# Learn Microsoft Access

Ben Welman

May 20, 2025

# Table of contents

Welco	
	Help
	Licence
Prefa	ce
T	
	xtbook Organization
So	ftware
. E	Basic
l. St	arting with Access
1.1	I. What Are Databases?
	1.1.1. Tables
	1.1.2. Queries $\ldots$
	1.1.3. Forms
	1.1.4. Reports
	1.1.5. Macros
	1.1.6. Modules $\ldots$
1.2	2. Starting and Closing Access
1.3	
1.4	
1.5	
1.6	
	1.6.1. Backup
1.7	7. Access File Formats
	ploring Candy database
2.1	L. Example Database Candy
	2.1.1. Table Customers
	2.1.2. Table Orders
	2.1.3. Table OrderDetails
	2.1.4. Table Boxes
	2.1.5. Table BoxDetails
	2.1.6. Table Pralines
2.2	$\mathcal{J}$
	Enable Macros Temporarily
	Add Publisher to Trusted Publishers List
	Move Database to a Trusted Location
	Security Settings for All Macros
2.3	1 0
2.4	4. Exploring Tables

## Table of contents

	2.5.	Exploring Forms
	2.6.	Using a Command Button
	2.7.	Searching for a Record
	2.8.	Exploring Queries
	2.9.	Exploring Reports
	2.10.	Sorting
		Filtering
		2.11.1. Simple Filter
		2.11.2. Modifying a Filter
		2.11.3. Filter on Two Criteria
		2.11.4. Number Filter         33
	2 1 2	Printing
		Exercises
	2.13.	Exercises
3.	Data	Entry and Modification 35
•		About Data Entry
	3.2.	Table Operations   35
	3.2.	Add a New Customer   36
	3.4.	Add a New Praline       30         37
	3.5.	Add a New Box       37         37         37
	3.6.	Data Restriction and Validation       37
	0.0.	
	3.1.	Exercises
4	Tabl	es 40
••		About Tables
	4.2.	Fields in Tables   40
	т.2.	4.2.1. Field Names       40
		4.2.1. Pield Walles
		4.2.2. Data Types
		4.2.3. Field Floperties
	4.9	
	4.3.	Validation         42
		4.3.1. Input Masks
		4.3.2. Validation Rule and Validation Text
	4.4.	Create a New table Manually
	4.5.	Importing an Excel Worksheet as a Table
	4.6.	Creating a List Box
	4.7.	Relations Between Tables 47
		4.7.1. One-to-Many Relationship
		4.7.2. One-to-One Relationship 49
		4.7.3. Many-to-Many Relationship
		4.7.4. Creating, Modifying, and Deleting Relationships
		4.7.5. Referential Integrity
		4.7.6. Creating the Boxes–BoxDetails Relationship
	4.8.	Exercises
5.	Que	ries 56
	5.1.	About Creating Queries
	5.2.	Customers and Orders
	5.3.	Criteria in Queries
	5.4.	Orders from Utrecht Customers
	5.5.	Customers with Box CHER

Table	of	contents
-------	----	----------

	5.6.	3. Summarizing and Calculations								
	5.7.									
	5.8.									
	5.9.									
	5.10.	10. First Order per Customer								72
		11. Parameter Query								
		12. Action Queries								
		5.12.1. Update Query								
		5.12.2. Make Table Query $\ldots$								
	5.13	13. Crosstab Query								
		14. Exercises								
6.	Forn									86
	6.1.	1. Creating Forms								86
		6.1.1. Main and Subforms								
		6.1.2. Linked Forms			•					
	6.2.	2. Controls and Layouts			•					88
	6.3.	J								
	6.4.	4. Main and Subform			•					92
	6.5.	5. Customer Data Entry Form			•					94
	6.6.	6. Praline Data Entry Form								96
	6.7.	7. Boxes Data Entry Form								97
	6.8.	8. Orders Data Entry Form								98
	6.9.	9. Total Order Amount								101
	6.10.	10. Form with Chart								104
	6.11.	11. Linking forms								108
		0	•••		•					
	6.12.	12. Exercises						•		110
7.	Rep	12. Exercises					•			113
7.	<b>Rep</b> 7.1.	12. Exercises	· ·	· ·	•					<b>113</b> 113
7.	<b>Rep</b> 7.1.	12. Exercises	· ·	· ·	•					<b>113</b> 113
7.	<b>Rep</b> 7.1. 7.2. 7.3.	12. Exercises	· ·	· · · · ·		· · · ·			 	<b>113</b> 113 113 115
7.	<b>Rep</b> 7.1. 7.2. 7.3.	12. Exercises	· ·	· · · · ·		· · · ·			 	<b>113</b> 113 113 115
7.	<b>Rep</b> 7.1. 7.2. 7.3. 7.4.	12. Exercises	· · ·	· · · · · · · ·		· · · ·			  	<b>113</b> 113 113 115 116
7.	<b>Rep</b> 7.1. 7.2. 7.3. 7.4. 7.5.	12. Exercises	· · · · · · · · · · · · · · · · · · ·	· · · · · · ·		· · ·		•	  	<b>113</b> 113 113 115 116 120
7.	<b>Rep</b> 7.1. 7.2. 7.3. 7.4. 7.5. 7.6.	12. Exercises	· · · · · · · · · · · · · · · · · · ·	· · · · · · ·		· · ·		•	  	<b>113</b> 113 113 115 116 120
11.	Rep 7.1. 7.2. 7.3. 7.4. 7.5. 7.6. Ad	12. Exercises	· · · · · · · · · · · · · · · · · · ·	· · · · · · ·		· · ·		•	  	<ul> <li>113</li> <li>113</li> <li>113</li> <li>115</li> <li>116</li> <li>120</li> <li>123</li> <li>128</li> </ul>
11.	Rep( 7.1. 7.2. 7.3. 7.4. 7.5. 7.6. Ad	12. Exercises	· · ·	<ul> <li>.</li> <li>.&lt;</li></ul>	•	· · ·		• • • •	  	<ul> <li>113</li> <li>113</li> <li>113</li> <li>115</li> <li>116</li> <li>120</li> <li>123</li> <li>128</li> <li>129</li> </ul>
11.	Rep 7.1. 7.2. 7.3. 7.4. 7.5. 7.6. Ad 8.1.	12. Exercises	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · ·	· · · · · · · · ·	· · ·	• • • • •	  	<ul> <li>113</li> <li>113</li> <li>113</li> <li>115</li> <li>116</li> <li>120</li> <li>123</li> <li>128</li> <li>129</li> <li>129</li> </ul>
11.	Rep 7.1. 7.2. 7.3. 7.4. 7.5. 7.6. Ad 8.1. 8.2.	12. Exercises	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · ·	· · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	<ul> <li>113</li> <li>113</li> <li>113</li> <li>115</li> <li>116</li> <li>120</li> <li>123</li> <li>128</li> <li>129</li> <li>129</li> <li>129</li> <li>129</li> </ul>
11.	Rep( 7.1. 7.2. 7.3. 7.4. 7.5. 7.6. Ad Integ 8.1. 8.2. 8.3.	12. Exercises	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · ·	· · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · ·	<ul> <li>113</li> <li>113</li> <li>113</li> <li>115</li> <li>116</li> <li>120</li> <li>123</li> <li>128</li> <li>129</li> <li>129</li> <li>131</li> </ul>
11.	Repo 7.1. 7.2. 7.3. 7.4. 7.5. 7.6. Ad Integ 8.1. 8.2. 8.3. 8.4.	12. Exercises	· · · · · · · · · · · · · · · · · · ·	· ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · ·	· · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · ·	<ul> <li>113</li> <li>113</li> <li>113</li> <li>115</li> <li>116</li> <li>120</li> <li>123</li> <li>128</li> <li>129</li> <li>129</li> <li>129</li> <li>131</li> <li>132</li> </ul>
II. 8.	Repo 7.1. 7.2. 7.3. 7.4. 7.5. 7.6. Ad Integ 8.1. 8.2. 8.3. 8.4. 8.5.	12. Exercises	· · · · · · · · · · · · · · · · · · ·	· ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · ·	· · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · ·	<ul> <li>113</li> <li>113</li> <li>115</li> <li>116</li> <li>120</li> <li>123</li> <li>128</li> <li>129</li> <li>129</li> <li>129</li> <li>131</li> <li>132</li> <li>132</li> <li>132</li> </ul>
II. 8.	Rep 7.1. 7.2. 7.3. 7.4. 7.5. 7.6. Ad Integ 8.1. 8.2. 8.3. 8.4. 8.5. Tool	12. Exercises	· · · · · · · · · · · · · · · · · · ·	<ul> <li>.</li> <li>.&lt;</li></ul>	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · ·	· · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · ·	<ul> <li>113</li> <li>113</li> <li>113</li> <li>115</li> <li>116</li> <li>120</li> <li>123</li> <li>128</li> <li>129</li> <li>129</li> <li>129</li> <li>131</li> <li>132</li> <li>132</li> <li>132</li> <li>133</li> </ul>
II. 8.	Repo 7.1. 7.2. 7.3. 7.4. 7.5. 7.6. Ad Integ 8.1. 8.2. 8.3. 8.4. 8.5. Tool 9.1.	12. Exercises	· · · · · · · · · · · · · · · · · · ·	· ·	· · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · ·	· · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · ·	<ul> <li>113</li> <li>113</li> <li>113</li> <li>115</li> <li>116</li> <li>120</li> <li>123</li> <li>128</li> <li>129</li> <li>129</li> <li>129</li> <li>131</li> <li>132</li> <li>132</li> <li>133</li> <li>133</li> </ul>
II. 8.	Repo 7.1. 7.2. 7.3. 7.4. 7.5. 7.6. Ad Integ 8.1. 8.2. 8.3. 8.4. 8.5. Tool 9.1. 9.2.	12. Exercises	· · · · · · · · · · · · · · · · · · ·	<ul> <li>.</li> <li>.&lt;</li></ul>	· · · · · · · · · · · · · · · · · · ·	· ·	· · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · ·	<ul> <li>113</li> <li>113</li> <li>113</li> <li>115</li> <li>116</li> <li>120</li> <li>123</li> <li>128</li> <li>129</li> <li>129</li> <li>129</li> <li>129</li> <li>131</li> <li>132</li> <li>132</li> <li>133</li> <li>133</li> <li>133</li> </ul>
II. 8.	Repo 7.1. 7.2. 7.3. 7.4. 7.5. 7.6. Ad Integ 8.1. 8.2. 8.3. 8.4. 8.5. Tool 9.1. 9.2.	12. Exercises	· · · · · · · · · · · · · · · · · · ·	<ul> <li>.</li> <li>.&lt;</li></ul>	· · · · · · · · · · · · · · · · · · ·	· ·	· · · ·	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · ·	<ul> <li>113</li> <li>113</li> <li>113</li> <li>115</li> <li>116</li> <li>120</li> <li>123</li> <li>128</li> <li>129</li> <li>129</li> <li>129</li> <li>129</li> <li>131</li> <li>132</li> <li>132</li> <li>133</li> <li>133</li> <li>133</li> </ul>

## Table of contents

10. Macros	139
10.1. About Creating Macros	139
10.2. Customer Search by Code	140
10.3. Customer Search by Name	142
10.4. Combo Box: Search by Box Code	146

# Welcome

This the free online version of the textbook **Learn Access**. This task-oriented textbook is developed especially for students in higher education. It has a lot of attention for creating queries, forms and reports. And contains many exercises.

## Data files

Data files are included with this textbook. You need these to complete the exercises. You can download them in one zip file: Download data files<sup>a</sup>.

<sup>a</sup>https://learnaccess.netlify.app/files-learnaccess.zip

#### Help

- Report an issue<sup>1</sup>
- Ask a question<sup>2</sup>

#### Licence



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License..

#### About the author



I studied Chemical Engineering at the University of Twente<sup>4</sup>. After teaching chemistry and mathematics at secondary schools for a several years, I began teaching informatics and statistics at Saxion University of Applied Sciences<sup>5</sup>. From there, I transitioned

<sup>&</sup>lt;sup>1</sup>https://github.com/bwelman/learnaccess/issues

<sup>&</sup>lt;sup>2</sup>https://github.com/bwelman/learnaccess/discussions

<sup>&</sup>lt;sup>3</sup>https://creativecommons.org/licenses/by-nc-sa/4.0

<sup>&</sup>lt;sup>4</sup>https://www.utwente.nl/en/

<sup>&</sup>lt;sup>5</sup>https://www.saxion.edu/

#### Welcome

internally to MeetingPoint, which focused on developing and supporting e-learning. In 1993 I founded Softwijs<sup>6</sup>, which I closed at the end of 2018.

Since my retirement, I've enjoyed my hobbies: billiards, bridge, brewing beer, baking bread, and traveling (especially with my camper and bike). In addition to creating textbooks, I also spend time exploring data analysis, particularly with  $\mathbb{R}^7$  and to a lesser extent with Python<sup>8</sup>.

Ben Welman

Other textbooks:

• Learn Excel<sup>9</sup>

<sup>&</sup>lt;sup>6</sup>https://bwelman.github.io/softwijs/

<sup>&</sup>lt;sup>7</sup>https://en.wikipedia.org/wiki/R\_(programming\_language)

<sup>&</sup>lt;sup>8</sup>https://www.python.org/

<sup>&</sup>lt;sup>9</sup>https://learnexcel.netlify.app/

# Preface

This textbook offers a comprehensive introduction to Microsoft Access and is designed specifically for students in higher education and users of Microsoft Access. The primary focus is on creating queries, forms, and reports rather than on database design. The chapters covering these components include numerous exercises to help you practice retrieving information from a database.

This textbook differs from other Access textbooks in several key ways:

- It is free and available online, making it accessible to a wide audience.
- Each topic is developed to accomplish a specific task—nothing more, nothing less. This results in clear, step-by-step instructions without unnecessary information that might distract from the learning objective.
- The tasks and exercises are based on the competencies and skills required in higher education. Many of the challenges are drawn from real-life experiences of students during internships and graduation projects.

# **Textbook Organization**

This textbook is centered around the candy365.accdb database of the fictional company SNOOPY, which sells a variety of chocolate boxes to customers. The chapters follow a logical structure for working with this database:

- Chapter 1 introduces the Access program and explains the general structure and contents of databases.
- Chapter 2 provides an overview of the candy365.accdb database, including its tables and contents. You will experiment with the database to explore different features of working with Access.
- Chapter 3 covers how to add and modify data, one of the most important tasks for users.
- Chapter 4 focuses on creating tables and establishing relationships between them.
- Chapter 5 teaches you how to retrieve specific information from a database using queries, another common end-user task.
- Chapter 6 introduces forms, which present organized, formatted views of table and query fields. Forms are also essential for data entry.
- Chapter 7 covers reports, which are typically formatted for printing. This chapter also includes label creation, a special type of report.
- Chapter 8 discusses ways to integrate Access with Excel and Word.
- Chapter 9 presents the tools available in Access to analyze, troubleshoot, and correct issues within a database.
- Chapter 10 shows how to automate common or complex tasks using macros.

# Software

This textbook is based on Access 365, but it is also suitable for earlier versions such as 2016, 2019, and even older versions like 2010 and 2007.

Part I. Basic

# 1. Starting with Access

### **OBJECTIVES**

- Understand what databases are and how they are structured.
- Get a brief introduction to database components (tables, queries, forms, reports, macros, modules).
- Learn how to start and close Access.
- Open an existing database and save it under a different name and/or file format.
- Get familiar with the layout of the Access program window.

Access is part of the Microsoft Office suite and allows you to create and manage databases. With databases, you can store, organize, and analyze data efficiently.

Data may include details such as customer information, orders, products, suppliers, and more. This data is stored in a structured way, enabling you to answer questions like:

- What was the revenue per customer over the past six months?
- Which products account for the majority of sales?

Access is often considered the most complex application in the Microsoft Office suite. This is primarily due to how databases are structured and function. Unlike Excel, PowerPoint, or Word, where you can start entering data right away, Access requires you to design and create a database structure before adding any data. For large or complex databases, this is typically done by database professionals.

## 1.1. What Are Databases?

A brief overview of what databases are and how they are organized.

A database is a collection of data related to a specific topic or purpose. It allows you to store and manage information from various sources, such as:

- Customer addresses in an address list
- Supplier phone numbers in a directory
- Product data in a filing system
- Customer orders in a folder

This information is usually stored in separate tables that make up the database. In its simplest form, a database may consist of just one table. However, tables can be related to one another, allowing you to link data across tables. Such a setup is known as a **relational database**.

In addition to creating and managing tables, you can build forms, queries, and reports. Forms make data entry more efficient. Queries allow you to retrieve specific information. Reports help present your results in a clear and readable format for screen or print.

A database in Access can contain the following components (also called **objects**):

#### 1. Starting with Access

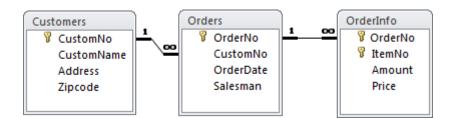


Figure 1.1.: Example of a relational database with three tables and their relationships.

- Tables
- Queries
- Forms
- Reports
- Macros
- Modules

Access stores all of these components in a single file.

There are two main tasks when working with databases:

- **Designing the database** This includes creating the structure: tables, queries, forms, and reports. Once completed, this task is usually done.
- Working with data Entering, updating, querying, reporting, and managing the data. This is an ongoing task.

#### 1.1.1. Tables

Data in Access is stored in tables. Each table usually represents a specific type of data. For example, a **Products** table might store product details, while a **Suppliers** table contains information about companies that provide those products. Determining which tables you need—and how they relate—requires careful planning and structure. Database designers often follow a process called **normalization** to ensure efficiency and avoid redundancy.

Data in a table is organized into columns (fields) and rows (records).

	Suppl	iers					
2	Id	*	Supplier 🚽	Contact 🚽	TelNo 👻	FaxNo	*
		1	Chocolate Paradise	Elizabeth Andersen	020-9876543	020-8765432	
		2	World of Nuts	Paul Dunton	030-1234567	030-7654321	
		3	Sweet Company	Madeleine Kelley	0543-654321	0543-123456	

Figure 1.2.: Example of a table with 3 records and 5 fields.

- **Field** A field is a data category, such as company name, contact person, phone number, or unit price. Each column in a table is a field.
- **Record** A record is a complete set of data for a single item—such as a product, person, or company. Each row in a table is a record.

## 1.1.2. Queries

A **query** is used to retrieve specific data from one or more tables. It's like asking the database a question, such as "Which products have a supplier located in Australia?" The data used to answer this question may come from multiple tables. The query gathers and displays the relevant records, also known as a **dynaset**.

When creating a query, you define the information you want to extract. You can also set conditions to filter the data. For example, you might create a query to view order details from a certain time period. Queries can also perform calculations, such as totals or averages.

OrderCode 👻	CustomerCode 👻	OrderDate 👻	Subtotal 👻
1	237	11/2/2009	38
3	158	11/2/2009	65.75
6	1	11/2/2009	218.25
7	275	11/2/2009	19
10	7	11/3/2009	76.5
11	148	11/3/2009	142.5
12	154	11/3/2009	162 5

Figure 1.3.: Example of a query showing order data.

Another type of query is an **action query**, which modifies the selected records.

## 1.1.3. Forms

While tables are great for viewing multiple records at once, **forms** are better suited for displaying individual records or presenting data in a customized layout.

Forms provide a user-friendly way to enter, update, and view records. When designing a form, you decide how the data should appear. When you open the form, Access retrieves and displays the data based on your design. Forms can pull data from multiple tables and include macros or modules for automation.

Chocolate Conoisseur						
Customer entry and modification						
Customer code:	1					
	First name:		Last r	name:		
Name:	Rita		Hend	lriks		
Address:	2e Zeine 99 A					
ZipCode:	7233 AL	C	City:	Zwolle		
Province:	OV	Reg	ion:	North	•	
Telephone:	03210-30149					

Figure 1.4.: Example of a form for entering or changing customer data. A drop-down list is provided for selecting the region.

#### 1. Starting with Access

Forms can include features like dropdown lists, color-coded highlights, validation messages, automatic calculations, and more. You can also switch between form view and datasheet view (a table-like layout) with a single click.

## 1.1.4. Reports

**Reports** are used to present data clearly on screen or paper, often including subtotals or grand totals across groups of records.

The elements in a form or report that display data are called **controls**. These can show field values, calculations, charts, and more. Reports can also be used to create labels.

ales per box p	per region	
BoxName	Region	Sales
All Seasons		
	North	\$210.00
	South	\$266.00
Summary for 'BoxNam	ne' = All Seasons (2 d	etail records)
Sum		\$476.00
Standard		0.89%
Autumn Surprise		
	North	\$946.00
	South	\$1,935.00
Summary for 'BoxNam	ne' = Autumn Surpris	e (2 detail records
Sum		\$2,881.00
Standard		5.40%
Butterscotch		
	North	\$2.052.50

Figure 1.5.: Example of a part of a report.

## i Note

#### Controls

Controls are elements on forms or reports that display or print data. Examples include buttons, list boxes, and check boxes.

They allow you to show field values, calculations, titles, messages, charts, or even other forms or reports.

## 1.1.5. Macros

A **macro** is a set of one or more actions, such as opening a form or printing a report. Macros help you automate routine tasks and can be attached to forms, reports, controls, keyboard shortcuts, or menus. You don't need programming skills to create them.

## 1.1.6. Modules

**Modules** contain code written in Visual Basic for Applications (VBA), the programming language built into Access. Modules let you automate complex or custom tasks. Creating VBA

modules does require some programming experience.

## 1.2. Starting and Closing Access

How you start Access depends on how it's installed. This course assumes a default installation of Microsoft 365 in English on a Windows 10 English system. On most computers, you can start Access via the Windows Start button:

#### Task 1.1.

1. Choose Start > Access.

The Access start screen appears. On the left (under Recent), you'll see your most recently opened files. On the right, you can choose from templates to create new databases.

2. To close Access, click the X button in the upper-right corner of the window.

i Note

Unlike many other programs, Access automatically saves all changes. You don't need to manually save your database.

## 1.3. Opening a Database

You can open an existing database in several ways:

- Double-click the Access file in Windows Explorer.
- In Access's start screen, click a recent file or choose Open Other Files.
- If Access is already open, choose File > Open.

#### Task 1.2. FILE: candy365.accdb

Use one of the methods above to open the database.

i Note

```
To close the current database but keep Access open, choose File > Close.
```

## 1.4. The Access Interface

The Access window includes several main parts from top to bottom: program controls, document area, and the status bar.

- **File** The File tab in the top-left corner provides common options like opening, saving, printing, and customizing Access through Options.
- **Quick Access Toolbar** This customizable toolbar contains shortcuts for frequently used actions. By default, it includes:

### 1. Starting with Access

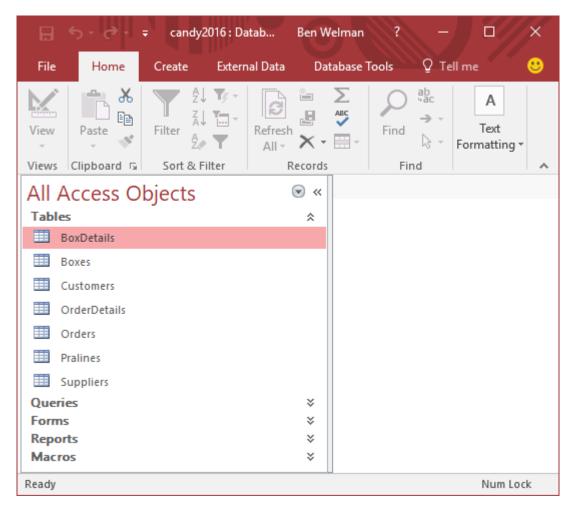


Figure 1.6.: Access program window.

- 🖬 Save
- 🕤 Undo
- 🧟 Redo
- **Ribbon** A panel at the top with organized commands grouped under tabs such as Home or Create. Some contextual tabs appear only when needed (e.g., the Table tab appears when a table is open). Frequently used commands are on the left; specialized ones are on the right.

## i Note

You can't remove the ribbon, but you can minimize it using CTRL + F1. Pressing it again restores the ribbon.

- **Tabs** Tabs appear at the top of the ribbon, grouping related commands. Some tabs only appear contextually.
- **Groups** Commands are grouped logically within each tab. To view more options, click the small arrow 5 in the bottom-right corner of a group.
- **Command button** Clicking a button usually applies an action immediately or opens a dialog box or drop-down menu.

Navigation Pane Located on the left, this pane lists all database objects.

**Document window** When you open an object, it appears in a separate tabbed document window.

# 1.5. Creating a New Database

Before creating a new database, you should first perform an information analysis and design its structure—typically using normalization. Once that's complete, building the database is relatively straightforward. While this course doesn't cover database creation in detail, here are some general steps:

From the Access start screen:

- Click Blank database to start with an empty database.
- Specify the file name and location.
- Access creates the new database with an empty table. (Table creation is discussed later.)

You can also use one of Access's many templates. These templates include predefined tables, queries, forms, and reports, which you can customize as needed—this can save time over starting from scratch.

# 1.6. Saving a Database As...

You might want to save your database under a different name or file format. Use Save As to do so.

This course uses a sample database named candy365.accdb. Since Access saves changes automatically, your original copy would quickly be altered. To avoid losing the original, create a working copy before making changes—either using Windows File Explorer or the Save As command.

#### Task 1.3. FILE: candy365.accdb

- 1. Open the database.
- 2. Choose File > Save As.

$\bigcirc$		
Info	Save As	
New	File Types	Save Database As
<b>Open</b> Save	Save Database As	Database File Types
Save As	Save Object As	Default database format.
Print Close		Save a copy that will be compatible with Access 2002-2003.
		Save a copy that will be compatible with Access 2000. Template (*.accdt)
Account Feedback		Save the current database as a database template (ACCDT). Advanced
Options		<ul> <li>Package and Sign         <ul> <li>Package the database and apply a digital signature.</li> <li>Make ACCDE             <ul></ul></li></ul></li></ul>

Figure 1.7.: Save As window.

- 3. Select Access Database (\*.accdb) and click Save As.
- 4. Choose a location, enter a new file name, and click Save.

#### 1. Starting with Access

## 1.6.1. Backup

A backup is simply a copy of your database at a given point in time. Though you can use [Save As], Access also offers a built-in **Backup** feature:

Choose File > Save As, select **Back Up Database**, and click Save As. You'll be prompted to choose a location and file name. By default, Access adds the current date to the original file name.

i Note

When restoring from a backup, it helps to know which database it came from and when it was created. Using the default file name is a good practice.

# **1.7. Access File Formats**

A summary of Access file formats:

- **ACCDB** The default format for Access 2007 and newer (2010, 2013, 2016, 2019, 365). Files are compatible across versions, but newer features may not work in older versions.
- **MDB** The format used by Access 2003 and earlier. Access 2007 and later can open and save MDB files, but newer features are not supported.

## **OBJECTIVES**

- Understand the sample database candy365.accdb.
- Learn how to handle security alerts.
- Explore the windows and navigation options within Access.
- Discover the display capabilities of tables, forms, reports, and queries.
- See how a command button works on a form.
- Learn how to search, sort, and filter records.
- Print tables, queries, forms, and reports.

This course is centered around the sample file candy365.accdb. This chapter explains the design and use of the tables in the database. You will experiment with this sample database to discover different aspects of working with Access. Some features are explored further in other chapters.

# 2.1. Example Database Candy

The example database candy365.accdb contains data from the company SNOOPY, which sells boxes of pralines to customers. Each box contains different types of pralines. The information is stored in six tables:

- Customers
- Orders
- OrderDetails
- Boxes
- BoxDetails
- Pralines

Customer data is stored in the Customers table. Customer orders are recorded in the Orders and OrderDetails tables. Information about each box of pralines, such as name and price, is stored in the Boxes table. The BoxDetails table shows which pralines and how many of each are in each box. The Pralines table contains details about each praline, including its name, chocolate type, filling, and even a picture.

#### i Note

Each table should include a column or set of columns that uniquely identifies each row. This is often a unique identification number, such as an article number or employee number. In database terminology, this is called the **primary key** of the table. The value of the primary key must be unique in the table; duplicate values are not allowed. Most tables have a primary key consisting of a single field, but sometimes a combination of fields is needed to ensure uniqueness.

In the Customers, Orders, Boxes, and Pralines tables, the key is a single field. In the

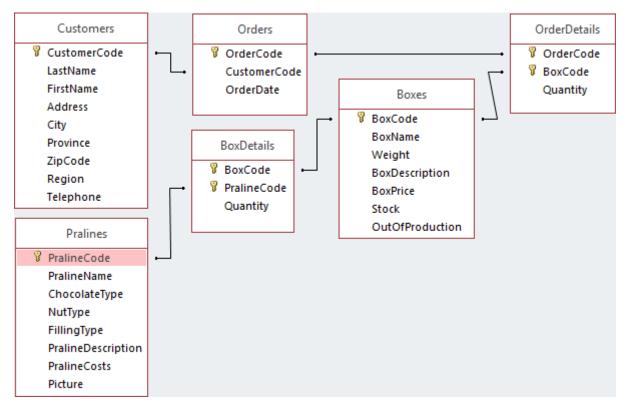


Figure 2.1.: Design database Candy.

OrderDetails and BoxDetails tables, two fields are required for the key. See Figure 2.1.

The six tables are briefly discussed below.

## 2.1.1. Table Customers

	CustomerCode •	LastName +	FirstNam +	Address -	City -	Province -	ZinCode +	Regio -	Telephone
	customercout +	costivanie ·	rinsulation -	Address .	City -	Province •	zipcoue ·	negio .	relephone
+	1	Hendriks	Rita	2e Zeine 99 A	Zwolle	ov	7233 AL	North	03210-3014
+	2	Pietersen	Steven	6e Helmersstra	Deventer	ov	2073 CB	North	060-129955
٠	3	Calfsbeek	Gerard	A'damseweg 5'	Rotterdar	ZH	2366 AC	South	06487-3098
٠	4	Jansen	Walter	A.Fokkerweg 2	Helmond	NB	2074 CM	South	060-114686
Ŧ	5	Mulder	M	A v Borgonstra	Arnhom	GL	3922 SW/	South	081-62094

The Customers table stores various details about each customer. Every customer has a unique customer code. The CustomerCode field is the primary key of the table. Each row in a table is called a record. Customers are sorted by the primary key.

Check that there are 325 customers in the Customers table.

## 2.1.2. Table Orders

In the Orders table, OrderCode is the primary key, and its value is unique. A specific Customer-Code can appear more than once in this table because a customer can place multiple orders. Each order is always associated with one customer.

Orders							
2		OrderCode 👻	CustomerCode 🕞	OrderDate 👻			
	+	1	237	11/2/2009			
	+	3	158	11/2/2009			
	+	6	1	11/2/2009			
	+	7	275	11/2/2009			
	+	10	7	11/3/2009			

Figure 2.3.: Table Orders.

Check that there are 784 orders in the Orders table.

## 2.1.3. Table OrderDetails

OrderDetails						
🕗 OrderCode 👻	BoxCode 👻	Quantity 👻				
1	PEAN	2				
3	CHER	2				
3	NOWI	1				
6	BUTT	1				
6	FANT	1				

Figure 2.4.: Table OrderDetails.

An order can contain multiple boxes, but each BoxCode can only appear once per order. However, a particular box can be ordered in multiple orders, so a BoxCode can be linked to multiple OrderCode values. The OrderDetails table has 1,537 records (order rows).

Check that two different boxes were ordered in the order with order code 3, as the table contains two rows with order code 3. A total of three boxes were ordered in this order: two boxes with box code CHER and one box with box code NOWI.

The combination of OrderCode and BoxCode is always unique. Therefore, the key in this table is a combination of these two fields.

#### i Note

The total number of boxes ordered across all orders is the sum of all values in the Quantity column. This number is not present in the table. Another chapter in this course will explain how to calculate it.

## 2.1.4. Table Boxes

In the Boxes table, BoxCode is the key. There are 18 different boxes, each with a unique BoxCode. This table serves as the Snoopy company's product list.

	Boxes								
2		BoxCode 👻	BoxName 👻	Weight 👻	BoxDescription -	BoxPric 🝷	Stocl -	OutOfPr 👻	
	+	ALLS	All Seasons	150	Strawberries, bluebe	\$14.00	700	No	
	+	AUTU	Autumn Surprise	500	Family packing with a	\$43.00	200	No	
	+	BUTT	Butterscotch	500	Our best butterscotcl	\$ 27.75	200	No	
	+	CHER	Classic Cherry	150	Classic box with who	\$16.25	500	No	
	+	FANT	Toffee Mocha Far	300	Δ dream collection of	\$ 18 00	400	No	

Figure 2.5.: Table Boxes.

BoxDetails							
BoxCode 👻	PralineCode 👻	Quantity 👻					
ALLS	B04	2					
ALLS	B05	2					
ALLS	F01	2					
ALLS	M01	2					
ALLS	M09	2					
ALLS	M12	2					
AUTU	B03	6					
ΛΠΤΠ	M05	6					

Figure 2.6.: Table BoxDetails.

## 2.1.5. Table BoxDetails

The BoxDetails table stores which types of pralines and how many of each are in a specific box. For example, you can see that box ALLS contains six types of pralines, two of each, for a total of twelve pralines. In this table, the key is a combination of BoxCode and PralineCode. The table has 84 records.

## 2.1.6. Table Pralines

Ē	Pralines							
2		PralineCoc +	PralineName 👻	ChocolateTy -	NutType 🗸	FillingType -	${\sf PralineDescription}~ {\boldsymbol{\star}}$	PralineCo 🔻
	+	B01	Candlelight Ecsta	Dark	Cashew	Mocha cream	Cashew in mocha crea	\$ 0.30
	+	B02	Butterscotch Blue	Butterscotch	None	Blueberry	Blueberry with a layer	\$ 0.25
	+	B03	Marzipan Oakleaf	Butterscotch	None	Marzipan	Marzipan in the shape	\$ 0.40
	+	B04	Butterscotch Stra	Butterscotch	None	Strawberry	Strawberry with a laye	\$ 0.23
	+	R05	Rutterscotch Rase	Rutterscotch	None	Rachharny	Rachharry with a lavor	¢ n 25

Figure	2.7.:	Table	Pralines.
--------	-------	-------	-----------

The Pralines table stores properties of each type of praline. The key is the PralineCode field. The table contains 41 records (praline types).

You can easily perform some calculations based on the data from these tables.

## 1. What is the average number of praline types per box?

Divide the total number of records in the BoxDetails table by the total number of box types:  $\frac{84}{18} = 4.7$ .

## 2. What is the average number of box types per order?

Divide the total number of order rows by the total number of orders:  $\frac{1537}{784} = 2.0$ .

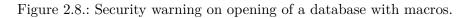
You can't easily calculate the average number of pralines in a box. To do this, you need the total number of pralines in all boxes, which is the sum of all values in the Quantity column in the BoxDetails table.

# 2.2. Security and Macros

How to handle security warnings when opening a database.

Some databases, such as candy365.accdb, contain macros. In most cases, these macros are designed to perform specific tasks in the database. However, virus writers can use macros to spread malware. When a database is not in a trusted location or signed by a trusted publisher, Access displays a security warning when opening the database, and macros are disabled.

SECURITY WARNING Some active content has been disabled. Click for more details. Enable Content



You can enable macros in one of the following ways.

## Enable Macros Temporarily

Click the Enable Content button in the area with the security alert.

This method is not recommended because you will have to enable macros every time you open the database and see this warning.

## Add Publisher to Trusted Publishers List

If the database creator has provided a digital certificate, you can add the creator to the Trusted Publishers list. Access will automatically enable all macros created by this publisher in all databases. This is a very secure method, but usually only larger companies use digital certificates. The database candy365.accdb does not have a digital certificate.

## Move Database to a Trusted Location

Access provides trusted locations (folders). Macros in databases stored in a trusted location are automatically enabled. This is the most convenient and recommended way to work safely without interruptions from security alerts for the database candy365.accdb. To do this:

Go to File > Options > Trust Center > Trust Center Settings... > Trusted Locations > Add new location...

In this example, the folder C:\temp is added as a trusted location.

Microsoft Office Trusted Location		?	×			
Warning: This location will be treated as a trusted source for opening files. If you change or add a location, make sure that the new location is secure. <u>Path</u> :						
C:\temp						
Subfolders of this location are also trusted Description:		<u>B</u> rov	vse			
Date and Time Created: 9/17/2017 5:16 PM	ОК	Car	ncel			

Figure 2.9.: Dialog box Microsoft Office trusted location.

## Security Settings for All Macros

Optionally, you can change how Access handles all macros in all databases:

Go to File > Options > Trust Center > Trust Center Settings... > Macro Settings

Trust Center	? ×
Trusted Publishers	Macro Settings
Trusted Locations	
Trusted Documents	<ul> <li>Disable all macros without notification</li> <li>Disable all macros with notification</li> </ul>
Trusted Add-in Catalogs	O Disable all macros except digitally
Add-ins	signed macros
ActiveX Settings	<ul> <li><u>Enable all macros (not recommended;</u> potentially dangerous code can run)</li> </ul>
Macro Settings	
Message Bar	
Privacy Options	
	OK Cancel

Figure 2.10.: Dialog box Trust Center.

It is not recommended to enable all macros, as this removes important security controls.

# 2.3. Exploring the Database

To explore, you must have the database candy365.accdb open.

## i Note

If a security warning is displayed, you should take action as described in Section 2.2.

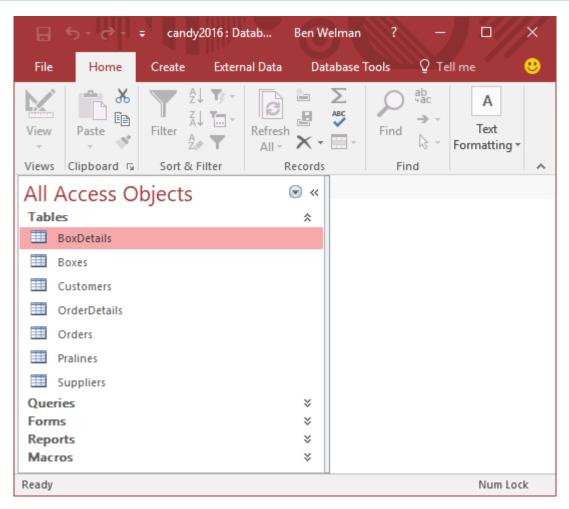


Figure 2.11.: Access with the Snoopy database with opened group Tables.

The Navigation Pane is on the left side and contains all objects in the database. Objects are divided into groups: Tables, Queries, Forms, Reports, Macros. In the example, the Tables group is expanded. The navigation pane and its groups can be expanded or collapsed.

- $\odot$  select which objects are shown
- $\boxtimes$  expand a group pane
- $\boxtimes$  collapse a group pane
- $\leq$  collapse the navigation pane
- $>\!\!>$  expand the navigation pane

Double-clicking an object in the Navigation Pane opens it and displays its contents in a document window. If you open several objects, they are displayed as tabbed documents.

Customers Orders						
2		OrderCode 👻	CustomerCode 👻	OrderDate 👻		
	÷	1	237	11/2/2009		
	÷	3	158	11/2/2009		
	+	6	1	11/2/2009		
	+	7	275	11/2/2009		

Figure 2.12.: Document window with the Customers and Orders tables in tabs.

## i Note

You can change this setting through File > Options > Current Database. In Figure 2.13, you can see the available document window options.

Document Window Options O Overlapping Windows Tabbed Documents Display Document Tabs

Figure 2.13.: Document Window Options.

If you select Overlapping Windows, objects are displayed in separate windows, overlapping each other.

If you need more horizontal space, you can collapse the navigation pane. If you need more vertical space, you can hide the ribbon by double-clicking a ribbon tab. Double-clicking again will show the ribbon.

# 2.4. Exploring Tables

Tables are the key components of a database, as all data is stored here.

The two most important views of a table are:

Datasheet View In this view, you can see and edit the contents of records, and add new records.

**Design View** In this view, you can see and modify the design of the table.

Task 2.1. FILE: candy365.accdb

- 1. Open the database.
- 2. Open the Customers table. The table opens in Datasheet View.
- 3. Switch the Customers table to Design View in one of the following ways:
  - Click the Design View button in the bottom right of the program window 🔛
  - Choose Home tab > View (Views group) > Design View
  - Right-click the table name in the navigation pane and choose Design View from the shortcut menu.

4. Switch the Customers table back to Datasheet View in one of the following ways:

- Click the Datasheet View button in the bottom right of the program window  $\square$ 

- Choose Home tab > View (Views group) > Datasheet View
- Right-click the table name in the navigation pane and choose Open
- 5. Close the Customers table using the Close button X at the top right of the document window.
- 6. Open the Orders table in Datasheet View.

### i Note

In the Orders table, there is an expand button + in front of the records. Clicking it shows the order details, which come from the [OrderDetails] table. The associated details can be found because both tables contain the [OrderCode] field.

7. Click the expand button on a few records to see which boxes and how many were ordered.

	Orders								
2		Or	derCode	Ψ.	CustomerCo	de	Ŧ	OrderDate	-
	+			1		- 1	237	11/2/2	009
	무			3		1	158	11/2/2	009
		Z	BoxCode	è 👻	Quantity	Ŧ	Clic	:k to Add 👻	
			CHER			2			
			NOWI			1			
		*							
	+			6			1	11/2/2	009
	+			7			275	11/2/2	009

Figure 2.14.: Orders details of order 3.

8. Close the Orders table.

## 2.5. Exploring Forms

Forms are especially important for displaying, adding, and editing data.

The main views of a form are:

Form View In this view, information is displayed, and you can edit or enter new data.

Datasheet View A view similar to that of the table.

**Design View** In this view, you can see and modify the design of the form.

Task 2.2. FILE: candy365.accdb

- 1. Open the database.
- 2. Open the Pralines form. The form opens in Form View.
- 3. Switch the Pralines form to Design View in one of the following ways:
  - Click the Design View button in the bottom right of the program window  $\bowtie$ .
  - Choose Home tab > View (Views group) > Design View

	Pralines		
	Pralines		
•	Praline name:	Candlelight Ecstasy	Praline code: B01
	Description:	Cashew in mocha cream with a layer butterscotch chocolate.	
	Chocolate type:	Butterscotch Puur Toffee Melk	
	Filling type:	Mocha cream 🗾	
	Nut type:	Cashew 🔽	
	Costs:	\$0.30	

Figure 2.15.: Form View of form Pralines. The data for the first record is displayed.

• • • • • • • • 1		2 * * * 1 * * *	• 3 • • • 1 • • •	4 • • •
Form Header				
Pralines				
✓ Detail				
Praline name:	PralineName		Praline code:	Praline
Description:	PralineDescriptio	pn		
Chocolate type	ChocolateType			
<sup>2</sup> Filling type:	FillingType	•		
Nut type:	NutType	<u> </u>		
Costs:	PralineC			
Form Footer				

Figure 2.16.: Design View of form Pralines.

• Right-click the form name in the navigation pane and choose Design View from the shortcut menu.

#### **i** Note

In Design View, you can change the layout of the form and add controls such as text boxes, labels, list boxes, checkboxes, and more.

- 4. Switch the Pralines form back to Form View using  $\blacksquare$ .
- 5. Browse through the records using the navigation buttons at the bottom left of the document window.
  - III First record
  - • Previous record
  - • Next record
  - • Last record
  - 🖬 New (empty) record
- 6. Close the form.

# 2.6. Using a Command Button

The Boxes form can be used for data entry as well as for viewing data. This form includes a command button labeled Sales.

#### Task 2.3. FILE: candy365.accdb

- 1. Open the database.
- 2. Open the Boxes form.
- 3. Click the Sales button on the form. The Box sales form is displayed. You will see all orders related to the selected box in the Boxes form, including the total sales for that box.
- 4. Close all forms.

## 2.7. Searching for a Record

A form can also be used to search for a record. In the following steps, you will search for a box with "autumn" in its name.

#### Task 2.4. FILE: candy365.accdb

- 1. Open the database.
- 2. Open the Boxes form.
- 3. Click in the Box name field and, on the ribbon, choose Home tab >  $\stackrel{\frown}{P}$  Find (Find group). The Find and Replace dialog box appears.

4. Type "autumn" in the Find What text box and select Any Part of Field in the Match box.

Find and Replace ? ×					
Find Replac	e				
Find What:	autumn	Find N	Vext		
Look In: Match: Search:	Current field Any Part of Field All Match Case Search Fields As Formatted	Can	cel		

Figure 2.17.: Dialog box find and replace.

- 5. Click Find Next. The content for the box "Autumn Surprise" is displayed.
- 6. Click Cancel to close the search window.
- 7. Close the form.

# 2.8. Exploring Queries

Queries are important for selecting or summarizing data from tables. A query is a specific question to the database to retrieve certain information.

The two most important views of a query are:

Datasheet View In this view, you can see the result (data) of the query.

Design View In this view, you can see and modify the design of the query.

Task 2.5. FILE: candy365.accdb

- 1. Open the database.
- 2. Open the Pralines per box query.

#### i Note

Access executes the query and displays a datasheet with the results, as shown in Figure 2.18. The results of this query are derived from multiple tables. To see which tables are used, switch to the query's Design View.

3. Switch the Pralines per box query to Design View in one of the following ways:

- Click the Design View button in the bottom right of the program window  $\bowtie$ .
- Choose tab Home > View (group Views) > Design View.
- Right-click the query name in the navigation pane and choose Design View from the shortcut menu.

4	Box	~	Praline 👻	PralineCosts 👻
	All Seasons		Butterscotch Raspberry	\$0.50
	All Seasons		Butterscotch Strawberry	\$0.46
	All Seasons		Sweet Blueberry	\$0.50
	All Seasons		Sweet Raspberry	\$0.52
	All Seasons		Sweet Strawberry	\$0.40
	All Seasons		Walnut Mocha Toffee	\$0.38
	Autumn Surprise		Marzipan Finch	\$1.92
	Autumn Surprise		Marzipan Maple Leaf	\$2.22
	Autumn Surprise		Marzipan Oakleaf	\$2.40
	Autumn Surprise		Marzipan Swallow	\$2.04
	Rutterscotch		Rutterscotch Cherry	\$1 0A

Figure 2.18.: Query pralines per box in datasheet view.

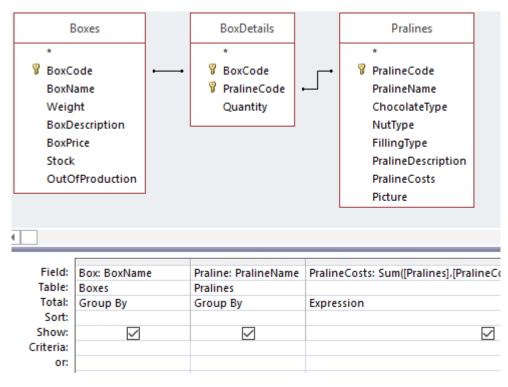


Figure 2.19.: Design query Pralines per box.

## i Note

The upper three rectangles represent the tables used in this query. The lines with arrows connect the field names by which the tables are related.

In the lower part of this window, the first row contains the names of the fields used. In the third column is an expression, a formula that performs operations on the fields. The expression here is Sum([Pralines].[PralineCosts]\*[BoxDetails].[Quantity]). The names in brackets indicate the tables and fields.

To see the entire expression, widen the column by dragging its right edge.

- 4. Switch the query back to Datasheet View.
- 5. Close the query

# 2.9. Exploring Reports

Reports allow you to print information from tables in a well-organized layout, either on screen or on paper. The displayed data may come from multiple tables and/or queries, and calculated values are possible. You can also add titles, headings, headers, and footers to a report.

The main views of a report are:

**Report View** In this view, you can see the contents of the report.

**Print Preview** Shows how the report will look when printed.

**Design View** In this view, you can see and modify the design of the report.

## Task 2.6. FILE: candy365.accdb

- 1. Open the database.
- 2. Open the Pralines per box report. The report opens in the Report View.
- 3. Switch the Pralines per box report to Print Preview in one of the following ways:
  - Click the Print Preview button in the bottom right of the program window 🗔
  - Choose Home tab> View (Views group) > Print Preview.
  - Right-click the report name in the navigation pane and choose Print Preview from the shortcut menu.
- 4. Switch the Pralines per box report to Design View in one of the following ways:
  - Click the Design View button in the bottom right of the program window 🕍
  - Choose Home tab> View (Views group) > Design View.
  - Right-click the report name in the navigation pane and choose Design View from the shortcut menu.

## i Note

In Design View, you can change the layout of the report and add controls such as text boxes, labels, list boxes, checkboxes, and more.

# Pralines per box

04-Aug-15



Box name	Praline name	Praline cost	
All Seasons			
	Butterscotch Raspberry	\$0.50	
	Butterscotch Strawberry	\$0.46	
	Sweet Blueberry	\$0.50	
	Sweet Raspberry	\$0.52	
	Sweet Strawberry	\$0.40	
	Walnut Mocha Toffee	\$0.38	
		\$ 2.76	
Autumn Surprise			
	Marzipan Finch	\$1.92	
	Marzipan Maple Leaf	\$2.22	
	Marzinan Oakleaf	\$2.40	

Figure 2.20.: Report pralines per box in report view.

-	Pralines per b	ox					
		(1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	2 • • • 1 • • • :	3 • • • • •	· · 4 ·		5 • • • 1 • • •
	FReport Hea	der					
	Pralines	per box	1	1			+000
	=Now()						
÷	F Page Heade	er (			I		
÷	Box name		Pralir	ne name		Pra	ine cost
	€ =[Box] Head	der					
-	Box						
	🗲 Detail		1				
•			Pralin	e		Pral	ineCosts
		er	1				
:						=Su	m([PralineCost
		r	1	1	1		
							=Page
İ	FReport Foot	ter	1	1	1		-5-
•					Tota	l costs:=Su	m([PralineCost

Figure 2.21.: Report pralines per box in design view.

- 5. Switch the report back to [Report View] using  $\blacksquare$ .
- 6. Close the report

## 2.10. Sorting

You can sort the records in a table based on the values in one or more fields, either in ascending or descending order.

In the next exercise, you will modify the Customers table to create an overview sorted first by city, then by last name, and then by first name.

	City 🚽	LastName 🚽	FirstName 🗃	CustomerCode 👻	Ad
+	Alkmaar	Speelman	Marjan	73	Burg.Be
+	Almelo	Molen, van de	Robert	15	Amarils
+	Almelo	Steenhuis	Peter	377	Meierij
+	Almelo	Werkman	Jan	132	Dr. van l
+	Almere	Oosting	Olivia	388	Monpla
+	Almere	Tijssen	Jacqueline	299	Kenned
+	Amersfoort	Brouwer	Wim	307	Kievitst
+	Amersfoort	Dijk, van	Nancy	302	Ketelwe
+	Amersfoort	Dijkstra	Paula	64	Bruister
+	Amersfoort	Fransen	Sam	277	Institute

Figure 2.22.: Table Customers, sorted on city, last name and first name.

#### Task 2.7. FILE: candy365.accdb

- 1. Open the database.
- 2. Open the Customers table. The table opens in Datasheet View.
- 3. Select the City column by clicking its header and drag it to the left to make it the first column.
- 4. Similarly, move the LastName and FirstName columns to the second and third positions, respectively.
- 5. Click the arrow on the right side of the City column header and select Sort A to Z from the shortcut menu.

#### i Note

Access reorganizes the records in alphabetical order by city and shows a small upward-pointing arrow  $(\neg 1)$  at the right side of the column header to indicate the sort order.

- 6. Remove the sorting by choosing Home tab > Remove Sort (Sort & Filter group).
- 7. To sort by multiple fields, select the City, LastName, and FirstName columns.
- 8. Choose Home tab > Ascending  $2 \downarrow$  (Sort & Filter group).

## i Note

Access reorganizes the records in alphabetical order by city and shows a small upward-pointing arrow (-1) at the right side of the column header to indicate the sort order.

9. Close the Customers table and choose No when asked to save changes.

# 2.11. Filtering

Filtering is a useful way to display only the data you want to see. You can use filters to display specific records in a form, report, query, or table. For example, you can easily create a filter to show all pralines with chocolate type "Milk" and filling type "Marzipan." Several filtering methods are discussed in the following exercises.

## 2.11.1. Simple Filter

Information needed: Show all pralines with chocolate type "White."

Task 2.8. FILE: candy365.accdb

- 1. Open the database.
- 2. Open the Pralines table. The table opens in Datasheet View.
- 3. Click a value "White" in the ChocolateType column.
- 4. Choose Home tab > Selection (Sort & Filter group) > Equals "White".

#### **i** Note

Access now shows the records (4 records) where the chocolate type is white. You can see that a filter has been applied:

- A small filter icon appears to the right of the ChocolateType column header  $(\mathbf{T})$
- The status bar shows Record: M 1 of 4 M Filtered Search

## 2.11.2. Modifying a Filter

Information needed: Show all pralines with chocolate type "Milk."

Here, you will modify the previous filter.

#### Task 2.9. FILE: candy365.accdb

- 1. Click the filter icon to the right of the ChocolateType column header.
- 2. In the dialog box, select "Milk" and deselect "White." Click OK. Now, 18 records are shown, all with chocolate type "Milk."

### 2. Exploring Candy database

Az↓ Z↓ A↓	<u>S</u> ort A to Z S <u>o</u> rt Z to A
*	Clear filter from ChocolateType
~	Text <u>F</u> ilters ►
	<ul> <li>(Select All)</li> <li>(Blanks)</li> <li>Butterscotch</li> <li>Dark</li> <li>Milk</li> <li>Toffee</li> <li>White</li> </ul>
	OK Cancel

Figure 2.23.: Dialog box filter.

### **i** Note

The options shown in Figure 2.23 depend on the field type. For a text field, there is a Text Filters submenu; for a numeric field, there is a Number Filters submenu.

### 2.11.3. Filter on Two Criteria

**Information needed:** Show all pralines with chocolate type "Milk" and filling type "Marzipan."

Here, you add a second selection criterion.

### Task 2.10. FILE: candy365.accdb

- 1. Click a value "Marzipan" in the FillingType column.
- 2. Choose Home tab > Selection (Sort & Filter group) > Equals "Marzipan".

Three records are displayed with chocolate type "Milk" and filling type "Marzipan."

3. Remove the filter by choosing Home tab >  $\mathbf{Y}$  Toggle Filter (Sort & Filter group).

### **i** Note

Filtering is removed and all records are displayed again.

Another way to remove the filter is by clicking the Filtered button in the status bar. The text on the button will change to Unfiltered. Clicking again reapplies the last used filter.

### 2.11.4. Number Filter

Information needed: Show all pralines with costs from \$0.25 to \$0.35.

Task 2.11. FILE: candy365.accdb

1. Click the arrow in the PralineCosts column header, then choose Number Filters > Between....

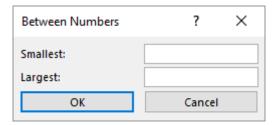


Figure 2.24.: Between Numbers dialog box.

- 2. Enter 0.25 for Smallest, 0.35 for Largest, then click OK. Now, 22 records are shown with praline costs from 0.25 to 0.35.
- 3. Close the Pralines table and choose No when asked to save changes.

# 2.12. Printing

Access uses the printers available in Windows for printing.

Depending on the view of an Access object, the result can be printed. For example, you can print datasheet views for tables and queries, report views for reports, and form views for forms.

i Note

There is no option to print the design view, but Access has a tool called Database Documenter that lets you print the design characteristics of database objects.

Print options are available through File > Print. You have the following options:

With Print Preview, you can customize the printout, such as page size, orientation, margins, and more.

# 2.13. Exercises

### Exercise 2.1. Select Pralines (expl001)

Create a filter to select pralines with chocolate type "Milk" or "Dark" whose price is at most 0.30.

### Exercise 2.2. Select Customers in Specific Cities (expl002)

Select the customers who live in Enschede, Hengelo, and Almelo.

### 2. Exploring Candy database



# Quick Print

Send the object directly to the default printer without making changes.



### Print

Select a printer, number of copies, and other printing options before printing.



#### Print Preview

Preview and make changes to pages before printing.

Figure 2.25.: Print options in Access.

## Exercise 2.3. Select Customers by Zip Code (expl003)

Select customers whose zip code begins with 20 and who live in Amsterdam.

# 3. Data Entry and Modification

### **OBJECTIVES**

- Using a data entry form.
- Editing, adding and deleting records in a table.
- Using a main form with a subform.
- Understanding options to enforce data entry validation.

Adding new data or changing existing data is one of the primary activities within Access. In this chapter, you will learn some basic skills for adding and/or modifying data.

# 3.1. About Data Entry

Records can be edited directly in a table, but this is not recommended for individuals unfamiliar with Access. Incorrect entries can easily be made, especially in tables related to other tables. The recommended method is to perform data entry through a form. Forms can be made user-friendly, provide explanatory text, and perform data validation in the background.

### 3.2. Table Operations

You can perform the most important data operations directly in the table, such as editing, adding, and deleting records. You can type directly into fields; standard copy and paste methods are also supported.

### **Editing Records**

Open the table, then use the mouse or arrow keys to navigate to the field you want to change. Click in the field and make the necessary changes. Once you make a change, a pencil icon will appear on the left side of the record,  $\checkmark$ . This indicates that the record is in **edit mode**. Once you leave the record, the icon disappears, and the changes are saved.

#### **Adding Records**

Navigate to the bottom of the table, to the row marked with an asterisk: \*. This row doesn't actually exist until you begin entering data, at which point it is created, and the asterisk moves down one row. The new record is saved automatically.

### **Deleting Records**

There are several ways to delete records. The two easiest methods are:

- Select the record, then press the Delete key.
- Right-click in the margin at the beginning of the record, then choose Delete Record. Access will ask for confirmation before deleting.

### 🛕 Warning

Access does not have a button or function to undo deletions.

# 3.3. Add a New Customer

The Registration form is designed exclusively for adding new records to the Customers table. Browse existing records is disabled in this form.

### Task 3.1. FILE: candy365.accdb

- 1. Open the database.
- 2. Open the Registration form.

### i Note

The Registration form contains blank fields for data entry. The insertion point indicates where the typed text will appear. You can move the insertion point by clicking on another field or by repeatedly pressing the TAB key.

3. Enter the data as shown in Figure 3.1.

*****		<i>The chocolate connoisseur</i> Registration
\$	Firs	tName LastName
	Name Eric	a Lauwers
	Address Slui	s 33
	ZipCode	9974 RH City Zoutkamp
	Province	GR Region North

Figure 3.1.: Data entry with form Registration.

### i Note

You will see the pencil icon again, indicating that the form is in edit mode and the data has not yet been saved.

The record is automatically saved when you close the form or navigate to another record.

- 4. Close the form using the close button X in the upper right corner of the document window. The records added through the Registration form are stored in the Customers table.
- 5. Open the Customers table, navigate to the last record, and verify that this is the record you just added.
- 6. Close the Customers table.

# 3.4. Add a New Praline

The Pralines form allows you to browse existing records and add new ones.

### Task 3.2. FILE: candy365.accdb

- 1. Open the database.
- 2. Open the Praines form.
- 3. In the status bar, click the New (blank) record button.
- 4. Enter the following data:
  - PralineName: Pecan Mocha Toffee
  - PralineCode: F03
  - PralineDescription: Sweat creamy mocha and pecan, surrounded with toffee
  - ChocolateType: Toffee
  - FillingType: Mocha cream
  - NutType: Pecan
  - PralineCosts: 0.25
- 5. Close the Pralines form. The records added through the Pralines form are stored in the Pralines table.
- 6. Open the Pralines table and ensure the record has been added.
- 7. Close the Pralines table.

# 3.5. Add a New Box

You can add a new box using the Boxes form. This form differs from the previous two because it includes another form within it. This structure is known as a main form with a subform.

- Main Form (Boxes) This form contains data about the box itself (not its contents), such as BoxName, BoxCode, BoxDescription, Weight, BoxPrice, and Stock. This data is stored in the Boxes table.
- **Subform (Boxes subform)** This form contains data about the pralines in the box, such as PralineCode and Quantity. This data is stored in the BoxDetails table.

### i Note

The content of the BoxCosts field is calculated by summing PralineCosts \* Quantity for all pralines in the box. Because this value can be calculated, it is not stored in a table

Task 3.3. FILE: candy365.accdb

- 1. Open the database.
- 2. Open the Boxes form.
- 3. In the status bar, click the New (blank) record button w.
- 4. Enter the following data:
  - BoxName: Winter Surprise
  - BoxCode: WINT
  - BoxDescription: Nuts and berries, covered with chocolate and toffee, ideal for long winter evenings by the fire.
  - Weight: 300 gram
- 5. Below "Content," in the Code field, click the selection arrow and select PralineCode B02 Butterscotch Blueberry. The remaining praline data, such as PralineName, ChocolateType, NutType, FillingType, and PralineCosts, will be filled in automatically. You only need to enter the data for Quantity (Nr.).
- 6. Enter 3 for Nr..
- 7. Add the following praines to the box:

Code	PralineName	Nr.
B05	Butterscotch Raspberry	3
P03	Cashew Perfect	3
F01	Walnut Mocha Toffee	3
F02	Pistachio Mocha Toffee	$\mathcal{3}$
P07	Classic Cherry	3

The BoxCosts of \$4.53 is automatically calculated.

- 8. Enter 25.00 for BoxPrice and 120 for Stock.
- 9. Close the Boxes form.
- 10. Check the Boxes table to ensure the new box has been added.

# 3.6. Data Restriction and Validation

Access offers various options to control how data is entered into the database. For example, you can limit the data that can be entered in a field by defining a validation rule for that field. If the entered data does not satisfy the rule, an error message will inform you of the permitted input. Another method for controlling data input is using an input mask, which defines a format for data entry in a field using characters and symbols.

You can apply these simple validation and restriction methods by setting properties for fields in tables or properties for controls on forms.

It is recommended to define data validation and restriction in the table design by setting field properties. This saves time because every time you use the field in a form, the field validation rules and other properties are also applied to the form. If the data entered through a form control is not dependent on a table field, then you need to define properties on the form for validating and restricting data.

### Task 3.4. FILE: candy365.accdb

A memo from Snoopy's marketing department states that the price of the Butterscotch box should be changed to \$7.75.

- 1. Open the database.
- 2. Open the Boxes form and navigate to the Butterscotch box.
- 3. Change BoxPrice from \$27.75 in \$7.75 and press the TAB key. A validation message appears stating that this price is not right and how you can fix it.

Box price too low, should be minimal 2\*BoxCosts. Contact marketing department for further information.

### i Note

This message appears because a validation rule is associated with the BoxPrice field.

- 4. Click OK.
- 5. Change the price back in \$27.75 and press the TAB key. Then, close the form.

### 3.7. Exercises

#### Exercise 3.1. Adding Pralines (data001)

Add the following pralines:

Field	Praline 1	Praline 2
PralineName	Cashew Mocha Toffee	Almond Mocha Toffee
PralineCode	F04	m F05
Description	Sweet creamy mocha and cashew surrounded with toffee.	Sweet creamy mocha and almond surrounded with toffee.
ChocolateType	Toffee	Toffee
FillingType	Mocha cream	Mocha cream
NutType	Cashew	Almond
Costs	\$ 0.24	\$ 0.19

#### Exercise 3.2. Add New Order (data002)

A customer places a new order with the data below. First, determine the best way to enter the order. Then, create this order. The OrderCode should be the next available order code.

Order placed on April 23, 2010, at 10:00 AM. Customer Rebecca Smit orders by phone: 3 boxes of CHER and 2 boxes of Marz.

A few hours later, the following change to this order is made by phone. Again, first, consider the best way to handle the change. Then, apply the change.

Order change on April 23, 2010, at 4:00 PM. The two boxes of Marz should be deleted.

### **OBJECTIVES**

- Understand the organization of tables and the relationships between them.
- Learn the various methods for creating tables.
- Explore field properties.
- Validate data input using input masks and validation rules.
- Understand referential integrity.

### 4.1. About Tables

In a database, data is stored in tables. Each table consists of fields for different types of data. For example, there might be a field for a person's last name and another for their first name. While data can be displayed in various ways, it is always stored in tables. In practice, you begin by conducting an information analysis to determine which tables and fields are needed. For larger databases, this is often done using a process called normalization. Typically, relationships are also established between tables, allowing data in one table to be linked to data in another.

Empty tables serve no purpose unless data is entered. If the data already exists in other files, such as Excel spreadsheets, it can be imported directly into Access.

### 4.2. Fields in Tables

Fields define the structure of a table. Each piece of data is stored in a field, and each field has properties that determine its behavior. To create a field, you need:

- A field name (required)
- A data type (required)
- A description (optional)

### 4.2.1. Field Names

The database design determines the kind of data each field will store. Often, field names are predetermined, but you should always use meaningful names. Although descriptions can be added, clear names are usually more effective.

Field names must:

- Be unique within a table
- Not start with a space
- Not contain the following characters: period (.), exclamation point (!), or brackets ([])

### 4.2.2. Data Types

The data type defines what kind of data can be stored in a field. This is a critical property. For instance, a Text field can contain letters and numbers, while a Number field can only store numeric data. A good rule of thumb: if you need to perform calculations with the field, use the Number data type. For example, a field for item prices should be a Number, while a product ID might be a Text field.

Data Types	Values	Explanation
Short Text	Alphanumeric	Up to 255 characters.
Long Text	Alphanumeric	Up to 1GB of text (formerly called Memo).
Number	Numeric	Range: $-2^{31}$ to $2^{31}$ - 1.
Large Number	Numeric	Range: $-2^{63}$ to $2^{63}$ - 1.
Date/Time	Dates and Times	Used for storing calendar dates and times.
Currency	Monetary values	Stored with four decimal points of precision.
AutoNumber	Auto-generated	Unique numbers automatically assigned per record.
Yes/No	Boolean	Yes/No, True/False, or On/Off.
OLE Object	External objects	Includes images, graphs, Excel sheets, Word docs, etc.
Hyperlink	Hyperlinks	URLs and email addresses.
Attachment	File references	Can store multiple files (e.g., photos) per record.
Calculated	Derived values	Results from an expression using other fields.

#### Table 4.1.: Data types in Access

### i Note

You can also choose Lookup Wizard as a data type. This launches the Lookup Wizard, which lets you define a field that displays values from another table or a custom list.

### 4.2.3. Field Properties

Field properties provide additional control over how data behaves. The available properties depend on the selected data type. The most common properties include:

- Field Size Controls the length of text or the numeric range. For text, it sets the maximum character count. For numbers, you can select Byte, Integer, Long Integer, Single, Double, or Decimal.
- **Format** Determines how data is displayed in tables, queries, forms, and reports. Formats vary by data type and can be customized or selected from a list.

Decimal Places Used with Number and Currency fields to set the number of decimal digits.

**Default Value** Automatically fills a field with a preset value when a new record is created. You can still modify the value manually.

Required Specifies whether a value must be entered. Options are Yes or No (default).

i Note

Special field properties like Input Mask, Validation Rule, and Validation Text help enforce data consistency and accuracy.

### 4.2.4. Number Field Sizes

Possible values for the Number data type include:

- Byte: 0 to 255
- Integer: -32,768 to +32,767
- Long Integer: -2,147,483,648 to +2,147,483,647
- Single: Approx.  $-3.4 \ge 10^{38}$  to  $+3.4 \ge 10^{38}$  (7-digit precision)
- Double: Approx. -1.797 x  $10^{308}$  to +1.797 x  $10^{308}$  (15-digit precision)
- Decimal: Approx. -9.999 x  $10^{27}$  to +9.999 x  $10^{27}$

### 4.3. Validation

To ensure users enter data in a specific format, you can apply input masks. You can also define validation rules to check if values meet certain conditions and, if not, display a helpful validation message.

#### 4.3.1. Input Masks

An input mask enforces a predefined format for data entry in a field. It consists of a combination of characters and symbols that define the pattern. For instance, an input mask can ensure that a phone number has exactly 10 digits. Input masks can be used with Text, Number, Currency, and Date/Time fields.

An input mask consists of one required and two optional parts, separated by semicolons:

- First part (required): Defines the structure, including placeholders and literal characters such as parentheses or hyphens.
- Second part (optional): Indicates whether the literal characters are stored. 0 stores them; 1 displays them without storing.
- Third part (optional): Sets the placeholder character (default is underscore \_).

i Note

Setting the second part to 1 can save database space.

Character	Meaning
0	Required digit (0-9)
9	Optional digit (0-9)
#	Optional digit, plus, minus, or space
L	Required letter
?	Optional letter
А	Required letter or digit
a	Optional letter or digit
&	Required character or space
С	Optional character or space
. , : ; - /	Decimal and thousands placeholders, date and time separators. The
	character you select depends on your Microsoft Windows regional
	settings.
>	Convert following characters to uppercase
<	Convert following characters to lowercase
!	Fill from left to right instead of right to left
$\setminus$	Escape next character
	Characters in quotes are displayed literally

Table 4.2.: Input mask symbols and meanings

You can create input masks manually or by using the Input Mask Wizard.

### Example 4.1. US Telephone Number

Input mask: (999) 000-000;0;-

Explanation:

- (999): optional three-digit area code
- 000–000: required seven-digit number
- ;0: stores literal characters
- ;-: uses hyphen (-) as the placeholder

### Example 4.2. Zip Code

Input mask: 0000 >LL

Explanation:

- $\bullet~$  0000: four required digits
- $\$ : literal space
- >LL: two required uppercase letters

### 4.3.2. Validation Rule and Validation Text

Validation rules restrict the values that can be entered into a field. If the entered data violates the rule, Access displays the validation text.

Validation Rule An expression that must evaluate to true. If not, the input is rejected.

Validation Text A message shown when a value violates the validation rule.

🔮 Tip				
Syntax Tips:				
• Field names go in square brackets: [OrderDate]				
• Dates use <b>#</b> symbols: <b>#12/31/2024#</b>				
• Text uses quotes: "New York"				
• Wildcards:				
<ul> <li>- ? = one character</li> <li>- * = any number of characters</li> <li>- # = one digit</li> </ul>				

Table 4.3 lists the most used operators.

Operator	Purpose	Example
<	Less than	<100
<=	Less than or equal	<=15
>	Greater than	>10
>=	Greater than or equal	>=0
=	Equal to	=21
<>	Not equal to	<>0
AND	Both conditions true	>=1 AND $<=9$
OR	Either condition true	"m" OR "f"
NOT	Condition not true	NOT $>10$
IN	Value in list	IN ("Berlin", "London", "Paris")
BETWEEN	Value in range	BETWEEN 1 AND 9
LIKE	Matches pattern	LIKE "Ams*"

### Example 4.3. Date Validation

- Rule: >=#1-1-2010#
- Text: Enter a date from January 1, 2010 onward.

#### Example 4.4. Invoice Number Validation

- Rule: Like "[0-9][0-9][0-9][0-9]"
- Text: Invoice number must be exactly four digits.

#### Example 4.5. Simple Email Validation

- Rule: Like "\*@\*.???"
- Text: Enter valid email address.

### 4.4. Create a New table Manually

Purpose: Create a new table from scratch and assign a primary key.

You need to create a new table named Transport Companies to store contact details for transportation companies.

### Task 4.1. FILE: candy365.accdb

- 1. Open the database.
- 2. Go to Create tab > Table (Tables group). A new table will be created with one default field named ID, which you can delete if not needed.
- 3. Switch to Design View. Since the table is not yet named, the Save As dialog box will appear.

Save As		?	$\times$
Table Name:			
Table1			
	ОК	Ca	ncel

Figure 4.1.: Save As dialog box.

- 4. Enter the name Transport Companies and click OK. The table Transport Companies will now open in Design View.
- 5. Right-click the ID field and choose Delete Rows. Confirm the deletion.
- 6. Add the following fields:

	Field Name	Data Type	
P	CompanyName	Short Text	Name of the transport company
	Car	Yes/No	Yes=transport by car, No=transport by plane
	Address	Short Text	Address of the company
	ZipCode	Short Text	Zipcode (4 digits and 2 letters)
	City	Short Text	City of the company
	ContactName	Short Text	Name of the contact
	ContactTelephone	Short Text	Telephone number of the contact

Figure 4.2.: Field setup for Transport Companies table.

Since the company name is unique, you can use it as the primary key.

- 7. Select the first row and click Design tab > Primary Key (Tools group). A key icon  $\[mathbf{g}\]$  will appear at the beginning of the row.
- 8. Switch to Datasheet View. When prompted to save the table, click Yes.
- 9. Enter the following records:
- 10. Close the Transport Companies table.

CompanyName 👻	Car 👻	Address -	ZipCode 👻	City 👻	ContactName 👻	ContactTele
Fast Express		Pluvierstraat 3	7654 DD	Rijndorp	Mary Magenta	0123-98765
Pegasus Service	◄	Reigersweg 17	8765 EF	Waalstad	George Green	
Speedy	✓	Mezenlaan 20	4567 AB	Slingeren	Roger Red	021-3456789

Figure 4.3.: Sample records in the Transport Companies table.

## 4.5. Importing an Excel Worksheet as a Table

Purpose: Import an Excel worksheet into Access as a new table.

Access integrates well with Excel, making it easy to import a worksheet as either a new table or into an existing one.

#### Task 4.2. FILE: candy365.accdb

- 1. Open the database.
- 2. Go to External Data tab > New Data Source (Import & Link group) > From File > Excel.

Get External Data - Excel Spreadsheet	?	×
Select the source and destination of the data		
Specify the source of the definition of the objects.		
Eile name: C:\Temp\	Browse	
<ul> <li>Specify how and where you want to store the data in the current database.</li> <li>Import the source data into a new table in the current database.</li> <li>If the specified table does not exist, Access will create it. If the specified table already exists, Access might contents with the imported data. Changes made to the source data will not be reflected in the database.</li> <li>Append a copy of the records to the table: BoxDetails</li> <li>If the specified table exists, Access will add the records to the table. If the table does not exist, Access will changes made to the source data will not be reflected in the database.</li> <li>Link to the data source by creating a linked table.</li> <li>Access will create a table that will maintain a link to the source data in Excel. Changes made to the source be reflected in the linked table. However, the source data cannot be changed from within Access.</li> </ul>	Il create it.	п
ОК	Cancel	

Figure 4.4.: Get External Data - Excel Spreadsheet dialog box.

- 3. Use the Browse button to locate the file transport.xlsx.
- 4. Select Import the source data into a new table in the current database and click OK. The **Import Spreadsheet Wizard** opens.

- 5. Check First Row Contains Column Headings and click Next. You can now adjust field properties.
- 6. Click Next. You'll be asked to define a primary key.
- 7. Choose No primary key and click Next.
- 8. Enter Transport as the table name and click Finish.
- 9. Click Close when the import is complete.
- 10. Open the new table to review the results, then close it.

### 4.6. Creating a List Box

Purpose: Configure a field to use a list box for data entry.

When only a limited set of predefined values is allowed in a field, a list box is a useful solution. For example, the Customers table has a Region field that should only contain North or South. You'll change the data type to a list box.

### 🛕 Warning

List boxes do not automatically update forms based on the modified table, such as the Registration form. To enable the list box in the form, open the form in Design View, delete the Region field, and add it again. This step is not included in this task.

#### Task 4.3. FILE: candy365.accdb

- 1. Open the database.
- 2. Open the Customers table in Design View.
- 3. Click in the Data Type box for the Region field.
- 4. Click the dropdown arrow and select Lookup Wizard....
- 5. Choose I will type in the values that I want. and click Next.
- 6. Enter "North" in the first cell and "South" in the second.
- 7. Click Finish.
- 8. Switch to Datasheet View and click Yes to save changes.
- 9. Verify that the list box appears in the Region field and works as expected.

### 4.7. Relations Between Tables

A relationship exists between two tables when a key field from one table is linked to a corresponding key field in another table. The related fields typically have the same name and data type. Relationships are established as part of the normalization process, which removes redundant data and organizes it across multiple tables.

When tables are related, you can create queries, forms, and reports that combine information from multiple tables and present it as a single entity.

Lookup Wizard				
	What values do you want to see in your lookup field? Enter the number of columns you want in the list, and then type the values you want in each cell.			
To adjust the width of a right edge of the columr			h you want, or dou	ble-click the
Number of <u>c</u> olumns:	1			
Col1				
South				
*				
	Cancel	< <u>B</u> ack	<u>N</u> ext >	Einish

Figure 4.5.: Lookup Wizard value entry dialog.

Region	*	Tel
North	•	032:
North		
South		

Figure 4.6.: Region list box in action.

### 4.7.1. One-to-Many Relationship

The one-to-many relationship is the most common type. In this relationship, a single row in Table A can be associated with multiple rows in Table B. However, each row in Table B relates to only one row in Table A. This is often referred to as a parent-child relationship.

For example, the Customers table is related to the Orders table via the CustomerCode field. A customer can place multiple orders, but each order belongs to one customer only. Therefore, Customers is on the "one" side, and Orders is on the "many" side of the relationship. Customers is the primary table, and Orders is the related table.

	Customers			Orders		
	CustomerCode 🕞	LastName	2	OrderCode 👻	CustomerCode 🗸	t OrderDate 👻
E	Ð 1	Hendriks 🔫	H	6		1 11/2/2009
E	Ð 2	Pietersen	H	301		1 12/14/2009
E	Ð 3	Calfsbeek	H	661		1 6/12/2010
F	F A	lancon	H	823		1 9/29/2010
			H	214		2 12/2/2009
			H	364		2 12/23/2009
			H	698		2 7/10/2010
			F	3/6		3 12/20/2009

Figure 4.7.: One-to-many relationship between Customers and Orders.

The field used to establish the relationship on the "one" side must contain unique values, usually a **primary key**. The field on the "many" side is known as the **foreign key**.

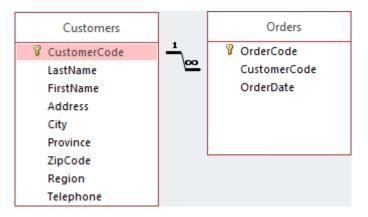


Figure 4.8.: Example of a one-to-many relationship.

In Access, a one-to-many relationship is shown by a line connecting the primary key to the foreign key, with a "1" on one end and an infinity symbol  $(\infty)$  on the many end.

### 4.7.2. One-to-One Relationship

In a one-to-one relationship, each row in Table A corresponds to no more than one row in Table B, and vice versa. This type of relationship is uncommon, as such related data is typically stored in a single table. Sometimes, it's used to split a large table into two smaller ones. Generally, this design should be avoided unless there is a specific reason.

### 4.7.3. Many-to-Many Relationship

In a many-to-many relationship, rows in Table A can match multiple rows in Table B, and vice versa. This type of relationship cannot be directly created in Access. Instead, you must use a third table—called a **junction table** which creates two one-to-many relationships.

For example, consider the relationship between Orders and Boxes. One order may include several boxes, and one box may be part of multiple orders. The OrderDetails table serves as the junction table. It links to Orders through OrderCode and to Boxes through BoxCode, each via a one-to-many relationship.

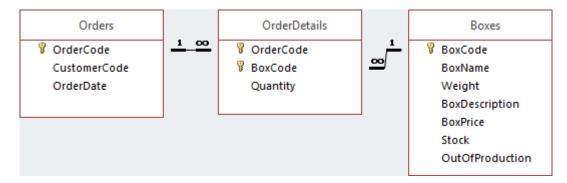


Figure 4.9.: Example of a many-to-many relationship.

### 4.7.4. Creating, Modifying, and Deleting Relationships

Relationships are managed in the *Relationships* window, which can be opened from Database Tools tab > Relationships (Relationships group).

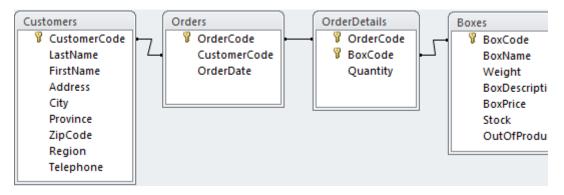


Figure 4.10.: Relationships window.

The ribbon includes commands for editing relationships.



Figure 4.11.: Relationships ribbon.

If no relationships have been created yet, the window will be empty. Click Show Table to add tables and queries to the window.

To create a relationship, drag the primary key from one table to the corresponding foreign key in another. This opens the *Edit Relationships* dialog box.

Edit Relationships			? ×
Table/Query: Boxes BoxCode	Related Table/Query: BoxDetails BoxCode	~	Create Cancel
✓ Enforce Referential Integrity		¥	Join Type Create New
Cascade Update Related Fields Cascade Delete Related Records Relationship Type: One-To-Many			

Figure 4.12.: Edit Relationships dialog box.

To enforce referential integrity, check the Enforce Referential Integrity box.

To modify a relationship, first select the relationship line—it will appear thicker—then doubleclick it to open the *Edit Relationships* dialog box.

To delete a relationship, select the line and press the DELETE key.

### 4.7.5. Referential Integrity

Referential integrity ensures consistency between related tables. When enabled, Access prevents changes that would break the link between related records.

You can enable referential integrity via the *Edit Relationships* dialog box by checking the Enforce Referential Integrity option.

### Important

### **Consequences of Enforcing Referential Integrity**

- You cannot enter a value in the foreign key field of the related table if that value doesn't exist in the primary table. For example, you can't enter an order for a customer who doesn't exist. Create the customer first.
- You cannot delete a record from the primary table if matching records exist in a related table. For example, you can't delete a Customers record if that customer has orders in the Orders table.
- You cannot change the primary key value in the primary table if related records exist in another table. For example, you can't change a customer's code if their orders still reference it.

If you attempt an action that violates these rules, Access displays an error message, like the one shown in Figure 4.13.

To enable referential integrity, these conditions must be met:

Microsoft	ft Access	×
	The Microsoft Access database engine cannot find a record in the table 'Customers' with key matching field(s) 'CustomerCo	de'.
	OK Help	

Figure 4.13.: Error message when entering an order for a non-existent customer.

- Both tables are in the same Access database.
- The linked field in the primary table must be a primary key or have a unique index.
- Linked fields must have the same data type, and numeric fields must match in size.
- Existing data must not violate referential integrity.

### Important

AutoNumber fields can be linked to Number fields as long as their Field Size property is set to Long Integer.

If existing data violates these rules, you must resolve the issues first. Common causes include:

- 1. The linked fields differ in data type or size. This can be fixed by modifying the table designs.
- 2. The related table contains values that don't exist in the primary table. This happens when records are deleted from the primary table but remain in the related one, creating "orphan" records. Use the Create tab > Query Wizard (Queries group) > Find Unmatched Query Wizard to identify and remove them.

### 4.7.6. Creating the Boxes-BoxDetails Relationship

To define a relationship, determine the primary and related tables, and the linking fields:

- Primary table: Boxes, field: BoxCode
- Related table: BoxDetails, field: BoxCode

### Task 4.4. FILE: candy365.accdb

- 1. Open the database.
- 2. Go to Database Tools tab > Relationships (Relationships group). The Relationships window will appear, showing existing relationships.

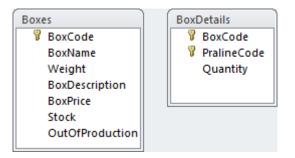


Figure 4.14.: Relationships window showing Boxes and BoxDetails.

3. Drag the BoxCode field from Boxes to the BoxCode field in BoxDetails.

Edit Relationships			? ×
Table/Query: Boxes	Related Table/Query:	$\sim$	Create
BoxCode	V BoxCode	<b>^</b>	Cancel Join Type
Enforce Referential Integrity Cascade Update Related Fields Cascade Delete Related Records		Create New	
Relationship Type:	One-To-Many		

Figure 4.15.: Edit Relationships dialog box.

- 4. Check Enforce Referential Integrity and click Create. The relationship will now be visible in the window.
- 5. Close the Relationships window and click Yes to save changes.

# 4.8. Exercises

### Exercise 4.1. Prevent Duplicate Names (tabl001)

The Suppliers table has a Supplier field. Currently, duplicate supplier names are allowed. Prevent this by setting the Indexed property of the field to Yes (No Duplicates).

### Exercise 4.2. Name Order Formatting (tabl002)

The customer with CustomerCode 15 has the last name Molen, van de and first name Robert. When printing labels, this results in Robert Molen, van de, which is incorrect. You want it to read Robert van de Molen. Simply changing the LastName to van de Molen will affect sorting. Think of a better solution. You may propose changes to the table design but do not apply them in the database.

### Exercise 4.3. Title Field with List Box (tabl003)

Some customers prefer to have their title included in mailings. Add a Title field to the Customers table. Use the Lookup Wizard to create a list with these values: ing., ir., drs., mr., dr.

### Exercise 4.4. Currency Format (tabl004)

Use the Transport table created earlier. If it's not in your database, either:

- Recreate the table manually, or
- Use the transport.accdb database, where it already exists.

Open the Transport table. Notice the transport costs are not formatted as currency. Change the Format property of the TransportationCosts field to display currency using your local symbol. Switch to Datasheet View to observe the result.

### i Note

The displayed currency symbol depends on your Windows regional settings.

### Exercise 4.5. Composite Key (tabl005)

In the Transport table, ID TransportCompany has repeated values, so it cannot serve as a primary key. Similarly, DeliveryProvince is also repeated. However, the combination of both fields is unique and can be used as a composite primary key. Set this composite key in the Transport table.

### i Note

In this exercise, you need the table Transport that you should have made earlier (see Section 4.4). When this table is not present in your database, you have two possibilities:

- You still perform the task by creating this table.
- You use database transport.accdb where the table Transport already was made for you.

### Exercise 4.6. Target Group Field (tabl006)

Customers should be classified into target groups: top customer, good customer, or small customer. Add a TargetGroup field to the Customers table using a list box with these options.

### Exercise 4.7. Create Relationship: BoxDetails-Pralines (tabl007)

Create a relationship between BoxDetails and Pralines, and enforce referential integrity.

### Exercise 4.8. Employees Table (tabl008)

In late 2010, the company decided to track which employee processed each order. An Excel file already contains employee data. Do the following:

- Import the Excel file employees.xlsx and name the table Employees. Assign a suitable primary key.
- dd an EmployeeCode field.
- Create a relationship between Orders and Employees, and enforce referential integrity.

### Exercise 4.9. Contact Registration (tabl009)

The marketing department wants to log all customer contacts (phone, mail, email, visits) to track time and costs. Each entry should include the contact type, date, start and end time, subject, and any costs.

Create a new table Contacts and link it to Customers, enforcing referential integrity.

If the Employees table exists (see Exercise 4.8), include which employee logged the contact by linking Contacts to Employees.

### Exercise 4.10. Complaints Registration (tabl010)

Customers sometimes submit complaints. These can be categorized as:

- Damaged packaging
- Box not delivered on time
- Expired best-before date
- Poor taste of pralines

Each complaint refers to an order. Create a Complaints table and relate it to Orders, enforcing referential integrity.

### Exercise 4.11. Check Referential Integrity (tabl011)

Verify whether referential integrity is enforced in relationships involving Customers, Orders, OrderDetails, Boxes, BoxDetails, and Pralines. If not, enable it.

### **OBJECTIVES**

- Understand the types of queries and their uses.
- Create a simple select query using the wizard.
- Apply conditions in a query.
- Use grouping and perform calculations.
- Work with parameter queries, update queries, make-table queries, and crosstab queries.

Retrieving information from a database is one of the most common tasks for end users. Queries are essential for this. While end users can create simple queries, complex ones may require assistance from professionals.

### 5.1. About Creating Queries

The most common query type is the **select query**, which retrieves specific data from the database. Queries can also perform calculations (such as sums or averages) and come in varioustypes:

- **Select query** Retrieves data from one or more tables and displays the results. It can also perform calculations (e.g., sum, average, count). This is the most widely used query type.
- **Parameter query** Prompts the user to enter one or more values before running. For example, it can show all orders placed on a specific date.
- **Update query** Modifies data in one or more fields for records that meet certain conditions, e.g., applying a 10% price increase to specific products.

Before creating a query, clearly define your information requirements.

### 5.2. Customers and Orders

**GOAL**: Create a sorted list of all customers showing their last name, first name, and order codes.

**ANALYSIS**: Customer names are in the Customers table. Order codes are in the Orders table. The Query Wizard is suitable for this straightforward task.

Task 5.1. FILE: candy365.accdb

- 1. Open the database.
- 2. Go to tab Create > Query Wizard (group Queries). The New Query dialog box opens.

- 3. Select Simple Query Wizard, then click OK.
- 4. Under Tables/Queries, select Table: Customers.

Simple Query Wizard	
	Which fields do you want in your query? You can choose from more than one table or query.
<u>T</u> ables/Queries	
Table: Customers	<u>•</u>
<u>A</u> vailable Fields:	Selected Fields:
CustomerCode	>       <
Ca	ncel < Back Next > Finish

Figure 5.1.: Simple Query Wizard: Selecting the Customers table.

- 5. Add fields LastName and FirstName.
- 6. Switch to Table: Orders and add OrderCode.
- 7. Click Next, choose Detail (shows every field of every record), then click Next.
- 8. Name the query Customername+Ordercodes, select Modify the query design, and click Finish.
- 9. Sort the LastName field in Ascending order.
- 10. Switch to Datasheet View to review the result.
- 12. Close the query and save the changes.

### 5.3. Criteria in Queries

This section explains conditions in a query, the Like operator, and wildcards.

In a query, you can define conditions so that only records meeting these conditions are displayed. The Criteria row in the design grid is used for this purpose.

A criterion is similar to a formula. It is a string that can consist of field references, operators, and constants (values that remain the same). Query criteria are also called expressions. The formulation of a criterion depends on the data type of the field (text, numeric, date/time, yes/no).

Simple Query Wizard	
	Which fields do you want in your query? You can choose from more than one table or query.
Tables/Queries	
Table: Orders	<u>·</u>
<u>A</u> vailable Fields:	Selected Fields:
CustomerCode OrderDate	LastName       FirstName       OrderCode       <
Car	ncel < Back Next > Einish

Figure 5.2.: Selected fields from Customers and Orders tables.

	Customers * CustomerCode LastName FirstName Address City Province		Orders * OrderCode CustomerCode OrderDate	
4				
	-			
Field:	LastName	FirstName	OrderCode	
Table:	Customers	Customers	Orders	
Sort:	Ascending			
Show:		$\checkmark$		
Criteria:				
or:				

Figure 5.3.: Query Design View for CustomerName+OrderCodes query.

LastName 👻	FirstName 👻	OrderCode 👻
Anders	Alice	484
Anders	Alice	208
Anders	Rolf	482
Anders	Rolf	390
Andriessen	Melissa	715
Andriessen	Melissa	418
Andriessen	Melissa	314
Andriessen	Melissa	164
Andriessen	Carolien	365
Appelo	Francien	515
Appelo	Francien	688

Figure 5.4.: Datasheet View of the CustomerName+OrderCodes query results.

Category	Operators
Arithmetic	+, -, *, /,  ^, Mod
Comparison	=, >, >=, <, <=, <>
Logical	And, Or, Not, Xor, Eqv
Concatenation	&, +
Special	Is Null, Is Not Null, Like, Between, In

You can create simple or very complex criteria using these operators.

The Like operator plays a special role. It compares a value with a specific pattern. This pattern can be a literal string, such as Like "North", or it can contain wildcards, such as Like "He\*". This makes the Like operator very powerful.

Date values must be enclosed in pound signs #. Some examples of date criteria: #12/5/2010#, >#9/1/2010#, >#9/1/2010# and <#9/15/2010#

Wildcards are placeholders for other characters, used when you don't know the entire search pattern but only a part of it. The three most common wildcards are:

- \*: For any number of characters. Examples: "A\*", "\*dam"
- ?: For any single character. Example: "b?k"
- #: For any single digit. Example "1#5"

External article: Examples of query criteria<sup>1</sup>

# 5.4. Orders from Utrecht Customers

This section covers manually adding fields and criteria to an existing query and saving the query under a new name.

To perform this task, you must have completed the steps in Section 5.2.

 $<sup>^{1}</sup> https://support.office.com/en-us/article/Examples-of-query-criteria-3197228c-8684-4552-ac03-aba746fb29d8$ 

**GOAL**: Create a list sorted by last name, first name, and order codes for all customers who live in the province of Utrecht and placed an order in December 2009.

**ANALYSIS**: All required information is available in the query created in Section 5.2. However, two additional conditions (criteria) are needed:

- The Province field (in the Customers table) should have the value "UT".
- The OrderDate field (in the Orders table) should have a value between December 1, 2009, and December 31, 2009.

### Task 5.2. FILE: candy365.accdb

- 1. Open the database.
- 2. Open the CustomerName+OrderCodes query in Design View.

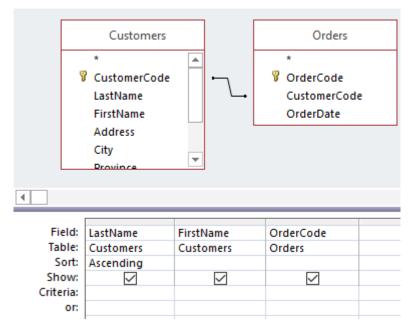


Figure 5.5.: Query Design View of the CustomerName+OrderCodes query.

3. Drag the Province field from the Customers table to the column next to OrderCode. Drag the OrderDate field from the Orders table to the next column.

	LastName Customers	FirstName Customers	OrderCode Orders	Province Customers	OrderDate Orders
	Ascending				
Criteria: or:					

Figure 5.6.: Query Design Grid with Province and OrderDate fields added.

- 4. Enter "UT" in the Criteria box under the Province field.
- 5. Enter Like "12\*2009" in the Criteria box under the OrderDate field.

Field:	LastName	FirstName	OrderCode	Province	OrderDate
Table:	Customers	Customers	Orders	Customers	Orders
Sort:	Ascending				
Show:	V	✓	✓	✓	
Criteria:				"UT"	Like "12*2009"

Figure 5.7.: Query Design Grid with criteria for Province and OrderDate.

### **i** Note

- For the date criteria you can use Like "12\*2009" if you only care about the month and the year. For more precise date ranges you should need Between #12/1/2009# And #12/31/2009#.
- The asterisk \* is a wildcard meaning any characters can be in its place. So, the string "12/\*/2009" would match any day in December 2009.
- Ensure your computer's date format settings (e.g., American: month-day-year) are consistent with how you enter dates in Access criteria. Access typically interprets dates based on regional settings but using the unambiguous #mm/dd/yyyy# format is best practice.
- 6. Switch to Datasheet View and verify that all customers are from the province of Utrecht and that all order dates are in December 2009.
- 7. Switch to Design View and uncheck the Show box for the Province and OrderDate columns.
- 8. Switch to Datasheet View.
- 9. Save the query under a different name using File > Save Object As and specify the name UtrechtOrdersDec2009.
- 10. click OK and close the query.

# 5.5. Customers with Box CHER

This section describes creating a select query involving three tables.

**GOAL**: Towards the end of the calendar year, it's noted that the sell-by date for CHER boxes is approaching. The sales department wants to organize a direct mail campaign for customers who have previously purchased CHER boxes. Create a list of all customers (last name, first name, and full address) who have bought at least one CHER box.

**ANALYSIS**: All customer information (first name, last name, address, zip code, city) is in the Customers table. The ordered boxes can be found in the BoxCode field in the OrderDetails table. To link an order to a customer, you also need the Orders table, which serves as the link between the Customers and OrderDetails tables. You'll need to specify the value CHER for the BoxCode field as a condition.

When using the Wizard to create the query, it is sufficient to add only the necessary fields from the Customers and OrderDetails tables. The Wizard will then automatically add the Orders table because it links the Customers and OrderDetails tables. If you create the query manually from scratch, you must remember to add the Orders table yourself. Therefore, using the Wizard is preferred for this task. i Note

It's not necessary to include a criterion stating that at least one box was ordered, because this condition is automatically enforced by the joins between linked tables when retrieving related data.

#### Task 5.3. FILE: candy365.accdb

- 1. Open the database.
- 2. Choose Create tab > Query Wizard (Queries group). The New Query dialog box will be displayed, where you can select the query type.
- 3. Select Simple Query Wizard and click OK.
- 4. In the Tables/Queries list box, select Table: Customers. The fields of the Customers table will be displayed in the Available Fields box (see Figure 5.1).
- 5. Add the following fields: FirstName, LastName, Address, ZipCode, City. Select the field, and click

#### 💡 Tip

You can also double click on a field to add or remove it.

- 6. In the Tables/Queries box, select Table: OrderDetails. The fields of the OrderDetails table will be displayed in the Available Fields box.
- 7. Add field BoxCode.

Simple Query Wizard	
	Which fields do you want in your query? You can choose from more than one table or query.
Tables/Queries	
Table: OrderDetails	<u>•</u>
<u>A</u> vailable Fields:	Selected Fields:
OrderCode Ouentity	<ul> <li>FirstName</li> <li>LastName</li> <li>Address</li> <li>ZipCode</li> <li>City</li> <li>BoxCode</li> </ul>
Car	ncel < Back Next > Finish

Figure 5.8.: Simple Query Wizard: Selected fields for Customers and CHER box query.

8. Click Next. Now you can specify if you want a detail or summary query.

- 9. Select Detail (shows every field of every record) and click Next.
- 10. Name the query CustomersAndCHER, select Modify the query design, and click Finish. The query will be saved and then appear in Design View.

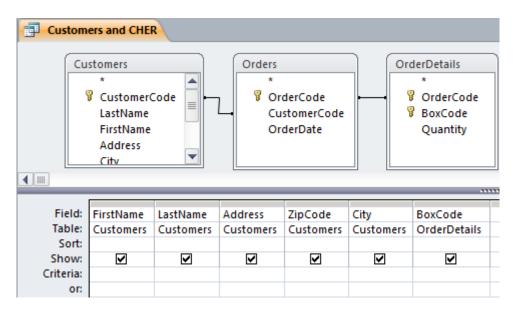


Figure 5.9.: Query Design View for Customers and CHER box query.

11. Enter "CHER" in the Criteria row under the BoxCode column and uncheck the Show box for this field.

Field:	FirstName	LastName	Address	ZipCode	City	BoxCode
Table:	Customers	Customers	Customers	Customers	Customers	OrderDetails
Sort:						
Show:				✓	<ul><li>✓</li></ul>	
Criteria:						"CHER"
or:						

Figure 5.10.: Query Design Grid with CHER criterion for BoxCode.

12. Switch to Datasheet View.

13. Close the query and save the changes.

## 5.6. Summarizing and Calculations

Simple select queries work with individual records. When you select certain customers from the Customers table with a query, you will see a record for each customer that meets the requirements. It is also possible to group your records and then perform calculations on these subgroups, similar to calculating totals and subtotals.



When a query is displayed in Design View, you will see the **Totals** on the ribbon under Query Design tab > Totals (Show/Hide group).

With this button, you can show or hide a Total row in the design grid. Access adds a Total box for each field, just below the table box.

	Field:	Box: BoxName	Praline: PralineName	PralineCosts: Sum([Pralines].[PralineCosts]*[BoxD
1	Table:	Boxes	Pralines	
1	Total:	Group By	Group By	Expression
	Sort:			
S	show:	✓	✓	
Cri	iteria:			
	or:			

Figure 5.11.: Query Design View with the Total row visible.

For each added field, you can choose an option from the drop-down list in the Total row. This option determines whether the field is used in a calculation, for grouping, or for filtering. The available options can be divided into three categories:

- **Grouping** The choice is Group By. The field is used to create smaller groups on which you can perform calculations. This is also the default option.
- **Filtering** The choice is Where. The checkmark in the Show box is also cleared and should remain so. In the Criteria box, you can specify the value or expression for filtering.
- **Calculations** The options are: Sum, Avg, Min, Max, Count, StDev, Var, First, Last, Expression. The selected calculation is then performed for the field.

Choice in Total box Description		
Group By	Groups records based on the values in this field.	
Sum	Adds together the values in this field.	
Avg	Averages the values in this field.	
Min	Retains the smallest value in this field.	
Max	Retains the largest value in this field.	
Count	Counts the number of records (regardless of the field).	
StDev Calculates the standard deviation of values in this field		
Var Calculates the variance of values in this field.		
First Retains the first value in this field for each group.		
Last	Retains the last value in this field for each group.	
Expression	Calculates a user-defined expression for this field.	
Where	Filters records based on values in this field.	

### Table 5.2.: Options for Summarizing.

### **Calculated Field**

A calculated field takes data from one or more fields and performs arithmetic operations to produce new information. You can perform simple arithmetic, like addition and multiplication, or use Access's built-in functions, such as Sum and Avg. You can only use fields that have been added to the query. To create a calculated field:

- 1. Click in an empty column in the Field row of the design grid.
- 2. Enter a name for the calculation (the result), followed by a colon (:).
- 3. Enter the expression for the calculation.

### **i** Note

- You can use field names in the expression. Field names must be enclosed in square brackets ([]). If a field name contains no spaces, Access will add the square brackets for you after you enter the name. If a field name contains spaces, you must type the square brackets yourself.
- When using one of the calculation options for summarizing, it is recommended to also provide a new name in front of the field name (e.g., AvgCost: PralineCosts); otherwise, Access generates a default name for the result in Datasheet View (e.g., AvgOfPralineCosts). This new name must also be followed by a colon.

Here are some examples. Study them carefully. Create and experiment with them.

### Example 5.1. Average Praline Costs per Chocolate Type

In this example, the ChocolateType field is used for grouping, creating a group for each chocolate type. The PralineCosts field is used to calculate the average praline costs for each group. The result is one record for each chocolate type, showing the average price.

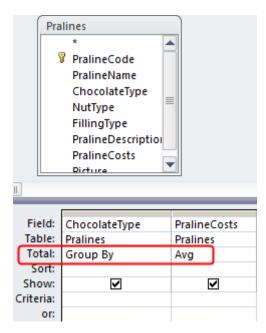


Figure 5.12.: Query Design: Average praline costs per chocolate type.

2	ChocolateType 👻	AvgOfPralineCosts -
	Butterscotch	\$0.26
	Dark	\$0.31
	Milk	\$0.28
	Toffee	\$0.20
	White	\$0.30

Figure 5.13.: Query Result: Average praline costs per chocolate type.

Because the column for the averages was not given a new custom name, Access generated the name AvgOfPralineCosts.

### Example 5.2. Box Price Statistics

In this example, the BoxPrice field is used four times with different calculations. Each column was given a new name. The result of the query is a single record containing the four calculation results.

	Boxes			
	*			
1	BoxCode			
	BoxName			
	Weight			
	BoxDescription			
	BoxPrice			
	Stock			
	OutOfProduction			
			1	
Field:	Min Price: BoxPrice	Max Price: BoxPrice	Avg Price: BoxPrice	Stdev Price: BoxPrice
Table:	Min Price: BoxPrice Boxes	Max Price: BoxPrice Boxes	Avg Price: BoxPrice Boxes	Stdev Price: BoxPrice Boxes
Table: Total:			-	
Table: Total: Sort:	Boxes	Boxes	Boxes	Boxes
Table: Total:	Boxes	Boxes	Boxes	Boxes
Table: Total: Sort:	Boxes Min	Boxes Max	Boxes Avg	Boxes StDev

Figure 5.14.: Query Design: Box price statistics.

Min Price 📼	Max Price 🕞	Avg Price 🕞	Stdev Price 👻
\$14.00	\$43.00	\$25.04	8.52

Figure 5.15.: Query Result: Box price statistics.

### Example 5.3. Average Box Price for Boxes Over 200 Grams

Calculate the average price of boxes weighing more than 200 grams. The Weight field is used for filtering to include only boxes over 200 grams.

# 5.7. Customers per Province

Purpose: Using a select query with grouping and a calculation with the COUNT function.

**GOAL**: Create a list showing the number of customers per province.

**ANALYSIS**: All necessary data is in the Customers table. You will need the Province field. A customer is uniquely identified by their CustomerCode, so you need to count the number of CustomerCodes in each province. This requires grouping by Province.

### Task 5.4. FILE: candy365.accdb

1. Open the database.

		Boxes		
	💡 BoxCode			
		BoxName		
		Weight		
		BoxDescript		
		BoxPrice		
		Stock		
		OutOfProduction		
	-			_
Fi	eld:	BoxPrice	Weight	
Tal	ble:	Boxes	Boxes	
Total: Av		Avg	Where	
_	ort:			
	ow:	$\checkmark$		
Crite			>200	
	or:			

Figure 5.16.: Query Design: Average box price for boxes over 200g.

AvgOfBoxPrice 👻
\$29.00

Figure 5.17.: Query Result: Average box price for boxes over 200g.

Add Tables				$\times$
Tables	Links	Queries	All	
Search				
BoxDetail Boxes Custome OrderDet Orders Pralines Suppliers	rs ails			
Add Selected Tables				

Figure 5.18.: Add Tables pane in Query Design View.

- 2. Go to Create tab > Query Design (Queries group). Access creates a new blank query window and displays the Add Tables pane on the right side.
- 3. Select the Customers table, then click Add Selected Tables. The Customers table has now been added to the query window.
- 4. Successively add the Province and CustomerCode fields to the design grid by double-clicking each field.

	* CustomerCode LastName FirstName Address City Province ZinCode	
Field Table Sort Show Criteria or	Province Customers	CustomerCode Customers 🗹

Figure 5.19.: Query Design Grid with Province and CustomerCode fields.

5. Click Query Design tab > Totals (Show/Hide group).

Field:	Province	CustomerCode
Table:	Customers	Customers
Total:	Group By	Group By
Sort:		
Show:	$\checkmark$	
Criteria:		
or:		

Figure 5.20.: Query Design Grid with Total row added.

6. In the Total row under the CustomerCode column, click and select Count from the dropdown list.

	Province	CustomerCode
	Customers	Customers
	Group By	Count
Sort:		
Show:	V	
Criteria:		
or:		

Figure 5.21.: Query Design Grid: Count of CustomerCode grouped by Province.

7. Switch to Datasheet View.

2	Province 👻	CountOfCustomerCode	-
	DR		21
	FL		4
	FR		17
	GL		40
	GR		24
	LI		13
	NB		40
	NH		53
	OV		28
	UT		39
	ZH		42
	ZL		5

Figure 5.22.: Query Result: Number of customers per province.

- 8. Close the query and answer Yes when prompted to save changes. The Save As dialog box will be displayed.
- 9. Name the query Number of customers per province and click OK.

# 5.8. Column Heading Modification

This section explains how to change column headings in a query.

To perform this task, you must have completed Section 5.7.

By default, Access uses field names as column headings in Datasheet View. For summarized data, Access generates a name (e.g., CountOfCustomerCode). It is recommended to use clearer, more descriptive names.

### Task 5.5. FILE: candy365.accdb

- 1. Open the database.
- 2. Open the Number of customers per province query in Design View.
- 3. Place the cursor in the Field row for the CustomerCode field, at the beginning of the field name, and type"Total customers: ". (Ensure there's a space after the colon if desired in the heading).

Field:	Province	Total customers: CustomerCode
	Customers	Customers
Total:	Group By	Count
Sort:		
Show:	V	

Figure 5.23.: Query Design Grid: Modifying the column heading for counted CustomerCode.

- 4. Switch to Datasheet View.
- 5. Close the query and save the changes.

Province 🔻	Total customers 🕞	
DR	21	L
FL	4	ł
FR	17	1
GL	40	)
GR	24	ł
LI	13	3
NB	40	)
NH	53	3
OV	28	3
UT	39	,
ZH	42	2
ZL	5	5

Figure 5.24.: Query Result: Number of customers per province with modified column heading.

# 5.9. Calculate Order Amounts

**GOAL**: Create a sorted list of order line items, showing for each order: order code, box code, number of boxes, box price, and the amount for each line item (i.e., for each type of box).

**ANALYSIS**: For each order, you can find the OrderCode, BoxCode, and the number of boxes (Quantity) in the OrderDetails table. The BoxPrice is in the Boxes table. The line item amount is not present in any table because this amount can be calculated from other data: Amount = Quantity \* BoxPrice.

Task 5.6. FILE: candy365.accdb

- 1. Open the database.
- 2. Go to Create tab > Query Design (Queries group). Access creates a new blank query window and displays the Add Tables pane (see Figure 5.18 if necessary).
- 3. Add the OrderDetails and Boxes tables to the query window.
- 4. Add the fields OrderCode, BoxCode, Quantity (from the OrderDetails table), and BoxPrice (from the Boxes table) to the grid by double-clicking each field.
- 5. Set the sort order for the OrderCode and BoxCode fields to Ascending. Click in the Field row of the first empty column and enter "Amount: Quantity\*BoxPrice". Access will automatically surround the field names with square brackets if they don't contain spaces; otherwise, you must type them.
- 6. Switch to Datasheet View.
- 7. Switch to Design View.

### i Note

All objects in Access have properties. These properties determine, among other things, the appearance of the object. You can set properties in the Property Sheet. You can toggle the visibility of the Property Sheet with Query Design tab > Property Sheet (Show/Hide group). A faster way is to use the keyboard shortcut F4.

To format the amounts as currency, you need to change the Format property of the calculated Amount field.

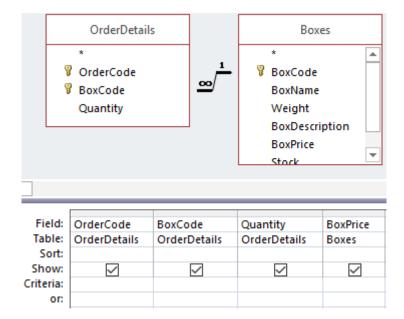


Figure 5.25.: Query Design: Tables and fields for calculating order row amounts.

Field:	OrderCode	BoxCode	Quantity	BoxPrice	Amount: [Quantity]*[BoxPrice]
Table:	OrderDetails	OrderDetails	OrderDetails	Boxes	
Sort:	Ascending	Ascending			
Show:	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
Criteria:					
or:					

Figure 5.26.: Query Design: Calculated field for Amount (Quantity\*BoxPrice).

OrderCode 👻	BoxCode 👻	Quantity 👻	BoxPrice 💌	Amount 👻
1	PEAN	2	\$ 19.00	38
3	CHER	2	\$ 16.25	32.5
3	NOWI	1	\$ 33.25	33.25
6	BUTT	1	\$ 27.75	27.75
6	FANT	1	\$ 18.00	18
6	HEAV	2	\$ 15.75	31.5
6	NOW/	2	\$ 22.25	66.5

Figure 5.27.: Query Result: Order row amounts, unformatted.

8. Ensure the Property Sheet is visible. Click anywhere in the Amount field in the design grid. In the Property Sheet, click in the Format box and choose Currency from the list box.

Property Sheet Selection type: Field Properties					
General Lookup					
Description					
Format Currency					
Decimal Places					
Input Mask					
Caption					

Figure 5.28.: Field Properties: Setting Format to Currency for the Amount field.

9. Switch to Datasheet View.

OrderCode 👻	BoxCode 👻	Quantity 👻	BoxPrice 👻	Amount -
1	PEAN	2	\$ 19.00	\$38.00
3	CHER	2	\$ 16.25	\$32.50
3	NOWI	1	\$ 33.25	\$33.25
6	BUTT	1	\$ 27.75	\$27.75
6	FANT	1	\$ 18.00	\$18.00
6	HEAV	2	\$ 15.75	\$31.50
6	NOWI	2	\$ 22.25	\$66.50

Figure 5.29.: Query Result: Order row amounts, formatted as currency.

- 10. Close the query and save the changes.
- 11. Name the query OrderRowAmount and click OK.

### 5.10. First Order per Customer

**GOAL**: Create a list of the first order for each customer. Show the customer code, customer name (last and first), and the date of their first order for every customer who has placed one or more orders.

**ANALYSIS**: The required data is in the Customers and Orders tables. You need to create a query showing customer data and order data. Finding the first order can be achieved by by using Min in the Total row for the OrderDate field, grouped by customer.

Task 5.7. FILE: candy365.accdb

- 1. Open the database.
- 2. Go to Create tab > Query Design (Queries group).
- 3. Add the Customers and Orders tables to the query window.
- 4. Add the fields CustomerCode, LastName, and FirstName (from Customers), and OrderDate (from Orders) to the grid by double-clicking each field.
- 5. Click Query Design tab > Totals (Show/Hide group).

6. Change the column title for OrderDate by typing "First order date:"in front of the field name in the Field row.

	Customers		Or	ders	
	* CustomerCode LastName FirstName Address Citv		* OrderCo Custom OrderDa	erCode	
Field:	CustomerCode	LastName	FirstName	First order da	ate: OrderDate
Table:	Customers	Customers	Customers	Orders	
Total:	Group By	Group By	Group By	Group By	
Sort:					
Show:	$\checkmark$	$\checkmark$	$\checkmark$		$\checkmark$
Criteria:					
or:					

Figure 5.30.: Query Design: Tables and fields for finding the first order date.

- 7. In the Total row under the OrderDate (or FirstOrderDate: OrderDate) column, click and select Min from the list. Ensure the Total row for CustomerCode, LastName, and FirstName is set to Group By.
- 8. Set the sort order for LastName and FirstName to Ascending.

Field:	CustomerCode	LastName	FirstName	First order date: OrderDate
Table:	Customers	Customers	Customers	Orders
Total:	Group By	Group By	Group By	Min
Sort:		Ascending	Ascending	
Show:	$\checkmark$	$\sim$	$\checkmark$	$\checkmark$
Criteria:				
or:				

Figure 5.31.: Query Design: Finding the minimum (first) order date per customer.

- 9. Switch to Datasheet View.
- 10. Close the query and save the changes.
- 11. Name the query First order dates and click OK.

### 5.11. Parameter Query

A parameter query displays a dialog box that prompts the user for additional information when it runs, such as criteria for retrieving records or a value to insert into a field. You can design the query to request multiple pieces of data, for example, a start and end date, to retrieve all records with dates between them.

Parameter queries are also useful as a basis for forms and reports. For example, based on a parameter query, you can create a monthly revenue report. When printing the report, a dialog

CustomerCode 👻	LastName 👻	FirstName 👻	First order date 📼
205	Anders	Alice	12/1/2009
122	Anders	Rolf	12/27/2009
48	Andriessen	Carolien	12/23/2009
232	Andriessen	Melissa	11/25/2009
70	Appelo	Francien	11/26/2009
134	Bakker	B. L.	12/20/2009
295	Bakker	Paula	12/3/2009
42	Bakker	V. J.	12/14/2009
100	Berents	Wim	12/5/2009

Figure 5.32.: Query Result: First order date for each customer.

box will prompt you for the month you want to print; you enter the month, and the correct report will be printed.

**GOAL**: At Snoopy, customer service representatives regularly receive phone inquiries about specific orders. You want to be able to quickly retrieve and display the data for a particular order. This can be achieved with a parameter query that asks for the order code when the query is run.

**ANALYSIS**: The necessary information about a particular order is in the Orders and OrderDetails tables. Prompting for the order code can be controlled via a criterion.

Task 5.8. FILE: candy365.accdb

- 1. Open the database.
- 2. Go to Create tab > Query Design (Queries group).
- 3. Add the Orders and OrderDetails tables to the query window.
- 4. Add the fields OrderCode, CustomerCode, OrderDate (from Orders), and BoxCode and Quantity (from OrderDetails) to the grid by double-clicking each field.

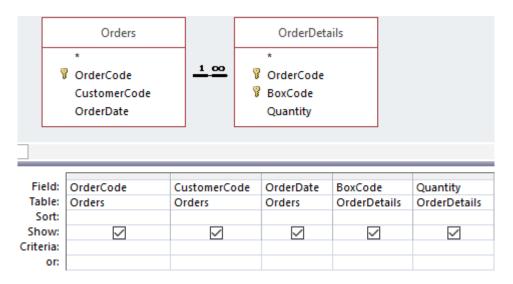


Figure 5.33.: Query Design: Tables and fields for the order information parameter query.

5. In the Criteria row under the OrderCode column, type [Enter order code]. The square brackets indicate to Access that this is a parameter.

Field:	OrderCode	CustomerCode	OrderDate	BoxCode	Quantity
Table:	Orders	Orders	Orders	OrderDetails	OrderDetails
Sort:					
Show:	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
Criteria:	[Enter order code]				
or:					

Figure 5.34.: Query Design: Adding a parameter criterion for OrderCode.

- 6. Switch to Datasheet View. The Enter Parameter Value dialog box will appear, prompting you to enter the order code.
- 7. Enter a value, e.g. 30, and click OK.

OrderCode 👻	CustomerCode 👻	OrderDate 👻	BoxCode 👻	Quantity	*
30	197	11/6/2009	BUTT		2
30	197	11/6/2009	PERF		1
30	197	11/6/2009	SWBI		2
30	197	11/6/2009	SWCR		1

Figure 5.35.: Query Result: Information for order with OrderCode 30 (parameter entry).

- 8. Close the query and save the changes.
- 9. Name the query Information specific order and click OK.

# 5.12. Action Queries

Most queries are select queries, used to search, collect, and display data, but not to change it. However, Access also has another category of queries, known as action queries, with which you can change data by deleting, updating, or adding records. The main advantage of an action query is that it can change a large number of records without requiring programming knowledge. The way you create and work with these queries is similar: first, create a select query, and then change the query type.

Access has four types of action queries:

- **Make Table** Selects one or more records and then creates a new table for them. This new table can be placed in the current database or created as a new table in a different database. You can use a Make Table query, for example, to copy outdated data to an archive database.
- **Append** Selects one or more records and then adds them to another existing table. For example, suppose you acquire new customers whose information has been stored in a separate table. With an append query, you can move these records to your main customers table.
- **Delete** Deletes one or more records. You specify a set of filter conditions, and then Access deletes the matching records. For example, you can remove products that are discontinued.
- **Update** Changes values in one or more records. Existing values in a field are replaced by new values, similar to a search and replace operation. You cannot undo the changes made by an update query, so it is advisable to first back up the database or the specific table before running the update query.

Important

Because these queries change data in the database, these queries could be a security risk. To provide protection, a number of checks are carried out in Access and the Trust Center. The trust center can disable the content. When opening such a database, Access displays a message with a security warning.

SECURITY WARNING Some active content has been disabled. Click for more details. Enable Content

Figure 5.36.: Access Message Bar showing a security warning.

If you want to enable the content, click Enable Content > Options and choose the desired option in the dialog box that appears. The database is opened again with full functionality. It is also advisable to make a backup of the tables that are changed. That 's easy to do with copy and paste.

### Example 5.4. Creating a Copy of a Table

- 1. In the Navigation Pane, right-click the name of the table and select Copy from the shortcut menu.
- 2. Right-click again in a blank area of the Navigation Pane and choose Paste. Give the new table a different name when prompted (e.g., TableName\_Backup).

To recover a table after an unintended change using a backup copy:

- 1. Delete or rename the changed table. For instance, in the Navigation Pane, right-click the name of the changed table and select Delete or Rename.
- 2. Right-click the backup copy of the table and choose Rename. Give the table its original name.

### 5.12.1. Update Query

This section provides an example of a simple update query that changes values in a field for all records satisfying a certain condition.

**GOAL**: The costs of all pralines with chocolate type "White" should be increased by 10%.

**ANALYSIS**: All necessary data is in the Pralines table. You need the ChocolateType and PralineCosts fields. Selecting the white chocolate types can be done by adding a criterion to the query. You can increase the PralineCosts by 10% by multiplying the current value by 1.1.

Important

Make a backup (copy) of the table, so you can restore the original situation. Another possibility is to backup (copy) the whole database.

#### Task 5.9. FILE: candy365.accdb

- 1. Open the database.
- 2. Go to Create tab > Query Design (Queries group).
- 3. Add the Praines table to the query window.

- 4. Add the ChocolateType and PralineCosts fields to the grid by double-clicking each field.
- 5. Change the query type by clicking Update in the Query Type group on the Query Design tab. The Sort and Show rows will disappear, and a new row, Update To, will appear.

		Pralines		
		*	*	
	8	PralineCode		
		PralineName		
		ChocolateType		
		NutType		
		FillingType		
		PralineDescription		
		PralineCosts		
L		Picture	*	
•				
Fie	ld:	ChocolateType	Pr	alineCosts
Tab	le:	Pralines	Pr	alines
Update 1	To:			
Criter	ria:			
	or:			

Figure 5.37.: Update Query Design: Initial fields before specifying update logic.

- 6. In the Criteria box under the ChocolateType column, enter "White".
- 7. In the Update To box under the PralineCosts column, type "[PralineCosts]\*1.1".

Field:	ChocolateType	PralineCosts
Table:	Pralines	Pralines
Update To:		[PralineCosts]*1.1
Criteria:	"White"	
or:		

Figure 5.38.: Update Query Design: Logic to increase PralineCosts by 10% for White chocolate.

### **i** Note

- Access will surround text criteria like "White" with double quotes if you don't type them.
- Field names in expressions must be enclosed in square brackets ([]).
- 8. Save the query by clicking the Save button () on the Quick Access Toolbar and name it "Increase costs white chocolates with 10%".
- 9. Click Run in the Results group on the Query Design tab. A dialog box will appear asking for confirmation and stating how many rows will be updated.
- 10. Click Yes to proceed with the update.
- 11. Close the query.

### 5.12.2. Make Table Query

Purpose: Creating a new table using a Make-Table query.

**GOAL**: The marketing department has a special offer for all customers in Friesland. For this, they need a table containing only the customer data for customers in the province of Friesland.

**ANALYSIS**: All necessary data is in the Customers table. Selecting customers from Friesland can be done with a criterion in the query. First, you need to create this select query and then change its type to Make Table.

Task 5.10. FILE: candy365.accdb

### Create select query

- 1. Open the database.
- 2. Go to Create tab > Query Design (Queries group).
- 3. Add the Customers table to the query window.
- 4. Add all fields by double-clicking the asterisk (\*) at the top of the Customers field list in the table object. Then, separately add the Province field again (this allows you to set criteria on it without affecting the display of all other fields from the asterisk).
- 5. Enter "FR" as the criterion for the separately added Province field and uncheck its Show box (because this field's data will already be shown via the Customers.\* selection).

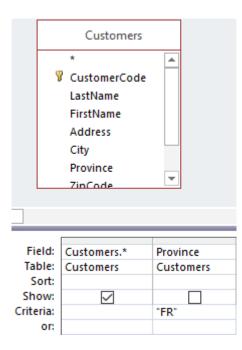


Figure 5.39.: Make-Table Query Design: Selecting Friesland customers.

- 6. Check the output of the query by switching to Datasheet View.
- 7. Save the query with the name Customers Friesland.

### Change the query type

8. Open the Customers Friesland query in Design View.

- 9. Click Make Table in the Query Type group on the Query Design tab. The Make Table dialog box will open.
- 10. Name the new table Friesland customers and ensure the option Current Database is selected for where the table should be created.

Make Table		?	×
Make New Tab		(	ОК
Table Name:	Friesland customers	Ca	ancel
Current Da		00	ancer
O Another Da	itabase:		
File Name:			
	Browse		

Figure 5.40.: Make Table dialog box: Specifying name and location for the new table.

- 11. click OK.
- 12. Click Run in the Results group on the Query Design tab. A dialog box will appear asking for confirmation and stating how many rows will be copied.
- 13. click Yes. The new table will be created.
- 14. Close the query. A dialog box will ask whether the changes to the query design (i.e., changing its type to a make-table query) should be saved.
- 15. click Yes.
- i Note

This prompt appears because the query type has been changed. This change is also reflected by the query's icon in the Navigation Pane under Queries. The query icon in front of the query name,  $\vec{s}$ , indicates it's an action query:

# 5.13. Crosstab Query

A crosstab query calculates a sum, average, or other aggregate function, and then groups the results by two sets of values: one down the left side of the datasheet (row headings) and another across the top (column headings). A crosstab query is similar in structure to an Excel PivotTable report and is often easier to read than a regular select query displaying the same data because the overview is more compact due to its horizontal and vertical alignment.

When you create a crosstab query, you specify which fields contain row headings, which field contains column headings, and which field contains values to summarize. You can use multiple fields for row headings (up to three), but only one field for column headings and one field for the values to summarize. It's also possible to use an expression for the row heading(s), column heading(s), and the values to summarize.

The easiest and fastest way to create a crosstab query is by using the Crosstab Query Wizard. For more complex queries, you can often start with this Wizard and then fine-tune the query in Design View.

**GOAL**: Calculate the number of customers by province and by region, and show the result in a crosstab table.

**ANALYSIS**: All necessary data can be found in the Customers table.

### Task 5.11. FILE: candy365.accdb

- 1. Open the database.
- 2. Choose Create tab > Query Wizard (Queries group). Select Crosstab Query Wizard and click OK. In the screen that appears, select the table or query that contains the fields for the crosstab.
- 3. Select Table: Customers and click Next. Now, select the field(s) for the row headings.
- 4. Select the Province field and move it to the Selected Fields list to be used as a row heading.

Crosstab Query Wizard				
Which fields' values do you want a row headings?	s Available		Select	ed Fields: nce
You can select up to three fields.	LastNan FirstNan Address City	ne	>>	
Select fields in the order you want information sorted. For example, y could sort and group values by Cou and then Region.	you ZipCode		<	
Sample:		$\wedge$		
	Province	Header1	Header2	Header3
	Province 1	TOTAL		
	Province2			
	Province3			
	Province4			
	Cancel	< <u>B</u> ack	<u>N</u> ext >	Einish

Figure 5.41.: Crosstab Query Wizard: Selecting Province for row headings.

- 5. click Next. Now, select the field for the column headings.
- 6. Select the Region field.
- 7. Click Next. Now, select the field whose values you want to aggregate and the aggregate function.
- 8. For the field to calculate, select CustomerCode. For the function, select Count. Deselect the option Yes, include row sums if you don't want totals for each province.
- 9. click Next.
- 10. Name the query Number of customers by province by region. Select View the query, and click Finish.

Crosstab Query Wizard				
Which field's values do you want as column headings?	Custome LastNan FirstNan Address	ne		
For example, you would select Employee Name to see each employee's name as a column headi	City ZipCode			
Sample:		$\sim$		
	Province	Region1	Region2	Region3
	Province 1	TOTAL		
	Province2			
	Province3			
	Province4			
	Cancel	< <u>B</u> ack	<u>N</u> ext >	Finish

Figure 5.42.: Crosstab Query Wizard: Selecting Region for column headings.

Crosstab Query Wizard				
		omerCode Iame	F	Functions: Avg Count
For example, you could calculate the sum of the field Order Amount for each employee (column) by country and region (row). Do you want to summarize each row?		lame ess ode hone		First Last Max Min StDev Sum Var
Sample:		$\wedge$	- 1/)	
	Province	Region1	Region2	Region3
	Province 1	Count(Customer	rCode)	
	Province2			
	Province3			
	Province4			
	Cancel	< <u>B</u> ack	<u>N</u> ext >	Finish

Figure 5.43.: Crosstab Query Wizard: Selecting CustomerCode and Count function for summarized values.

Province	•	North	-	South	-
DR			21		
FL			4		
FR			17		
GL					40
GR			24		
LI					13
NB					40
NH			53		
OV			28		
UT					39
ZH					42
ZL					5

Figure 5.44.: Crosstab Query Result: Number of customers by province and region.

# 5.14. Exercises

### Exercise 5.1. Customers from Friesland and Groningen (quer001)

Create a query to select customers in the provinces of Friesland (FR) and Groningen (GR). Show LastName, FirstName, and City. Name the query quer001.

### Exercise 5.2. Direct Mail Campaign for MARZ Box (quer002)

Suppose it is the end of December 2010. The expiration date of the MARZ box is approaching, and there is still plenty in stock. You want to start a direct mail campaign targeting customers who ordered at least one MARZ box between August and November 2010 (inclusive). Show the name and address of these customers. Name the query quer002.

### Exercise 5.3. Praline Boxes with Low Price (quer003)

Create a list of boxes whose price is \$17.50 or less. Show the fields BoxCode, BoxName, and BoxPrice. Name the query quer003.

### Exercise 5.4. Milk and Dark Pralines with Low Cost (quer004)

Create a list of pralines with chocolate type "Milk" or "Dark" (Pure) and whose costs are 30 cents or less. Show the fields PralineCode, PralineName, ChocolateType, and PralineCosts. Name the query quer004.

### Exercise 5.5. Customers from Enschede, Hengelo or Almelo (quer005)

Create a list of all customers (name and address) who live in Enschede, Hengelo, or Almelo. Name the query quer005.

### Exercise 5.6. Customers from Amsterdam with Zip Code 20\* (quer006)

Create a list of customers from Amsterdam (name and address) whose zip code begins with 20. Name the query quer006.

### Exercise 5.7. Pralines Without Filling (quer007)

Create a list of all pralines without filling. Name the query quer007.

### Exercise 5.8. Customers Outside Amsterdam (quer008)

Create a list of all customers who do not live in Amsterdam. Name the query quer008.

### Exercise 5.9. Number of Pralines per Box (quer009)

Create a list of all box names with their total number of individual pralines per box. The column showing the number of pralines should have an appropriate title. The list should be sorted in ascending order by box name. Name the query quer009. (Hint: This usually involves summing the Quantity field from BoxDetails for each box).

### Exercise 5.10. Number of praline types per box (quer010)

Create a list of box names and their number of praline types per box. The column showing the number of pralines should have an appropriate title. The list should be sorted in ascending order of box name. Name the query quer010.

### Exercise 5.11. Number of Orders per Region(quer011)

Create a list showing the number of orders per region. The column with the numbers should have an appropriate title. Name the query quer011.

### Exercise 5.12. Total Sales per Customer (quer012)

Create a list with the total sales amount per customer. Show customer code, customer name, and the total sales. The sales column should have an appropriate title and currency formatting. Sort by sales in descending order. Name the query quer012.

### Exercise 5.13. Customers with Orders (quer013)

Create a list of customers (code and name) who have placed at least one order. Name the query quer013.

# Exercise 5.14. Customers with Last Name Jansen or Janssen in North Region (quer014)

Create a list of customers with the last name Jansen or Janssen who are in the "North" region. Show relevant details. Name the query quer014.

### Exercise 5.15. Pralines with Filling and Without Nuts (quer015)

Create a list of pralines that have a filling but do not contain nuts. Show relevant praline details. Name the query quer015.

### Exercise 5.16. Boxes Heavier than 150 Grams with Max Price \$35 (quer016)

Create a list of boxes that are heavier than 150 grams and have a maximum price of \$35. Show only BoxCode, BoxName, and BoxPrice. Sort the boxes by price in ascending order. Name the query quer016.

### Exercise 5.17. Boxes Priced \$17-\$25 with Stock of at least 400 (quer017)

Create a list of boxes with a price between \$17 and \$25 (inclusive) and of which there are at least 400 in stock. Show the code, name, price, and stock of these boxes. Name the query quer017.

### Exercise 5.18. Total Sales per Province (quer018)

Create a list of total sales per province. The sales column should have an appropriate title and currency formatting. Sort by sales in descending order. Name the query quer018.

### Exercise 5.19. Customers with Orders the Week Before Christmas (quer019)

Create a list of customers who placed an order in the last week before Christmas 2010 (December 20, 2010, through December 24, 2010). Show the code, name, and address of these customers. Each customer should appear only once in the overview, even if they placed multiple orders during this period. Name the query quer019.

### Exercise 5.20. Boxes with Low Sales (quer020)

Create a list of boxes with low sales, defined as less than \$2000 in total sales. Show the box code and its total sales. Sort by sales in ascending order. The sales column should have an appropriate title and currency formatting. Name the query quer020.

# Exercise 5.21. Customers with Zip Code Starting 22 and Total Sales less than \$50 (quer021)

Create a list of customers whose zip code starts with 22 and who have total sales of \$50 or less. Show relevant customer and sales information. Name the query quer021.

### Exercise 5.22. Customers for "Sweet and Bitter" Box in 2009 (quer023)

Create an alphabetical list of customers who bought at least one "Sweet and Bitter" box in 2009. Ensure there are no duplicate customer records in the result. Name the query quer023.

### Exercise 5.23. Cities with at Least 2 Customers (quer024)

Create an alphabetical list of cities that have at least 2 customers. Name the query quer024.

### Exercise 5.24. Parameter Query for Praline Chocolate Type (quer025)

There are several chocolate types for pralines: Butterscotch, Milk, Dark, Toffee, and White. Create a parameter query that prompts the user to enter a chocolate type when the query runs, and then displays the pralines of that type. Show ChocolateType, PralineName, PralineCosts, and PralineDescription. Name the query quer025.

### Exercise 5.25. Update Query for Price Reduction (quer026)

Due to a decline in the world market price of chocolate, the prices of all boxes can be reduced by 15%.

- 1. Start by making a backup copy of the entire database or at least the Boxes table.
- 2. Design an update query to implement this reduction. Name the query quer026 and run it.
- 3. After verifying the result, you would typically keep this change. For exercise purposes, restore the original situation from your backup.

- Making a backup is crucial because you cannot easily undo the execution of an update query.
- Be aware that every time you run this query, the prices will be reduced by an additional 15% from their current values.

### Exercise 5.26. Crosstab Boxes per Weight (quer027)

Create a crosstab query with box names as row headings, the weight of the boxes as column headings, and the stock quantity as the summarized value in the table. Name the query quer027.

### Exercise 5.27. pdate Query for Price Increase (quer028)

Due to an increase in the chocolate price on the world market, the prices of all boxes must be increased by 10%.

- 1. Start by making a backup copy of the table or the entire database.
- 2. Design an update query to implement this price increase. Name the query quer028 and execute it.
- 3. For exercise purposes, restore the original situation from your backup.
- Making a backup is important because you cannot revert the execution of an update query.
- Be aware that each time you execute this query, the prices will be further increased by 10%.

### Exercise 5.28. Crosstab Sales per Customer per Month (quer029)

Create a crosstab query to provide an overview of total sales per customer per month in 2010. To do this, first create a select query named SalesPerCustomerPerMonth2010 that calculates these monthly sales. Then, use this select query as the source for the crosstab query. Display customers using their customer number and months using month numbers (e.g., 1 for January, 2 for February). Also, display the total sales for each customer in 2010 (row sums). Name the crosstab query quer029.

### **OBJECTIVES**

- Types of forms: main form, subform and linked form.
- Creating forms: automatic and manual.
- Working with controls.
- Performing calculations in a form.
- Forms with charts.

Forms provide organized and formatted views of data from tables and queries. They are especially useful for data entry.

# 6.1. Creating Forms

While it's possible to enter data directly into tables, this is generally only practical for very simple tables. Typically, forms are used for data entry. Basic forms often display data from a single table or query. More advanced forms, such as main-subform combinations or linked forms, draw data from related tables or queries..

Beyond simply giving your forms an attractive appearance, the key advantage is the ability to apply various validations during data entry. Forms also support a variety of controls such as list boxes, checkboxes, buttons, and more.

You can build a form from scratch, let Access generate one for you, or use the Form Wizard and then adjust the generated form manually.

### 6.1.1. Main and Subforms

A **subform** is a form embedded within another form, known as the **main form**. Subforms are typically used to display data from tables or queries with a one-to-many relationship. The "one" side appears in the main form, and the "many" side in the subform.

### Example 6.1. Orders by Customer

- Main form: Displays customer data from the Customer table.
- Subform: Displays the customer's orders from the Orders table.

The main and subform are linked so that the subform only shows records related to the current main form record.

# **Customers and Orders**

CustomerCode	1
FirstName	Rita
LastName	Hendriks
Address	2e Zeine 99 A
ZipCode	7233 AL
City	Zwolle

Orders:

$\angle$	OrderCode 👻	OrderDate 👻	Subtotal 👻	
	6	11/2/2009	\$218.25	
	301	12/14/2009	\$86.00	
	661	6/12/2010	\$23.00	
	823	9/29/2010	\$32.25	
Re	cord: I4 4 1 of 4		No Filter Se	arch

Figure 6.1.: ustomers main form with Orders subform.

### 6.1.2. Linked Forms

Linked forms are separate forms that interact with each other. One form—usually the main form—contains a command button that opens another form when clicked. The records in both forms are synchronized. The button includes VBA code to open the second form, which is automatically generated by the wizard. No programming knowledge is required.

Whether you use linked forms or subforms depends on user preferences. If users only need basic customer data, there's no need to display order information simultaneously. A button that opens the order details form is often more convenient.

Order data	)	<b></b> C	us	tomer (	Orders	
CustomerCode	1 Hendriks	•		derCode derDate	6	
FirstName	Rita			der data	11/2/2003	
ZipCode	7233 AL		2	BoxCode 👻	Quantity -	
City	Zwolle			FANT	1	
ord: I → 1 of 326 →	No Filter Sei			HEAV NOWI	2	
				PEAN SWBI	1	
			*	5000	2	
			Re	cord: I4 🖂 1 of	6 🕨 🕅 🌬	K No F

Figure 6.2.: Two linked forms with synchronized customer and order data.

# 6.2. Controls and Layouts

### 6.2.0.1. Controls

Controls are elements in a form or report that allow you to enter, display, or manipulate data. The **Text Box** is the most commonly used control for displaying data. Other frequently used controls include command buttons, checkboxes, and combo boxes.

Controls can be:

- Bound: Connected to a field in a table or query; used to display data from that source.
- Unbound: Not connected to any data source; used for static text, images, lines, etc.
- Calculated: Displays a value based on an expression rather than a data field.

### 6.2.0.2. Layouts

Layouts help align controls horizontally and vertically, giving your form a consistent look. Two layout types are commonly used:

• **Tabular Layout:** Controls are arranged in rows and columns like a spreadsheet, with labels across the top.

1	Hendriks	Rita	
CustomerCode	LastName	FirstName	

Figure 6.3.: Example of a tabular layout.

• Stacked Layout: Controls are arranged vertically with labels to the left.

CustomerCode	1
LastName	Hendriks
FirstName	Rita

Figure 6.4.: Example of a stacked layout.

### 6.2.0.3. Edit Controls

Since you often need to rearrange controls, it's important to understand how to move and modify them. Many operations can be performed on multiple controls at once. Make sure the Property Sheet is visible before editing.

- Select a control: Click on it.
- Select multiple controls: Hold down the Shift key and click each control, or drag a rectangle around them.
- **Deselect:** Click an empty area of the form.
- Move: Drag the control using the upper-left handle. You can also set an exact position in the Property Sheet.
- Resize: Drag the middle or corner handles, or set dimensions via the Property Sheet.
- Align: Use tab Arrange > Align (Sizing & Ordering group).
- **Delete:** Press the Delete key.

### 6.3. Automatically Generated Form

When new records are added to a table, typically all fields must be filled. A form containing all fields is the easiest way to do this. The fastest method is to select the table and use Form from the [Create] tab to auto-generate a form. You can then modify it as needed.

In this task, you'll create a form for a warehouse employee to update box stock and indicate if a box is out of production. Other box details should not be editable through this form. The desired result looks like this:

### Task 6.1. FILE: candy365.accdb

- 1. Open the database.
- 2. Select the Boxes table (no need to open it).
- 3. Click Create > Form (Forms group). A form is created and displayed in Layout View.

E Stock-	-in-trade
BoxCode	ALLS
BoxName	All Seasons
Weight	150
BoxDescription	Strawberries, blueberries and raspberries, to enjoy in all seasons, both bitter and milk.
BoxPrice	\$14.00
Stock	700
OutOfProduction	No

Figure 6.5.: Stock-in-trade form for warehouse data updates.

### i Note

If Access detects a one-to-many relationship between your selected table/query and another table, it adds a subdatasheet for that related table. If multiple such tables exist, no subdatasheet is added.

In this case, Boxes has a one-to-many relationship with OrderDetails, so a subdatasheet appears showing related OrderDetails records.

If you don't want this, you can delete the subdatasheet in Layout View by selecting it and pressing Delete.

- 4. Select the subdatasheet and press Delete.
- 5. Switch to Design View.

Form Header	
Boxes	
✓ Detail	
BoxCode	BoxCode
BoxName:	BoxName
Weight	Weight
BoxDescription	BoxDescription
BoxPrice	BoxPrice
Stock	Stock
OutOfProduction:	OutOfProduction
	3
Form Footer	

Figure 6.6.: Stock-in-trade form in Design View.

### i Note

Automatically created forms include two controls per field: a Text Box and a linked Label.

8oxCode:	BoxCode	_
Label	Text Box	

Figure 6.7.: Text box control with a linked label.

By default, all controls are arranged in a stacked layout, meaning you can't freely move them. To move a control freely, remove its layout using Arrange > Remove Layout (Table group).

6. In the Form Header, change the label from "Boxes" to "Stock-in-trade". Set Font Weight to Bold and Font Size to 20 pt.

### 💡 Tip

You can show or hide the Property Sheet using the [Property Sheet] button or the F4 hotkey. To find properties more quickly, sort them alphabetically using the  $2\downarrow$  button in the top right.

7. Select all text boxes and set the Width property to 3". Click elsewhere to deselect.

Prop	erty	Shee	et				×
Selection	n type: I	Multiple	selec	tion			₽↓
					$\sim$		
Format	Data	Event	Oth	er	All		
Format						$\sim$	^
Decimal	Places			Aut	0		
Visible				Yes			
Show D	ate Pick	er		For	dates		
Width				3"			
Height							

Figure 6.8.: Setting the Width property for text boxes.

### **i** Note

The Locked property makes a control read-only.

- 8. Select all text boxes except Stock and OutOfProduction. Set their Locked (tab Data) property to Yes.
- 9. While the fields are still selected, change their background color to light gray by either:
  - Setting Back Color directly.
  - Right-clicking and choosing Fill/Back Color from the shortcut menu.
- 10. Switch to Form View to check the result.

		6. I	Form	$\mathbf{s}$		
Prop	oerty	Shee	et			×
Selection	n type: I	Multiple	selec	tior	ı	₽↓
					$\sim$	
Format	Data	Event	Oth	er	All	
Control	Source					
Text For	mat			Pla	in Text	
Input M	lask					
Default	Value					
Validati	on Rule					
Validati	on Text					
Filter Lo	okup			Dat	tabase Def	ault
Enabled	ł			Yes		
Locked				Yes		

Figure 6.9.: Setting the Locked property for selected text boxes.

- 11. Close the form and save the changes.
- 12. Name the form Stock-in-trade and click OK.

### 6.4. Main and Subform

The easiest way to create a main form with a subform is by using the Form Wizard. This wizard generates both forms and handles the linking automatically.

The goal is to create a form that displays customer information (code, name, address) and, additionally, shows their corresponding orders (order code, date, and subtotal).

Customer data will appear in the main form and is sourced from the Customers table. The subform will show order data: OrderCode, OrderDate, and a calculated Subtotal. The subtotal is derived from the fields Quantity and BoxPrice in the Boxes table.

To calculate the subtotal, you'll use a query that contains all necessary fields and serves as the data source for the subform. A ready-made query named Orders with subtotals is already available. It includes the CustomerCode field to link the main and subform.

#### Task 6.2. FILE: candy365.accdb

- 1. Open the database.
- 2. Go to tab Create > Form Wizard (Forms group).
- 3. From Table: Customers, add the following fields: CustomerCode, FirstName, LastName, Address, ZipCode, City, Telephone.
- 4. Next, select Query: Orders with subtotals and add: OrderCode, OrderDate, Subtotal. Click Next.
- 5. Review the grouping structure and click Next. Select Datasheet as the subform layout.
- 6. Click Next. You'll be prompted to enter titles for the forms.
- 7. Set the form titles as follows:

Form Wizard	
	Which fields do you want on your form? You can choose from more than one table or query.
<u>T</u> ables/Queries	
Table: Customers	~
<u>A</u> vailable Fields:	Selected Fields:
CustomerCode  LastName FirstName Address City Province ZipCode Region  V	>       <
Car	ncel < Back Next > Finish

Figure 6.10.: Form Wizard: Selecting fields from the Customers table.

Form Wizard	
How do you want to view your data?	
by Customers by Orders with subtotals	CustomerCode, FirstName, LastName, Address, ZipCode, City, Telephone
	OrderCode, OrderDate, Subtotal
	●Form with subform(s) ○Linked forms
Cance	el < <u>B</u> ack <u>N</u> ext > <u>F</u> inish

Figure 6.11.: Form Wizard: Choosing how to display data for the subform.

- Form title: "Customers and Orders"
- Subform title: "Customers and Orders Subform"
- 8. Click Finish.

Cus	tomerCode	1				
Firs	tName	Rit	ta			
Las	tName	He	endr	iks		
Ado	dress	2e	Zei	ne 99 A		
ZipCode		72	7233 AL			
City	/	Zw	/olle	2		
Ord	lers:					
2	OrderCode 👻	OrderDate	Ŧ	Subtotal -		
	6	11/2/2	009	\$218.25		
	301	12/14/2	009	\$86.00		
	661	6/12/2	010	\$23.00		
	823	9/29/2	010	\$32.25		

Figure 6.12.: Customers and Orders form with datasheet subform.

▶ ▶ ▶ 🗮 🐒 No Filter Search

### **i** Note

You can modify the layout and design of both the main form and subform as needed.

## 6.5. Customer Data Entry Form

In this task, you'll create a form for adding new customers and editing existing ones. You'll also practice moving and aligning controls to match a specific layout:

Task 6.3. FILE: candy365.accdb

- 1. Open the database.
- 2. Select the Customers table (no need to open it).

Record: I4 4 1 of 4

3. Click Create > Form (Forms group). The form opens in Layout View.

Chocolate Conoisseur							
Customer e	ntry and	mo	dific	ation			
Customer code:	1						
	First name:		Last	name:			
Name:	Rita		Hen	driks			
Address:	2e Zeine 99	A					
ZipCode:	7233 AL		City:	Zwolle			
Province:	OV	Re	gion:	North	•		
Telephone:	03210-3014	9					

Figure 6.13.: Customer data entry form.

#### i Note

Since Customers has a one-to-many relationship with Orders, a subdatasheet based on Orders is added automatically.

- 4. Delete the subdatasheet by selecting it and pressing Delete.
- 5. Switch to Design View.
- 6. Remove the form logo in the header.
- 7. Save the form as Customer entry and modification.
- 8. Increase the height of the Form Header by dragging the top of the Detail section downward.
- 9. Change the title label from "Customers" to "Chocolate Connoisseur". Set the font to MS Sans Serif, 14 pt, italic, and bold.
- 10. Insert a new label via Design > Label (group Controls) and place it below the title. Type "Customer entry and modification" and format it with MS Sans Serif, 14 pt, bold, black.
- 11. Select all controls in the Detail section by dragging a rectangle around them.
- 12. Click Arrange > Remove Layout (Table group) to unstack the layout.
- 13. Deselect the controls.
- 14. Arrange the controls to match the desired layout.

### 🅊 Tip

- Align controls using Arrange > Align (group Sizing & Ordering).
- Add a new Label called Name: before the first and last name fields.
- Adjust size using Width and Height.
- Position controls using Top and Left.
- Set background color using Back Color.

- 15. Switch to Form View to test the layout
- 16. ave and close the form.

## 6.6. Praline Data Entry Form

This task involves creating a form to add or edit pralines. You'll also replace three text boxes with combo boxes for better usability. The final layout should look like Figure 6.14.

Pralines				
Praline code:	B01	Praline name	Candleli	ght Ecstasy
Chocolate type:	Butterscotch Milk Dark Toffee		Picture:	
Filling type:	Mocha cream	•		
Nut type:	Cashew	•	Costs:	\$0.30
Description:	Cashew in moo chocolate.	cha cream with a	layer butt	terscotch

Figure 6.14.: Praline data entry form with combo boxes.

#### Task 6.4. FILE: candy365.accdb

- 1. Open the database.
- 2. Select the Pralines table (no need to open it).
- 3. Click tab Create > Form (Forms group).
- 4. Switch to Design View.
- 5. Delete the BoxDetails subdatasheet.
- 6. Remove the form logo in the header.
- 7. Save the form with as Praline information.
- 8. Change the header text format to: Calibri, 22 pt, bold, white color on a green background.
- 9. Remove the stacked layout by selecting all controls in the Detail section and choosing Arrange > Remove Layout (Table group).
- 10. Deselect all controls.
- 11. Delete the fields ChocolateType, FillingType, and NutType. Reorganize the remaining fields to match the layout.
- 12. Use Design > Line (group Controls) to draw a horizontal line under PralineCode and PralineName. Set line thickness to 2 pt and match the header background color.
- 13. Add a combo box for ChocolateType using the List Box Wizard with these settings:

- Enter values manually.
- One column with values: Butterscotch, Milk, Dark and Toffee.
- Store value in ChocolateType.
- Label: "Chocolate type:"
- 14. Create a Combo Box for FillingType with values: Amaretto, Blueberry, Cherry cream, Coconut, Marzipan, Mocha cream, None.
- 15. Create a Combo Box for NutType with values: Almond, Cashew, Hazelnut, Macadamia, Paranut, Pecan, Pistachio, Walnut, None.
- 16. Switch to Form View and test the form.
- 17. Close the form and save the changes.

# 6.7. Boxes Data Entry Form

In this task, you'll replace a Text Box on an existing subform with a Combo Box that uses a query as its data source.

The existing Boxes form allows you to select pralines only by their PralineCode. The goal is to enable selection by praline name as well.

The Boxes form consists of a main form (Boxes) and a subform (Boxes subform). Praline selection happens in the subform. Therefore, only the subform needs to be modified—specifically, replacing the PralineName text box with a Combo Box.

However, the Boxes subform is based on the BoxDetails table, which does not contain the PralineName field. To support name selection, you first need to create a query that includes both PralineName and PralineCode, and sorts the results alphabetically by name. This new query will be the data source for the combo box.

### Task 6.5. FILE: candy365.accdb

### Query: Choose praline

- 1. Open the database.
- 2. Go to Create > Query Design (Queries group). Add the Pralines table. Add the fields PralineName and PralineCode. Sort by PralineName in ascending order.
- 3. Save the query as Choose praline and close it.

### Modify the Subform

- 4. Open the Boxes subform in Design View.
- 5. Delete the PralineName field from the Detail section.
- 6. Go to Design > Combo Box (Controls group) and draw a rectangle where the PralineName field was. The Combo Box Wizard starts.
- 7. Answer the wizard prompts as follows:
  - Choose: I want the values to come from another table or query.
  - Select the Choose praline query.
  - Add the fields PralineName and PralineCode.

6. Forms

		Pralines		
		*		]
	ÿ	PralineCode PralineName ChocolateType NutType FillingType PralineDescription		
			_	
			_	
	eld:	PralineName	$\sim$	PralineCode
Tal	ble:	Pralines		Pralines
S	ort:	Ascending		
Sh	ow:	$\checkmark$		$\checkmark$
Crite	ria:			
	or:			

Figure 6.15.: Query design for selecting pralines by name.

- Sort by PralineName (ascending).
- Adjust column widths if needed.
- Store the value from PralineCode into the Code field.
- Accept the default label (this will be removed later).

8. Delete the auto-generated label, as a label already exists in the form header.

			Chocolate Nut	PralineName	Code
					Ø Detail
Costs Jan	Ċosts	Filling	Chocolate Nut	Code	Code 💌
				Footer	

Figure 6.16.: Subform with Combo Box for praline selection.

9. Select the Combo Box and check the Limit To List property in the Property Sheet (tab Data). If it's not set to Yes, change it.

### **i** Note

Setting "Limit To List" to Yes ensures that only predefined praline names can be selected, new names cannot be entered manually.

- 10. Save and close the Boxes subform.
- 11. Open the main Boxes form and test the combo box functionality.

## 6.8. Orders Data Entry Form

To enter new orders and view existing ones, you'll create an order form that includes both main form data and order details in a subform.

# Orderdata

OrderCode	6	OrderDate	11/2/2009
CustomerCode	1	Telephone	03210-30149
FirstName	Rita	LastName	Hendriks
Address	2e Zeine 99 A		
ZipCode	7233 AL	City	Zwolle
Region	North	Province	OV

## Order details

	BoxCode	BoxName	Quantity	BoxPrice	Total	
•	BUTT	Butterscotch	1	\$27.75	\$27.75	
	FANT	Toffee Mocha Fantasy	1	\$18.00	\$18.00	
	HEAV	Heavenly Hazelnuts	2	\$15.75	\$31.50	=
	NOWI	Northwind Surprise	2	\$33.25	\$66.50	
	PEAN	Peanut Pralines	1	\$19.00	\$19.00	
	SWBI	Sweet and Bitter	2	\$27.75	\$55.50	
*						-
Recor	d: I4 → 1 of	6 🕨 🕨 👫 No Filter	Search			

Figure 6.17.: Order data entry form with subform for order details.

- The main form will display data from the Customers and Orders tables.
- The **subform** will show data from OrderDetails and Boxes, including a calculated Total field.

To supply data to these forms:

- Create a query Orders-Customers that combines the necessary fields from [Orders] and [Customers].
- Create a query OrderDetails-Boxes combining data from [OrderDetails] and [Boxes], with the [Total] calculated as Quantity \* BoxPrice.

Then use the Form Wizard to create the form and subform based on these queries.

#### Task 6.6. FILE: candy365.accdb

### Create Query: Orders-Customers

- 1. Open the database.
- 2. Go to Create > Query Design (Queries group). Add [Orders] and [Customers].
- 3. Add all fields from tbothtables.

### **i** Note

You can use \* to add all fields from a table.

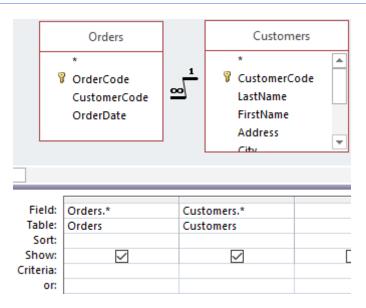


Figure 6.18.: Query design combining Orders and Customers.

4. Save the query as Orders-Customers and close it.

### Create Query OrderDetails-Boxes

- 5. Go to tab Create > Query Design (group Queries). Add tables OrderDetails and Boxes.
- 6. Add all fields from both tables.
- 7. Add a calculated field: Total: [Quantity]\*[BoxPrice].
- 8. Save the query as OrderDetails-Boxes and close it.

	OrderDetails				Boxes
8	oracroode		<u></u>	•	* A BoxCode BoxName Weight ToxDescription
		_		-	
Field:	OrderDetails.*	Boxe	s.*	Tot	al: [Quantity]*[BoxPrice]
Table:	OrderDetails	Boxe	s		
Sort:					
Show:	$\checkmark$		~		$\checkmark$
Criteria:					
or:					

Figure 6.19.: Query design combining OrderDetails and Boxes, including a calculated Total field.

#### Create the Form and Subform

- 9. Go to tab Create > Form Wizard.
- 10. First, select Orders-Customers and add all fields except Orders.CustomerCode.
- 11. Then select OrderDetails-Boxes and add: [Boxes.BoxCode], [BoxName], [Quantity], [Box-Price], [Total]
- 12. Click Next, and choose to view the data by Orders-Customers. This makes it the main form, with OrderDetails-Boxes as the subform.
- 13. Click Next again and select the Tabular layout for the subform.
- 14. Choose a style and click Next.
- 15. Enter the form names:
  - Main form: Orderdata main'
  - Subform: Orderdata sub'
- 16. Click Finish. The form opens automatically.
- 17. Adjust the layout of both forms to achieve the desired final design.

### 6.9. Total Order Amount

In this section, you'll enhance the form created in Section 6.8 by calculating and displaying the total amount of an order in the subform footer.

OrderCode	6	OrderDate	11/2/2009
CustomerCode	1	Telephone	03210-30149
FirstName	Rita	LastName	Hendriks
Address	2e Zeine 99 A		
ZipCode	7233 AL	City	Zwolle
Region	North ~	Province	OV

# Order details

	BoxCode	BoxName	Quantity	BoxPrice	Total 📤
	BUTT	Butterscotch	1	\$27.75	\$27.75
	FANT	Toffee Mocha Fantasy	1	\$18.00	\$18.00
	HEAV	Heavenly Hazelnuts	2	\$15.75	\$31.50
	NOWI	Northwind Surprise	2	\$ 33.25	\$66.50
	PEAN	Peanut Pralines	1	\$19.00	\$19.00
		1	Order to	tal:	\$218.25
Recor	d: I4 → 1 of	6 🕨 🕨 🎎 🐁 No Filter	Search	•	Þ

Figure 6.20.: Orders with total amount.

OrderCode	6	OrderDate	11/2/2009
CustomerCode	1	Telephone	03210-30149
FirstName	Rita	LastName	Hendriks
Address	2e Zeine 99 A		
ZipCode	7233 AL	City	Zwolle
Region	North 🗸	Province	OV

### Order details

	BoxCode	BoxName	Quantity	BoxPrice	Total 📤
	BUTT	Butterscotch	1	\$ 27.75	\$27.75
	FANT	Toffee Mocha Fantasy	1	\$18.00	\$18.00
	HEAV	Heavenly Hazelnuts	2	\$15.75	\$31.50
	NOWI	Northwind Surprise	2	\$33.25	\$66.50
	PEAN	Peanut Pralines	1	\$19.00	\$19.00
		r	Order to	tal:	\$218.25
Record	d: I4 → 1 of	6 🕨 🕨 🎉 🐁 No Filter	Search	•	

Figure 6.21.: Order form with calculated total amount.

### Task 6.7. FILE: candy365.accdb

- 1. Open the database.
- 2. Open the Orderdata sub form in Design View.
- 3. In the Property Sheet, set the selection type to Form. Change the Default View to Continuous Forms. This setting is necessary because otherwise the total cannot be calculated.
- 4. Increase the height of the Form Footer section by dragging its bottom edge downward.
- 5. Go to Design > Text Box (Controls group) and draw a text box in the footer where the total should appear.
- 6. Set the Control Source of the text box to =Sum([Total]) and its Format to Currency.
- 7. Save and close the subform.
- 8. Open the main form Orderdata main and verify the total appears and updates correctly.

Prop Selection			et				× 2↓
Form					$\sim$		
Format	Data	Event	Oth	ner	All		
Caption				Ord	erdata sub		
Default	View			Con	tinuous For	ms	$\sim$
Allow Form View				Yes			
Allow Datasheet View				Yes			
Allow Layout View				Yes			
Picture Type Embedded							

6. Forms

Figure 6.22.: Setting Default View to Continuous Forms for total calculation.

# 6.10. Form with Chart

## GOAL

Create a form with a column chart showing sales per box, per month, for the year 2010.

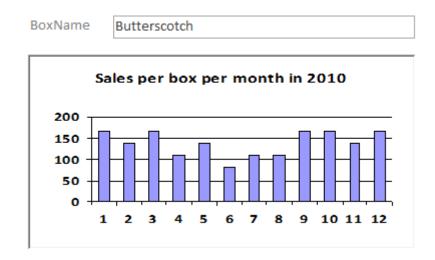


Figure 6.23.: Form displaying a column chart of sales per box per month (2010).

## ANALYSIS

- BoxName: from the Boxes table.
- Sales: calculated as [Quantity]\*[BoxPrice].
- **BoxPrice**: from Boxes table.
- Quantity: from OrderDetails table.
- **OrderDate**: from Orders table; Month and Year can be calculated with the functions Month([OrderDate]) and Year([OrderDate]).

You will:

- Create a query that outputs BoxName, Sales, Month, and Year.
- Group by [BoxName] and then by [Month].
- Filter on [Year] = 2010.

Task 6.8. FILE: candy365.accdb

### Create Query

- 1. Open the database.
- 2. Go to tab Create > Query Design (Queries group) and add the tables Boxes, OrderDetails, and Orders.
- 3. Add field BoxName from table Boxes, and create the calculated fields: Month: Month([OrderDate]), Sales: [Quantity]\*[BoxPrice] en Year: Year([OrderDate]).
- 4. Click Design > Totals (Show/Hide group).
- 5. Change in the column Sales the Group By operator in Sum and in the column Year in Where. Add 2010 as criteria for Year.

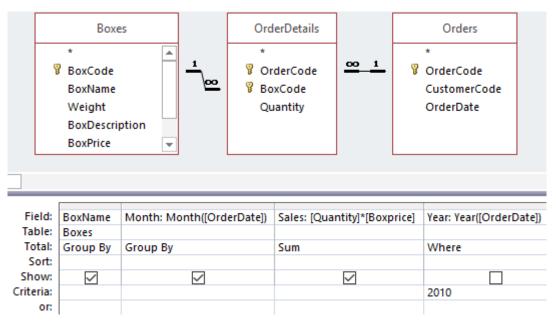


Figure 6.24.: Query design: sales per box per month in 2010.

6. Save the query as Sales per box per month in 2010 and close it.

#### i Note

After reopening the query in Design View, Access may reformat the [Sales] column to:

- Field: Sales: Sum([Quantity]\*[BoxPrice]).
- Total:Expression.

#### Create Form

- 7. Click tab Create > Form Wizard. Use table [Boxes] and select BoxName. Click Next.
- 8. In the wizard:
  - Choose Datasheet layout.
  - Title: "Sales per box per month in 2010".
  - Save the form with the same name as the title.

#### 6. Forms

- 9. Open the form in Design View and expand the Detail section.
- 10. Click tab Design > Chart (Controls group), , and draw a rectangle. The Chart Wizard starts.
- 11. In the wizard:
  - Use query Sales per box per month in 2010
  - Fields: Month, Sales.
  - Chart type: Column Chart.

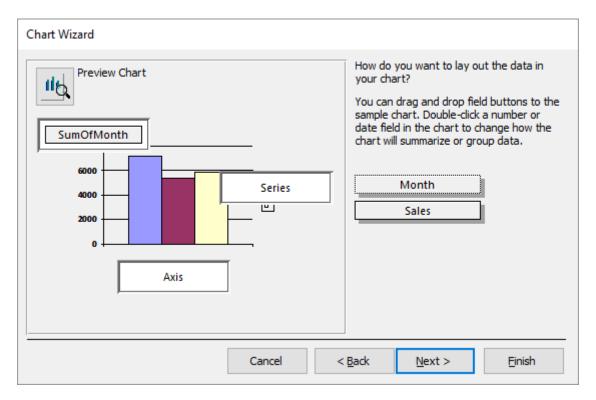


Figure 6.25.: Initial chart layout generated by the Chart Wizard.

- 12. Drag SumOfMonth to the horizontal axis. It changes to Month.
- 13. Drag Sales to the vertical axis. It updates to SumOfSales.
- 14. Click Next. Now you should select the field in the form that should be linked to the field in the chart. In both cases it is BoxName.
- 15. Click Next. Set the chart title: "Sales per box per month in 2010". Disable the legend and click Finish.

#### 🔔 Warning

In Design View, a placeholder chart appears, not live data.

- 16. Save and close the form.
- 17. Open the form in Form View, browse some records, then close the form.

## 6. Forms

Chart Wizard	
Preview Chart SumOfSales 6000 4000 4000 5eries 5eries Month	How do you want to lay out the data in your chart? You can drag and drop field buttons to the sample chart. Double-click a number or date field in the chart to change how the chart will summarize or group data. <u>Month</u> Sales
Cancel <	Back Next > Einish

Figure 6.26.: Final chart layout with proper data axes.

Chart Wizard			
XXX AAA XXX XXX XXX BBB XXX XXX	If you want the chart to change link the document and the chart	e from record to record, select the fields th t.	at
XXX CCC XXX XXX XXX XXX XXX XXX	Form Fields	Chart Fields:	
	BoxName	V BoxName V	
		×	
	Cancel	< Back Next >	inish

Figure 6.27.: Linking form data to chart fields.

# 6.11. Linking forms

The easiest way to create linked forms is by using the Form Wizard. This wizard generates all necessary forms, creates the required links, and adds a command button on the main form to open the related form.

The goal is to create a form that displays customer information, with a command button to open a separate form that shows the customer's order details. See Figure 6.28.

Order data			ustomer	Orders
CustomerCode LastName FirstName Address ZipCode City	1       Hendriks       Rita       2e Zeine 99 A       7233 AL       Zwolle		OrderCode OrderDate Order data BUTT FANT HEAV NOWI PEAN SWBI * Record: 14 4 1	6 11/2/2009 V Quantity V 1 1 2 2 1 2 1 2 1 2 1 2
		Basard	l l 1 of 1	N N N" Filterad Search

Figure 6.28.: Two linked forms with synchronized data.

The main form displays customer data, which comes from the Customers table. The linked form shows the customer's orders, with order code and date, from the Orders table. It also includes a subform that shows order details, sourced from the OrderDetails table.

#### Task 6.9. FILE: candy365.accdb

- 1. Open the database.
- 2. Go to tab Create > Form Wizard (group Forms).
- 3. Select Table: Customers.
- 4. Add the following fields:
  - From Customers: CustomerCode, LastName, FirstName, Address, ZipCode, City.
  - From Orders: OrderCode, OrderDate.
  - From [OrderDetails: BoxCode, Quantity.
- 5. Click Next. Choose to view data by Customers and select Linked Forms as the layout.
- 6. Click Next. You'll be prompted to name the forms.
- 7. Use the following names:

### $6. \ Forms$

Form Wizard			
	Which fields do you want on your form? You can choose from more than one table or query.		
<u>T</u> ables/Queries			
Table: Customers	$\sim$		
<u>A</u> vailable Fields:	Selected Fields:		
CustomerCode  LastName FirstName Address City Province ZipCode Region  V	> >> <<		
Car	ncel < Back Next > Finish		

Figure 6.29.: Form Wizard: Selecting customer-related fields.

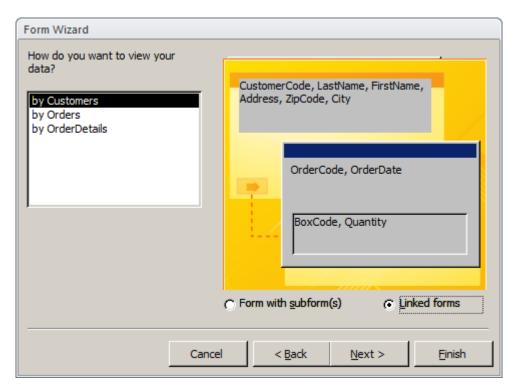


Figure 6.30.: Form Wizard: Selecting linked forms display option.

- Main form: CustomerOrders
- First linked form: CustomerOrdersSub1
- Subform: CustomerOrdersSub2

8. Click Finish.

The CustomerOrders main form appears. However, the command button to open the linked form is hidden beneath the form title and cannot be clicked.



Figure 6.31.: Command button hidden beneath form title—requires layout adjustment.

- 9. Open the CustomerOrders form in Design View.
  - Delete the CustomerOrders label.
  - Select the command button and, in the Property Sheet, change the caption to "Order data".
- 10. Optionally, adjust the layout of all three forms.
- 11. Switch to Form View and test the command button.
- 12. Save and close the form.

💡 Tip

By default, Access opens each object (tables, queries, forms, reports) in a separate tab. However, with linked forms, this isn't ideal, you won't see both forms at once. To show forms in overlapping windows instead of tabs:

- 1. Go to File > Options > Current Database.
- 2. Under Document Window Options, select Overlapping Windows.

# 6.12. Exercises

## Exercise 6.1. Customers (form001)

Create a form named form001 that looks like the following example.

Customers			
Customer code:	1		
Last name:	Hendriks		
First name:	Rita		
Address:	2e Zeine 99 A		
City:	Zwolle		
Province:	OV		
Zip code:	7233 AL		
Region:	North		
Telephone:	03210-30149		

Figure 6.32.: Target layout for form001 – Customers form.

## Exercise 6.2. Pralines in Boxes (form002)

Create a main form with a subform that shows which boxes contain a specific bonbon and in what quantity. Name the forms:

- Main form: form002 main
- Subform: form002 sub

Pralines in	Boxes	
Praline code	B03	
Praline name	Marzipan Oakleaf	Picture
Chocolate type	Butterscotch	
Nut type	None	
Filling type	Marzipan	
Pralir	\$ 0.40	
Praline description	Marzipan in the shape of a big o butterscotch chocolate stem.	ak leave with a
Pralines in boxes		
🔬 BoxCode 👻 C	uantity 👻	
AUTU	6	
BUTT	5	
MARZ	3	
*		
Record: I + 1 of 3	► N N N	

Figure 6.33.: Target layout for form002 – Pralines in Boxes.

## Exercise 6.3. Boxes per Customer (form003)

Create a main form with a subform. The main form displays box data; the subform shows total sales per customer for the selected box.

First, create a query that calculates total box sales per customer. Name everything as follows:

- Query: Boxes per customer
- Main form: form003 main

• Subform: form003 sub

Boxes per cu	istomer					
BoxCode	ALLS	BoxNan	ne	All Seas	ons	
Weight	150	BoxPric	e	\$14.00		
BoxDescription	Strawberries, enjoy in all se				to	
Stock	700	OutOfP	roduction	No		
Boxes per customer	Z Customer	Code 👻	LastName	- Quai	ntity 👻	
		16	Stoffers		2	≡
		33	Huizinga		2	
		37	Lange, de		2	
		40	Winters		2	
		56	Meijer		2	
		90	Franssen		1	
		91	Drost		2	
			Rolvink		2	
	Record: M 🚽 1	of 21	► H H≣   🐨	No Filter	Search	

Figure 6.34.: Target layout for form003 – Boxes per Customer.

#### **OBJECTIVES**

- What reports are used for.
- How to create labels using the Wizard.
- Creating an auto-generated report with manual adjustments
- Creating a grouped report.
- Creating a label report with chocolate images

Reports are structured summaries typically intended for printing. Labels are also a form of report.

# 7.1. Creating Reports

Reports are used to present summaries or overviews of large data sets, especially when preparing them for printing.

For example, to clearly print a sales summary on paper, you would use a report. Reports can include subtotals and grand totals. You can build reports manually, but using the Wizard is much more efficient.

Reports can also be dynamic through the use of parameters. When opening a report, you may be prompted to enter criteria. For example, the Sales per box report asks for a start and end date when opened.

# 7.2. Creating Labels

This section explains how to create address labels using the Label Wizard.

#### Task 7.1. FILE: candy365.accdb

- 1. Open the database.
- 2. Select the Customers table.
- 3. Go to Create > Labels (group Reports).
- 4. Choose Metric units, manufacturer Avery, and product Avery L7160, see Figure 7.1.
- 5. Click Next. On the next screen, you can customize the font and color of the label text.
- 6. Accept the default settings and click Next.

Label Wizard				
	This wizard creates stand What label size would you Product number: Avery L7160 Avery L7161 Avery L7162 Avery L7163 Avery L7164 Unit of Measure O English O Measure Filter by manufacturer:	I like? Dimensions: <b>163,5 x 38,1 mm</b> 63,5 x 46,6 mm 99,0 x 33,9 mm 99,0 x 38,1 mm 63,5 x 72,0 mm Labe etric	Number across:	
	Filter by manufacturer:     Avery       Customize     Show custom label sizes			
	Canc	el < <u>B</u> ack	Next > Einish	

Figure 7.1.: Label selection: Avery L7160 (metric, A4)

Label Wizard	
/	What would you like on your mailing label? Construct your label on the right by choosing fields from the left. You may also type text that you would like to see on every label right onto the prototype.
Available fields:	Prototype label:
CustomerCode LastName FirstName Address City Province	
	Cancel < Back Next > Finish

Figure 7.2.: Inserting fields in the Label Wizard.

#### i Note

You can insert fields by double-clicking the field name or using the > button. The field name appears in curly braces on the sample label. You can type regular text and use Enter to create line breaks.

7. Create the following sample label (with one space between first and last name, and two spaces between zip code and city):

{FirstName} {LastName}
{Address}
{ZipCode} {City}

- 8. Click Next. Choose to sort the labels by ZipCode.
- 9. Click Next. On the final screen, give the report the name Labels Customers and click Finish.

Roel Faber	Joost Wever	Daan Leegstra
Kernkade 1 G	Markt	Geenraderweg 24
07754 Amsterdam	1002 EK Amstelveen	1047 AR Maastricht
Wilma Oosterhof	Helen Peters	Samantha Huisman
Koppelstraat 70	Akkerpad 9	Edisonweg 9
1070 WD Amsterdam	1092 HC Meppel	1116 NW Amsterdam
Bert Evertsma	André Oosterbrink	

Figure 7.3.: Print Preview of customer address labels.

Industrieven 6

Honistraat 66

10. Close the report.

## 7.3. Auto-Generated Report

Access can generate a basic report from a table or query. While the result is functional, it often needs manual refinement.

#### Task 7.2. FILE: candy365.accdb

- 1. Open the database.
- 2. Select the Sales per region per box query.

Hooistraat 66

3. Go to Create > Report (Reports group). The report is generated and opens in Layout View.

You'll notice that the Region field is repeated for each record and that the sales amounts are not formatted as currency. We'll fix this.

- 4. Close the report and confirm saving when prompted.
- 5. Name the report Sales per region per box and click OK.

- 6. Reopen the report in Design View.
- 7. Select the Region field in the Detail section. In the Property Sheet, set Hide Duplicates to Yes.

i Note

The Property Sheet can be made visible and invisible by clicking on button Property Sheet or by using hotkey F4.

- 8. Select the Sales field and set its Format to Currency.
- 9. Switch to Report View. Now the Region is shown only once per group and the sales values are formatted properly.
- 10. Save and close the report.

# 7.4. Grouped Report

#### GOAL

Create a report that shows, for a specified period, the sales per box, including order details. For example, see Figure 7.4, which shows a portion of the report for November 2009.

Sales per box	per order p	ber pe	riod
Daublanca	All Casara		
BoxName	All Seasons		
OrderCode	OrderDate	Quantity	Sales
45	11/8/2009	2	28
59	11/10/2009	2	28
74	11/12/2009	1	14
89	11/14/2009	1	14
119	11/18/2009	2	28
162	11/24/2009	1	14
178	11/27/2009	2	28
192	11/29/2009	2	28
Summarv for 'BoxNar	me' = All Seasons (8	detail reco	rds)
Sum		13	182
BoxName	Autumn Sur	prise	
OrderCode	OrderDate	Quantity	Sales
18	11/4/2009	2	86
34	11/6/2009	1	43
лл	11/8/2009	1	12

Figure 7.4.: Report November 2009 (partial view).

ANALYSIS

The report needs the following fields:

- BoxName
- OrderCode
- OrderDate
- Quantity
- Sales, calculated as [Quantity] \* [BoxPrice]

A query named Sales per box per order per period already exists and includes all these fields.

#### Task 7.3. FILE: candy365.accdb

- 1. Open the database.
- 2. Select the query Sales per box per order per period.
- 3. Go to tab Create > Report Wizard (Reports group).

Report Wizard	
	Which fields do you want on your report? You can choose from more than one table or query.
<u>T</u> ables/Queries	
Query: Sales per box per order per	period 🗸
<u>A</u> vailable Fields:	Selected Fields:
BoxName OrderCode OrderDate Quantity Sales	>       <
Ca	ncel < Back Next > Finish

Figure 7.5.: Selecting fields to include in the grouped report.

- 4. Add all fields from the query, then click Next.
- 5. Remove the default grouping (OrderCode) and add BoxName as the grouping level.
- 6. Click Next. Set sorting to ascending by OrderCode.
- 7. Click Summary Options.... Enable summing for both Quantity and Sales.
- 8. Click OK, then Next.
- 9. Choose the Outline layout.
- 10. Click Next. Enter the report name: Sales per box per order per period, then click Finish.

Report Wizard	
Do you want to add any grouping levels?	BoxName OrderCode, OrderDate, Quantity, Sales
Grouping Options Cancel	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish

Figure 7.6.: Choosing grouping levels for the report.

Report Wizard		
What sort order and summary inform	ation	do you want for detail records?
		ou can sort records by up to four fields, in either scending or descending order.
	1	OrderCode 🗸 Ascending
	2	<ul> <li>✓ Ascending</li> </ul>
	3	✓ Ascending
	4	✓ Ascending
		Summary Options
Car	ncel	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish

Figure 7.7.: Defining sort order and summary options.

Summary Options		
What summary values would Field Quantity Sales	you like calculated?	OK Cancel Show © Detail and Summary Summary Only Calculate percent of total for sums

Figure 7.8.: Enabling calculated totals for Quantity and Sales.

Report Wizard		
How would you like to lay out your report?		
-	Layout	Orientation
	○ <u>S</u> tepped	●Portrait
	O Block	○Landscape
	Outline	
		A
E CERTIFICATION OF THE REAL PROPERTY OF THE REAL PROPERTY OF THE REAL PROPERTY OF THE REAL PROPERTY OF THE REAL		
	Adjust the field widt a page.	h so all fields fit on
Cancel	< Back Next :	> <u>F</u> inish
Calicei	< Dark Wext	Ellist

Figure 7.9.: Choosing the report layout style.

- 11. When prompted, enter the start and end dates. For example:
  - Start Date: 11/1/2009
  - End Date: 11/30/2009
- 12. Close the report after verifying the output.

## 7.5. Praline Pictures

In this task, you'll create a report that includes images of pralines, with the praline code and name displayed above each image. This will be done using labels, with one praline per label.

#### Task 7.4. FILE: candy365.accdb

- 1. Open the database.
- 2. Select the Pralines table.
- 3. Go to tab Create > Labels (Reports group).
- 4. Choose Metric as the unit, manufacturer Zweckform, and product Zweckform 3415.

Figure 7.10.: Label selection: Zweckform 3415 (metric format).

- 5. Click Next. On the next screen, you can change the font and text color.
- 6. Accept the default settings and click Next.
- 7. Create the following prototype label ( with one space between the fields): {PralineCode} {PralineName}
- 8. Click Next and sort the labels by PralineCode.

- 9. Click Next. Name the report Labels Pralines and click Finish.
- 10. The report is generated and displayed in Print Preview. Switch to Design View.
- 11. Click Design > Bound Object Frame (group Controls),  $\square$ , and draw a square about 1 inch by 1 inch in the Detail section.

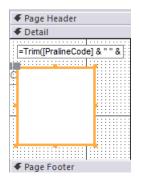


Figure 7.11.: Inserting a Bound Object Frame for images.

- 12. With the frame selected, adjust the following properties in the Property Sheet:
  - In the [Format] tab, set Width and Height to 1" (Access may slightly adjust the dimensions).
  - In the [Data] tab, set the Control Source to the field Picture.

Format Data Event Ot	her All
Visible	Yes
Size Mode	Zoom
Width	1
Height	1
Top	0.375

Figure 7.12.: Setting dimensions of the image frame to 1x1 inch.

Format	Data	Event	Oth	er	All	Ì	
Control	Source			Pict	ure		×
Source	Item						
Source	Doc						

Figure 7.13.: Binding the image frame to the Picture field.

- 13. Select the label that is located behind the image frame.
- 14. Delete the label using the Delete key.
- 15. Switch to Print Preview.

The images are now displayed, but their vertical alignment may vary. To ensure consistency, the text fields should have a fixed height so all images align evenly.

- 16. Return to Design View, select the text box, and set the following properties:
  - Height: 0.4"
  - Can Grow: No
  - Can Shrink: No
- 17. Align the text box and the image frame to the left.



✓ Page Header ✓ Detail	
=Trim([PralineCod	e] & " " &
✓ Page Footer	

Figure 7.14.: Selecting the default label underneath the image frame.

Format Data Event O	ther All
Format	L
Decimal Places	Auto
Visible	Yes
Width	1.375"
Height	0.4"
Тор	0.0833"
Left	0.1
Back Style	Normal
Kight Padding	0.0208
Hide Duplicates	No
Can Grow	No
Can Shrink	No
Display When	Always
Reading Order	Context

Figure 7.15.: Setting fixed height and disabling growth/shrink for the text box.

✓ Page Header ✓ Detail	
=Trim([PralineCode] & [PralineName])	&""
······	1
Page Footer	

Figure 7.16.: Aligning the text box and image frame.

18. Switch back to Print Preview to verify the layout.

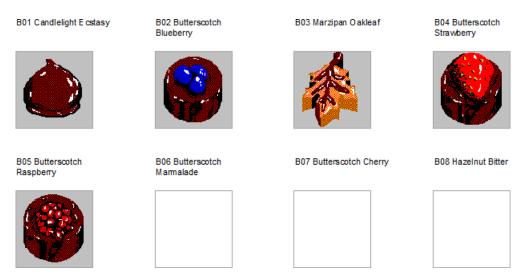


Figure 7.17.: Final result: labels with praline name, code, and image.

19. Save and close the report.

# 7.6. Exercises

## Exercise 7.1. Sales per Region per Box (rapp001)

Create a report showing sales per region per box. The report should also include the total sales for each region and show the percentage of total sales across all regions. Name the report rapp001.

Region North	
BoxName	Sales
All Seasons	\$210.00
Autumn Surprise	\$946.00
Butterscotch	\$3,052.50
Classic Cherry	\$942.50
Heavenly Hazelnuts	\$882.00
International	\$1,156.00
Island Assortment	\$1,085.00
Love Hearts	\$2,275.00
Marzipan Surprise	\$709.50
Mountain Assortment	\$933.75
Northwind Surprise	\$4,355.75
Peanut Pralines	\$722.00
Perfects	\$383.25
Romantic Pralines	\$1,712.50
Sweet and Bitter	\$2,025.75
Sweet and Creamy	\$1,196.00
Toffee Mocha Fantasy	\$1,386.00
Tropical Surprise	\$714.00
Summary for 'Region' = North (18 d	
Sum	\$24,687.50
Standard	46.3%

# Sales per region per box with totals

Figure 7.18.: Example output of report [rapp001]: Sales per region per box.

## Exercise 7.2. Sales per Box per Region (rapp002)

Create a report that shows, for each box, the sales figures per region. Name the report rapp002.

ales per box	per region		
BoxName	Region	Sales	
All Seasons			
	North	\$210.00	
	South	\$266.00	
Summary for 'BoxNa	ame' = All Seasons (2 d	etail records)	
Sum \$476.00			
Standard		0.89%	
Autumn Surprise			
	North	\$946.00	
	South	\$1,935.00	
Summary for 'BoxNa	ame' = Autumn Surpris	e (2 detail record	
Sum		\$2,881.00	
Standard		5.40%	
Butterscotch			
	North	\$2.052.50	

Figure 7.19.: Example output of report [rapp002]: Sales per box per region.

## Exercise 7.3. Box and Praline Costs (rapp003)

Create a report that lists all pralines in each box, including the cost of each praline. Also, include the total praline cost per box. Name the report rapp003.

Box costs wit	h Praline costs	
Box	Praline	PralineCosts
All Seasons	Butterscotch Raspberry	\$0.50
	Butterscotch Strawberry	\$0.46
	Sweet Blueberry	\$0.50
	Sweet Raspberry	\$0.52
	Sweet Strawberry	\$0.40
	Walnut Mocha Toffee	\$0.38
Sum		\$2.76
Autumn Surprise	Marzipan Finch	\$1.92
	Marzipan Maple Leaf	\$2.22
	Marzipan Oakleaf	\$2.40
	Marzipan Swallow	\$2.04
Sum		\$8.58
Rutterscotch	Rutterscotch Cherry	\$1 0.1

Figure 7.20.: Example output of report [rapp003]: Praline costs per box.

#### Exercise 7.4. Box and Praline Costs Grouped (rapp004)

Create a copy of the rapp003 report and name the copy rapp004. Modify the report so that each box starts on a new page. To do this, set the Force New Page property of the group header to Before Section.

Also adjust the layout slightly: move the Sum text box to the right and draw a small horizontal line above the sum.

Box costs with	n Praline costs		
Box	Praline		PralineCosts
All Seasons	Butterscotch Raspberr	у	\$0.50
	Butterscotch Strawber	ry	\$0.46
	Sweet Blueberry		\$0.50
	Sweet Raspberry		\$0.52
	Sweet Strawberry		\$0.40
	Walnut Mocha Toffee		\$0.38
	-	Sum	\$2.76
Autumn Surprise	Marzipan Finch		\$1.92
	Marzipan Maple Leaf		\$2.22
	Marzipan Oakleaf		\$2.40
	Marzipan Swallow		\$2.04
	-	Sum	\$8.58
Butterscotch	Butterscotch Cherny		¢1 0/I

Figure 7.21.: Example output of report [rapp004]: Grouped praline costs per box with page breaks.

## Exercise 7.5. Sales per Box per Order per Period (rapp005)

Create a copy of the Sales per box per order per period report and name it rapp005. Modify the layout so that the total sales per box are printed next to the box name.

Sales per box	per order	per per	riod
BoxName All Seaso	ns	13	182
OrderCode	OrderDate	Quantity	Sales
45	11/8/2009	2	28
59	11/10/2009	2	28
74	11/12/2009	1	14
89	11/14/2009	1	14
119	11/18/2009	2	28
162	11/24/2009	1	14
178	11/27/2009	2	28
192	11/29/2009	2	28
BoxName Autumn	Surprise	23	989
OrderCode	OrderDate	Quantity	Sales
18	11/4/2009	2	86

Figure 7.22.: Modified report [rapp005]: Totals printed next to box names

## Exercise 7.6. Box Data on Separate Pages (rapp006)

Create a copy of the rapp005 report and name it rapp006. Modify the design so that each box starts on a new page.

## Exercise 7.7. Box Labels (rapp007)

Create a label report for all boxes following the example below. Use Avery J8163 labels (1½" x 3.9/10"), font: Consolas, size: 12 pt, black. Name the report rapp007.

All Seasons

Code : ALLS Weight: 150 gram

### Exercise 7.8. Yearly Sales per Region (rapp008)

Create a report based on a parameter query with grouping.

The report should show the number of boxes sold per region for a user-specified year. When opening the report, the user should be prompted to enter the year. Name the report rapp008.

Yearly s	ales per r	egion
Region North	BoxCode	Quantity
	ALLS	15
	AUTU	16
	BUTT	72
	CHER	29
	FANT	49
	LIEAN/	41

Figure 7.23.: Report [rapp008]: Yearly box sales per region (example: 2009).

Part II.

Advanced

# 8. Integration with Office

### OBJECTIVES

- Exchange data between Access and Excel or Word.
- Create a mail merge letter in Word using an Access address table.
- Export a table to a new RTF document.
- Export a table to Excel.

Access, Excel, PowerPoint, and Word are often used as standalone programs. However, in many business scenarios, collaboration between them is essential—and Microsoft Office offers plenty of options to support this integration.

# 8.1. About Office Integration

The Microsoft Office applications are designed to work together, allowing seamless data exchange across programs. Still, many users treat Access, Excel, and Word as entirely separate tools—each with its own strengths:

- Access for managing structured data (databases).
- Excel for calculations and numerical analysis.
- Word for word processing.

Access allows you to export data from tables, queries, and forms to Excel for calculations that may not be possible within Access. Similarly, Access can import data from Excel worksheets.

You can also export Access data to Word in RTF format (Rich Text Format). Additionally, Access tables or queries can be used as data sources in Word's Mail Merge function, which can be initiated from either Word or Access.

# 8.2. Mail Merge

In this task, you'll send a form letter to customers announcing a new voucher box called Snow White. The address data will come from the Customers table.

#### Task 8.1. FILE: candy365.accdb

- 1. Open the database.
- 2. Select the Customers table.
- 3. Right-click the table and choose Export > Word Merge.
- 4. Choose Link your data to an existing Microsoft Word document and click OK.

#### 8. Integration with Office

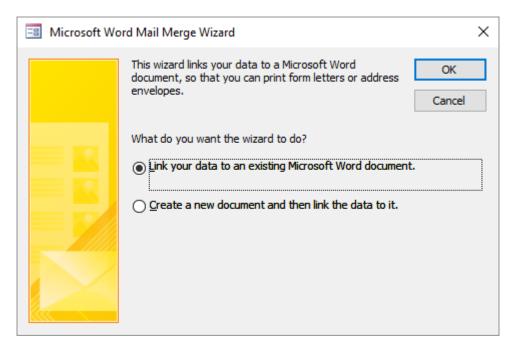


Figure 8.1.: Microsoft Word Mail Merge Wizard.

5. Select the practice file snow-white.docx and click Open. Microsoft Word opens with the document loaded. The Mailings tab is active, and the Mail Merge panel appears on the right. You're now in step 3 of 6.



Figure 8.2.: Progress wizard mail merge: step 3 of 6.

- 6. Click Next: Write your letter.
- 7. Place the cursor at the top of the letter. Go to Mailings > Insert Merge Field > FirstName.
- 8. Then add the following fields in the format below:

```
<<FirstName>> <<LastName>>
<<Address>>
<<ZipCode>> <<City>>
```

- 9. In the Mail Merge panel, click Next: Preview your letters (Step 4 of 6). The first customer's letter will be displayed.
- 10. Then click Next: Complete the merge (Step 5 of 6).

**i** Note

You now have two options:

- Print to send letters directly to the printer.
- Edit individual letters to generate one document with all letters.

8. Integration with Office

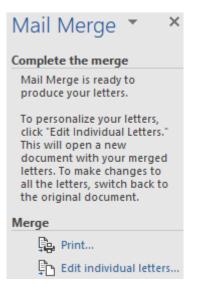


Figure 8.3.: Finishing mail merge.

11. Click Edit individual letters.

Merge to New Doc	ment	?	×
Merge records <ul> <li>All</li> <li>Curr<u>e</u>nt record</li> <li><u>F</u>rom:</li> </ul>	<u>I</u> o:		
C	(	Canc	el

Figure 8.4.: Selection of records.

- 12. In the prompt, choose to generate letters for the first 10 customers. Click OK. A new Word document with 10 letters is created.
- 13. Save the document as Invitation new box and close Word.

# 8.3. Export to Word

Goal: Export an Access table to a new RTF document.

If you want to insert the content of a table or query into an existing Word document, the easiest method is to copy and paste. However, you can also export directly to a new Word file. This creates a Rich Text Format (.rtf) file that can be opened in Word.

Task 8.2. FILE: candy365.accdb

- 1. Open the database.
- 2. Select the Boxes table.
- 3. Right-click the table and choose Export > Word RTF File. The Export RTF File dialog appears.

- 4. Enter the file name and path, then click OK. The Save Export Steps dialog appears.
- 5. You don't need to save the export steps. Click Close.
- 6. Open the exported file in Word to verify.

# 8.4. Export to Excel

Goal: Export an Access table to a new Excel workbook.

Exporting a table from Access to Excel is straightforward.

### Task 8.3. FILE: candy365.accdb

- 1. Open the database.
- 2. Select the Boxes table.
- 3. Right-click the table and choose Export > Excel. The Export Excel Spreadsheet dialog appears.
- 4. Specify a file name and path, select Export data with formatting and layout, then click OK.
- 5. In the next dialog (Save Export Steps), click Close—saving is optional.
- 6. Open the file in Excel to check the export.

# 8.5. Exercises

#### Exercise 8.1. Form Letter (intg001)

The marketing department wants to promote the new voucher box to customers who have previously purchased a box containing white chocolate pralines.

Create the required query, and then use it as the source for a form letter.

- Criteria: If a box contains at least one white chocolate praline and was ordered by a customer, that customer should receive the letter.
- Ensure that each customer receives no more than one letter.

## Exercise 8.2. Export Customer Data (intg002)

Export all customer information to an Excel file.

# 9. Tools

## **OBJECTIVES**

- Analyze tables and detect data inconsistencies
- Compact and repair databases to improve performance .

# 9.1. About Tools

Access provides several tools to help manage and optimize your database. This chapter covers the following:

- Table analysis
- Compacting and repairing databases

Additional useful tools in Access include:

- Performance Analyzer
- Database Documenter
- Encrypt with Password
- Creating a Switchboard

# 9.2. Analyze Table

Goal: Detecting inconsistencies in tables.

When the same piece of information is stored in multiple places, it's called **redundancy**. This is problematic because if that information changes, you must update it in every location. If you don't, your database becomes **inconsistent**—meaning the same data exists in contradictory forms.

Access includes an Analyze Table wizard that helps detect redundancy and split a table into multiple related tables for more efficient data storage. This process is called **normalization**.

You can allow the wizard to decide how to split the data, or you can make those decisions manually. After splitting, the wizard can help you clean up the original data. At the end, you can choose to create a query that merges the normalized tables into a view that resembles the original.

#### 9. Tools

In Figure 9.1, you see an example of a table named Contacts with lots of duplicated data this table is not normalized. Access can split it into multiple tables so that redundant data is minimized.

Tontacts					
ID 👻	ContactName 👻	CompanyName 👻	Tel 👻	Subject 👻	ContactDate 👻
1	Ralph Jansen	Jansen Inc.	053-4321976	Purchase database	2/12/2010
2	Ralph Jansen	Jansen Inc.	053-4321976	Implementation software	2/9/2010
3	Peter Smith	Global Personel	074-2233654	Installation network	2/10/2010
4	Ralph Jansen	Jansen Inc.	053-4321976	Training personel	3/15/2010
5	Peter Smith	Global Personel	074-2233654	Training personel	3/22/2010

Figure 9.1.: Original Contacts table with redundant data (not normalized).

#### Task 9.1. PRACTICE FILE: tools.accdb

- 1. Open the practice file.
- 2. Select the Contacts table.
- 3. Go to Database Tools > Analyze Table (group Analyze). The **Table Analyzer** wizard opens with an introduction explaining redundant data.
- 4. Click Next. The next screen provides information on how the wizard can resolve redundancy.
- 5. Click Next again. The wizard asks which table contains redundant data.
- 6. Select the Contacts table and click Next. Now you'll choose whether you or the wizard will decide how to separate the data.
- 7. Select the option to make the decisions yourself, then click Next.

#### i Note

In this step, you can create new tables and assign fields to them.

- 8. Select the ContactName field and drag it outside the current table area. A new table is created containing that field. The wizard prompts you to name the new table.
- 9. Name the new table Contactpersons and click OK.

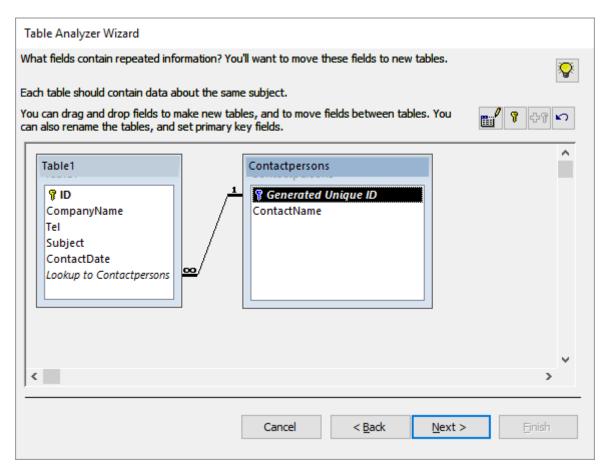
#### i Note

- A relationship is automatically created between the original and the new table.
- You can adjust the table positions and sizes by dragging them or their borders.
- 10. Drag the fields CompanyName and Tel to the Contactpersons table.
- 11. Click Next. The wizard asks if you want to create a query.
- 12. Choose not to create a query and click Finish. Now three tables exist:
  - Contacts (original)

Table Analyzer Wizard	
What fields contain repeated information? You'll want to move these fields to new tables.	<b>)</b>
Each table should contain data about the same subject.	
You can drag and drop fields to make new tables, and to move fields between tables. You can also rename the tables, and set primary key fields.	)
Table1 Tomas ContactName CompanyName Tel Subject ContactDate	
	×
Cancel < <u>B</u> ack <u>N</u> ext > <u>Finish</u>	

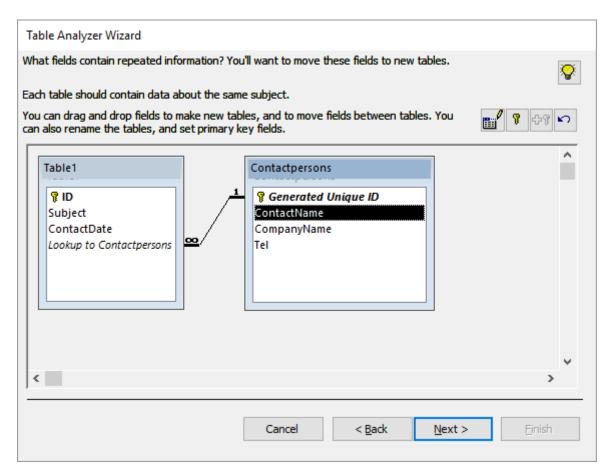
 $Figure \ 9.2.: \ Table \ Analyzer \ Wizard - manual \ field \ selection \ and \ normalization \ options.$ 

9. Tools



9. Tools

Figure 9.3.: New table Contactpersons created by dragging the ContactName field.



9. Tools

Figure 9.4.: Final result: fields moved and tables normalized with relationships.

- Contactpersons (new)
- Tabel1 (new)

#### 🛕 Warning

You may see a warning message stating that the command TileHorizontally is unavailable. If so, click OK.

- 13. Close and save the tables.
- 14. Delete the original Contacts table.
- 15. Rename Tabel1 to Contacts.

# 9.3. Compact and Repair Databases

Goal: Defragmenting and repairing database files.

Over time, as you add, edit, and delete records, your database can become fragmented, resulting in slower performance. Deleting records or objects may leave behind unused space. Although Access attempts to repair issues when you open a database, it doesn't always catch every problem. In such cases, you should manually run the **Compact and Repair** utility.

Compacting also affects AutoNumber fields. If records at the end of a table with an AutoNumber field are deleted, the next new record will continue numbering from the last used value.

To manually compact and repair a database, go to Database Tools > Compact and Repair Database (group Tools).

# 10. Macros

#### **OBJECTIVES**

- Understand what macros can do.
- Learn how to create macros.
- Learn how to connect macros to a form.

# 10.1. About Creating Macros

A macro is a small program that performs automated tasks. For example, a macro can open a form or search for a record. Macros are often linked to controls such as command buttons. When you click the button, the macro executes its list of commands.

In Access, a macro consists of a sequence of actions. Many actions require additional arguments to specify how they should be performed.

2	Macro1 ×	Actior	n Catalog	×
+	Add New Action 🗸	Search		P
			rogram Flow Comment Group If Submacro	
		4 💼 A	ctions	
		▶ 💼	Data Entry Operations	
		▷ 💼	Data Import/Export	
		▶ 💼	Database Objects	
		Þ 💼	Filter/Query/Search	
		Þ 💼	Macro Commands	
		Þ 💼	System Commands	
		Þ 💼	User Interface Comman	ds
		▶ 💼	Window Management	
		⊳ 💼 In	this Database	

Figure 10.1.: Macro editor with action catalog shown on the right.

You don't need any programming experience to create a macro. Each macro is saved as an object within the database.

## **b** Caution

#### Security note:

Access classifies macro actions into two types:

- Safe actions that are always allowed
- **Potentially unsafe actions**, such as Print, which could send unwanted jobs to the printer

By default, Access only shows safe actions. To view all actions, including potentially unsafe ones, enable Show All Actions (tab Design > group Show/Hide) in the macro window.

# 10.2. Customer Search by Code

## Goal

Add a list box to a form to search for a customer by their customer code. When a code is selected, the form displays the matching record.

## Analysis

Start with a standard form. Add a Combo Box that lists customer codes. Then create a macro with two actions:

- 1. Move focus to the customer code field
- 2. Search for the matching record

#### Task 10.1. FILE: candy365.accdb

- 1. Open the database.
- 2. Select the Customers table (do not open it).
- 3. Go to Create > Form (Forms group). A form is created and opened in Layout View.
- 4. Save the form as Search customer by code, using the button  $\square$  on the Quick Access toolbar.
- 5. Switch to Design View.
- 6. Go to Design > Combo Box (group Controls) and draw a combo box,  $\square$ , in the right section of the form header.



Figure 10.2.: Form with newly added combo box placeholder.

After drawing the framework, the Combo Box Wizard is automatically started.

7. The Combo Box Wizard will open automatically. Answer the prompts as follows:

#### 10. Macros

- The combo box should get values from a table or query.
- Choose the Customers table.
- Include only the CustomerCode field.
- Sort ascending by CustomerCode.
- Accept the default column width.
- Store the selected value for later use.
- Label: "Search by customer code:".

After completing the wizard you are back in the Design View.

- 8. If the label overlaps the combo box, reposition them by dragging the top-left handle of the controls.
- 9. Close and save the form.
- 10. Go to tab Create > Macro (Macros & Code group).

2	Macro1	×	Actior	n Catalog	×
*	Add New Action	$\sim$	Search		P
			4 📫 Pr	rogram Flow Comment Group If Submacro	
			🔺 📕 🗛	ctions	
			▷ 💼	Data Entry Operations	
			▷ 💼	Data Import/Export	
			▷ 💼	Database Objects	
			▷ 💼	Filter/Query/Search	
			▷ 💼	Macro Commands	
			▷ 💼	System Commands	
			▷ 💼	User Interface Comman	ds
			▷ 💼	Window Management	
			🕴 🗾 In	this Database	

Figure 10.3.: Macro window ready for defining actions.

10. In the Add New Action dropdown, choose GoToControl. In the Control Name field, enter CustomerCode.

-	GoToControl		
	Control Name	CustomerCode	
÷	Add New Action		$\sim$

Figure 10.4.: First macro action: GoToControl targeting CustomerCode field.

11. Add another action: FindRecord. Set Find What to =[searchcode]. Leave the other fields at their default values.

GoToControl	
Control Name	CustomerCode
FindRecord	
Find What	=[searchcode]
Match	Whole Field
Match Case	No
Search	All
Search As Formatted	No
Only Current Field	Yes
Find First	Yes
+ Add New Action	~

Figure 10.5.: Complete macro to search customer by code.

- 12. Close and save the macro as Search customer by code.
- 13. Reopen the form Search customer by code in Design View.
- 14. Select the combo box and set its Name property (tab Other) to searchcode.

Prop Selection		Shee Combo E							× 2↓
searchc	ode				$\sim$				
Format	Data	Event	Oth	her	All				
Name				sea	rchcode	ſ			
Label Name					rch on o	us	tomer	code:_	Label
Datasheet Caption									
Control	Tip Text								
Tab Ind	ex	0							

Figure 10.6.: Setting the Name property of the combo box to 'searchcode'.

- 15. While the combo box is still selected, go to the [Event] tab in the Property Sheet. For After Update, select the macro Search customer by code.
- 16. Save and close the form.
- 17. Open the form and test the combo box functionality.

# 10.3. Customer Search by Name

#### Goal

This task builds on the previous one, Task 10.1. You'll add a second combo box that lets you find customers by name (last name followed by first name).

		Shee							×
Selection	i type:	Compo B	ox						Ź↓
searchc	ode				$\sim$				
Format	Data	Event	Oth	ner	All				
On Click	c								
Before l	Jpdate								
After Update					rch cus	stome	er by c	ode	~
On Dirty	/								,
On Cha	nge								
On Not	in List								

Figure 10.7.: Assigning macro to the combo box's After Update event.

### Analysis

Start by copying the form Search customer by code. Add a second Combo Box that displays a sorted list of full names. Then create a macro to search the correct record and synchronize the two boxes.

### Task 10.2. FILE: candy365.accdb

- 1. Open the database.
- 2. Right-click the Search customer by code form, choose Copy, then right-click again and choose Paste.
- 3. Name the new form Search customer by name, and open it in Design View.
- 4. Drag the top edge of the [Detail] section downward to make space in the header.

Form Hea	der												
	Cust	ome	ers		Search	i on ci	stome	er code	:: U	nboun	d	<	
					1		±						
Ø Detail		-	-				Ŧ						
Custon	nerCod	e Cu	stome	rCode									

Figure 10.8.: Expanding the form header to make room for an additional combo box.

- 5. Add a second Combo Box below the first one. In the Combo Box Wizard:
  - Get values from a table or query.
  - Choose the Customers table.
  - Add fields in this order: [LastName], [FirstName], [CustomerCode]
  - Sort by last name, then first name (ascending).
  - Accept default width and hide the key column (CustomerCode).
  - Store the value for later use.
  - Label: "Search by customer name:".

After completing the wizard you are back in the Design View.

## $10.\ Macros$

<b>∉</b> For	m Head	er											
		Cust	ome	arc									
		Lusi	onne	15		Search	i on cu	istome	er code	8: U	nboun	d	$\sim$
						Search	i on cu	stome	er nam	e: U	nboun	d	$\sim$
🗲 Det	ail												
C	istom	erCod		stome	rCode								

Figure 10.9.: Form layout with two combo boxes: one for code, one for name.

- 6. Align the labels and combo boxes neatly.
- 7. Select the second combo box. In the Property Sheet (tab Other), set Name to searchname. In the After Update event (tab Event), enter Search customer by name.

i Note	
The macro does not exist yet, you'll create it in the next step.	

- $8. \ Save \ and \ close \ the \ form.$
- 9. Go to tab Create > Macro (Macros & Code group).

Macro1     Add New Action	× Action Catalog	_
	<ul> <li>Search</li> <li>Program Flow</li> <li>Comment</li> <li>Group</li> <li>If</li> <li>Submacro</li> <li>Actions</li> <li>Data Entry Operations</li> <li>Data Import/Export</li> <li>Database Objects</li> <li>Filter/Query/Search</li> <li>Macro Commands</li> <li>System Commands</li> <li>System Commands</li> <li>User Interface Commands</li> <li>Window Management</li> </ul>	<u>'</u>

Figure 10.10.: Creating a new macro to handle name-based search.

- 10. Add the action GoToControl, and set Control Name to CustomerCode.
- 11. Add the action FindRecord, and set Find What to =[searchname].

#### $10.\ Macros$

-	GoToControl		
	Control Name	CustomerCode	
t	Add New Action		$\sim$

Figure 10.11.: Macro step: GoToControl set to the CustomerCode field.

i Note
To keep both combo boxes synchronized, you'll need to use SetValue to update the first combo box when the second is used.
12. Enable Show All Actions in the macro design window to access SetValue.

13. Add the SetValue action with the following parameters:

- Item: [Form]![Search customer by name]![searchcode]
- Expression: [Form]![Search customer by name]![CustomerCode]

GoToControl	
Control Name	CustomerCode
FindRecord	
Find What	=[searchname]
Match	Whole Field
Match Case	No
Search	All
Search As Formatted	No
Only Current Field	Yes
Find First	Yes
🛕 SetValue	
ltem	= [Forms]![Search customer by name]![searchcode]
Expression	= [Forms]![Search customer by name]![CustomerCode]

Figure 10.12.: Macro with SetValue to synchronize both combo boxes.

- 14. Save the macro as Search customer by name and close the macro window.
- 15. Open the form Search customer by name and test both combo boxes.

### i Note

If you want the first combo box to update when the second is used, you'll need to add a similar SetValue action to the other macro as well. However, doing so may affect the original form Search customer by code.

# 10.4. Combo Box: Search by Box Code

### Goal

This task is similar to searching by customer code. You'll create a form that displays box details and includes a combo box to search by BoxCode.

### Analysis

Start with a standard form. Add a Combo Box, and create a macro that:

- 1. Moves focus to the BoxCode field.
- 2. Searches for the selected record.

#### Task 10.3. FILE: candy365.accdb

- 1. Create a new form with a combo box, and name it Search box by code.
- 2. Create a macro named Search box by code that performs the lookup.
- 3. Open the form and test that the combo box finds and displays the correct record.