-level triggers have been created?

Answer: The `ON-POPULATE-DETAILS` *and* `ON-CHECK-DELETE-MASTER` *triggers have been created.*

A description of these triggers will follow Question e.

e) Judging from their names, what do you think these triggers do?

Answer: They help the relation coordinate the population of records in the master and detail blocks and manage the deletion of records.

These three triggers, along with three program units (`CHECK_PACKAGE_FAILURE`, `CLEAR_ALL_MASTER_DETAILS`, and `QUERY_MASTER_DETAILS`), are created automatically whenever a relation object is created. This is true whether you use the wizards to create the relation or do it manually.

The `ON-CLEAR-DETAILS` and `ON-POPULATE-DETAILS` triggers work together to ensure that the records in the detail block correspond to those in the master block. This means that if you are looking at Instructor ID 101 in the master block, then in the detail block you should only see the sections Instructor 101 has taught.

The `ON-CLEAR-DETAILS` trigger fires whenever the user goes from one record in the master block to another. This could be done by scrolling to the next record with a button on the toolbar or by issuing an entirely new query. This trigger simply calls the `CLEAR_ALL_MASTER_DETAILS` procedure that flushes the records from any of this master block's detail blocks. So, in simple terms, it clears out the detail block.

The ON-POPULATE-DETAILS trigger checks to ensure that the current master record exists in the database and then calls the QUERY_MASTER_DETAILS procedure that populates the detail block. The way in which QUERY_MASTER_DETAILS populates the detail block will differ depending on the value of the Deferred property of the relation. You will experiment with the Deferred property in Question k. In short, this trigger and procedure work together to put records into the detail block.

The ON-CHECK-DELETE-MASTER trigger fires whenever a user tries to delete a record in the master block. If detail records exist, then deletion of the master record is prevented. The nature of this trigger changes depending on the value of the relation object's Delete Record Behavior property.

Chances are you will never have to change or manipulate any of the triggers and procedures that coordinate the master and detail blocks. However, it is good to have an understanding of how and when they fire.

f) What has happened to the triggers under the INSTRUCTOR block?

Answer: The ON-CHECK-DELETE-MASTER *trigger has been changed to a* PRE-DELETE *trigger.*

g) Judging from the code you see and what you know about the term *cascading* from SQL, what will happen if you run the form and delete a master record?

Answer: All the records in the detail block will be marked for deletion.

As its name suggests, the Delete Record Behavior property controls how to coordinate the deletion of a record in the master block with the corresponding records in the detail block. The effects of the three choices for Delete Record Behavior are summarized as follows:

1) Cascading will delete the master record and all corresponding detail records.
2) Isolated will delete the master record, but does not delete any of its detail records. It is important to note that if the constraints in the database do not allow the deletion of a master record when detail records exist, then it will prevent Forms from doing so as well.
3) Non-Isolated (default) will not delete the master record if detail records exist.

Run the form. Navigate to SECTION_ID in the SECTION frame and issue a query.

h) Were the SECTION records returned to the form? How about the INSTRUCTOR records?

Answer: Yes, both sets of records were returned to the form.

i) Were the SECTION records returned to the form this time? What does the Prevent Masterless Operations property do?

Answer: No, the SECTION records were not returned this time. The Prevent Masterless Operations property does not allow you to perform operations on the detail block if a master record is not in the master block.

Specifically, it will not allow the user to query or insert records into the detail block unless there is a record present in the master block.

j) Were the items in the SECTION frame populated?

Answer: No they were not.

k) What then does the Deferred property defer?

Answer: It defers or holds off on executing the query for the detail block.

In the previous questions, the master block was queried and populated immediately after the detail block was queried and populated. With the Deferred property set to Yes, the form waits for more actions from the user before querying and populating the detail block.

l) What happened after you issued the second query?

Answer: The detail records were returned to the form.

m) How could this be useful?

Answer: This can be useful if the user wants to query by example for the detail block, or if the user wants to insert detail records without querying existing records.

■ FOR EXAMPLE:

What if a user wanted to see the sections that Instructor ID 101 taught that had a capacity of 15? By setting Deferred to Yes and Auto Query to No, the user can retrieve the Instructor 101 record first, then retrieve all of the sections that had a capacity of 15. Take the following steps to test this:

1) Check that the INSTRUCTOR_SECTION relation's Deferred property is still set to Yes and Auto Query is set to No.
   Run the form.
2) Click Enter Query button on the toolbar and enter 101 in the INSTRUCTOR_ID item.

go to contents

3) Click `Execute Query` on the toolbar. Note that only the master records have been returned.
4) Navigate to any field in the `SECTION` frame and click the `Enter Query` button on the toolbar. Enter `15` in the `CAPACITY` item.
5) Click the `Execute Query` button on the toolbar again. Note that only the records with a `CAPACITY` of 15 have been returned to the detail block.

n) Will you have to issue a second query this time? If not, what should you do to populate the items in the `SECTION` frame?

Answer: No, simply navigate to the `SECTION` *frame to populate the items within.*

In this case, the query and subsequent population of the detail records is still deferred. However, as soon as the user navigates to an item in the detail block, the query is issued and it gets populated. The user does not have to enter more query criteria or explicitly click the `Execute_Query` button.

o) How could this be useful?

Answer: This could be useful if the user wants to view the master records before deciding whether or not to view the details.

■ FOR EXAMPLE:

Assume that the SECTION block will cause a long-running query. The user may not always want to wait for the query to complete. Here, he can query an instructor record and view it to decide whether or not it is necessary to also see the section records. For the user, this operation is fast and flexible. Fast because the return of the instructor record is not hindered by the slowness of the query for the section records. Flexible because he has a choice of whether or not to view the section records at all.

LAB 4.1 SELF-REVIEW QUESTIONS

In order to test your progress, you should be able to answer the following questions:

1) The blocks in a master-detail form are which of the following?

a) ____ Two non-base-table blocks linked by a relation
b) ____ Two base-table blocks linked by a relation
c) ____ Neither the master nor the detail block contains non-base-table items
d) ____ The blocks are based on the same database object and coordinated by triggers

go to contents

2) What can't you do with master-detail blocks?
a) ____ Create them with a wizard
b) ____ Create them manually
c) ____ Edit their relation object's properties
d) ____ Display their relation object on a different canvas

3) What database objects are relation objects analogous to?
a) ____ Many-to-one relationships
b) ____ Parent-child tables
c) ____ Primary-foreign key constraints
d) ____ Database links

4) Which of the following is not true of the data items in a detail block?
a) ____ They are base-table items
b) ____ They are copied from the master block
c) ____ They can appear on the same canvas as the items in the master block
d) ____ It is common to give them a Tabular style layout

5) Which of the following is true about the relation object?
a) ____ It is owned by the detail block
b) ____ It is owned by the master block
c) ____ One of its properties can affect the way records are deleted
d) ____ a & c
e) ____ b & c

6) What will happen when the `Deferred` property of a relation is set to `No`?
a) ____ The master block is populated after the detail block
b) ____ The detail block is populated along with the master
c) ____ The detail block is disabled
d) ____ The master and detail blocks are put into Enter Query mode.

7) What will happen when the `Prevent Masterless` property is set to `Yes`?
a) ____ The detail records will be deleted along with corresponding master records
b) ____ The detail block's operations are put into Normal mode
c) ____ The detail block cannot be queried or inserted into unless a master record is present
d) ____ The master operations are prevented from coordinating the detail processing

Quiz answers appear in Appendix A, Section 4.1.

go to contents

CHAPTER 4

TEST YOUR THINKING

1) Build a master-detail form between the ZIPCODE and STUDENT tables. On the Data Block Wizard's master-detail page, uncheck Auto-join data blocks and create the relation yourself. The items in the ZIPCODE block should be laid out in Form style. The items in the STUDENT block should be laid out in Tabular style. Display five records for the STUDENT block.

2) Build a master-detail form between the COURSE and SECTION tables. The items in the COURSE block should be laid out in Form style. The items in the SECTION block should be laid out in Tabular style. Display five records for the SECTION block. Configure the relation so that the user can enter query criteria for both blocks.

3) Build a master-detail form between the STUDENT and ENROLLMENT tables. The items in the STUDENT block should be laid out in Form style. The items in the ENROLLMENT block should be laid out in Tabular style. Display five records for the ENROLLMENT block. Configure the relation so that the detail records are returned as soon as the master block is queried.

4) Is it possible to build a master-detail-detail form between the INSTRUCTOR, SECTION, and ENROLLMENT tables? Try to build it. Choose the proper layout styles for the items in each block. Don't be overly concerned if the canvas is very long. Experiment with the Deferred and Auto Query properties of each relation.

go to contents