PROGRESS OPENEDGE 10

OpenEdge Reporting:
Query/Results for Windows
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>Preface–1</td>
</tr>
<tr>
<td>1. Results Overview</td>
<td>1–1</td>
</tr>
<tr>
<td>Results overview</td>
<td>1–2</td>
</tr>
<tr>
<td>Database basics</td>
<td>1–2</td>
</tr>
<tr>
<td>The structure of a table</td>
<td>1–4</td>
</tr>
<tr>
<td>Indexes</td>
<td>1–6</td>
</tr>
<tr>
<td>Table relationships</td>
<td>1–7</td>
</tr>
<tr>
<td>Types of relationships</td>
<td>1–8</td>
</tr>
<tr>
<td>2. Results Basics</td>
<td>2–1</td>
</tr>
<tr>
<td>Before you begin</td>
<td>2–2</td>
</tr>
<tr>
<td>Putting your Results data to work for you</td>
<td>2–2</td>
</tr>
<tr>
<td>Starting Results</td>
<td>2–3</td>
</tr>
<tr>
<td>Starting Results from a program group</td>
<td>2–3</td>
</tr>
<tr>
<td>Starting Results from the OpenEdge Desktop</td>
<td>2–4</td>
</tr>
<tr>
<td>Accessing the Results main window</td>
<td>2–6</td>
</tr>
<tr>
<td>Exiting Results</td>
<td>2–7</td>
</tr>
<tr>
<td>3. Query Basics</td>
<td>3–1</td>
</tr>
<tr>
<td>Before you begin</td>
<td>3–2</td>
</tr>
<tr>
<td>Creating a query</td>
<td>3–2</td>
</tr>
<tr>
<td>Query creation overview</td>
<td>3–2</td>
</tr>
<tr>
<td>Details to create a query</td>
<td>3–4</td>
</tr>
<tr>
<td>Record displays in multi-table queries</td>
<td>3–8</td>
</tr>
<tr>
<td>Saving a query</td>
<td>3–11</td>
</tr>
<tr>
<td>Closing a query</td>
<td>3–12</td>
</tr>
<tr>
<td>Opening an existing query</td>
<td>3–12</td>
</tr>
<tr>
<td>Getting query information</td>
<td>3–13</td>
</tr>
</tbody>
</table>
Printing queries ......................................................... 3–13
Using print preview ............................................... 3–14
Deleting a query ...................................................... 3–16

4. Using Query Views .................................................. 4–1
   Query views overview ........................................... 4–2
   Browse view ....................................................... 4–2
      Navigating through records .................................. 4–3
      Using frames ................................................... 4–4
      Modifying frame properties .................................. 4–6
   Form view .......................................................... 4–7
      Navigating through records .................................. 4–8
      Modifying records .............................................. 4–8
      Rearranging fields ............................................. 4–12
      Using frames ................................................... 4–13
      Modifying frame properties .................................. 4–15
   Report view ....................................................... 4–16
      Adding cover and final pages, headers, and footers ...... 4–17
      Selecting page sizes ............................................ 4–21
      Adjusting report dimensions .................................. 4–22
      Printing reports ................................................ 4–23
   Label view ........................................................ 4–24
      Selecting label types ........................................... 4–27
      Modifying label formats ....................................... 4–28
      Printing labels ................................................ 4–29
   Export view ....................................................... 4–30
      Generating an ASCII file ...................................... 4–31
      Modifying export types ....................................... 4–33

5. Changing the Query Selection Criteria ........................... 5–1
   Adding and removing tables from a query ..................... 5–2
   Changing table relationships in multi-table queries ......... 5–4
      Types of table relationships .................................. 5–4
   Sorting records ................................................. 5–7
      Sort order and break groups .................................. 5–9
      Specifying the sort order ..................................... 5–10
   Creating filters .................................................. 5–12
      Data Selection ................................................ 5–12
      Query by example ............................................. 5–13
      Using Data Selection to refine a query ...................... 5–13
      Using Query by Example to refine a query ................. 5–21
      Removing a WHERE clause created using Query by Example .. 5–22
   Using the Data Governor ....................................... 5–23
Figures

Figure 1–1: Paper filing system compared to a database ............................... 1–3
Figure 1–2: Relationship between a database and its tables ...................... 1–4
Figure 1–3: Fields and records in the Customer table .............................. 1–5
Figure 1–4: Structure of a field ............................................................. 1–5
Figure 1–5: Simple and compound indexes ......................................... 1–6
Figure 1–6: Relationship between Customer and Order tables ................. 1–7
Figure 1–7: One-to-one relationships .................................................. 1–8
Figure 1–8: One-to-many relationships ............................................... 1–9
Figure 1–9: Many-to-many relationships ............................................ 1–9
Figure 2–1: Results main window ......................................................... 2–6
Figure 3–1: Relationship type field default behavior ............................ 3–9
Figure 4–1: Browse view example ....................................................... 4–2
Figure 4–2: Browse view displaying one query frame ............................ 4–4
Figure 4–3: Browse view displaying two frames .................................. 4–5
Figure 4–4: Form view example .......................................................... 4–7
Figure 4–5: Query with one frame ...................................................... 4–13
Figure 4–6: Query with two frames .................................................... 4–14
Figure 4–7: Sample on-screen report format ...................................... 4–16
Figure 4–8: Data displayed using Print Preview ................................. 4–17
Figure 4–9: Label view icon ............................................................... 4–24
Figure 4–10: Export view icon ............................................................ 4–30
Figure 4–11: Export query ................................................................. 4–30
Figure 4–12: ASCII file generated by an Export query ......................... 4–31
Figure 5–1: Example of a partial relationship ..................................... 5–5
Figure 5–2: Query displaying ordered items by customer ...................... 5–7
Figure 5–3: Query displaying ordered items by order number and date .... 5–8
Figure 5–4: Changing sort order example .......................................... 5–9
Figure 6–1: Default query display for records from multiple tables ......... 6–3
Figure 6–2: Master and Detail sections in a query view ......................... 6–4
Figure 6–3: Calculated fields example .............................................. 6–6
Figure 6–4: Percent of Total calculation example ............................... 6–9
Figure 6–5: Total-Paid field example ............................................... 6–10
Figure 6–6: Counter field example ..................................................... 6–12
Figure 6–7: Stacked Array field example .......................................... 6–13
Figure 6–8: Adding a Lookup field example ...................................... 6–14
Figure 6–9: Expression example ......................................................... 6–23
Figure 6–10: Round Amount field example ....................................... 6–26
Figure 6–11: Add Field - Date Function dialog box ............................ 6–27
Figure 6–12: Subtotals - example one ............................................... 6–39
Figure 6–13: Subtotals - example two ............................................... 6–40
Figure 6–14: Sort order by Cust-num field - example one .................... 6–42
Figure 6–15: Sort order by Totals Only Summary - example two .......... 6–43
Figure 7–1: Format dialog box ............................................................ 7–5
Figure 7–2: Format dialog box ................................. 7–9
## Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2–1</td>
<td>Query views</td>
<td>2–2</td>
</tr>
<tr>
<td>3–1</td>
<td>Query views</td>
<td>3–3</td>
</tr>
<tr>
<td>3–2</td>
<td>Partial relationship sample query data</td>
<td>3–9</td>
</tr>
<tr>
<td>3–3</td>
<td>Complete relationship sample query data</td>
<td>3–10</td>
</tr>
<tr>
<td>3–4</td>
<td>Print Preview navigation buttons</td>
<td>3–15</td>
</tr>
<tr>
<td>4–1</td>
<td>Browse view keys</td>
<td>4–3</td>
</tr>
<tr>
<td>4–2</td>
<td>Navigation buttons</td>
<td>4–8</td>
</tr>
<tr>
<td>4–3</td>
<td>Rearranging fields in the Form view</td>
<td>4–12</td>
</tr>
<tr>
<td>4–4</td>
<td>Report variable functions</td>
<td>4–19</td>
</tr>
<tr>
<td>4–5</td>
<td>Custom Page attributes</td>
<td>4–22</td>
</tr>
<tr>
<td>4–6</td>
<td>Custom label attributes</td>
<td>4–28</td>
</tr>
<tr>
<td>4–7</td>
<td>Custom export attributes</td>
<td>4–34</td>
</tr>
<tr>
<td>4–8</td>
<td>Code formats</td>
<td>4–36</td>
</tr>
<tr>
<td>6–1</td>
<td>Results data analysis techniques</td>
<td>6–2</td>
</tr>
<tr>
<td>6–2</td>
<td>Calculated fields</td>
<td>6–7</td>
</tr>
<tr>
<td>6–3</td>
<td>String functions</td>
<td>6–20</td>
</tr>
<tr>
<td>6–4</td>
<td>Numeric functions</td>
<td>6–24</td>
</tr>
<tr>
<td>6–5</td>
<td>Date functions</td>
<td>6–27</td>
</tr>
<tr>
<td>6–6</td>
<td>Logical functions</td>
<td>6–31</td>
</tr>
<tr>
<td>7–1</td>
<td>Field properties</td>
<td>7–3</td>
</tr>
<tr>
<td>7–2</td>
<td>Character field properties</td>
<td>7–6</td>
</tr>
<tr>
<td>7–3</td>
<td>Date field properties</td>
<td>7–7</td>
</tr>
<tr>
<td>7–4</td>
<td>Decimal and integer field properties</td>
<td>7–8</td>
</tr>
<tr>
<td>A–1</td>
<td>Query menu</td>
<td>A–2</td>
</tr>
<tr>
<td>A–2</td>
<td>Table menu</td>
<td>A–3</td>
</tr>
<tr>
<td>A–3</td>
<td>Field menu</td>
<td>A–3</td>
</tr>
<tr>
<td>A–4</td>
<td>Data menu</td>
<td>A–4</td>
</tr>
<tr>
<td>A–5</td>
<td>Options menu</td>
<td>A–4</td>
</tr>
<tr>
<td>A–6</td>
<td>View menu</td>
<td>A–5</td>
</tr>
<tr>
<td>A–7</td>
<td>Help menu</td>
<td>A–6</td>
</tr>
<tr>
<td>B–1</td>
<td>Operators</td>
<td>B–2</td>
</tr>
</tbody>
</table>
Preface

This Preface contains the following sections:

- Purpose
- Audience
- Organization
- Using this manual
- Typographical conventions
- OpenEdge messages
- Third party acknowledgements
Purpose

This guide introduces you to Progress Results®. Query/Results, more commonly referenced throughout this guide as Results, is an interactive, menu-driven tool that lets you query, report, and maintain information stored in Progress-supported databases.

Audience

This guide is an end-user manual that explains how to use the default Results application. If you purchased Results through a third-party application vendor, the vendor might have modified Results. These modifications might include adding or removing features, or even incorporating Results into another application. If you have a modified Results application, your vendor might modify the help system to reflect these changes or provide you with additional documentation.

Organization

Chapter 1, “Results Overview”
Briefly describes the benefits of using Results, then explains basic database concepts.

Chapter 2, “Results Basics”
Explains how to start and stop Results and use the online help system.

Chapter 3, “Query Basics”
Explains how to use Results to query information in a database, then print and save your queries.

Chapter 4, “Using Query Views”
Explains how to use the browse, form report, label, and export query views to process and analyze data.

Chapter 5, “Changing the Query Selection Criteria”
Explains how to change or refine the set of information a query retrieves from a database.
Chapter 6, “Analyzing Data”

Explains how to use calculated field, totals, and subtotals to analyze data.

Chapter 7, “Modifying Display Fields”

Explains how to add and remove display fields and change display field properties.

Appendix A, “Results Menus”

Describes the Results menus.

Appendix B, “Comparison Operators”

Describes the comparison operators you can use in a WHERE clause.

Using this manual

You need not read the entire manual in detail. A good strategy might be to skim through once quickly, then go back and read for detail only those chapters relevant to your situation and needs.

For the latest documentation updates see the OpenEdge Product Documentation category on PSDN http://www.psdn.com/library/kbcategory.jspa?categoryID=129.

Typographical conventions

This manual uses the following typographical conventions:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bold</strong></td>
<td>Bold typeface indicates commands or characters the user types, or the names of user interface elements.</td>
</tr>
<tr>
<td><em>Italic</em></td>
<td>Italic typeface indicates the title of a document, provides emphasis, or signifies new terms.</td>
</tr>
<tr>
<td>SMALL, BOLD CAPITAL LETTERS</td>
<td>Small, bold capital letters indicate OpenEdge™ key functions and generic keyboard keys; for example, GET and CTRL.</td>
</tr>
</tbody>
</table>
OpenEdge Reporting: Query/Results for Windows

OpenEdge messages

OpenEdge displays several types of messages to inform you of routine and unusual occurrences:

- **Execution messages** inform you of errors encountered while OpenEdge is running a procedure; for example, if OpenEdge cannot find a record with a specified index field value.

- **Compile messages** inform you of errors found while OpenEdge is reading and analyzing a procedure before running it; for example, if a procedure references a table name that is not defined in the database.

- **Startup messages** inform you of unusual conditions detected while OpenEdge is getting ready to execute; for example, if you entered an invalid startup parameter.
After displaying a message, OpenEdge proceeds in one of several ways:

- Continues execution, subject to the error-processing actions that you specify or that are assumed as part of the procedure. This is the most common action taken after execution messages.

- Returns to the Progress Procedure Editor, so you can correct an error in a procedure. This is the usual action taken after compiler messages.

- Halts processing of a procedure and returns immediately to the Progress Procedure Editor. This does not happen often.

- Terminates the current session.

OpenEdge messages end with a message number in parentheses. In this example, the message number is 200:

```
** Unknown table name table. (200)
```

If you encounter an error that terminates OpenEdge, note the message number before restarting.

** Obtaining more information about OpenEdge messages**

On Windows platforms, use OpenEdge online help to obtain more information about OpenEdge messages. Many OpenEdge tools include the following Help menu options to provide information about messages:

- Choose Help → Recent Messages to display detailed descriptions of the most recent OpenEdge message and all other messages returned in the current session.

- Choose Help → Messages and then enter the message number to display a description of a specific OpenEdge message.

- In the Progress Procedure Editor, press the help key (F2 or CTRL-W).

** Third party acknowledgements**

OpenEdge includes Imaging Technology copyrighted by Snowbound Software 1993-2003.

[www.snowbound.com](http://www.snowbound.com)
OpenEdge includes software developed by the Apache Software Foundation (
http://www.apache.org/). Copyright © 1999 The Apache Software Foundation. All rights reserved (Xerces C++ Parser (XML) and Xerces2 Java Parser (XML)); Copyright © 1999-2002 The Apache Software Foundation. All rights reserved (Xerces Parser (XML)); and Copyright © 2000-2003 The Apache Software Foundation. All rights reserved (Ant). The names “Apache,” “Xerces,” “ANT,” and “Apache Software Foundation” must not be used to endorse or promote products derived from this software without prior written permission. Products derived from this software may not be called “Apache” , nor may “Apache” appear in their name, without prior written permission of the Apache Software Foundation. For written permission, please contact apache@apache.org. Software distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product.


OpenEdge includes software developed by Vermont Creative Software. Copyright © 1988-1991 by Vermont Creative Software.

OpenEdge includes code licensed from RSA Security, Inc. Some portions licensed from IBM are available at http://oss.software.ibm.com/icu4j/.

OpenEdge includes the UnixWare platform of Perl Runtime authored by Kiem-Phong Vo and David Korn. Copyright © 1991, 1996 by AT&T Labs. Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software. THIS SOFTWARE IS BEING PROVIDED “AS IS”, WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHORS NOR AT&T LABS MAKE ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

OpenEdge includes the RSA Data Security, Inc. MD5 Message-Digest Algorithm. Copyright ©1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.

OpenEdge includes software developed by the World Wide Web Consortium. Copyright © 1994-2002 World Wide Web Consortium, (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). All rights reserved. This work is distributed under the W3C® Software License [http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231] in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Preface–6
OpenEdge includes Sonic software, which includes software developed by Apache Software Foundation (http://www.apache.org/). Copyright © 1999-2000 The Apache Software Foundation. All rights reserved. The names “Ant”, “Axis”, “Xalan”, “FOP”, “The Jakarta Project”, “Tomcat”, “Xerces” and/or “Apache Software Foundation” must not be used to endorse or promote products derived from the Product without prior written permission. Any product derived from the Product may not be called “Apache”, nor may “Apache” appear in their name, without prior written permission. For written permission, please contact apache@apache.org.

OpenEdge includes Sonic software, which includes the JMX Technology from Sun Microsystems, Inc. Use and Distribution is subject to the Sun Community Source License available at http://sun.com/software/communitysource.

OpenEdge includes Sonic software, which includes software developed by the ModelObjects Group (http://www.modelobjects.com). Copyright © 2000-2001 ModelObjects Group. All rights reserved. The name “ModelObjects” must not be used to endorse or promote products derived from this software without prior written permission. Products derived from this software may not be called “ModelObjects”, nor may “ModelObjects” appear in their name, without prior written permission. For written permission, please contact djacobs@modelobjects.com.

OpenEdge includes Sonic software, which includes files that are subject to the Netscape Public License Version 1.1 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/NPL/. Software distributed under the License is distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License. The Original Code is Mozilla Communicator client code, released March 31, 1998. The Initial Developer of the Original Code is Netscape Communications Corporation. Portions created by Netscape are Copyright © 1998-1999 Netscape Communications Corporation. All Rights Reserved.

OpenEdge includes software Copyright © 2003-2006, Terence Parr All rights reserved. Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. Software distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License agreement that accompanies the product.

OpenEdge includes ICU software 1.8 and later - Copyright © 1995-2003 International Business Machines Corporation and others All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit
persons to whom the Software is furnished to do so, provided that the above copyright notice(s)
and this permission notice appear in all copies of the Software and that both the above copyright
notice(s) and this permission notice appear in supporting documentation.

OpenEdge includes software developed by the OpenSSL Project for use in the OpenSSL
Toolkit (http://www.openssl.org/). Copyright © 1998-2007 The OpenSSL Project. All
devices reserved. This product includes cryptographic software written by Eric Young
(eay@cryptsoft.com). This product includes software written by Tim Hudson
(tjh@cryptsoft.com). Copyright © 1995-1998 Eric Young (eay@cryptsoft.com) All
rights reserved. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse
or promote products derived from this software without prior written permission. For written
permission, please contact openssl-core@openssl.org. Products derived from this software may
not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written
permission of the OpenSSL Project. Software distributed on an "AS IS" basis, WITHOUT
WARRANTY OF ANY KIND, either express or implied. See the License for the specific
language governing rights and limitations under the License agreement that accompanies the
product.

OpenEdge includes Sonic software which includes a version of the Saxon XSLT and XQuery
Processor from Saxonica Limited that has been modified by Progress Software Corporation.
The contents of the Saxon source code and the modified source code file (Configuration.java)
are subject to the Mozilla Public License Version 1.0 (the "License"); you may not use these
files except in compliance with the License. You may obtain a copy of the License at
http://www.mozilla.org/MPL/ and a copy of the license (MPL-1.0.html) can also be found
in the installation directory, in the Docs7.5/third_party_licenses folder, along with a copy of the
modified code (Configuration.java); and a description of the modifications can be found in the
Progress SonicMQ and Progress Sonic ESB v7.5 README files. Software distributed under
the License is distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND,
either express or implied. See the License for the specific language governing rights and
limitations under the License. The Original Code is The SAXON XSLT and XQuery Processor
from Saxonica Limited. The Initial Developer of the Original Code is Michael Kay
http://www.saxonica.com/products.html). Portions created by Michael Kay are Copyright
© 2001-2005. All rights reserved. Portions created by Progress Software Corporation are
Copyright © 2007. All rights reserved.

OpenEdge includes software developed by IBM. Copyright © 1995-2003 International
Business Machines Corporation and others. All rights reserved. Permission is hereby granted,
free of charge, to any person obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software,
and to permit persons to whom the Software is furnished to do so, provided that the above
copyright notice(s) and this permission notice appear in all copies of the Software and that both
the above copyright notice(s) and this permission notice appear in supporting documentation.
Software distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either
express or implied. See the License for the specific language governing rights and limitations
under the License agreement that accompanies the product. Except as contained in this notice,
the name of a copyright holder shall not be used in advertising or otherwise to promote the sale,
use or other dealings in this Software without prior written authorization of the copyright holder.
OpenEdge includes Sonic software, which includes software developed by ExoLab Project
(http://www.exolab.org/). Copyright © 2000 Intalio Inc. All rights reserved. The names
“Castor” and/or “ExoLab” must not be used to endorse or promote products derived from the
Products without prior written permission. For written permission, please contact
info@exolab.org. Exolab, Castor and Intalio are trademarks of Intalio Inc.

OpenEdge includes Sonic software, which includes software Copyright © 1999 CERN -
European Organization for Nuclear Research. Permission to use, copy, modify, distribute and
sell this software and its documentation for any purpose is hereby granted without fee, provided
that the above copyright notice appear in all copies and that both that copyright notice and this
permission notice appear in supporting documentation. CERN makes no representations about
the suitability of this software for any purpose. It is provided "as is" without expressed or
implied warranty.

OpenEdge includes Sonic software, which includes software developed by the University
Corporation for Advanced Internet Development http://www.ucaid.edu Internet2 Project.
Copyright © 2002 University Corporation for Advanced Internet Development, Inc. All rights
reserved. Neither the name of OpenSAML nor the names of its contributors, nor Internet2, nor
the University Corporation for Advanced Internet Development, Inc., nor UCAID may be used
to endorse or promote products derived from this software and products derived from this
software may not be called OpenSAML, Internet2, UCAID, or the University Corporation for
Advanced Internet Development, nor may OpenSAML appear in their name without prior
written permission of the University Corporation for Advanced Internet Development. For
written permission, please contact opensaml@opensaml.org.

OpenEdge includes DataDirect products for the Microsoft SQL Server database which contains
a licensed implementation of the Microsoft TDS Protocol.

OpenEdge includes Sonic software, which includes code licensed from Mort Bay Consulting
Pty. Ltd. The Jetty Package is Copyright © 1998 Mort Bay Consulting Pty. Ltd. (Australia) and
others.
OpenEdge Reporting: Query/Results for Windows

Preface–10
Results Overview

This chapter provides a brief overview of Results, and introduces you to the basic database concepts that are essential to understanding Results. This chapter presents:

- Results overview
- Database basics
- Indexes
- Table relationships
Results overview

In today’s fast-paced business environment, the amount of information you need to store, organize, and manage is growing daily. Progress Results works in conjunction with your Progress-supported database to help you sort, maintain, and analyze information quickly and efficiently. Results helps you deal effectively with information overload and puts your information to work for you. You can use Results to update information stored in your database, create reports, generate mailing labels, or even export information from your database to common software applications, including spreadsheet and word processing packages.

Note: If you are already familiar with database concepts, you consider skimming this chapter before proceeding to Chapter 2, “Results Basics.”

Database basics

A database is a collection of data organized in logically related tables that can be accessed or retrieved. A table is a collection of logically related records organized into rows and columns. One way to understand how a database works is to think of it as an electronic filing system because it organizes data in the same manner as a paper filing system.

When setting up a paper filing system, you keep all related information about a subject in a filing cabinet. The database equivalent to a filing cabinet is a table. Both a filing cabinet and a table allow you to group related information in one location. For example, in a paper filing system you might store information about customers in a filing cabinet called Customer. In a database, you store all this information in a Customer table.

How do you organize related information within a filing cabinet? By placing it in folders. In a database, a record functions as a folder. A record is a complete set of related information or data. For example, within the Customer filing cabinet, you would find folders for each customer. In the Customer table, you would find the records for each customer.
Figure 1–1 compares a paper filing system to a database filing system.

For example, if you are running a business called All Around Sports, you might have a sales and inventory database that contains several tables for storing information about your customers, the products they have ordered, and your overall product inventory.
Figure 1–2 shows the database for All Around Sports.

![Database Tables Diagram](image)

**Figure 1–2: Relationship between a database and its tables**

**The structure of a table**

Another way you can think of a database table is to view it as a paper table that you use to organize and present data. A paper table is made up of rows and columns, and the intersection of each row and column contains one specific piece of data.

A database table is similar to a paper table, but the terminology differs a little. A row in a database table is called a *record* and a column is called a *field*. 
Figure 1–3 shows a section of the Customer table.

This section of the Customer table contains four records, and each record contains a complete set of information about one customer. In this example, each record contains a customer number, name, and street. These items are called fields.

A field contains two elements: the name of the field and the data stored in the field. Note that the name of the field is common to all the fields in the same column. However, the data contained within a field varies from record to record. For example, the data in the field Cust-Num ranges from 101 to 104. Figure 1–4 shows the structure of a field.

Figure 1–4 shows a section of the Customer table.
Indexes

The last component of a database is called an index. When you open the drawer of a filing 
cabinet and thumb through the tabs on the folders, you are using a kind of index. What is on the 
paper tab represents the information found in the folder. For example, you expect the tabs in the 
Customer filing cabinet to contain the name of the customer whose information is in that folder. 
Similarly, an electronic index is a component defined for a specific database table that serves as 
the basis for faster searching, sorting, or otherwise processing the records in that table.

In a database, an index is a list that contains a value for each record in the table. When you create 
an index, you choose the fields used to derive the index value for each record. For example, if 
you choose the Name field as an index, Progress creates an index for the Customer table that 
consists of a list of customer names, just like the list of tabs in the paper filing system.

A simple index is based on the value of one field, while a compound index is based on two or 
more fields. Figure 1–5 shows an example of a simple and a compound index.

![Simple and compound indexes](image-url)

Figure 1–5: Simple and compound indexes
Table relationships

The most powerful advantage that a database has over a paper filing system is that it can process data based on the relationship between database tables. The relationship is based on shared data. As shown in Figure 1–6, the Customer record contains the customer number, name, and address fields. The Order record contains the customer number, order number, delivery date, etc. The Order table is related to the Customer table because they contain a common field: Customer No.

![Relationship between Customer and Order tables](image)

Figure 1–6: Relationship between Customer and Order tables

With relationships between the Customer and Order tables established, you can perform requests for information, called queries. For example, you can query the database to find:

- All the orders for a given customer.
- All the customers whose orders are promised before November 10, 2004.
- The number of outstanding orders for each sales region. (The number of unshipped orders for each region.)

As you can see by the relationship between the Customer and Order tables, relationships between tables are based on standard business practices. These relationships are determined when the database is designed.
Types of relationships

You can create three types of relationships among tables: one-to-one, one-to-many, many-to-many. The following sections describe these relationships.

One-to-one relationship

A one-to-one relationship occurs when one record can relate to one and only one instance of a record in another table. For example, a business might decide to assign one office to exactly one employee. Thus, one employee can have only one office. The same business might also decide that a department can have only one manager. Thus, one manager can manage only one department. Figure 1–7 illustrates this relationship.

Figure 1–7: One-to-one relationships
One-to-many relationship

A one-to-many relationship occurs when one record can relate to multiple records in another table. For example, one customer can place many orders, one student can take many courses, and a sales representative can have many customer accounts. Figure 1–8 illustrates this relationship.

![Figure 1–8: One-to-many relationships](image)

Many-to-many relationship

A many-to-many relationship occurs when multiple records can relate to multiple records in another table. Likewise, those related records have many records in the first table. For example, an order can contain many items, and an item can appear in many different orders; an employee can work on many projects, and a project can have many employees working on it. Figure 1–9 illustrates this relationship.

![Figure 1–9: Many-to-many relationships](image)
This chapter explains the basics of what Results is and how it helps you maintain and analyze information stored in your Progress-supported databases. Specifically, this chapter presents:

- Before you begin
- Putting your Results data to work for you
- Starting Results
- Accessing the Results main window
- Exiting Results
Before you begin

Your organization can purchase Results directly through Progress Software Corporation (PSC) or through a third-party application vendor. If you purchase Results through a third-party vendor, your vendor might modify Results. These modifications might include adding and removing features or even incorporating Results into another application.

This guide describes how to use the Results application that PSC ships. If you have a modified Results application, your vendor might modify or provide you with additional documentation.

Putting your Results data to work for you

Results is a software tool that allows you to create queries to search for specific data in one or more databases. Once the query accesses the data, you can use a variety of query views to display and update the data, generate reports and mailing labels, or export the data to other applications. Table 2–1 lists the available query views.

<table>
<thead>
<tr>
<th>View</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browse</td>
<td>Lists records online in a tabular form similar to a spreadsheet. You can scroll forward and backward through the data.</td>
</tr>
<tr>
<td>Report</td>
<td>Creates multi-line and master-detail reports for printing. Provides calculated values and totals for thorough data analysis. You can control the formatting of the headers, footers, and page breaks.</td>
</tr>
<tr>
<td>Form</td>
<td>Allows you to view, update, add, and delete individual records in the database.</td>
</tr>
<tr>
<td>Label</td>
<td>Formats and prints records in a variety of label formats, including mailing and inventory labels.</td>
</tr>
<tr>
<td>Export</td>
<td>Formats records to export to other applications.</td>
</tr>
</tbody>
</table>
Results is flexible. Instead of presenting you with a predetermined set of queries, Results lets you define and save queries that suit your particular needs. You can query all the records in a particular table or you can query any number of fields in one or more related tables. If, for example, your database contains Customer and Order tables and these tables are related, you can instruct Results to find all the orders for each customer in the database.

Results provides you with a filter to refine the contents of your queries. For example, you can use a filter to refine the previous query to find all the orders for each customer in Massachusetts.

Starting Results

How you start Results depends on whether your application was modified by a third-party vendor. If your application was

- Not modified, you will most likely start Results from the ADE Desktop.
- Modified, you will most likely start it from within a program group or from within another application.

The following sections describe how to start Results from a program group and from the ADE Desktop. If your Results application is contained within another application or you are not sure how to start Results, see your system administrator.

Starting Results from a program group

The following sections describe how to start Results from a program group and from the ADE Desktop. If your Results application is contained within another application or you are not sure how to start Results, see your system administrator.
To start Results from the OpenEdge program group:

1. Choose the Results icon from the OpenEdge Desktop.

2. If you are starting Results for the first time, a dialog box prompts you to create the .qc7 file. Choose Yes.

   The Progress Results window appears. If your Results application was modified, you might see additional startup screens before the Results main window appears.

**Note:** The query directory (.qc7) file contains descriptions to the filenames that Results generates.

---

Starting Results from the OpenEdge Desktop

The following section describes how to start Results from the OpenEdge Desktop.

**Note:** Depending on how the Results application is configured, the appearance of the OpenEdge Desktop and Results icons might differ from their appearance in these instructions.

To start Results from the OpenEdge Desktop:

1. Double click the OpenEdge icon on your Windows desktop.

   The OpenEdge Desktop appears:

   ![OpenEdge Desktop](image)

   **Note:** The specific OpenEdge product icons that display can vary, depending on the OpenEdge products installed at your site. The OpenEdge Desktop that appears in this step only displays some of the available OpenEdge products that can potentially appear on the OpenEdge Desktop.
2. Single click the Results icon on the OpenEdge Desktop.

3. If you're starting Results for the first time, a dialog box prompts you to create the .qc7 file. Choose Yes.

4. If you are not connected to a database, the following dialog box appears:

5. Choose OK to connect to a database. The Connect Database dialog box appears:

6. Specify the connection information, then choose OK.

   After you connect to a database, the OpenEdge Results window appears. Note that if your Results application was modified, you might see additional startup screens before the Results window appears.

   **Note:** The query directory (.qc7) file contains descriptions to the filenames that Results generates.
Accessing the Results main window

Use the Results main window to create and open queries. Figure 2–1 shows the Results main window with the components of the window identified.

By default, the OpenEdge Results window contains the following items:

- **Menu bar** — Lets you access the menu options you use to run Results. See Appendix A, “Results Menus,” for a description of each menu.
- **Tool bar** — Provides quick access to the most commonly used Results menu options.
- **Status bar** — Contains informational messages about the current status of Results.
Exiting Results

The following section describes how to exit Results.

To exit Results:

1. Choose Query → Exit.
   
   If you created or modified the query and did not save it, a dialog box prompts you to save your query.

2. Choose Yes to save the query or No to close the query without saving it.
   
   If you are saving the query for the first time, the Save As dialog box appears.

3. Type a description of your query that consists of up to 48 alphanumeric characters (A-Z, a-z, and 0-9), then choose OK.
   
   Be sure to pick a description that reflects the type of query you created. For example, you might type Customer for a query that accesses all the fields in the Customer table.
Query Basics

This chapter explains the basics of creating and using queries. Specifically, this chapter presents:

- Before you begin
- Creating a query
- Saving a query
- Closing a query
- Opening an existing query
- Getting query information
- Printing queries
- Deleting a query
Before you begin

A query is a set of instructions that specifies the criteria by which information is extracted from a database. A query can retrieve data from one or more tables, from all the records in a table, or just a subset of records.

Once you have located the data you need through the use of a query, you can use a variety of sorting and filtering features to fine tune and analyze the data. For example, to send catalogs to your current customers only, you might create a query to locate the names and addresses of all your customers who ordered merchandise in the last year. Then you can use the query to generate mailing labels, create printed reports, update the information stored in the database, and even create a data file to export the data to another application.

The remaining sections of this chapter provide procedural detail to create and use queries.

Creating a query

This section presents:

- Query creation overview
- Details to create a query
- Record displays in multi-table queries

Note: The instructions in this guide are based on the demonstration database, the sports database, that is shipped with the Results application. If you have access to this sample database, you can use it while you learn how to use Results.

Query creation overview

The primary tasks involved in creating a query are:

1. Select the database tables to be included in the query. When you create a query, you select the tables that contain the information you want Results to retrieve.

2. Select the view you want to use to process the information retrieved by the query. A view is an on-screen display of only part of the information in a database, as specified in a query. A view can display data from one or more tables or from other views.
Table 3–1 lists these views.

Table 3–1: Query views

<table>
<thead>
<tr>
<th>View</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browse</td>
<td>Lists the tables in all connected databases from which you can add or remove to build a query. This view is the initial, basic view for all other views described in this table derive data.</td>
</tr>
<tr>
<td>Report</td>
<td>Creates multi-line and master-detail reports for printing. Provides calculated values and totals. You can format the headers, footers, and page breaks.</td>
</tr>
<tr>
<td>Form</td>
<td>Allows you to view, update, add, and delete individual records in the database.</td>
</tr>
<tr>
<td>Label</td>
<td>Formats and prints records in a variety of label formats, including mailing and inventory labels.</td>
</tr>
<tr>
<td>Export</td>
<td>Formats records to export to other applications.</td>
</tr>
</tbody>
</table>

3. Select the fields you want to display.

When you create a query, Results retrieves all the fields in each table you include in the query. However, you might want to limit the number of display fields, which are the fields that you select to appear in the view, whether on-screen or printed. For example, if you create a query that accesses the Customer table, you might want to view only those fields that contain address information. Limiting the display fields does not alter the contents of the query; it simply limits the number of fields you view.

Once you create the initial query, you can modify its contents, select a new query view, or select different display fields.
Details to create a query

You can create a query with the available database data that you want to examine.

To create a query:

**Note:** The steps for selecting the display fields do not apply to the Label view. See Chapter 4, “Using Query Views,” for instructions on creating queries for the Label view.

1. Choose **Query → New** and a view for the query. The **Add/Remove Tables** dialog box appears:

   ![Add/Remove Tables dialog box](image)

   The **Available Tables** selection list shows the tables that comprise the database that you are connected to. If you are connected to more than one database, **Results** indicates the database in which each table resides.
2. Select a table from the **Available Tables** selection list, then choose **Add**. The table appears in the **Selected Tables** selection list. The **Available Tables** selection list now displays only tables that have a relationship with the table you selected:

![Add/Remove Tables](image)

3. Repeat **Step 2** to add more tables, as necessary.

After you add tables, the **Selected Tables** selection list displays the relationships, or joins, among the tables in the query. For example, if you select the **Customer**, **Order**, and **Order-Line** tables, the **Selected Tables** selection list appears as follows:

![Add/Remove Tables](image)

*Displays the table’s relationship with the preceding table in the list*
4. When you are done adding tables, choose **OK**. The **Add/Remove Fields** dialog box appears:

![Add/Remove Fields Dialog Box]

*Indicates the table that contains the field*

The **Available Fields** selection list displays all the fields in the tables you selected. If you are connected to more than one database, **Results** indicates the database where each field in the list resides.

5. Select one or more fields from the **Available Fields** selection list, then choose **Add**:

![Add/Remove Fields Dialog Box with Selected Fields]

To select multiple fields, press **CTRL** and click individual fields, or press **SHIFT** and drag the mouse to select a range of fields. The fields are displayed in the **Selected Fields** selection list in the order in which they will be displayed in the view (from left to right).
6. To reorder the display fields, select a field and choose Move Up or Move Down. Repeat this process until the fields are listed in the order in which you want them to appear in the view.

7. Choose OK to display the query in the Progress Results window.

The appearance of the query depends on the type of view you chose in Step 1. The following query illustrates the Browse view. See Chapter 4, “Using Query Views,” for more information about changing and using query views:

The status bar identifies the query view you are using and the database to which you are connected. If you are connected to more than one database, the status bar lists the first database to which you connected.

Each record in the view contains the information from the display fields you selected for the query. In this example, the records are composite records because they contain fields from more than one table. The data is displayed from left to right in the order you specified when you created the query.
The customer *Lift Line Skiing* appears several times because of the type of relationships between the tables. Each customer can have one or more orders, thus the relationship between the Customer and Order tables is a one-to-many relationship. Each order can have several line items, so the Order and Order-Line tables also have a one-to-many relationship. Results displays a composite record for each line item a customer orders.

For more information about relationships, see Chapter 1, “Results Overview.”

**Record displays in multi-table queries**

There are two defaults that determine how records are displayed in queries that contain two or more tables:

- **Sort order** — Uses the primary index in the first table you included in the query to determine the order in which Results retrieves records from the tables.

- **Relationship types** — Displays only those records that contain information in both tables.

The following sections explain these defaults.

**Sort order default**

When you create a query that contains more than one table, Results uses the primary index for the first table you included in the query to determine the order in which Results retrieves and displays the records. See Chapter 1, “Results Overview,” for more information about indexes.

You can change the order in which records are sorted by selecting one or more fields to control the sort order for the query. For more information, see Chapter 5, “Changing the Query Selection Criteria.”
**Relationship types default**

When you create a query that specifies a field that is contained in more than one table, by default Results retrieves only the records that contain data in the related fields.

*Figure 3–1* helps to illustrate this relationship.

<table>
<thead>
<tr>
<th>Customer table</th>
<th>Order table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cust-num</td>
<td>Name</td>
</tr>
<tr>
<td>1</td>
<td>Lift Line Skiing</td>
</tr>
<tr>
<td>2</td>
<td>Urpon Frisbee</td>
</tr>
<tr>
<td>3</td>
<td>Hoops Croquet Co.</td>
</tr>
<tr>
<td>4</td>
<td>Go Fishing Ltd.</td>
</tr>
<tr>
<td>5</td>
<td>Match Point Tennis</td>
</tr>
<tr>
<td>6</td>
<td>Fanatical Athletes</td>
</tr>
</tbody>
</table>

*Figure 3–1: Relationship type field default behavior*

In *Figure 3–1*, notice that the Order table doesn’t contain any orders for Customer 5. Thus, a query to display the orders for each customer does not display a record for customer. This type of a relationship is called a *partial relationship*, or *inner join*.

The sample query identified in *Table 3–2* illustrates the results of a partial relationship between the Customer and Order tables.

**Table 3–2: Partial relationship sample query data**

<table>
<thead>
<tr>
<th>Cust-num</th>
<th>Name</th>
<th>Order-num</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lift Line Skiing</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>Urpon Frisbee</td>
<td>94</td>
</tr>
<tr>
<td>3</td>
<td>Hoops Croquet Co.</td>
<td>55</td>
</tr>
<tr>
<td>4</td>
<td>Go Fishing Ltd.</td>
<td>21</td>
</tr>
<tr>
<td>6</td>
<td>Fanatical Athletes</td>
<td>135</td>
</tr>
</tbody>
</table>
If you want to display all the records in the Customer table whether or not the Order table contains a related record, you can change the type of relationship between the tables to a complete relationship, or an outer join. The sample query identified in Table 3–3 illustrates the results of a complete relationship between the Customer and Order tables.

Table 3–3: Complete relationship sample query data

<table>
<thead>
<tr>
<th>Cust-num</th>
<th>Name</th>
<th>Order-num</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lift Line Skiing</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>Urpon Frisbee</td>
<td>94</td>
</tr>
<tr>
<td>3</td>
<td>Hoops Croquet Co.</td>
<td>55</td>
</tr>
<tr>
<td>4</td>
<td>Go Fishing Ltd.</td>
<td>21</td>
</tr>
<tr>
<td>5</td>
<td>Match Point Tennis</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Fanatical Athletes</td>
<td>135</td>
</tr>
</tbody>
</table>

For details on how to change table relationships in multi-table queries, see Chapter 5, “Changing the Query Selection Criteria.”
Saving a query

Once you create a query, you can save and reuse the query whenever you want to access the same information in your database.

To save a query:

1. Choose Query→Save. The Save As dialog box appears:

2. Type a short description for the query.

   The description can consist of up to 48 alphanumeric characters and spaces. You’ll use the query description to locate and open queries, so pick a description that reflects the type of query you create. For example, you might type Customer Orders for a query that accesses all the orders for each customer in the Customer table.

   **Note:** Results lets you use descriptions to locate and access query files, so you are not limited to eight-character filenames. However, Results automatically generates filenames for these files so that your computer can store and access them properly. The query directory (.qd7) file contains a directory that maps the descriptions you provide with the actual filenames Results generates. See the *OpenEdge Reporting: Query/Results Administration and Development* for more information about the query directory.

3. Choose OK.

   Once you have saved a query, you can use the Save As command to save the query without changing its description.
Closing a query

You will generally want to close a query without exiting Results.

To close a query without quitting Results:

1. Choose Query → Close.
   
   If you created or modified the query and did not save it, a dialog box prompts you to save.

2. Choose OK to save the query and close it, or No to close the query without saving it.
   
   If you are saving the query for the first time, the Save As dialog box appears.

3. Type a description of your query that consists of up to 48 alphanumeric characters, then choose OK.

Opening an existing query

Opening a query that has already been created is another frequent Results activity.

To open an existing query:

1. From the Results main window, choose Query → Open. The Open dialog box appears.

2. Select the query you want to open, then choose OK.
3. By default, this dialog box lists all the queries for databases to which you are currently connected. Although you cannot open queries for databases to which you are not currently connected, you can list them by selecting the **Show Queries on Disconnected Databases** toggle box.

### Getting query information

To help you keep track of the contents of your queries, you can request Results to display a list containing the tables included in the query, the display fields used, the sort order for the query, and if you printed or viewed a report through print preview, the amount of time it took to run the report.

To display this list, choose **Options → Query Information**.

### Printing queries

You can print the contents of your queries and perform a print preview from the Report, Labels, and Export query views.

To print a query:

1. Choose **Query → Print → To Printer**. The **Print** dialog box appears.
2. Select the print settings you want to use, then choose **OK**.
Using print preview

Print Preview lets you display the query instead of sending it to the printer.

To use Print Preview, choose **Query → Print Preview**. The query appears in a **Print Preview** window:

You can use several techniques to scan through records in **Print Preview**, including the horizontal and vertical scroll bars, the navigation buttons, and **PAGE UP** and **PAGE DOWN**. The following sections describe these methods.

**Scroll bars**

Use the vertical scroll bar to scan up and down through records on the current page. Use the horizontal scroll bar to scroll left and right to view information that is wider than the window.
Navigation buttons

Table 3–4 identifies the navigation buttons you can use to change the current page displayed in the print preview.

Table 3–4: Print Preview navigation buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prompts for a page to display.</td>
</tr>
<tr>
<td></td>
<td>Displays the first page of records.</td>
</tr>
<tr>
<td></td>
<td>Displays the previous page of records.</td>
</tr>
<tr>
<td></td>
<td>Displays the next page of records</td>
</tr>
<tr>
<td></td>
<td>Displays the final page of records.</td>
</tr>
</tbody>
</table>

Keyboard

Before you can use the keyboard to navigate in Print Preview, click anywhere inside the preview to give it focus.

Results provides the following navigation keys:

- **Page up** — Scrolls up through the current page one screen at a time.
- **Page down** — Scrolls down through the current page one screen at a time.
- **Cursor up** — Scrolls to the previous record on the page.
- **Cursor down** — Scrolls to the next record on the page.
Deleting a query

Periodically, you might want to delete a query that is of limited or no use.

To delete one or more queries:

1. Choose Query→Delete. The Delete dialog box appears.

2. Choose the query you want to delete, then choose OK.

To select multiple queries, press CTRL and click individual queries, or press SHIFT and drag the mouse to select a range of queries:

3. By default, this dialog box lists all the queries for databases that you are currently connected to. To list additional queries for databases that you are not currently connected to, activate the Show Queries on the Disconnected Databases toggle box.
Results provides query views that process and analyze data. This chapter presents:

- Query views overview
- Browse view
- Form view
- Report view
- Label view
- Export view
Query views overview

You can display queries in a variety of interchangeable views that let you accomplish specific tasks. These views include **Browse**, **Form**, **Report**, **Label**, and **Export**. To change the view, either choose a different view from the **View** menu or choose a view button from the tool bar. For example, if you are currently viewing a query as a browse, you might want to change it to a report that you can format and print.

When you switch between views, Results maintains both the contents of the query and the display fields you selected, so none of your data is lost. However, the **Labels** view allows you to select an additional set of display fields specifically for creating labels. In this case, Results maintains two lists of display fields: the list you select for creating labels and the list you use for the remaining query views. Results automatically toggles between the two lists depending on the view you choose. See the “Label view” section on page 4–24 for more information about switching to the **Label** view.

Browse view

A **browse** is an onscreen view that displays records in a tabular format similar to a spreadsheet. You can use a browse view to scan through large numbers of records quickly. Figure 4–1 shows an example of a browse.

![Figure 4–1: Browse view example](image)
The query in the browse presented in Figure 4–1 uses a relationship among the Customer, Order, and Order-Line tables to list the items each customer ordered. Each composite record displays the Cust-Num and Name fields from the Customer table; the Order-num field from the Order table, and the Item-num and Extended-Price fields from the Order-Line table.

Navigating through records

Use the Left and Right arrow buttons and the vertical scroll bar to scan through records. The Left arrow button scrolls to the first record in the browse and selects it, and the Right arrow button scrolls to the last record in the browse and selects it. If the browse is wider than the window, a horizontal scroll bar appears so that you can scroll left or right.

You can also use several keyboard keys to navigate through records in the browse view. To use a key, move the input focus to the browse by placing the pointer in the browse and clicking the left mouse button, then press one of the keys described in Table 4–1.

Table 4–1: Browse view keys

<table>
<thead>
<tr>
<th>Key</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page up</td>
<td>Scrolls to the first record in the window and selects it.</td>
</tr>
<tr>
<td>Page down</td>
<td>Scrolls to the last record in the window and selects it.</td>
</tr>
<tr>
<td>Home</td>
<td>Scrolls to the first record in the browse and selects it.</td>
</tr>
<tr>
<td>End</td>
<td>Scrolls to the last record in the browse and selects it.</td>
</tr>
<tr>
<td>Cursor up</td>
<td>Scrolls up one record in the browse and selects it.</td>
</tr>
<tr>
<td>Cursor down</td>
<td>Scrolls down one record in the browse and selects it.</td>
</tr>
</tbody>
</table>
Using frames

The query is contained within a portion of the window called a frame. By default, a browse uses one frame for each query you create. Figure 4–2 shows an example of a browse view that displays one query frame.

Figure 4–2: Browse view displaying one query frame

If a query includes more than one table, you can tell Results to split the contents of the tables into two frames, allowing you to view the data more easily. See the “Using Master and Detail sections to display records” section on page 6–3 for more information.
Figure 4–3 illustrates a browse view that displays two query frames.

When you split the contents of a query into two frames, Results provides a set of navigation buttons for each frame. The frames will overlap if there is not enough space to display them in the window at the same time. To display both frames simultaneously, modify the frame properties as described below.

You can also enlarge the window by clicking the maximize button or dragging the corner resize handle in the window border.
Modifying frame properties

This section describes how to modify frame properties for one or both frames to adjust the position and size of each frame.

To modify the frame properties:

1. Choose **Options** → **Frame Properties**. The **Frame Properties** dialog box appears:

   ![Frame Properties dialog box]

2. Select the frame you want to modify from the **Frames** selection list.

3. Change one or both of the following attributes:
   - **Row** — Determines how many rows separate the menu line from the top of the frame. Use this feature to stagger the location of the frames when you want to display two frames simultaneously.
   - **Browse height** — Determines the number of records displayed in a query at one time.

4. Choose **OK**.
Form view

A form is an on-screen view that displays one record at a time. You typically use forms to add, delete, and update one or more records in a table. Figure 4–4 shows an example of a Form.

Figure 4–4: Form view example

This form displays records in the Customer table one at a time. Press TAB to move through the display fields within the record.
Navigating through records

Use the navigation buttons to scan through records:

Table 4–2: Navigation buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>first</td>
<td>Displays the first record in the query.</td>
</tr>
<tr>
<td>previous</td>
<td>Displays the previous record in the query.</td>
</tr>
<tr>
<td>next</td>
<td>Displays the next record in the query.</td>
</tr>
<tr>
<td>last</td>
<td>Displays the final record in the query.</td>
</tr>
</tbody>
</table>

Modifying records

You can use forms to update, copy, add, and delete records contained in a single table.

For example, if a query consists of the fields in the Customer table, you can use the Form view to modify the Customer table. However, if a query consists of fields from more than one table, you cannot modify the tables unless you use the Master-Detail feature to break the tables into individual frames. See the “Using Master and Detail sections to display records” section on page 6–3 for more information.

The procedures in the following sections explain how to modify records contained within a single table.
Updating records

The following procedure describes how to update sections explain how to modify records contained within a single table.

To change the contents of one or more fields in a record:

1. Display the record you want to modify.
2. Choose Update.

Results displays the fields in the Update window. The first field is highlighted by default:

3. Select the contents of the field you want to modify, then type the new information in the field. The new information overwrites the old information.
4. When you are done modifying fields, choose OK to save your modifications and close the Update window.

As you update fields, you might find that your modifications are restricted by a variety of factors. For example, you cannot update a calculated field because it is a display only field. Or perhaps your company assigns unique customer numbers so you cannot update the Cust-Num field.
Copying records

The following section explains how to create a new record based on the contents of an old record. Be sure to follow the guidelines at your site for creating new records based on existing records.

To create a new record based on the contents of an old record:

1. Display the record you want to copy.
2. Choose Copy.

Results copies the record and displays the copy in the Update window. Results copies all the fields in the record whether or not the fields are displayed.

3. Since you are creating a new record, you will probably have to update several fields to reflect the new information.
4. When you are done modifying fields, choose OK to save your new record and close the update window.
Creating records

The following procedure describes how to create a new record.

To create a new record:

1. Choose Add.

   Results creates a new record that contains all the fields in the table, then displays the record in the Add window. Note that only the fields you chose to display in the query are displayed in the window:

   ![Add window](image)

   Some of the fields might already contain values, as in the Credit-Limit field. These initial values are determined by the business practices established by your company and enforced by the database.

2. Type values in the empty fields.

3. When you are done filling in fields, choose OK to save your record and close the Add window.

Deleting records

The following procedure describes how to delete a record.

To delete a record:

1. Display the record you want to delete.

2. Choose Delete. A dialog box prompts you to verify that you want to delete the record.

3. Choose Yes. Results deletes the entire record, including fields that are not displayed in the query.
You might find that you cannot delete a particular record. For example, you might not be able to delete a customer that has outstanding orders. Such restrictions are determined by your company’s business practices and are enforced by your database.

**Rearranging fields**

Table 4–3 describes the type of field rearranging you can perform in the **Form** view.

**Table 4–3: Rearranging fields in the Form view**

<table>
<thead>
<tr>
<th>To...</th>
<th>Then...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change the order in which fields display...</td>
<td>Click and drag the field you want to move.</td>
</tr>
<tr>
<td>Lengthen the frame that contains the form...</td>
<td>Drag a field to the bottom of the frame.</td>
</tr>
<tr>
<td>Shorten the frame that contains the form...</td>
<td>Drag a field to the top of the frame.</td>
</tr>
</tbody>
</table>

**Note:** Moving a field within a form does not alter either the tab order for the fields or the order in which they are displayed in other views. See the “Adding and removing display fields” section on page 7–2 for more information about changing the tab order of fields.
Using frames

A query is contained within a frame. By default, a form uses one frame for each query you create. Figure 4–5 shows a query displayed within a frame:

![Query frame](image)

**Figure 4–5: Query with one frame**

However, if a query includes more than one table, you can tell Results to split the contents of the tables into two frames, allowing you to view the data more easily. See the “Using Master and Detail sections to display records” section on page 6–3 for more information.
Figure 4–6 illustrates a query that uses two frames.

When you split the contents of a query into two frames, Results provides a set of navigation buttons for each frame. The frames will overlap if there is not enough space to display them in the window at the same time. To display both frames simultaneously, modify the frame properties as described below.

You can also enlarge the window by clicking the maximize button or dragging the corner resize handle in the window border.
Modifying frame properties

This section describes how to modify properties for one or both frames to adjust the position and size of each frame.

To modify the properties:

1. Choose Options→Frame Properties. The Frame Properties dialog box appears:

![Frame Properties dialog box](image)

2. Select the frame you want to modify from the Frames selection list.

3. Change one or both of the following attributes:
   - **Row** — Determines how many rows separate the menu line and the top of the frame. Use this feature to stagger the location of the frames when you want to display several frames simultaneously.
   - **Read-only** — When activated, prevents you from modifying records in the database.

4. Choose OK.

If you activated Read-Only, then Add, Copy, Update, and Delete are grayed out.
Report view

Use the Report view to create detailed printed reports.

Figure 4–7 shows a sample on-screen report format.

An on-screen report lists the field labels and display formats for the display fields you included in the query. Display formats control the way data appears on-screen and in printed reports. For example, the format for the Name field, (X20), means this field can display up to 20 characters. See the “Changing display field properties” section on page 7–3 for more information.

Figure 4–8 uses Print Preview to approximate how the same report looks when printed.
Adding cover and final pages, headers, and footers

You can include the following elements in a report:

- **Cover page** — A separate page at the front of a report.
- **Final page** — A separate closing page at the end of a report.
- **Headers** — A note at the top of a page. You can use one or all of the following headers: a centered header printed on the first page of the report; a left, centered, or right header printed on each page of the report.
- **Footers** — A note at a bottom of a page. You can use one or all of the following footers: a centered footer printed on the last page of the report; a left, centered, or right footer printed on each page of the report.
To add a cover page, final page, headers, and footers to your document:

1. Choose **Options → Header and Footers**. The **Headers and Footers** dialog box appears:

![Headers and Footers dialog box]

2. Choose one of the following buttons to create a header, footer, or page:

<table>
<thead>
<tr>
<th>![Icon]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Cover Page Icon]</td>
<td>Create a cover page.</td>
</tr>
<tr>
<td>![Special Header Icon]</td>
<td>Create a special header for the first page.</td>
</tr>
<tr>
<td>![Left Aligned Header Icon]</td>
<td>Create a left-aligned header for all pages.</td>
</tr>
<tr>
<td>![Centered Header Icon]</td>
<td>Create a centered header for all pages.</td>
</tr>
<tr>
<td>![Right Aligned Header Icon]</td>
<td>Create a right-aligned header for all pages.</td>
</tr>
<tr>
<td>![Left Aligned Footer Icon]</td>
<td>Create a left-aligned footer for all pages.</td>
</tr>
</tbody>
</table>
Using Query Views

3. Enter the information that you want to print within the header or footer. To enter the information, do one or both of the following:

- Type text in the **Layout** box at the top of the window.
  
  For example, you might want to type **Customer Orders** in a header for a report that lists each customer’s orders. Note that you can enter information that spans several lines by inserting a carriage return at the end of each line.

- Double click a function in the **Function List**.

  *Functions* insert a variable value into the element you create when you print the report. For example, you can use a function to print the current date in every header in your report. Table 4–4 lists the available functions.

<table>
<thead>
<tr>
<th>Table 4–4: Report variable functions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Function</strong></td>
</tr>
<tr>
<td>TODAY</td>
</tr>
<tr>
<td>TIME</td>
</tr>
<tr>
<td>NOW</td>
</tr>
<tr>
<td>COUNT</td>
</tr>
<tr>
<td>PAGE</td>
</tr>
</tbody>
</table>
This is an example of a header that centers the words “Customer Orders” and the current date on each page of the report:

Table 4–4: Report variable functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER</td>
<td>Inserts the user’s username.</td>
</tr>
<tr>
<td>VALUE</td>
<td>Inserts the current value of any field you select. You cannot use this function for cover, first only, last only, and final pages.</td>
</tr>
</tbody>
</table>

4. Repeat Steps 2 and 3 to create additional headers, footers, and pages.

5. Choose OK when you are done creating elements, choose OK.
Selecting page sizes

By default, Results prints reports on 8.5x11-inch pages. However, Results provides a variety of paper sizes from which to choose, including legal, tabloid, A3, A4, A5, B5, and 35-mm slide. You can also adjust the report dimensions to print on other paper sizes. See the “Adjusting report dimensions” section on page 4–22 for more information.

To choose a different page size:

1. Choose Options → Standard Page. The Standard Page dialog box appears:

![Standard Page dialog box]

2. Select the paper size you want to use, then choose OK.
Adjusting report dimensions

Use custom page to control the position of elements on the printed page. This feature is useful for sizing reports to print on non-standard sized pages.

To adjust the report dimensions:

1. Determine the page size that most closely matches the page dimensions you need.

2. Choose Options→Custom Page. The Custom Page dialog box appears:

Modify the attributes in this dialog box to reformat your report. Table 4–5 describes the Custom Page attributes.

Table 4–5: Custom Page attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lines before header</td>
<td>If you include a header in the report, determines the number of blank lines inserted before the header.</td>
</tr>
<tr>
<td>Left margin</td>
<td>Determines the left margin, in character spaces.</td>
</tr>
<tr>
<td>Lines between header and body</td>
<td>If you include a header in your report, determines the number of blank lines inserted between the header and the body of the report.</td>
</tr>
</tbody>
</table>
3. Change the format attributes as desired.

4. When you are done reformatting the report, choose OK to save your changes.

**Printing reports**

The following procedure describes how to print a report.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spaces Between Columns</td>
<td>Determines the number of character spaces between fields printed horizontally across the page.</td>
</tr>
<tr>
<td>Line Spacing</td>
<td>Determines the number of lines down from each previous record the records are displayed.</td>
</tr>
<tr>
<td>Lines Between Body and Footer</td>
<td>If you include a footer in the report, determines the number of blank lines inserted between the body of the report and the footer.</td>
</tr>
<tr>
<td>Page Length</td>
<td>Determines the total number of lines printed on a page.</td>
</tr>
</tbody>
</table>

**Table 4–5: Custom Page attributes**

To print a report:

1. Choose **Query** → **Print** → **To Printer**. The **Print** dialog box appears.

2. Select the print settings you want to use, then choose OK.
Label view

Figure 4–9 identifies the Label view.

![Label view icon](image)

**Figure 4–9: Label view icon**

Use the Label view to create and print labels. Since many labels, like mailing labels, often use the same fields, Results lets you pick a set of default display fields. The default display fields are selected by your system manager based on your company’s business practices. For example, if you want to create mailing labels, you might tell Results to select the default display fields from the Customer table.

To create labels:

2. Select a table from the Available Tables selection list, then choose Add:

![Add/Remove Tables dialog box](image)

The table appears in the Selected Tables selection list. As you add tables, the Available Tables selection list is updated to display only tables that have a relationship with the table you selected.
3. Repeat Step 2 to select additional tables.
4. Choose **OK** when you are done selecting tables.

A dialog box prompts you to choose whether you want Results to select the display field for you.

5. Do one of the following:

- Choose **Yes** to select the default display fields.

  Results selects the fields based on the criteria set by your site manager. Thus, some tables might contain numerous display fields and other tables might contain no default display fields. If you want to change the default display fields for a particular table, see your site manager.

- Choose **No** to display the **Add/Remove Fields** dialog box and select the display fields yourself.

The **Add/Remove Fields** dialog box appears. The dialog box lists the display fields in the order in which they will appear in the labels:
6. To accept the list of display fields as is, choose OK. To modify the list, insert fields, delete fields, or change the order of the fields as follows:

- To insert a display field, place the cursor where you want to insert the field, then choose Insert Field. The Insert Field dialog box appears. Select a field, then choose OK.

- To delete a display field, select the field then press either CTRL-X or DELETE.

- To delete all the display fields, choose Clear All.

- To change the order of a display field, select the field then press CTRL-X. Position the cursor where you want to insert the field, then press CTRL-V.

- To restore the list of display fields to its original condition, choose Restore Layout.

Results displays the label information:

![Label dimensions and display fields](image)

**Note:** The Field→Aggregates menu selection is not available with the Label view. The Field→Properties menu selection is only available for calculated fields.
The following figure uses **Print Preview** to show how the labels look when they are printed:

![Print Preview](image)

**Selecting label types**

By default, Results prints labels approximately 3.5 inches wide and 1 inch high in a single-column format. However, Results provides a variety of standard label sizes, including single-column and multi-column formats, Avery labels, large mailing labels, envelope labels, postcard labels, and Rolodex labels.

**To choose a label type:**

1. Choose Options→ **Standard Label**. The **Standard Label** dialog box appears:

![Standard Label](image)
2. Select the label type you want to use, then choose OK.

Modifying label formats

If you find that none of the standard label types is suitable, you can modify the format for one of the existing label types.

To modify the label format:

1. Determine which label type is most suitable. The label type you choose will serve as the basis for your new label.

Choose Options → Custom Label. The Custom Label dialog box appears:

Use the custom label attributes in the dialog box to reformat the label. Table 4–6 describes the custom label attributes.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left margin</td>
<td>Determines the left margin, in character spaces.</td>
</tr>
<tr>
<td>Label width</td>
<td>Determines the label width, in characters.</td>
</tr>
<tr>
<td>Number across</td>
<td>Determines the number of labels printed horizontally.</td>
</tr>
</tbody>
</table>
2. Change the format attributes as desired.

3. When you are done reformatting the label, choose **OK** to save your changes.

### Printing labels

The steps to print labels are relatively simple.

**To print labels:**

1. Choose **Query → Print → To Printer**. The **Print** dialog box appears.

2. Select the print settings you want to use, then choose **OK**.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>Determines the label height, in lines.</td>
</tr>
<tr>
<td>Omit blank lines</td>
<td>Determines whether a line is omitted when the corresponding field within a record is blank.</td>
</tr>
<tr>
<td>Spaces between</td>
<td>Determines the number of character spaces between labels printed horizontally.</td>
</tr>
<tr>
<td>Copies of each</td>
<td>Determines the number of times each label is printed. Duplicate labels are printed vertically.</td>
</tr>
<tr>
<td>Lines between</td>
<td>Determines the number of lines between labels printed vertically.</td>
</tr>
</tbody>
</table>
Export view

The following icon identifies the Export view:

![Export view icon]

**Figure 4–10: Export view icon**

Use the Export view to generate ASCII files to use with other software applications. For example, you can use the Export view to create an ASCII file that contains fields from the Customer table, then use the contents of the ASCII file to customize form letters you create from within a word processing application. You can also use the contents of an ASCII file from within a graphics application to create presentations.

Figure 4–11 shows an export query, and Figure 4–12 shows the ASCII file that is generated by the data in Figure 4–11.

![Export query]

**Figure 4–11: Export query**

This export query contains display fields from the Customer table. The format attributes determine how the contents of these fields are formatted for the ASCII file. See the “Modifying export types” section on page 4–33 for more information about format attributes.
Figure 4–12 uses **Print Preview** to show the ASCII file generated by the Export query. You can use **Print Preview** to view ASCII export data.

---

**Generating an ASCII file**

Once you create the initial export query, you can generate an ASCII file by selecting the export type for the ASCII file, then creating the file. The following sections explain how to do this.

**Selecting an export type**

Applications often differ in the way they expect ASCII files to be formatted. Thus, **Results** provides several export types to format data for a variety of applications:

- **ASCII.**
- **ASCII with field-name header.**
- **ASCII fixed-width (SDF).**
- **ASCII fixed-width (SDF) with field-name header.**
- **Comma-separated value (CSV).**
- **DIF with dates as strings.**
• DIF with dates as numbers.
• Microsoft Word (creates an ASCII data merge file).
• Microsoft Word for Windows (creates an ASCII data merge file).
• Progress.
• SYLK.
• System Data Format.

**Note:** If none of the export types is suitable, you can temporarily modify an existing export type by overriding the values of one or more of its format attributes. See the “Modifying export types” section on page 4–33 for more information.

By default, Results uses the export type called **Progress** to generate ASCII files.

To choose a different export type:

1. Determine which export type you want to use.

   See the documentation for the application in which you plan to incorporate the data file to determine which export format to use.

2. Choose **Options**→**Standard Export**. The **Standard Export** dialog box appears:

   ![Standard Export dialog box](image)

3. Select the export type you want to use, then choose **OK**.
Creating an ASCII file

The following procedure describes how to create an ASCII file.

To create an ASCII file:

1. Choose **Query** → **Print** → **To File**. The **Print to File** dialog box appears:

By default, Results names the file `export.txt` and saves it in your working directory. You can change the defaults, as follows:

- To rename the file, type a new file name in the **File** field.
- To change the directory in which the file is located, either type the full pathname and filename in the **File** field, or choose **Files** and change the directory.
- To append the file to the end of an existing ASCII file, first activate the **Append to Existing File** toggle box, then in the **File** field, enter the name of the existing file to which you want to append the file.

Modifying export types

If none of the standard export types is suitable, you can temporarily modify an export type by overriding its default settings for one or more format attributes.
To modify an export type:

1. Determine which export type most closely matches the format you need.

2. Choose Options→Custom Export and select the attribute you want to modify. Table 4–7 describes these attributes.

Table 4–7: Custom export attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output headers</td>
<td>A Yes or No value that indicates whether Results will export the field name as the first line of the export.</td>
</tr>
<tr>
<td>Fixed width</td>
<td>A Yes or No value that indicates whether the width of the fields is fixed. This value must be set to No unless you select ASCII Fixed-width (SDF) as your export type. The field width determines how Results exports the field data. For example, it defines the width of a character field. See the “Changing display field properties” section on page 7–3 for information on formats.</td>
</tr>
<tr>
<td>Record start</td>
<td>The initialization string at the start of a record. The initialization string is the set of characteristics that precedes the record data on each line of the export file. For example, an asterisk (*) might indicate the beginning of a line.</td>
</tr>
<tr>
<td>Record end</td>
<td>The record termination string. The record termination string is the set of characters used to indicate the end of each record value.</td>
</tr>
<tr>
<td>Field delimiter</td>
<td>The field delimiter string. The field delimiter string is the set of characters used to indicate the beginning and end of each field value.</td>
</tr>
<tr>
<td>Field separator</td>
<td>The field separator string. The field separator string is the set of characters that separates each field value within a record.</td>
</tr>
</tbody>
</table>
Depending which attribute you select, the following occurs:

- The **Output Headers and Fixed Width** options toggle between **Yes** and **No**. When you set the value to **Yes**, a check mark appears next to the option name within the menu.

- The **Record Start**, **Record End**, **Field Delimiter**, and **Field Separator** options display a dialog box that allows you to enter the codes for the string. See “Modifying strings” section on page 4–35 for more information about modifying these attributes.

**Modifying strings**

You can enter up to 16 codes for each string. Enter these codes from left to right and top to bottom:
To enter a code, do one of the following:

- Type the code using one of the formats in Table 4–8.

**Table 4–8: Code formats**

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>’X’</td>
<td>Literal character enclosed in single quotes.</td>
<td>’C’</td>
</tr>
<tr>
<td>^X</td>
<td>Control character preceded by a circumflex.</td>
<td>^Z</td>
</tr>
<tr>
<td>##h</td>
<td>One or two hex digits followed by the letter h.</td>
<td>4Fh</td>
</tr>
<tr>
<td>###o</td>
<td>One, two, or three octal digits followed by the letter o.</td>
<td>374o</td>
</tr>
<tr>
<td>###</td>
<td>A decimal number consisting of one, two, or three digits.</td>
<td>027</td>
</tr>
<tr>
<td>XXX</td>
<td>Control symbol. Choose the Pick From ASCII Table button.</td>
<td>Not applicable.</td>
</tr>
</tbody>
</table>

- Choose Pick From ASCII Table and choose a control symbol from the ASCII Codes dialog box.
A query’s selection criteria determine which records the query returns from the database and the order in which it returns them. This chapter explains several ways you can change the selection criteria for a query.

This chapter describes:

- Adding and removing tables from a query
- Changing table relationships in multi-table queries
- Sorting records
- Creating filters
- Using the Data Governor
Adding and removing tables from a query

Results supports adding and removing tables for an existing query.

To add and remove tables from a query:

1. Choose Tables → Add/Remove Tables.

Results displays the Add/Remove Tables dialog box:

2. Add and remove tables as follows:
   - To add a table, select the table from the Available Tables selection list and choose Add.
   - To remove a table, select the table from the Selected Tables selection list and choose Remove.

3. When you are done adding tables to and removing tables from the query, choose OK.

Results displays the Add/Remove Fields dialog box, so you can add or remove display fields. Note that when you remove a table, Results automatically removes the display fields associated with the table.

4. To add a display field, select the field from the Available Fields selection list and choose Add.
5. In the **Selected Fields** selection list, the fields are listed in the order they will be displayed in the query view (from left to right). To reorder the display fields, select a field and choose either **Move Up** or **Move Down**.

Select the fields, then choose **Add** to include them in the layout. Notice that the field list includes not only database fields, but also includes any calculated fields you defined for the query. Use **Move Up** and **Move Down** to position the order of the fields in the layout:

If you want to include information in a calculated field, but did not create the fields before accessing **Label** view, choose **Field → Add Calculated Field** from the menu bar, then create the field. Results inserts the calculated field in the label layout at the cursor position. See the “Creating calculated fields” section on page 6–8 for information about creating calculated fields.

**Note:** Because the **Form** view lets you change the position of the fields by dragging the fields in the display, **Move Up** and **Move Down** do not change the position of the fields. They simply change the tab order for the fields.

6. Choose **OK** to save your modifications to the query.
Changing table relationships in multi-table queries

When you use a multi-table query, Results uses table relationships to determine the set of records it returns.

Types of table relationships

Results contains two types of table relationships:

- Partial

  When you create a query, Results automatically establishes a partial relationship between each pair of tables in the query. A partial relationship, also called an inner join, accesses only those records in the first table that have related records in the second table.

- Complete

  If you want to access all the records in the first table whether or not they have related records in the second table, change the relationship to a complete relationship. A complete relationship is also called an outer join.

  If you set a complete relationship between a pair of tables and use sort ordering to control the order in which the query retrieves records, Results automatically uses the primary index in the first table in the relationship as the first sort field. See the “Sorting records” section on page 5–7 for more information.
For example, you might have a query that displays the invoices for your customers by invoice date, as shown in Figure 5–1.

![Figure 5–1: Example of a partial relationship](image)

Sorted in ascending order by Invoice-Date

If you change the table relationship for the query presented in Figure 5–1 to a complete relationship, Results will automatically use the **Cust-Num** field (primary index in the first table) then the **Invoice-Date** field to sort the query. Thus the query will display the records by invoice date for each customer.

Note that the values for aggregate subtotals and totals only associated with fields from the second table will change to reflect the new sort order when you change the table relationship to a complete relationship. For example, if you created an aggregate total to total the amount of the invoices by date, then changed the relationship between the Customer and Invoice tables to a complete relationship, the aggregate total would then total the amount of the invoices by date for each customer.

See the “Record displays in multi-table queries” section on page 3–8 for examples of partial and complete relationships.
To change the relationship between one or more pair of tables:

1. Choose **Table**→**Relationship Types**. The **Relationship Types** dialog box appears:

   ![Relationship Types Dialog Box]

   **Note:** Partial relationships are displayed in white and complete relationships are displayed in reverse video.

2. To change the type of relationship between a pair of tables, simply select the pair of tables you want to modify. Results automatically toggles to the opposite relationship type.

3. When you are done modifying relationship types, choose **OK**.
Changing the Query Selection Criteria

Sorting records

By default, Results uses the primary index from the first table you included in the query to determine the order in which it retrieves and displays records.

Figure 5–2 displays the items ordered by each customer. Note that Results automatically uses the primary index, **Cust-Num**, within the Customer table to sort the records in ascending order by customer number.

![Sorted in ascending order by Cust-Num](image)

**Figure 5–2:** Query displaying ordered items by customer

You can alter the order in which the query retrieves records by choosing one or more fields as the criteria for sorting the records.
For example, you might prefer to see the list sorted by order number, starting with the most recent order. In this case, you use the **Order-Num** field to sort the records in descending order, as shown in Figure 5–3.
Then you might decide that you want to retrieve the orders for each customer, starting with the most recent order. In the Sports database, the order number increases with each new order. Therefore, you use two fields to sort the records: Cust-Num and Order-Num. Cust-Num is sorted in ascending order and Order-Num is sorted in descending order, as shown in Figure 5–4.

You can sort on all database fields and on the following types of calculated fields: Math, String Function, Numeric Function, Date Function, and Logical Function. See Chapter 6, “Analyzing Data,” for more information on the different types of calculated functions.

Sort order and break groups

When you use a field to sort records in a query, Results automatically creates a break group based on the field you used to sort the records. A break group is a set of records that have a common value in a certain database field. Figure 5–4 used the Cust-Num and Order-Num fields to retrieve the orders for each customer, starting with the most recent order. Thus it contains two break groups: one for Cust-Num and one for Order-Num.

Break groups are useful because they allow you to perform calculations on a subset of records in the query. For example, you can count the number of orders for each individual customer or even calculate the total Extended-Price for the items each customer ordered. See the “Creating aggregate totals and subtotals” section on page 6–41 for more information about performing calculations on break groups.
Specifying the sort order

This section describes how to change the sort order of records in Results.

To change the order in which Results sorts records:

1. Choose Data→Sort Ordering. The Sort Order Fields dialog box appears.

Notice that the Available Fields selection list contains all the fields in the query. You can use any field to sort records, including some calculated fields and fields that are not displayed in the query view. The available calculated fields appear at the top of the Available Fields list:

![Sort Order Fields Dialog]

2. You can use up to 16 fields to sort the records in the query. To choose the sort fields, select the fields from the Available Fields selection list, then choose Add.
3. The fields appear in the **Selected Fields** selection list. Note that the fields appear in the order in which Results uses them to sort the records:

![Screenshot of Sort Order Fields dialog box]

- **Sorted by Cust-Num, then Order-Num**

4. If you selected only one sort field or you are satisfied with the order of the sort fields, go to **Step 5**. Otherwise, to change the order of the sort fields, select one field at a time and choose either **Move Up** or **Move Down**.

5. By default, the fields are sorted in ascending order. To sort a field in descending order, select the field, then choose **Descending**:

![Screenshot of Sort Order Fields dialog box]

6. When you are done setting the sort criteria, choose **OK** to update the sort order and redisplay the query.
Creating filters

When you create a query, Results retrieves all the records in the tables you select. However, sometimes you might want to retrieve only a subset of these records. For example, you might want to refine the selection criteria for a query that retrieves customer invoices to retrieve only those invoices for customers in Massachusetts. You might further refine the query to retrieve only those invoices for customers in Massachusetts who have not paid for their merchandise.

To do this, you use a filter to create a set of instructions, called a \textit{WHERE clause}, to instruct the query to retrieve only those records where the State field equals Massachusetts and the Balance field is greater than 0. In this example, the \textit{WHERE clause} is written as follows:

\begin{verbatim}
State = "MA" AND Balance > 0
\end{verbatim}

In this example, \texttt{State} and \texttt{Balance} are fields; \texttt{MA} is compared to \texttt{State} using the equality operator (\texttt{=}); \texttt{0} is compared to \texttt{Balance} using the greater than operator (\texttt{>}); the logical operator \texttt{AND} bonds the two sections of the clause, called \textit{expressions}, together. See Appendix B, “Comparison Operators,” for a description of the available comparison operators.

Results makes creating a \textit{WHERE clause} easy. Just select the elements you want to include in the clause and Results generates and formats the clause for you. If, however, you prefer to write and format clauses yourself, Results lets you create a \textit{WHERE clause} manually.

Results provides you with two similar techniques to generate a \textit{WHERE clause}:

- Data selection
- Query by example

\section*{Data Selection}

Data selection lets you generate a \textit{WHERE clause} either manually, or by selecting fields and operators and allows Results to format the \textit{WHERE clause} for you. The \textit{WHERE clause} can be based on any field from within any table in the query, whether or not the field is a display field. Data selection also gives you the opportunity to set the selection criteria at run time. This feature is available for all views.
Query by example

Query by example uses a blank form to help you generate a WHERE clause based on one or more display fields. This feature is available only in Form view.

The following sections describe how to use these techniques.

Using Data Selection to refine a query

You can use Data Selection to create a WHERE clause automatically or manually. The following sections describe these techniques.

Using Data Selection to create a WHERE clause automatically

Use this procedure to create a WHERE clause automatically in Results.

To refine a query using Data Selection:

1. Choose Data → Selection. If there is more than one table available, the Available Tables dialog box appears.

2. Select the table that contains the fields you want to use to limit the query, then choose OK.

If you are creating a WHERE clause that contains more than one expression and these expressions contain fields from different tables, create the expressions in increments. Select the first table and create the first expression, then select another table to create the second expression, and so on until you complete the clause.
The **Data Selection** dialog box appears:

3. If you want **Results** to prompt you to set your selection criteria at run time, activate the **Ask At Runtime** toggle box.

   By default, **Data Selection** does not use **Ask At Runtime**. This means that the values you enter in the **WHERE clause** are the only values used to select data. To change the selection criteria, you must re-enter **Data Selection** and create a new **WHERE clause**. **Ask At Runtime** lets you modify the values for one or more fields in the **WHERE clause** each time you run the query. For example, you can tell **Results** to prompt you for a value to compare to the **State** field each time you run a query. See the “Setting Data Selection at run time” section on page 5–19 for more information.

4. Double click the field you want to use to refine the selection criteria.

   **Results** displays the field in the **Selection Criteria** text box at the bottom of the screen.
5. Select the comparison operator you want to use to compare a value to the field. See Appendix B, “Comparison Operators,” for a description of these operators.

One of the following occurs:

- If you activated the Ask At Runtime toggle box, the Ask At Run Time Prompt dialog box appears:

![Ask At Run Time Prompt dialog](image)

Type a question that will prompt you to enter a value for the field when you open or run the query. For example, if you are comparing a value to the State field, you might type Enter a State. Then choose OK.

- If you did not activate the Ask At Runtime toggle box, the Enter Constant dialog box appears:

![Enter Constant dialog](image)

Type the value you want to compare to the field, then choose OK. Do not place quotation marks around the value. If quotation marks are required, Results will automatically insert the quotation marks for you when it formats the WHERE clause.

Results formats the WHERE clause and displays it in the Selection Criteria area.
6. Do one of the following:

- If the WHERE clause is complete, choose OK and redisplay the query based on your selection criteria.

- If you want to add another expression to the WHERE clause based on a field in the current table, determine whether you want to use the AND or the OR logical operator to join the next expression to the expression you just created. Then choose either AND or OR at the bottom of the dialog box.

The AND operator means both the combined expressions must be true for the record to be selected. If you use State = "MA" AND Balance > 0, Results selects only those records where the customer is from Massachusetts and the customer has a positive balance.

The OR operator means only one of the combined expressions must be true for the record to be selected. If you use State = "MA" OR Cust-Num < 50, Results selects all the records between 1 and 49 plus those records where the customer is from Massachusetts.

- If the expression you want to add to the WHERE clause is based on a field in a different table, choose OK to save the WHERE clause and redisplay the query based on your selection criteria. Then start with Step 1 in this procedure to create the expression.

When you open the table to create the new expression, the previous expression is not displayed in the Selection Criteria text box. Each table displays its own portion of the WHERE clause only. Results automatically uses the AND operator to join expressions from different tables together and create the WHERE clause for you.

7. Repeat Step 3 through Step 6 in this procedure to add an expression to the WHERE clause based on a field in the same table.
Using Data Selection to create a WHERE clause manually

Once you are familiar with automatically creating WHERE clauses using Data Selection, you can create and edit WHERE clauses manually.

To create a WHERE clause manually:

1. Choose Data→Selection.

   The Available Tables dialog box appears.

2. Select the table that contains the fields you want to use to limit the query, then choose OK.

   If you are creating a WHERE clause that contains more than one expression and these expressions contain fields from different tables, create the expressions in increments. Select the first table and create the first expression, then select another table to create the second expression, and so on until you complete the clause.

   The Data Selection dialog box appears.

3. Position the pointer in the Selection Criteria text box and click to place the insertion point in the upper-left corner of the editor.

4. Type the WHERE clause in the Selection Criteria editor. See Appendix B, “Comparison Operators,” for a description of the comparison operators you can use to compare values within the WHERE clause.
Follow these guidelines to create your WHERE clause:

- Use the complete name for each field in the WHERE clause. A complete field name consists of the database name, the table name, and the field name separated by periods (.). For example, type `sports.Customer.City` where sports is the name of the database, customer is the name of the table, and city is the name of the field.

- Enclose all text strings in quotation marks (" "). For example, `sports.customer.city BEGINS "M"`:

5. When you are done entering the WHERE clause, choose **Now** to check the syntax for the clause. An alert box appears.

6. Choose **Cancel**. If the syntax is incorrect, fix the error and recheck the syntax.

7. Choose **OK** to save the WHERE clause and close the **Data Selection** dialog box.
Setting Data Selection at run time

When you use Data Selection to create a WHERE clause, you can choose the Ask At Runtime option. Choosing Ask At Runtime means that instead of entering a comparison value when you create the WHERE clause, Results prompts you to enter the value when you run the query. This means that you can enter the value when you run the query in an application outside of Results or when you change to the Browse or Form view. When you use the Report, Labels, and Export views, Results does not prompt you to enter a value until you either print the query or use Print Preview.

To enter a value, type the value in the Ask At Runtime dialog box, then click OK. Note that you do not have to place quotation marks around the value. If the value requires quotation marks, Results automatically inserts them:

![Ask At Runtime dialog box]

To reset the value from within the Form and Browse views, choose Data→Re-Ask Questions, then type a value in the Ask At Runtime dialog box. To reset the value from within the Report, Labels, and Export view, simply reprint the query or use Print Preview.

Removing a WHERE clause created using Data Selection

Use the following procedure to remove a WHERE clause created using Data Selection.
To remove a WHERE clause that you created using Data Selection:

1. Choose Data → Selection.

2. Select the table that contains the expression you want to remove. The Data Selection dialog box appears:

3. Select the WHERE clause, then press DELETE to remove the clause.

4. Choose OK to save your changes and close the Data Selection dialog box.

5. If you created an expression for another table, repeat Steps 1 through 4 for every table that contains an expression.
Using Query by Example to refine a query

Query by Example is available in the Form view. Use Query by Example when you want to refine the selection criteria for the query by using one or more display fields in the form. If you already created a WHERE clause for the same table using Data Selection, Query by Example automatically deletes the WHERE clause and replaces it with the WHERE clause you create using Query by Example.

To use Query by Example:

1. Choose Query by Example at the bottom of the form. A blank form appears:

You can use any of these fields to modify the selection criteria for the query.

2. Go to the first field you want to use and type the selection criteria for the field.

The selection criteria for a field consists of an operator and the value you want the operator to work on. For example, to retrieve all the unpaid invoices, type > 0 in the Balance field.

You can use the following comparison operators: =, <>, <, >, <=, >=, Begins, Matches, and Contains. See Appendix B, “Comparison Operators,” for a description of these operators. If you do not enter a comparison operator in the selection criteria, Query by Example automatically uses the equality operator.

You do not have to place quotation marks ("") around text strings unless the first word in the text string happens to be the name of a comparison operator. For example, to select all the records that pertain to customers in Massachusetts, type “MA” in the State field.
3. If you want to further refine the selection criteria, enter the selection criteria for one or more additional fields. For example, to display unpaid invoices for the first 49 customers, type $\leq 49$ in the Cust-Num field and $> 0$ in the Balance field:

4. When you are finished defining the selection criteria, choose OK.

Results saves the selection criteria as a WHERE clause and retrieves the matching records. Note that once you create a WHERE clause, Results maintains the WHERE clause until you change or delete it.

**Removing a WHERE clause created using Query by Example**

There are two ways to remove a WHERE clause you create through Query by Example:

- Choose the Clear Criteria button using Query by Example.
- Use Data Selection to locate the WHERE clause, select it, and delete it.

**Note:** Although Data Selection displays WHERE clauses you create using Query by Example, Query by Example does not display WHERE clauses you create using Data Selection.
Using the Data Governor

The Data Governor is an option that limits the number of records a query displays. By default, when you run a query, Results displays all the records in the selected tables that meet the display and data selection criteria. However, the Data Governor allows you to save time and system resources by displaying only a specific number of records. For example, if you have 10,000 orders, you might want to limit the number shown to 30 while you develop and test the query. This allows Results to process 30 orders instead of 10,000 orders when you print or preview the query.

For example, if you run a query that displays the customer name and balance, Results displays all the customers in the selected tables. You can use the Data Governor to display only the first 20 records.

You can only use the Data Governor in the Report, Label, and Export views.

To use the Data Governor:

1. Choose Data→Governor. The Data Governor dialog box appears:

   ![Data Governor dialog box]

2. Enter the number of records you want to display.

3. Activate the Include in Generated Code toggle box if you want to include this limit in the generated code.

   This means that when you run the query from an application, Results displays only the number of records specified in the Number of Records field. If you do not activate this toggle box, Results displays all the records when you run the query from an application.

4. Choose OK.
Results provides a variety of techniques for analyzing the data in your queries. This chapter presents:

- Overview
- Using Master and Detail sections to display records
- Calculated fields
- Aggregate fields
- Using a Totals-Only Summary
Overview

Table 6–1 identifies and briefly describes each Results data analysis technique that is described in this chapter.

**Table 6–1: Results data analysis techniques**

<table>
<thead>
<tr>
<th>This data analysis technique...</th>
<th>Allows you to...</th>
<th>And, for more information, go to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master and Detail sections</td>
<td>Split the query view into Master and Detail sections to reduce the display of repetitive information.</td>
<td>“Using Master and Detail sections to display records” section on page 6–3</td>
</tr>
<tr>
<td>Calculated fields</td>
<td>Create calculated fields to calculate values based on constants and the values of other fields in the query.</td>
<td>“Calculated fields” section on page 6–6</td>
</tr>
<tr>
<td>Aggregate fields</td>
<td>Create aggregate totals and subtotals for one or more columns of fields within a report.</td>
<td>“Aggregate fields” section on page 6–37</td>
</tr>
<tr>
<td>Totals-Only Summary</td>
<td>Display data in break groups as totals only.</td>
<td>“Using a Totals-Only Summary” section on page 6–42</td>
</tr>
</tbody>
</table>
Using Master and Detail sections to display records

When you create a query that contains more than one table, Results automatically displays all the fields from the tables in one section. To help you analyze data, you can separate the information contained in these tables into two sections, called Master and Detail.

For example, if your query uses fields from the Customer, Order, and Order-Line tables to list the items each customer ordered, Results places the records in one section by default, as in the Figure 6–1.

In this example, the query retrieves a record for each item number within each order. Many of the records contain identical information. To reduce the amount of repetitive information displayed in the view, you can split the view into two sections, Master and Detail. A Master section begins a new order and its corresponding Detail section contains the specific items ordered. You can use the Master-Detail option with the Browse, Report, and Form query views.
Splitting composite records into **Master** and **Detail** sections does not alter the way in which the query retrieves the records. As the example query view shows in Figure 6–2, it simply alters the way in which Results displays the records within the query view.

**Figure 6–2:** Master and Detail sections in a query view

In Figure 6–2, the **Master** section contains fields from the Customer and Order tables and the **Detail** section contains fields from the Order-Line table.
To split composite records into Master and Detail sections:

1. Choose Options → Master-Detail. The Master-Detail dialog box appears:

2. Determine how you want to split the tables, then select a break point and choose OK.

When you split composite records into Master and Detail sections the Form or Browse views, Results automatically places each section in its own frame. See the “Browse view” section on page 4–2 and “Form view” section on page 4–7 for more information about frames.

Displaying composite records in one section

This section describes how to display composite records in one section.

To remove the Master and Detail sections and display composite records in one section:

1. Choose Options → Master-Detail.

2. Select <<one section>>, then choose OK.
Calculated fields

So far you have displayed fields that already exist in the database. However, you can create your own column of data, called *calculated fields*, to help you analyze the information in your query.

For example, you can create a field called **Total Price** that automatically multiplies the price of each item a customer orders by the number of items ordered, as shown in Figure 6–3.

![Calculated fields example](image-url)

**Figure 6–3:** Calculated fields example
Calculated fields are display fields only, so neither the field nor its contents are saved to the database. Table 6–2 shows the types of calculated fields you can create.

Table 6–2: Calculated fields

<table>
<thead>
<tr>
<th>Calculated field types</th>
<th>Description</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent of total</td>
<td>Expresses the percent of total that the value of a field represents within the query. For example, you can display the percent of total for each sales representative’s monthly sales figures.</td>
<td>Available in the Report and Export views only.</td>
</tr>
<tr>
<td>Running total</td>
<td>Displays the running total of any numeric field or expression. A running total increments for each line of the query. That is, the value displayed in that column on any line in the query is the sum of the field value for that and all previous lines in the query. For example, if the value in the first record is 1 and the value in the second record is 3, the running total for the first line is 1 and the running total for the second record is 4 (1 plus 3).</td>
<td>Available in the Report and Export views only.</td>
</tr>
<tr>
<td>Counter</td>
<td>Displays a running counter. Use counters for line numbering and counting records.</td>
<td>Available in the Report and Export views only.</td>
</tr>
<tr>
<td>Stacked array</td>
<td>Displays the elements in an array in a vertical column.</td>
<td>Available in the Report view only.</td>
</tr>
<tr>
<td>Lookup</td>
<td>Displays the value of a field from a table not included in the query.</td>
<td>Available in all the query views except Labels.</td>
</tr>
<tr>
<td>Math</td>
<td>Defines a math expression based on the value of one or more fields. Use the following operations to calculate the value of the two fields: add, subtract, multiply, divide, or raise to a power. For example, you can create a field named <strong>Total Price</strong> to multiply the value of the <strong>Price</strong> and <strong>Qty</strong> fields for each record in the query.</td>
<td>Available in all views.</td>
</tr>
<tr>
<td>String function</td>
<td>Defines a display field based on a string expression.</td>
<td>Available in all views.</td>
</tr>
</tbody>
</table>
Creating calculated fields

You can create calculated fields from within any query view except the Labels view. See Table 6–2 for a list of the views in which individual calculated fields are available.

To create a calculated field:

1. Choose Field→Add Calculated Field, then select the field you want to create.

   Results displays the Add Field dialog box for the type of calculated field you selected. The following sections describe these dialog boxes.

2. Use the Add Field dialog box to specify the parameters for the field, then choose OK. The Properties dialog box appears.

   For example, you can choose Assistant to modify how the data in the field is displayed. You can also change the name of a calculated field by selecting Rename. See the “Renaming calculated fields” section on page 6–36 for more information.

3. Set the properties for the field you created. See the “Changing display field properties” section on page 7–3 for more information.

### Table 6–2: Calculated fields

<table>
<thead>
<tr>
<th>Calculated field types</th>
<th>Description</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeric function</td>
<td>Defines a display field based on a numeric expression.</td>
<td>Available in all views.</td>
</tr>
<tr>
<td>Date function</td>
<td>Defines a display field based on a date calculation.</td>
<td>Available in all views.</td>
</tr>
<tr>
<td>Logical function</td>
<td>Defines a logical display field based on the value of one or more fields in the query.</td>
<td>Available in all views.</td>
</tr>
</tbody>
</table>
Adding a Percent of Total field

The Add Field - Percent of Total dialog box lets you specify the field for which you want to calculate the percent of total.

To specify a field, select the field and choose OK:

For example, Figure 6–4 uses Percent of Total to calculate each sales representative’s percent of total for the month of January.

Figure 6–4: Percent of Total calculation example
Adding a Running Total field

The **Add Field - Running Total** dialog box lets you select the field for which you want to calculate a running total.

To specify a field, select the field and choose **OK**:

Figure 6–5 lists the running total for the **Total-Paid** field.

Figure 6–5: Total-Paid field example
Adding a Counter

The Add Field - Counter dialog box lets you specify the parameters for the Counter field:

To create the Counter field, set the following parameters:

- **Starting number** — Specifies the number from which to begin counting.
- **Number to add** — Specifies the increment between the numbers.
- **Sections used in** — Select the sections you want to count: Master, Detail, or both. See “Using Master and Detail sections to display records” section on page 6–3 for more information about Master and Detail sections.
The example presented in Figure 6–6 uses the **Counter** field to count the total number of orders in the **Master** section.

![Counter field example](image)

**Figure 6–6: Counter field example**

**Adding a Stacked Array field**

When a field consists of several elements, each of which contains its own value, the field is called an *array*. By default, the values in the array are listed horizontally. To list them in a vertical column, use the **Stacked Array Calculated** field.

The **Add Field - Stacked Array** dialog box lets you specify the array you want to stack vertically.
To specify the array, select the array and choose **OK**:

In **Figure 6–7**, the **Stacked Array** field displays the contents for **Month-Quota** in a vertical column.

**Figure 6–7:** Stacked Array field example
Adding a Lookup field

The **Add Field - Lookup Source** dialog box allows you to display the value of a field in a table not currently in the query.

To use this option, both the table that contains the field you want to display and one of the tables in your query must contain a field that stores the same kind of information. This similarity between the two fields lets you establish a relationship between the two tables.

As shown in Figure 6–8, you might use a field in the Customer table called **Cust-Num** and a field in the Invoice table called **ID-Num** to identify your customers. In this case, they both store a set of numbers that identify your customers.

![Figure 6–8: Adding a Lookup field example](image)

See the “Record displays in multi-table queries” section on page 3–8 and the “Changing table relationships in multi-table queries” section on page 5–4 for more information about table relationships.
To look up a field in a table not included in the query:

1. Select a field from one of the tables in your query that contains the same type of information as a field in the table with which you are establishing a relationship.

2. Choose OK. The Add Field - Lookup Matching Field dialog box appears:

3. Select the table that contains the field you want to look up.

4. Select a field that contains the same type of information as the field you selected in Step 1. This establishes a relationship between the two tables.
5. Choose **OK**. The **Add Field - Lookup Display Field** dialog box appears:

6. Select the field that contains the information you want to display.

7. Type a value to display in place of the lookup field when Results cannot find a matching field value in one or more records. By default, Results displays a question mark (?) when it cannot find a value.

8. Choose **OK**.

The following example displays a lookup field called **Region** from a table called **State** that is not included in the query:
Adding a Math expression

The Add Field - Math dialog box allows you to create a math expression based on the value of two fields. When you select the Math option, Results displays the Add Field - Math dialog box:

To create a Math expression:

1. Select the mathematical type of function to perform. Results lists the available display fields in the Function/Field selection list and displays an empty expression based on the function you choose:
2. Double click one of the following to create the first part of the expression:
   - **A field** — Uses the value of the field in the first part of the expression.
   - **Constant value** — Lets you enter a value to use in the first part of the expression. For example, you can use constant value field to insert a number to calculate within the expression.
   - **Subexpression** — Lets you enter a subexpression as the first value in the expression. A subexpression is an expression within another expression.

Results creates the first part of the expression and displays it in the **Expression** field at the bottom of the dialog box.

3. If the expression contains a second value, double click a field, constant value, or subexpression to create the last part of the expression.

Results creates the remaining part of the expression and displays the entire expression in the **Expression** selection list:
4. Choose **OK** to save the expression.

The following example contains a calculated field named **Total Price** that multiplies the value of the **Price** and **Qty** fields for each record in the query:
Adding a String Function

The Add Field - String Function dialog box allows you to create a string expression:

1. In the Function/Field selection list, double click the function you want to perform. Table 6–3 describes these functions.

Table 6–3: String functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combine strings</td>
<td>Combines two or more strings into one string.</td>
</tr>
<tr>
<td>Current time</td>
<td>Returns the current time.</td>
</tr>
<tr>
<td>Display as time</td>
<td>Displays a string in time format.</td>
</tr>
<tr>
<td>Greater of two strings</td>
<td>Returns the greater (larger) of two strings.</td>
</tr>
<tr>
<td>If first expression is true return second else third string</td>
<td>Defines an expression that makes a decision based on string values. If the first string expression is true, the second expression is returned. If the first string expression is false, the third expression is returned.</td>
</tr>
</tbody>
</table>
Table 6–3: String functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length of string</td>
<td>Returns the number of characters in a string. For example, the length of the string “Progress” is 8.</td>
</tr>
<tr>
<td>Lesser of two strings</td>
<td>Returns the lesser (smaller) of two strings.</td>
</tr>
<tr>
<td>Name of month</td>
<td>Returns the current month.</td>
</tr>
<tr>
<td>Name of weekday</td>
<td>Returns the current day.</td>
</tr>
<tr>
<td>String constant or field</td>
<td>Lets you build an expression with either a string constant or the value of a character field.</td>
</tr>
<tr>
<td>Substring</td>
<td>Returns a substring. A substring is part of a string. For example, “hold” is a substring of “credit hold.”</td>
</tr>
<tr>
<td>User ID</td>
<td>Returns the user ID.</td>
</tr>
</tbody>
</table>

Results does the following:

- Lists the available display fields in the Function/Field selection list.
- Displays an empty expression based on the function you chose in Step 1.

2. Double click one of the following to create the first part of the expression:

- **A field** — Uses the value of the field in the first part of the expression.
- **Constant value** — Lets you enter a value to use in the first part of the expression. For example, you could use constant value to insert a comma or a space within a string.
- **Subexpression** — Lets you enter a subexpression as the first value in the expression. A subexpression is simply an expression within another expression.

Results creates the first part of the expression and displays it in the Expression field at the bottom of the dialog box.
3. If your expression contains a second value, double click a field, constant value, or subexpression to create the last part of the expression.

Results creates the remaining part of the expression and displays the entire expression in the Expression field:

![Expression field](image)

4. Choose **OK** to save the expression.

The following example shows the results. It contains a calculated field named **City/State** that uses the Combine String function to combine the values of the **City** and **State** fields into one field:

![Combined field example](image)
Note that the City/State field uses a constant value that contains a comma and space to separate the city and state. This constant value is added to the expression through a subexpression. Figure 6–9 illustrates the expression used in this example.

\[
\text{string (city)} + \text{string(constant value)} + \text{string(state)}
\]

**Figure 6–9: Expression example**

**Adding a Numeric Function**

Use the **Numeric Function** to create a numeric expression. When you select the **Numeric Function** option, Results displays the **Add Field - Numeric Function** dialog box:
To create a numeric expression:

1. In the Function/Field selection list, double click the function you want to perform. Table 6–4 describes these functions.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute value</td>
<td>Returns the absolute value of a number. The absolute value is the positive</td>
</tr>
<tr>
<td></td>
<td>value of a given number. For example, the absolute value of -3 is 3; the</td>
</tr>
<tr>
<td></td>
<td>absolute value of 5 is 5.</td>
</tr>
<tr>
<td>Cube root</td>
<td>Returns the cube root of a number. The cube root of a number is a number</td>
</tr>
<tr>
<td></td>
<td>that when multiplied by itself three times, equals the number. For example,</td>
</tr>
<tr>
<td></td>
<td>the cube root of 8 is 2 (2<em>2</em>2).</td>
</tr>
<tr>
<td>Greater of two numbers</td>
<td>Returns the largest number from a list of numbers. For example, if a list</td>
</tr>
<tr>
<td></td>
<td>consists of two numbers, 24 and 55, the function returns 55.</td>
</tr>
<tr>
<td>If first expression is</td>
<td>Defines an expression that makes a decision. If the first expression is</td>
</tr>
<tr>
<td>true return second else</td>
<td>true, the second expression is returned. If the first expression is false,</td>
</tr>
<tr>
<td>third number</td>
<td>the third expression is returned.</td>
</tr>
<tr>
<td>Log base e</td>
<td>Returns the logarithm based on the value e. The value of e is 2.7182818.</td>
</tr>
<tr>
<td>Log base n</td>
<td>Returns the natural logarithm for a number for a given base. For example,</td>
</tr>
<tr>
<td></td>
<td>2^5 is 32. The natural logarithm for 32 with base 2 is 5.</td>
</tr>
<tr>
<td>Numeric constant or field</td>
<td>Defines a field’s value</td>
</tr>
<tr>
<td>Remainder</td>
<td>Returns the remainder of a division operation.</td>
</tr>
<tr>
<td>Round</td>
<td>Rounds a value to the nearest whole number. For example, rounding 5.2</td>
</tr>
<tr>
<td></td>
<td>returns 5 and rounding 5.7 returns 6.</td>
</tr>
</tbody>
</table>
Table 6–4: Numeric functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smaller of two numbers</td>
<td>Returns the smaller of two numbers. For example, if the function compares the numbers 18 and 29, it returns the number 18.</td>
</tr>
<tr>
<td>Square root</td>
<td>Returns the square root of a number. For example, the square root of 9 is 3.</td>
</tr>
<tr>
<td>Truncate</td>
<td>Truncates a number.</td>
</tr>
</tbody>
</table>

Results does the following:

- Lists the available display fields in the Function/Field selection list.
- Displays an empty expression based on the function you chose in Step 1 within the Expression scroll box.

2. Double click one of the following to create the first part of the expression:

- A field — Uses the value of the field in the first part of the expression.
- Constant value — Lets you enter a value to use in the first part of the expression. For example, you can enter a number to calculate within the expression.
- Subexpression — Lets you enter a subexpression as the first value in the expression. A subexpression is simply an expression within an expression.

Results creates the first part of the expression and displays it in the Expression field at the bottom of the dialog box.
3. If the expression contains a second value, double-click a field, constant value, or subexpression to create the last part of the expression. Results creates the remaining part of the expression and displays the entire expression in the **Expression** field:

![Expression field example](image)

4. Choose **OK** to save your expression.

Figure 6–10 contains a calculated field named **Round Amount** that rounds the amount of each customer’s invoice to the nearest whole dollar.

![Invoice data](image)

**Figure 6–10:** Round Amount field example
Adding a Date Function

The Add Field - Date Function dialog box, shown in Figure 6–11, allows you to define a date calculation.

![Add Field - Date Function dialog box](image)

Figure 6–11: Add Field - Date Function dialog box

To add a Date Function:

1. In the Function/Field selection list, double click the function you want to perform. Table 6–5 describes these functions.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add days to date value</td>
<td>Adds the number of days you specify to a date and returns the value.</td>
</tr>
<tr>
<td>Current date</td>
<td>Returns the current date.</td>
</tr>
<tr>
<td>Day of month</td>
<td>Returns the day of the month for the date you specify.</td>
</tr>
<tr>
<td>Day of week</td>
<td>Returns the day of the week for the date you specify in numeric format, where the value “1” begins with Sunday.</td>
</tr>
</tbody>
</table>

Table 6–5: Date functions (1 of 2)
### Table 6–5: Date functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Difference between two dates</strong></td>
<td>Returns the difference between two dates.</td>
</tr>
<tr>
<td><strong>Earlier of two dates</strong></td>
<td>Returns the earlier of two dates. For example, if the function compares the dates 01/05/95 and 01/09/99, it returns 01/05/95.</td>
</tr>
<tr>
<td><strong>If first expression is true return second else third date</strong></td>
<td>Defines an expression that makes a decision. If the first part of the expression is true, the function returns the second date; if the first part of the expression is not true, the function returns the third date.</td>
</tr>
<tr>
<td><strong>Later of two dates</strong></td>
<td>Returns the later of two dates. For example, if the function compares the dates 01/05/95 and 01/09/99, it returns 01/09/99.</td>
</tr>
<tr>
<td><strong>Month of year</strong></td>
<td>Returns the number of the month for the date you specify.</td>
</tr>
<tr>
<td><strong>Subtract days from date value</strong></td>
<td>Subtracts the number of days you specify from a date and returns the value.</td>
</tr>
<tr>
<td><strong>Year value</strong></td>
<td>Returns the year for the date you specify. For example, the date 04/05/99 is returned as 1999.</td>
</tr>
</tbody>
</table>
Results does the following:

- Lists the available display fields in the **Function/Field** selection list.
- Displays an empty expression based on the function you chose in Step 1 within the **Expression** editor box:

2. Double click one of the following to create the first part of the expression:

- **A field** — Uses the value of the field in the first part of the expression.
- **Constant value** — Lets you enter a date to use in the first part of the expression.
- **Current date** — Uses the current date each time you run the query.
- **Subexpression** — Lets you enter subexpression as the first value in the expression.

Results creates the first part of the expression and displays it in the **Expression** field at the bottom of the dialog box.
3. Double click a field, constant value, current date, or subexpression to create the last part of the expression.

Results creates the remaining part of the expression and displays the entire expression in the Expression field:

4. Choose OK to save your expression.

The following example contains a calculated field named Delivery Time that uses the Differences Between Two Dates function to calculate the difference between the Ordered and Shipped dates. Note that a question mark displays in the Delivery Time field for those records that Results cannot calculate because the order has not shipped:
Adding a Logical Function

The **Add Field - Logical Function** dialog box allows you to create a logical expression:

To create a logical expression:

1. In the **Function/Field** selection list, double click the function you want to perform. Table 6–6 describes these functions.

   **Note:** Many of the functions in Table 6–6 contain comparison operators. See Appendix B, “Comparison Operators.”

<table>
<thead>
<tr>
<th>Table 6–6: Logical functions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Function</strong></td>
</tr>
<tr>
<td>Begins</td>
</tr>
<tr>
<td>Dates equal</td>
</tr>
</tbody>
</table>
### Logical functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dates greater or equal</td>
<td>Returns the logical value “yes” for dates larger or equal to the value you specify or the logical value “no” for dates smaller than the value you specify.</td>
</tr>
<tr>
<td>Dates greater than</td>
<td>Returns the logical value “yes” for dates larger than the value you specify or the logical value “no” for dates smaller or equal to the value you specify.</td>
</tr>
<tr>
<td>Dates less or equal</td>
<td>Returns the logical value “yes” for dates smaller or equal to the value you specify or the logical value “no” for dates larger than the value you specify.</td>
</tr>
<tr>
<td>Dates less than</td>
<td>Returns the logical value “yes” for dates smaller than the value you specify or the logical value “no” for dates larger or equal to the value you specify.</td>
</tr>
<tr>
<td>Dates not equal</td>
<td>Returns the logical value “yes” for dates not equal to the value you specify or the logical value “no” for dates equal to the value you specify.</td>
</tr>
<tr>
<td>Logical AND</td>
<td>Lets you join two logical expressions together. The function returns the value “yes” when both expressions are true and the value “no” when either one or both expressions are false.</td>
</tr>
<tr>
<td>Logical OR</td>
<td>Lets you join two logical expressions together. The function returns the value “yes” when either expression is true and the value “no” when neither expression is true.</td>
</tr>
<tr>
<td>Matches</td>
<td>Returns the logical value “yes” for strings that match the characters you specify or the value “no” for strings that do not match the characters you specify.</td>
</tr>
<tr>
<td>Negate logical expression (NOT)</td>
<td>Returns the logical value “no” for logical expressions that are true and the logical value “yes” for logical expressions that are false.</td>
</tr>
<tr>
<td>Numbers equal</td>
<td>Returns the logical value “yes” for numbers equal to the value you specify or the logical value “no” for numbers not equal to the value you specify.</td>
</tr>
<tr>
<td>Numbers greater or equal</td>
<td>Returns the logical value “yes” for numbers larger or equal to the value you specify or the logical value “no” for numbers smaller than the value you specify.</td>
</tr>
</tbody>
</table>
## Logical functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numbers greater than</td>
<td>Returns the logical value “yes” for numbers larger than the value you specify or the logical value “no” for numbers smaller or equal to the value you specify.</td>
</tr>
<tr>
<td>Numbers less or equal</td>
<td>Returns the logical value “yes” for numbers smaller or equal to the value you specify or the logical value “no” for numbers larger than the value you specify.</td>
</tr>
<tr>
<td>Numbers less than</td>
<td>Returns the logical value “yes” for numbers smaller than the value you specify or the logical value “no” for numbers larger or equal to the value you specify.</td>
</tr>
<tr>
<td>Numbers not equal</td>
<td>Returns the logical value “yes” for numbers not equal to the value you specify or the logical value “no” for numbers equal to the value you specify.</td>
</tr>
<tr>
<td>Strings equal</td>
<td>Returns the logical value “yes” for strings equal to the value you specify or the logical value “no” for strings not equal to the value you specify.</td>
</tr>
<tr>
<td>Strings greater or equal</td>
<td>Returns the logical value “yes” for strings larger or equal to the value you specify or the logical value “no” for strings smaller than the value you specify.</td>
</tr>
<tr>
<td>Strings greater than</td>
<td>Returns the logical value “yes” for strings larger than the value you specify or the logical value “no” for strings smaller or equal to the value you specify.</td>
</tr>
<tr>
<td>Strings less or equal</td>
<td>Returns the logical value “yes” for strings smaller or equal to the value you specify or the logical value “no” for strings larger than the value you specify.</td>
</tr>
<tr>
<td>Strings less than</td>
<td>Returns the logical value “yes” for strings smaller than the value you specify or the logical value “no” for strings larger or equal to the value you specify.</td>
</tr>
<tr>
<td>Strings not equal</td>
<td>Returns the logical value “yes” for strings not equal to the value you specify or the logical value “no” for strings equal to the value you specify.</td>
</tr>
</tbody>
</table>
2. Double click one of the following to create the first part of the expression:

- **A field** — Uses the value of the field in the first part of the expression.
- **Constant value** — Lets you enter a date to use in the first part of the expression.
- **Current date** — Uses the current date each time you run the query.
- **Subexpression** — Lets you enter subexpression as the first value in the expression. A subexpression is an expression within an expression.

Results creates the first part of the expression and displays it in the **Expression** field at the bottom of the dialog box.

3. If your expression contains a second value, double click a field, constant value, current date, or subexpression to create the last part of the expression.

Results creates the remaining part of the expression and displays the entire expression in the **Expression** field:

4. Choose **OK** to save your expression.
5. By default, Results uses the display values \textit{yes} and \textit{no} to represent true and false. To substitute a different set of display values for true and false, choose \textbf{Assistant} in the \textbf{Properties} dialog box and modify the display values. See “Changing display field properties” section on page 7–3 for more information.

The following example contains a calculated field named \textbf{On Time} that uses the Dates \textbf{Less or Equal} function to determine whether the date an order is shipped is equal to or less than the promise date for the order:
Renaming calculated fields

When you create a calculated field, Results automatically names the field with the prefix `qbf-` followed by a unique number. For example, Results might name a field `qbf-004`. While `qbf-004` might be a logical name to Results, it is not intuitive. Therefore, you can rename a calculated field with a more descriptive name. For example, if `qbf-004` calculates the number of days between the current date and the ship date, rename it `Days_To_Ship`.

To rename a calculated field:

1. Choose Field → Properties. The Properties dialog box appears:

![Properties dialog box]

2. Select the calculated field whose name you want to change from the Name drop-down list.

3. Choose Rename. The Rename Field dialog box appears:

![Rename Field dialog box]
4. Enter the new calculated field name, then choose OK to rename the field and return to the Properties dialog box.

The field name can be up to 32 characters long and consist of alphabetic characters (A-Z or a-z), digits (0-9), and special characters ($, & , #, -, and _). In addition, field names must begin with a letter (a-z or a-z). The field name must also be unique within the query and cannot contain spaces.

5. Choose OK.

Removing calculated fields

Use the same procedure to remove a calculated field that you use to remove a display field. See “Adding and removing display fields” section on page 7–2 for more information.

To remove a calculated field:

2. Select the calculated field you want to remove from the Selected Fields selection list, then choose Remove.
3. Choose OK to save your changes.

Aggregate fields

An aggregate field calculates the value of a column of data. For example, you can use an aggregate value to count the total number of customer orders in a report. Results provides the following aggregate values:

- **Total** — The combined value for the column of data.
- **Count** — The number of rows of data in the column.
- **Minimum** — The smallest value in the column of data.
- **Maximum** — The largest value in the column of data.
- **Average** — The average value of all the data in the column.
Use aggregate values to calculate the value of all the data in the column or to calculate the value of one or more groups of data within the column, called break groups. When you calculate the values of all the data in a column, you create an aggregate called a total; when you calculate the value of a break group, you create a subtotal. The following sections describe totals and subtotals.

**Totals**

When you create a total for a particular field, Results calculates the value for the entire column and places it in the summary area beneath the column at the bottom of the last page of the report. The following example shows a report that calculates the number of customer orders and the average price of the items ordered:

Counts the number of customer orders and places the result in the summary section of the report. Calculates the average price of all the items ordered and places the result in the summary section of the report.
Subtotals

If you use a sort order to create break groups within the report, you can calculate the subtotal for each break group. See “Sorting records” section on page 5–7 for information about sorting with break groups.”

For example, a report that is sorted in ascending order by Cust-Num and descending order by Order-Num, contains two break groups: one for customers and one for orders. You can use either the break group associated with Cust-Num or the break group associated with Order-Num to create subtotals.

Figure 6–12 shows a report that contains a subtotal associated with the Order-Num break group to calculate the total price for all the items in each order.
The report presented in Figure 6–13 contains an additional subtotal based on the **Cust-Num** break group to calculate the total price of all the items each customer ordered. This subtotal includes all the items in all the orders for each customer.

As shown in this example, you can use any combination of subtotals for each break group.

You can create an aggregate field based on any database field and on the following types of calculated fields: **Math, String Function, Numeric function, Date Function, and Logical Function.**

### Figure 6–13: Subtotals - example two

<table>
<thead>
<tr>
<th>Item</th>
<th>Cust-Num</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 lift Line Skiing</td>
<td>00031</td>
<td>1,205.50</td>
</tr>
<tr>
<td>1 lift Line Skiing</td>
<td>00084</td>
<td>4.32</td>
</tr>
<tr>
<td>1 lift Line Skiing</td>
<td>00039</td>
<td>508.30</td>
</tr>
<tr>
<td>1 lift Line Skiing</td>
<td>00046</td>
<td>19,760.66</td>
</tr>
<tr>
<td>1 lift Line Skiing</td>
<td>00046</td>
<td>14,215.46</td>
</tr>
</tbody>
</table>

**Total for order 79:** 26,783.84

**Total for order 36:** 492.25

**Total for order 6:** 764.40

**Total of orders for customer 1:** 1,827.50

**Total of orders for customer 1:** 45,569.87
Creating aggregate totals and subtotals

This section describes how to create one or more aggregate totals and subtotals using Results.

To create one or more aggregate totals and subtotals:

1. Choose Field→ Aggregates. The Field Aggregates dialog box appears:

2. Create a total or subtotal as follows:

   • To create a total, select the field for which you want to calculate the total, select summary line from the Break-By Fields selection list, and select the aggregate value you want to create.

     For example, to count all the orders in the report, select Order-Num from the Query Fields selection list, select summary line from the Break-By Fields selection list, then select Count.

   • To create a subtotal, select the field for which you want to calculate the subtotal, select the sort field you want to use from the Break-By Fields selection list, and select the aggregate value you want to create.

     For example, to calculate the total price of the items each customer ordered, select the Price field from the Query Fields selection list, select the Cust-Num field from the Break-By Fields selection list, then select Total.
3. Repeat Step 2 to create additional aggregates.

4. When you are finished creating aggregates, choose OK.

Using a Totals-Only Summary

In the **Report** and **Export** views, **Totals-Only Summary** provides a quick way to calculate and display the combined value of a column of data for a specific break group. See “Sorting records” section on page 5–7 for more information.

Figure 6–14 shows a query that does not use **Totals Only Summary**. In this example, the query displays the orders for each customer. The query uses the **Cust-Num** field to sort the orders for each customer in ascending order. Thus, it contains one break group—a break group for customers.

![Figure 6–14: Sort order by Cust-num field - example one](image-url)
As Figure 6–15 shows, the query contains a break group based on the *Cust-Num* field. Therefore, you can use **Totals Only Summary** to calculate and display the combined value of the *Extended-Price* field for each set of customer orders.

**Figure 6–15:** Sort order by Totals Only Summary - example two

When you use **Totals Only Summary**, Results summarizes the information and displays one record for each group containing the total for the group. Note that when you use **Totals Only Summary** the Export view, Results displays the preceding records in the group in addition to the record containing the total.
To use Totals Only Summary:

1. Verify that you have at least one break group. If you have more than one break group, **Total Only Summary** automatically uses the break group based on the last field in your sort list to perform the total.

2. Choose **Options** → **Totals Only Summary**. The **Totals Only Summary** dialog box appears:

   ![Totals Only Summary dialog box](image)

   3. Select one or more fields to total, then choose **OK**.
Modifying Display Fields

This chapter describes techniques to modify display fields. Specifically, this chapter presents:

- Adding and removing display fields
- Changing display field properties
Adding and removing display fields

The Add/Remove Fields option lets you change the fields displayed in the Query view.

To add and remove display fields from the Query view:

1. Choose Fields→ Add/Remove Fields. Results displays the Add/Remove Fields dialog box:

2. Add and remove fields as follows:
   - To add a field to the query layout, select the field from the Available Fields selection list and choose Add or double click on the desired field.
   - To remove a field from the query layout, select the field from the Selected Fields selection list and choose Remove.

3. The fields are listed in the Selected Fields selection list in the order they will be displayed in the query layout (from left to right). To reorder the fields, select a field and choose either Move Up or Move Down. Repeat this process until the fields are listed in the order you want them to appear in the query layout.

4. Choose OK to save your modifications to the query layout.
Changing display field properties

Field properties determine the way the display fields present data on the screen and in printed reports. Table 7–1 describes these properties.

Table 7–1: Field properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>Determines the display label assigned to the field. Changing the Display label for the field does not alter the name of the field stored in the database.</td>
</tr>
<tr>
<td>Format</td>
<td>Determines how the data stored in the field is displayed. For example, display formats indicate how many spaces Results should put in the display, dollar signs, decimal points, or other special characters.</td>
</tr>
<tr>
<td>Hide Repeating Values</td>
<td>Determines whether Results displays repeating values for the fields. (Results displays repeating values for the fields by default.)</td>
</tr>
<tr>
<td>Row</td>
<td>Determines the coordinate location for fields in reports and forms.</td>
</tr>
<tr>
<td>Column</td>
<td>Determines the coordinate location for the field in reports and forms.</td>
</tr>
</tbody>
</table>
To change the properties for one or more display fields:

1. Choose **Field → Properties**. The **Properties** dialog box appears:

   ![Properties dialog box](image)

2. Click the down arrow next to the name field to show the list of display fields included in the query, then select the display field you want to modify.

3. Change the format, label, and repeating values attributes for the field as follows:

   - To change the display format for the field, either type a new format in the **Format** field or choose **Assistant**.
     
     If you choose **Assistant**, Results displays the **Assistant** dialog box for the type of field you are modifying. These types include character, date, decimal, integer, and logical. See “Changing the format for a display field” section on page 7–5 for more information about changing display field formats.
   
   - To change the display label, select the existing label and type a new label. The new label overwrites the old label.
   
   - Activate the **Hide Repeating Values** toggle box.
   
   - To change the coordinates where a field is displayed in reports and forms, enter a numeric value greater than 1 in the **Row and Column** fields. The numbers you enter determine the number of rows and columns the field is offset from its current position. One column is equal to one character space and one row is equal to one line, where the line is one character high.
4. When you are done changing display properties, choose OK in the Properties dialog box to save your changes.

Changing the format for a display field

When you choose Assistant in the Properties dialog box, Results displays the Format dialog box for the type of display field you selected. The following sections describe how to change the formats for these field types.

**Note:** Any change you make to the display format modifies how the field is displayed. Changing the display format does not alter how the field is stored in the database.

Changing the format for a character field

Figure 7–1 shows the Format dialog box you use to edit character fields.

![Figure 7–1: Format dialog box](image)

The format for the field in this example is \( X(20) \) where \( X \) represents the symbol type for the field and \( (20) \) the maximum length of the field. The symbol \( X \) means that the field can display any type of character. You can modify both the symbol type and maximum length of the field.
To modify the maximum number of characters the field can display, type a number in the Width field. To modify the symbol type, choose one of the radio buttons listed in Table 7–2, then choose OK.

Table 7–2: Character field properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow any character</td>
<td>Displays any type of character, including blank spaces.</td>
</tr>
<tr>
<td>Allow letters and numbers only</td>
<td>Displays only letters and numbers. Blank spaces are not allowed.</td>
</tr>
<tr>
<td>Allow letters only</td>
<td>Displays letters only. Blank spaces are not allowed.</td>
</tr>
<tr>
<td>Allow letters and convert to caps</td>
<td>Displays capital letters only. Blank spaces are not allowed.</td>
</tr>
<tr>
<td>Allow numbers only</td>
<td>Displays numbers only. Blank spaces are not allowed.</td>
</tr>
</tbody>
</table>

Changing the format for a date field

Use the Format dialog box to edit date fields:
Modifying Display Fields

The **Format** field displays the current format for the field. In this example, the format for the field is **99/99/99**. To change the display format for the field, choose one of the radio buttons listed in Table 7–3, then choose OK.

### Table 7–3: Date field properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>99/99/99</td>
<td>Displays the date in mm/dd/yy format.</td>
</tr>
<tr>
<td>99.99.99</td>
<td>Displays the date in mm-dd-yy format.</td>
</tr>
<tr>
<td>99.99.99</td>
<td>Displays the date in mm.dd.yy format.</td>
</tr>
<tr>
<td>99/99/9999</td>
<td>Displays the date in mm/dd/yyyy format.</td>
</tr>
<tr>
<td>99-99-9999</td>
<td>Displays the date in mm-dd-yyyy format.</td>
</tr>
<tr>
<td>99.99.9999</td>
<td>Displays the date in mm.dd.yyyy format.</td>
</tr>
<tr>
<td>999999</td>
<td>Displays the date in mmmddyy format.</td>
</tr>
<tr>
<td>99999999</td>
<td>Displays the date in mmmddyyyy format.</td>
</tr>
</tbody>
</table>

### Changing the format for a decimal or integer field

Use the **Format** dialog box to edit decimal and integer fields:

![Format dialog box](image)

---

7–7
The format for the field in this example is ->,>>,>>9.99, where the dash (-) places a minus sign in front of the number if the number is negative, or a blank in front of the number if the number is positive; the greater than signs (>) are replaced with a digit if that digit is not a leading zero, 9 is replaced with a digit, and the period (.) represents a decimal point.

To modify the default display format for the field, change one or more attributes listed in Table 7–4, then choose OK.

Table 7–4: Decimal and integer field properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of digits</td>
<td>Specifies the maximum number of characters the field can display.</td>
</tr>
<tr>
<td>Number of decimal places</td>
<td>Specifies the number of decimal places the field can display.</td>
</tr>
<tr>
<td>Use thousand’s separator &quot;&quot;,&quot;</td>
<td>Displays a comma between the thousands and hundreds columns.</td>
</tr>
<tr>
<td>Suppress leading zeros</td>
<td>Does not display leading zeros.</td>
</tr>
<tr>
<td>Show as zero</td>
<td>Displays leading zeros as zeros.</td>
</tr>
<tr>
<td>Show as asterisk</td>
<td>Displays leading zeros as asterisks (*).</td>
</tr>
<tr>
<td>Leading text string:</td>
<td>Specifies the text to display before the number.</td>
</tr>
<tr>
<td>Trailing text string:</td>
<td>Specifies the text to display after the number.</td>
</tr>
<tr>
<td>Show leading sign always</td>
<td>Always displays a positive (+) or negative (-) sign before the number.</td>
</tr>
<tr>
<td>Show leading sign on negatives only</td>
<td>Displays a negative (-) sign in front of negative numbers only.</td>
</tr>
<tr>
<td>Show trailing sign always</td>
<td>Always displays a positive (+) or negative (-) sign after the number.</td>
</tr>
<tr>
<td>Show trailing sign on negatives only</td>
<td>Displays a negative (-) sign after negative numbers only.</td>
</tr>
<tr>
<td>Show negative numbers in parenthesis</td>
<td>Displays negative numbers within a set of parentheses (()).</td>
</tr>
</tbody>
</table>
Changing the format for a logical field

Figure 7–2 presents the Format dialog box you use to edit logical fields.

Table 7–4:  Decimal and integer field properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show negatives with trailing 'CR'</td>
<td>Displays the credit symbol (CR) after negative numbers.</td>
</tr>
<tr>
<td>Show negatives with trailing 'DB'</td>
<td>Displays the debit symbol (DB) after negative numbers.</td>
</tr>
<tr>
<td>Show negatives with trailing 'DR'</td>
<td>Displays the debit symbol (DR) after negative numbers.</td>
</tr>
<tr>
<td>Show negatives with trailing 'cr'</td>
<td>Displays the credit symbol (cr) after negative numbers.</td>
</tr>
<tr>
<td>Show negatives with trailing 'db'</td>
<td>Displays the debit symbol (db) after negative numbers.</td>
</tr>
<tr>
<td>Show negatives with trailing 'dr'</td>
<td>Displays the debit symbol (dr) after negative numbers.</td>
</tr>
<tr>
<td>Allow positive numbers only</td>
<td>Displays positive numbers only.</td>
</tr>
</tbody>
</table>

Logical fields can have one of two possible values: TRUE or FALSE. The Format field shows the terms used to display these values. To change these terms, type a new term in the Display This When TRUE and Display This When FALSE fields, then choose OK.
This appendix describes the standard menu options. Specifically, this chapter presents:

- Query menu
- Table menu
- Field menu
- Data menu
- Options menu
- View menu
- Help menu

**Note:** If your application has been modified, the menu options you have might not be the same as those discussed in this appendix.
Query menu

The Query menu manipulates queries and exits Results. Table A–1 describes the Query menu.

Table A–1: Query menu

<table>
<thead>
<tr>
<th>Query menu option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>Creates a new query.</td>
</tr>
<tr>
<td>Open</td>
<td>Opens a query.</td>
</tr>
<tr>
<td>Save</td>
<td>Saves a query.</td>
</tr>
<tr>
<td>Save As...</td>
<td>Names and saves a query.</td>
</tr>
<tr>
<td>Close</td>
<td>Closes a query without exiting Results.</td>
</tr>
<tr>
<td>Delete...</td>
<td>Deletes one or more queries.</td>
</tr>
<tr>
<td>Generate...</td>
<td>Generates 4GL code for the query. (Might be available to the system administrator only.)</td>
</tr>
<tr>
<td>Print</td>
<td>Prints a query. You can also use the Print option to save the query to the Clipboard or to a file.</td>
</tr>
<tr>
<td>Print Preview</td>
<td>Displays how the printed document will look.</td>
</tr>
<tr>
<td>Site Admin</td>
<td>Sets printer options, access limitations, and define table relationships. (Might be available to the system administrator only.)</td>
</tr>
<tr>
<td>Customize</td>
<td>Modifies the Results environment. (Might be available to the system administrator only.)</td>
</tr>
<tr>
<td>Exit</td>
<td>Exits the Results application.</td>
</tr>
<tr>
<td>Add/Remove Fields...</td>
<td>Adds and removes display fields from the query view.</td>
</tr>
<tr>
<td>Add Calculated Field...</td>
<td>Creates a variety of calculated fields to display in the query view.</td>
</tr>
<tr>
<td>Aggregates...</td>
<td>Creates aggregates for the current report.</td>
</tr>
<tr>
<td>Properties</td>
<td>Modifies field names, labels, data formats, and coordinates for the display fields in the query view.</td>
</tr>
</tbody>
</table>
Table menu

The Table menu manipulates tables. Table A–2 describes the menu that appears when you select this option.

<table>
<thead>
<tr>
<th>Table menu option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add/Remove Tables...</td>
<td>Adds and removes tables from the open query.</td>
</tr>
<tr>
<td>Relationship Types...</td>
<td>Defines partial or complete relationships among tables in the query.</td>
</tr>
</tbody>
</table>

Field menu

The Field menu creates and manipulates fields. Table A–3 describes the menu that appears when you select this option.

<table>
<thead>
<tr>
<th>Field menu option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add/Remove Fields...</td>
<td>Adds and removes display fields from the query view.</td>
</tr>
<tr>
<td>Add Calculated Field...</td>
<td>Creates a variety of calculated fields to display in the query view.</td>
</tr>
<tr>
<td>Aggregates...</td>
<td>Creates aggregates for the current report.</td>
</tr>
<tr>
<td>Properties</td>
<td>Modifies field names, labels, data formats, and coordinates for the display fields in the query view.</td>
</tr>
</tbody>
</table>
Data menu

The Data menu manipulates data. Table A–4 describes the menu that appears when you select this option.

Table A–4: Data menu

<table>
<thead>
<tr>
<th>Data menu option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selection...</td>
<td>Defines the selection criteria Results uses to retrieve records from the database.</td>
</tr>
<tr>
<td>Re-ask Questions...</td>
<td>Prompts you to redefine the data selection criteria in the Form and Browse views.</td>
</tr>
<tr>
<td>Sort Ordering...</td>
<td>Determines the order in which Results retrieves records from the database.</td>
</tr>
<tr>
<td>Governor...</td>
<td>Limits the number of records retrieved from the database.</td>
</tr>
</tbody>
</table>

Options menu

The Options menu manipulates query and page information. Table A–5 describes the menu that appears when you select this option.

Table A–5: Options menu (1 of 2)

<table>
<thead>
<tr>
<th>Options menu option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Information...</td>
<td>Displays a list containing the tables included in the query, the display fields used, the sort order for the query, and if you either printed or viewed a report through print preview, the amount of time it took to run the query, and the connected databases.</td>
</tr>
<tr>
<td>Headers and Footers...</td>
<td>Creates report headers, footers, cover pages, and final pages.</td>
</tr>
<tr>
<td>Master-Detail...</td>
<td>Sets a break point between the tables used in the Master and Detail sections of a query.</td>
</tr>
</tbody>
</table>
The View menu control in which view Results displays the data and manipulate the screen layout. Table A–6 describes the menu that appears when you select this option.

Table A–6: View menu

<table>
<thead>
<tr>
<th>View menu option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>As Browse</td>
<td>Lists records on-screen in a tabular form similar to a spreadsheet.</td>
</tr>
<tr>
<td>As Report</td>
<td>Creates reports for printing.</td>
</tr>
<tr>
<td>As Form</td>
<td>Views, updates, adds, and deletes records in the database.</td>
</tr>
<tr>
<td>As Label</td>
<td>Uses the Labels view to create and print labels.</td>
</tr>
</tbody>
</table>
Help menu

The Help menu provides help information. Table A–7 describes the menu that appears when you select this option.

<table>
<thead>
<tr>
<th>Help menu option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help Topics</td>
<td>Choose this option to access the main table of contents, the index, and the search engine for help information about the specific tool.</td>
</tr>
<tr>
<td>Messages</td>
<td>Choose this option to quickly search for information about Progress and operating system error messages. In the Messages dialog box, enter the message number to display the description.</td>
</tr>
<tr>
<td>Recent Messages</td>
<td>Choose this option to display the most recent Progress message and its description. You can scroll backwards through all messages since the beginning of your session.</td>
</tr>
<tr>
<td>About RESULTS</td>
<td>Choose this option to display the version of Progress Results, copyright date, and system and memory information.</td>
</tr>
</tbody>
</table>
Comparison Operators

You use comparison operators within \texttt{WHERE} clauses to refine the selection criteria for a query. This appendix presents:

- Before you begin
- Comparison operator details
Before you begin

You use comparison operators within \texttt{WHERE} clauses to refine the selection criteria for a query. Table B–1 in this appendix describes the available comparison operators. See the “Creating filters” section on page 5–12 section in Chapter 5, “Changing the Query Selection Criteria,” for more information about creating \texttt{WHERE} clauses.

Comparison operator details

Table B–1 identifies and describes the comparison operators you can use with \texttt{WHERE} clauses you use in Results.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>Equality selects records where the value of the tested field is equal to the specified value.</td>
<td>\texttt{State = &quot;Massachusetts&quot;} Selects all the customers from Massachusetts.</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>Inequality selects records where the value of the tested field is different to the specified value.</td>
<td>\texttt{Balance&lt;&gt; 0} Selects all the customers with a credit balance not equal to $0.00.</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less Than selects records where the value of the tested field is smaller than the specified value.</td>
<td>\texttt{Cust-Num &lt; 50} Selects all the customers 1 through 49.</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater Than selects records where the value of the tested field is larger than the specified value.</td>
<td>\texttt{Balance &gt; 0} Selects all the records with a balance of more than $0.00.</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less Than or Equal To selects records where the value of the tested field is less than or equal to the specified value.</td>
<td>\texttt{Amount &lt;= 1000.00} Selects all the invoices that total less than or equal $1000.00.</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater Than or Equal To selects records where the value of the tested field is greater than or equal to the specified value.</td>
<td>\texttt{Invoice-Date &gt;= 05/27/94} Selects all the invoices dated 05/27/94 or later.</td>
</tr>
</tbody>
</table>
## Comparison Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
</table>
| Begins    | Selects records where the value of the tested field begins with a specific character. | Name Begins "J"  
Selects all the customers that start with the letter “J”. |
| Matches   | Selects records where the value of the tested field matches one or more characters. Use “wildcard” characters to extend the range of pattern matching capabilities. Use an asterisk (*) to match any group of characters and a period (.) to match any single character in the specified position. | Name Matches "Urpon Frisbee"  
Selects all the customer records that match the name “Urpon Frisbee.”  
Name Matches *and*  
Selects all the customer records that contain the substring “and.”  
Postal-Code Matches 02.10  
Selects records with the Postal-Code field matching 02010, 02110, 02210, 02310, 02410, 02510, 02610, 02710, 02810, and 02910. |
| Contains  | Selects records where a particular field contains a specific word. Use an asterisk (*) at the end of a word fragment to extend the range of pattern matching capabilities for the word. | Comment Contains "credit"  
Selects those records where the Comment field contains the word “credit.”  
Comment Contains "cred*"  
Selects those records where the Comment field contains a word that begins with “cred”. |
| Range     | Selects records where the value of the tested field falls within a specified range. Use the Inclusive checkbox to set whether or not the range is to be considered inclusive. If the range is inclusive, it contains the beginning and ending values. | Cust-Num Range 10 20 Inclusive  
Selects the records for customer numbers 10 through 20.  
Cust-Num Range 15 25 Exclusive  
Selects the records for customer numbers 16 through 24. |
## Index

### A
- Adding display fields 7–2
- tables to a query 5–2
- ADE, *See* Application Development Environment
- Adjusting report dimensions 4–22
- Aggregate fields
  - average 6–37
  - count 6–37
  - creating 6–37, 6–41
  - maximum 6–37
  - minimum 6–37
  - subtotal 6–39
  - total 6–37, 6–38
- Analyzing data 6–1
- Application Development Environment
desktop 2–4
- Arrays 6–7, 6–12
- ASCII files
generating 4–31
- Ask At Runtime
dialog box 5–19
toggle box 5–14

### B
- Bars
  - scroll 3–14
- Break groups 5–9, 6–39
- Browse views 3–3, 4–2
  - described 2–2
  - frames 4–4
  - navigating through records 4–3

### C
- Calculated fields
  - counter 6–7, 6–11
  - creating 6–8
  - date function 6–8, 6–27
  - described 6–6
  - logical function 6–8, 6–31
  - lookup 6–7, 6–14
  - math 6–7, 6–17
  - numeric function 6–8, 6–23
  - percent of total 6–7, 6–9
  - removing 6–37
  - renaming 6–36
  - running total 6–7, 6–10
  - stacked array 6–7, 6–12
  - string function 6–7, 6–20
Changing
  display field properties 7–3
  relationships 5–4
Character fields 7–5
Closing queries 3–12
Comparison operators B–1
Complete relationships 3–10
  changing 5–4
  described 3–10
  effect on sort order 5–4
Complete table relationships, See Complete relationships
Composite records 3–7, 6–5
Compound indexes 1–6
Connecting databases 2–5
Copying records 4–10
Creating
  aggregate fields 6–37, 6–41
  ASCII files 4–33
  calculated fields 6–6, 6–8
  labels 4–24
  queries 3–2
  records 4–11
  WHERE clauses 5–12

D
Data 1–2
  analyzing 6–1
  indexes, See Indexes
  queries, See Queries
  sort order 3–8
Data governor 5–23
Data menu A–4

Data selection 5–13
  creating WHERE clauses 5–12
  removing WHERE clauses 5–19
  setting at runtime 5–19
Databases
  connecting 2–5
  described 1–2
  indexes, See Indexes
  modifying 4–8
  tables, See Tables
Date fields 7–6
Dates
  calculated fields 6–8, 6–27
Decimal fields 7–7
Defaults
  queries 3–8
Deleting
  queries 3–16
  records 4–11
Detail section 6–3
Display fields
  adding 7–2
  changing labels 4–16
  changing properties 7–3
  changing the order 7–2
  character 7–5
  creating aggregate fields 6–37
  creating calculated 6–6
  date 7–6
  decimal 7–7
  described 3–3
  format 4–16, 7–5
  integer 7–7
  label 4–16
  logical 7–9
  modifying 7–1
  numeric 7–7
  rearranging in form view 4–12
  removing 7–2
  removing calculated 6–37

Index–2
Exiting
RESULTS 2–7

Export types
modifying 4–33
selecting 4–31

Export views 3–3, 4–30
described 2–2
modifying export types 4–33
selecting export types 4–31

Expressions
calculated fields 6–7, 6–17

Field menu A–3

Fields
See Display fields
aggregate, See Aggregate fields
calculated, See Calculated fields
classifier 7–5
database
structure 1–5
date 7–6
decimal 7–7
described 1–5
display, See Display fields
integer 7–7
logical 7–9
rearranging 4–12
selecting for queries 3–6
values 6–7, 6–14

Files
ASCII 4–31

Filters 5–12

Footer 4–17

Form views 3–3, 4–7
described 2–2
frames 4–13
modifying databases 4–8

navigating through records 4–8
Formats for display fields 7–5
Frames
browse views 4–4
form view 4–13
changing properties 4–15

Generating ASCII files 4–31

Headers 4–17

Help
menu A–6

Indexes 3–8
compound 1–6
described 1–6
simple 1–6

Inner joins, See Relationships, partial

Integer fields 7–7

Joins, See Relationships

Label types
reformatting 4–28
selecting 4–27

Label views 3–3, 4–24
described 2–2
reformatting labels 4–28
selecting label types 4–27
OpenEdge Reporting: Query/Results for Windows

Labels
  creating 4–24
  printing 4–29

Logical fields 7–9

Logicals
  calculated fields 6–8, 6–31

M

Many-to-many relationships 1–9
Master sections 6–3
Menu bar 2–6
Menus A–1

Modifying
  display fields 7–1
  export types 4–33

N

Navigation buttons
  print preview 3–15
Numeric fields 7–7

O

One-to-many relationships 1–9
One-to-many table relationships 3–8
One-to-one relationships 1–8
Opening queries 3–12
Operators B–1
Options menu A–4
Order, sort
  See Sort order

P

Page size 4–22

Pages
  cover 4–17
  final 4–17
  footers 4–17
  headers 4–17
  sizes 4–21, 4–22

Partial relationships 3–9, 5–4

Print preview 3–14

Printing
  labels 4–29
  queries 3–13
  reports 4–23

Program groups
  starting RESULTS 2–3

Properties
  display fields 7–3
  frames
    form view 4–15

Q

Queries 3–2
  adding tables 5–2
  closing 3–12
  creating 3–2
  data selection 5–13
  defaults 3–8
  deleting 3–16
  described 1–7
  frames, See Frames
  getting information 3–13
  opening 3–12
  printing 3–13
  relationships 3–9
  removing tables 5–2
  saving 3–11
selection criteria 5–1
views 3–3
Query By Example 5–21
creating WHERE clauses 5–21
removing WHERE clauses 5–22
Query directory (.qd7) 3–11
Query menu A–2
Query selection criteria
changing 5–1

R
Rearranging display fields 4–12
Records 1–2
composite 3–7, 6–5
copying 4–10
creating 4–11
deleting 4–11
described 1–5
modifying 4–8
navigating
form views 4–3, 4–8
sorting 5–7
Reformatting labels 4–28
Relationships 3–8, 5–4
changing 5–4
complete 3–10
described 1–7
many-to-many 1–9
one-to-many 1–9
one-to-one 1–8
partial 3–9
selecting for queries 3–5
types 1–8
defaults 3–8
Removing
calculated fields 6–37
display fields 7–2
tables from a query 5–2
WHERE clauses
created through data selection 5–19
created through Query By Example 5–22
Renaming calculated fields 6–36
Report views 3–3, 4–16
described 2–2
Reports
adjusting dimensions 4–22
cover pages 4–17
final pages 4–17
footers 4–17
headers 4–17
printing 4–23
selecting page sizes 4–21
RESULTS 2–1
described 2–2
exiting 2–7
main window 2–6
starting 2–3

S
Saving queries 3–11
Scroll bars 3–14
Selecting
export types 4–31
label types 4–27
page sizes, reports 4–21
Setting data selection at run time 5–19
Simple indexes 1–6
Sort order
break groups 5–9
defaults 3–8
Sorting
records 5–7
Starting RESULTS 2–3
Status bars 2–6, 3–7

Strings
  calculated fields 6–7, 6–20
  modifying 4–35

Switching query views 4–2

T

Table menu A–3

Table relationships 3–8
  See Relationships

Tables
  adding 5–2
  removing 5–2
  selecting for queries 3–5
  structure 1–4

Tool bar 2–6

Totals
  aggregate fields 6–38
  running 6–7, 6–10
  totals only summary 6–42

U

Using
  print preview 3–14

V

View menu A–5

Views
  browse 4–2, 4–3
  described 2–2
  export 4–30
    modifying export types 4–33
    selecting export types 4–31
    using 4–30
  form 4–7
    modifying databases 4–8
    navigating through records 4–8
  label 4–24
    reformatting labels 4–28
    selecting label types 4–27
  report 4–16
    formatting 4–17
    printing 4–23
    selecting page size 4–21
    switching 4–2

W

WHERE clause
  creating 5–12
  removing 5–19, 5–22