MONASH University

FIT1013 Digital Futures: IT for Business
Week 10 : Creating a Database, Defining Table Relationships

CLAYTON, FACULTY OF INFORMATION TECHNOLOGY
MONASH UNIVERSITY

On completion of your study this week, you should aim to:

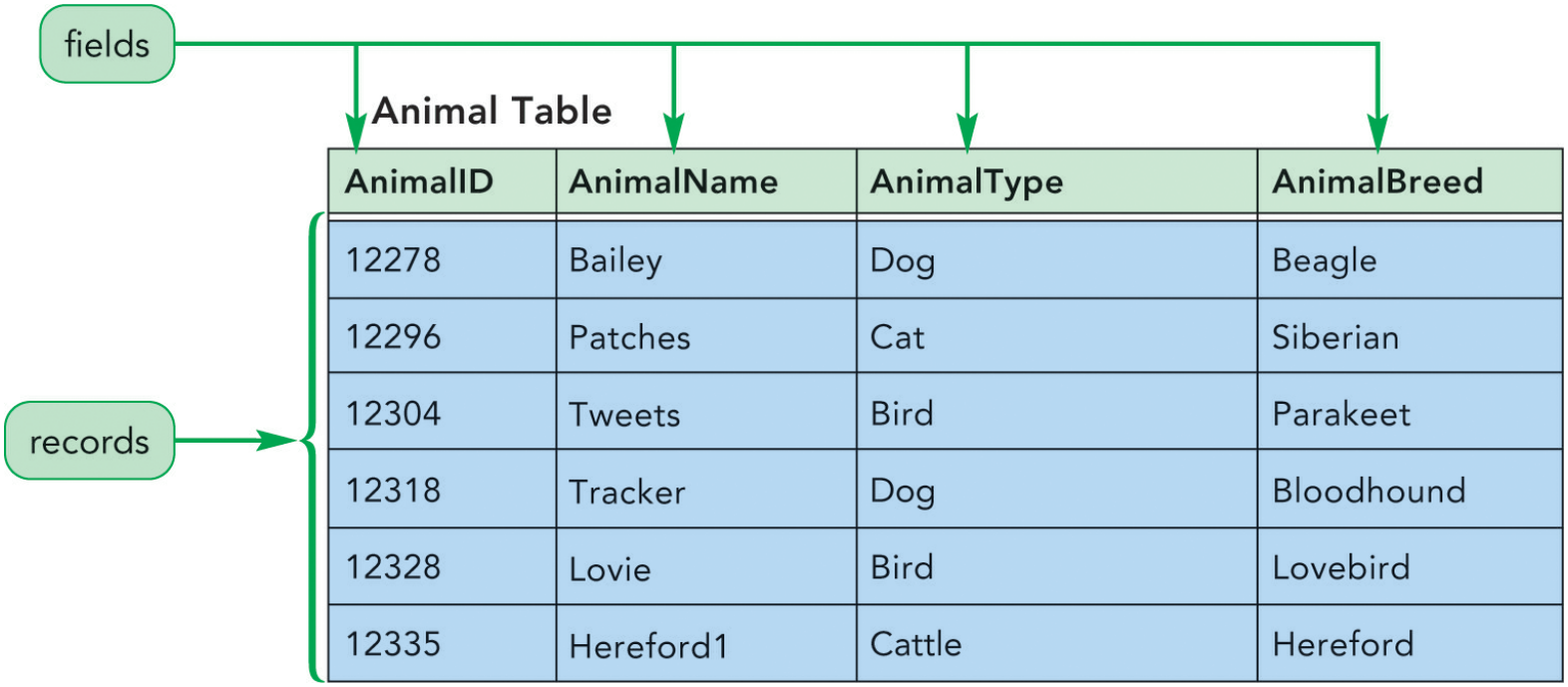
- Create a database
- Create a simple query, form and report
- Learn the guidelines for designing databases
- Create a table in Design view
- Define a relationship between two tables

GROUP OF EIGHT AUSTRALIA

Introduction to Database Concepts

- Organizing Data
  - A **field** is a single characteristic or attribute of a person, place, object, event, or idea
    - Patient ID, first name, last name, address, phone number, visit date, reason for visit, and invoice amount
  - Related fields are grouped together into a **table**
    - A collection of fields that describes a person, place, object, event, or idea
    - The specific content of a field is called the **field value**
      - this set of field values is called a **record**

Introduction to Database Concepts (Cont.)

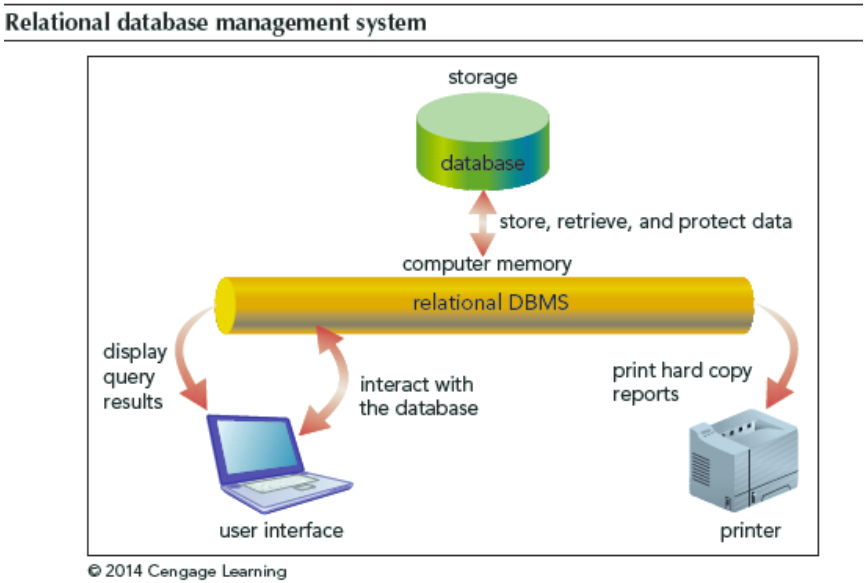


Databases and Relationships

- A **relational database** is a collection of related tables
- Records in the separate tables are connected through a **common field**
- A **primary key** is a field, or a collection of fields, that uniquely identify each record in a table
- Including the primary key from one table as a field in a second table to form a relationship between the two tables, it is called a **foreign key** in the second table

Relational Database Management Systems

- A **database management system (DBMS)** is a software program that lets you create databases and then manipulate the data they contain
- In a **relational database management system**, data is organized as a collection of tables. A relational DBMS controls the storage of databases and facilitates the creation manipulation, and reporting of data



Relational Database Management Systems

- A relational DBMS provides the following functions:
  - Allows you to **create database structures** containing fields, tables, and table relationships
  - Lets you easily add new records, change field values in existing records, and delete records
  - Contains a built-in **query language**, which lets you obtain immediate answers to the questions (or queries) you ask about your data
  - Contains a **built-in report generator**, which lets you produce professional-looking, formatted reports from your data
  - Protects databases through **security**, **control**, and **recovery** facilities

Relational database

- Collection of **tables** (tables are also called **relations**)
- Each table in the database contains information related to **one** particular subject (e.g. customer, product, student)
- The information in the table is contained in columns called **fields**
- Each field contains a **single** piece of information
- Tables are **linked** together using a **common field**

Customer table

Customer ID	First Name	Last Name	Phone
11005	Owen	Hawes	616-392-0622
11008	Melissa	Caputo	269-985-1122
11014	Amol	Mehta	616-396-1972
11015	John	Weiss	616-637-7783
11027	Karen	O'Brien	517-483-9244
11053	Hwan	Tang	616-396-8401

A table of Customers

Fields: Customer ID, First Name, Last Name, Phone

records

Customer table

Customer ID	First Name	Last Name	Phone
11005	Owen	Hawes	616-392-0622
11008	Melissa	Caputo	269-985-1122
11014	Amol	Mehta	616-396-1972
11015	John	Weiss	616-637-7783
11027	Karen	O'Brien	517-483-9244
11053	Hwan	Tang	616-396-8401

Customers table lists Customers of Belmont Landscapes – a landscaping company

Contract table – another table in the database - Provides information about the contracts between customer and company

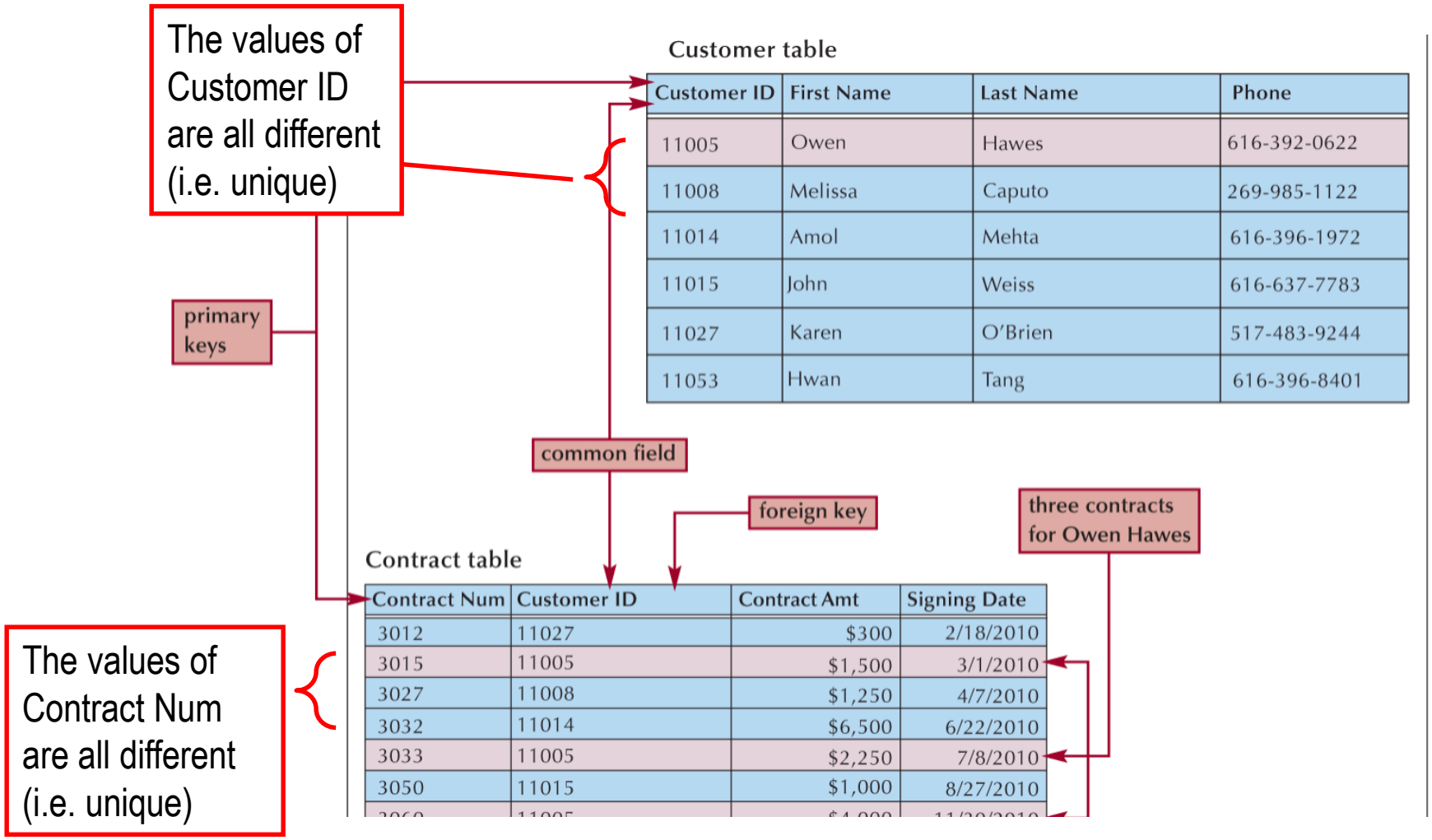
Fields: Contract Num, Customer ID, Contract Amt, Signing Date

records

Contract table

Contract Num	Customer ID	Contract Amt	Signing Date
3012	11027	$300	2/18/2010
3015	11005	$1,500	3/1/2010
3027	11008	$1,250	4/7/2010
3032	11014	$6,500	6/22/2010
3033	11005	$2,250	7/8/2010
3050	11015	$1,000	8/27/2010
3060	11005	$4,000	11/30/2010
3062	11053	$20,000	1/11/2011

Relationship between tables for Customers and Contracts



Customer table

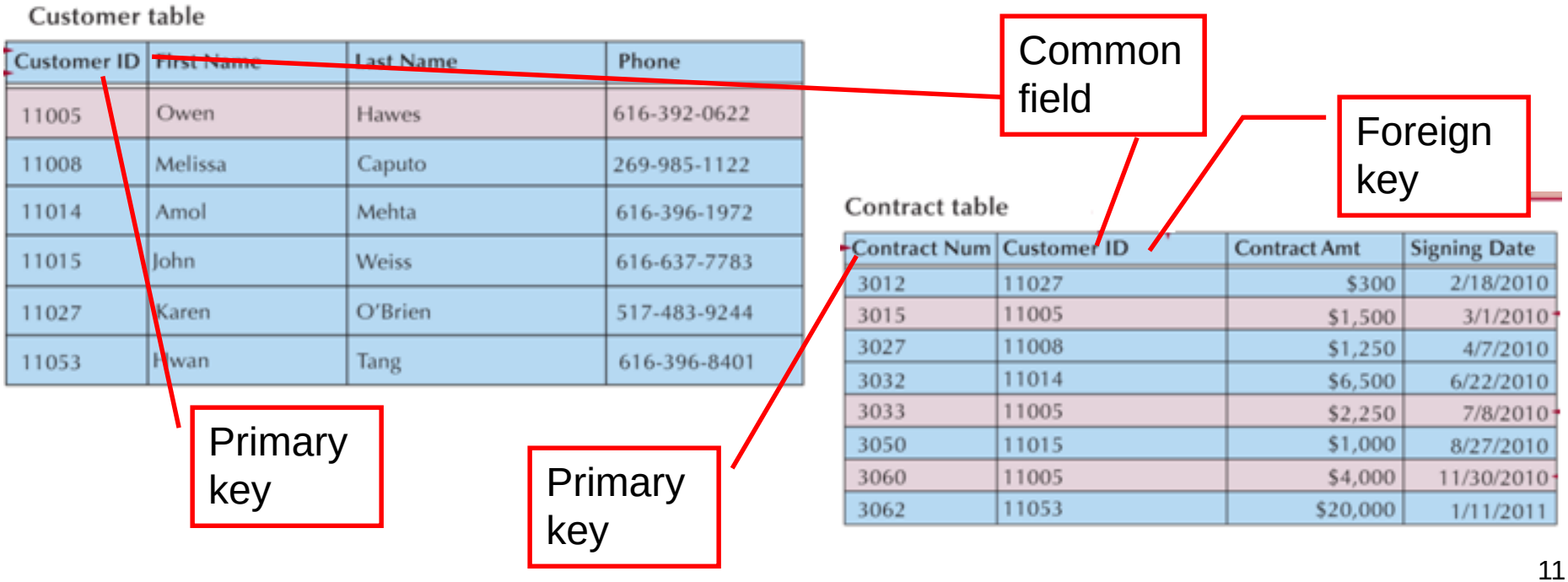
Customer ID	First Name	Last Name	Phone
11005	Owen	Hawes	616-392-0622
11008	Melissa	Caputo	269-985-1122
11014	Amol	Mehta	616-396-1972
11015	John	Weiss	616-637-7783
11027	Karen	O'Brien	517-483-9244
11053	Hwan	Tang	616-396-8401

Contract table

Contract Num	Customer ID	Contract Amt	Signing Date
3012	11027	$300	2/18/2010
3015	11005	$1,500	3/1/2010
3027	11008	$1,250	4/7/2010
3032	11014	$6,500	6/22/2010
3033	11005	$2,250	7/8/2010
3050	11015	$1,000	8/27/2010

Definitions:

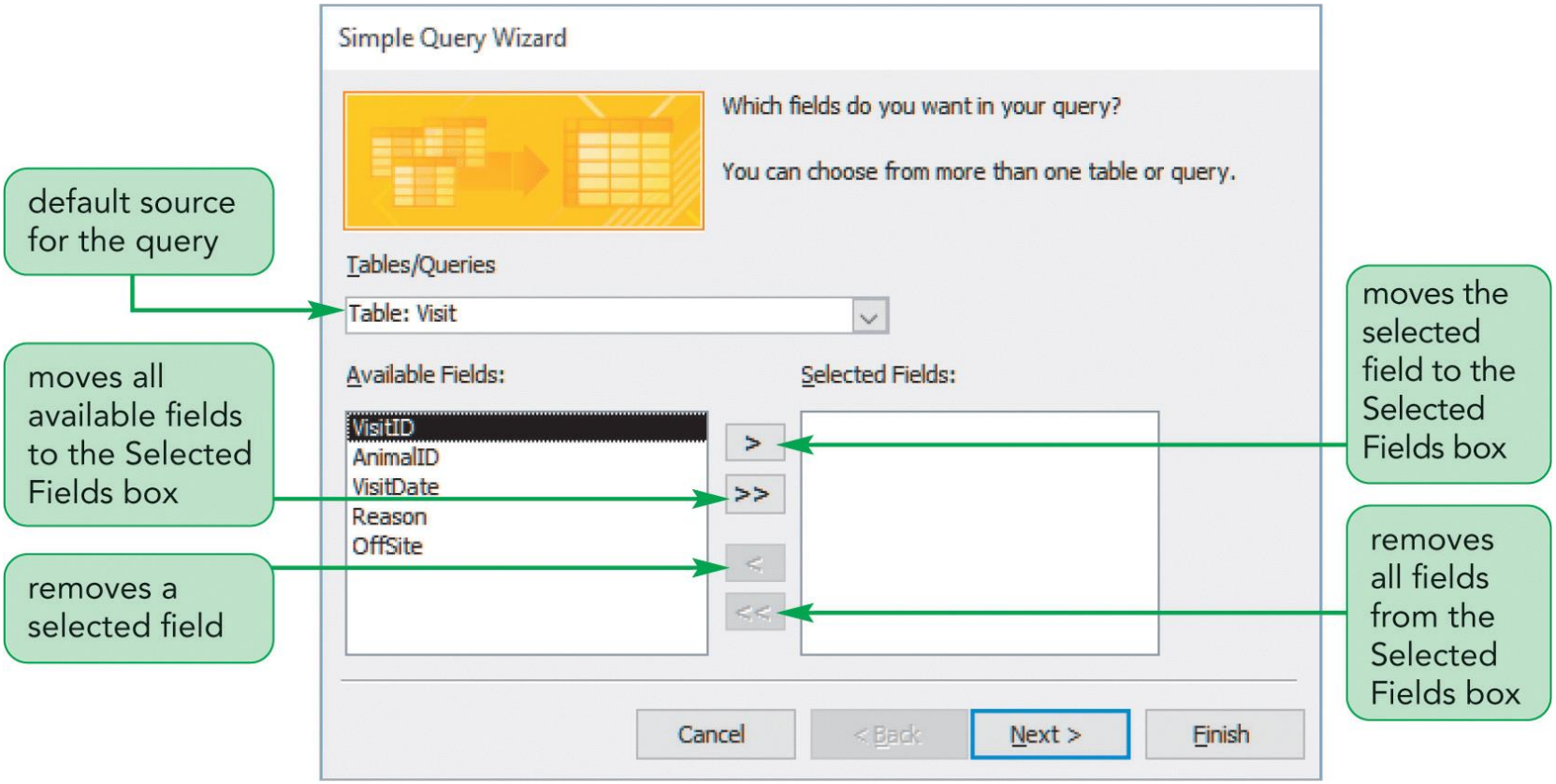
- **Primary key:** A field (or collection of fields) which uniquely identifies each record in a table.
- **Common field:** a field that appears in two tables - defines a relationship between the tables.
- **Foreign key:** a field in a table which is the same as the primary key in a related table



Summary of definitions so far..

- **field** a single characteristic or attribute of a person, place, object, event, or idea
- **field value** the specific value, or content, of a field
- **table** a collection of related fields that describe a person, place, object, event, or idea (sometimes called a relation)
- **record** a set of field values
- **primary key** a field, or a collection of fields, whose values uniquely identify each record in a table
- **foreign key** a field in a table which is the same as the primary key in a related table
- **common field** a field that appears in separate tables in a database. Defines a relationship between two tables.

Creating a Simple Query (Cont.)



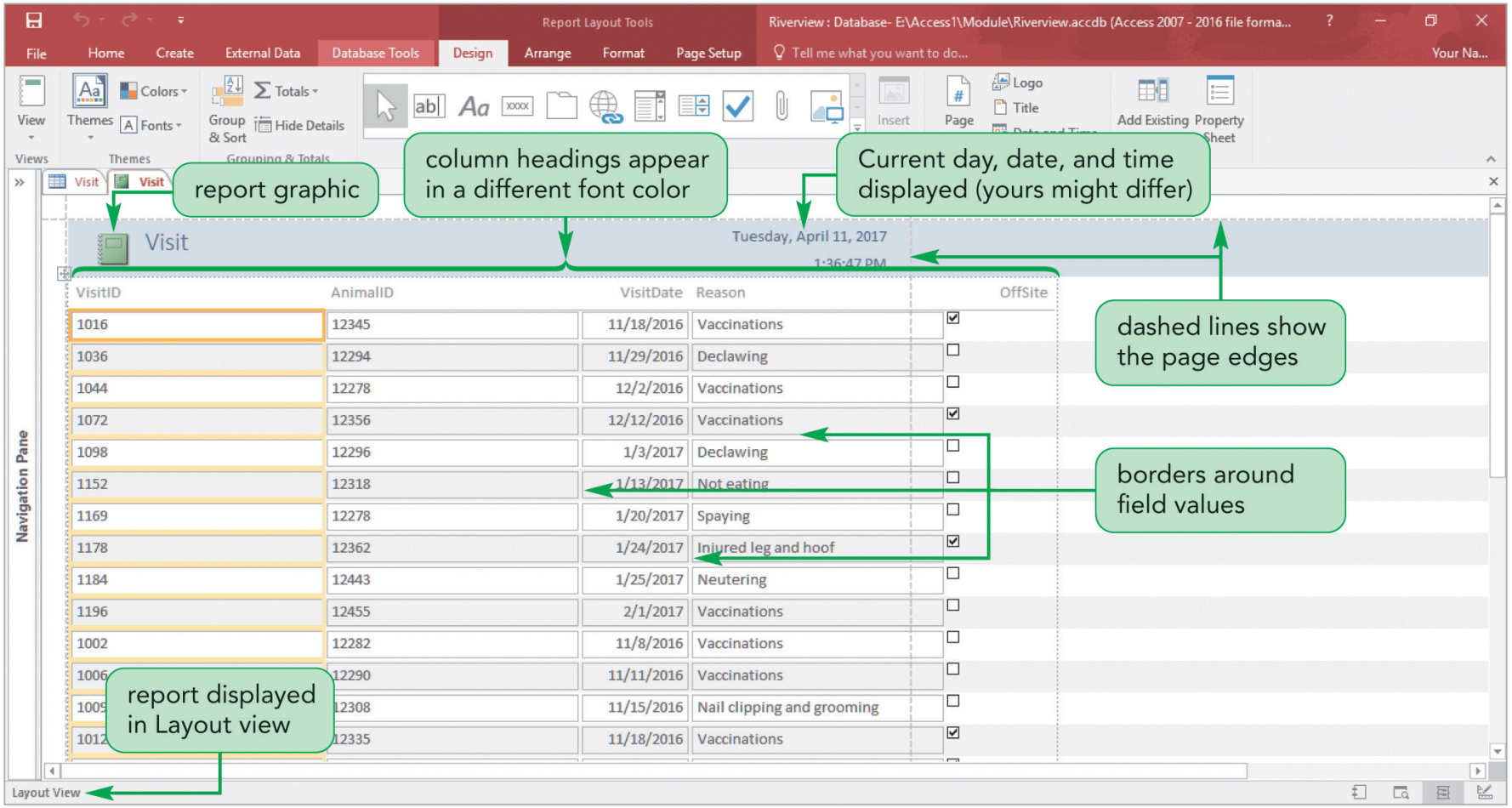
Creating a Simple Form

- Forms display one record at a time
  - Provide another view of the data that is stored in the table
  - Allowing you to focus on the values for one record
- Access displays the field values for the first record in the table
- Each field appears on a separate line
- As indicated in the status bar, the form is displayed in Layout view
  - In **Layout view**, you can make design changes to the form while it is displaying data, so that you can see the effects of the changes you make immediately

Creating a Simple Report

- A report is a formatted printout (or screen display) of the contents of one or more tables or queries
- Reports show each field in a column, with the field values for each record in a row, similar to a table or query datasheet
- Reports offers a more visually appealing format for the data, with the column headings in a different color, borders around each field value, a graphic of a report at the top left, and the current day, date, and time at the top right
- Printing a Report
- Print reports to distribute to others who need to view the report's contents

Creating a Simple Report (Cont.)



VisitID	AnimalID	VisitDate	Reason	OffSite
1016	12345	11/18/2016	Vaccinations	☑
1036	12294	11/29/2016	Declawing	☐
1044	12278	12/2/2016	Vaccinations	☐
1072	12356	12/12/2016	Vaccinations	☑
1098	12296	1/3/2017	Declawing	☐
1152	12318	1/13/2017	Not eating	☐
1169	12278	1/20/2017	Spaying	☐
1178	12362	1/24/2017	Injured leg and hoof	☑
1184	12443	1/25/2017	Neutering	☐
1196	12455	2/1/2017	Vaccinations	☐
1002	12282	11/8/2016	Vaccinations	☐
1006	12290	11/11/2016	Vaccinations	☐
1009	12308	11/15/2016	Nail clipping and grooming	☐
1012	12335	11/18/2016	Vaccinations	☑

Creating a Simple Report (Cont.)

Figure 1-25 Report after resizing the VisitID column

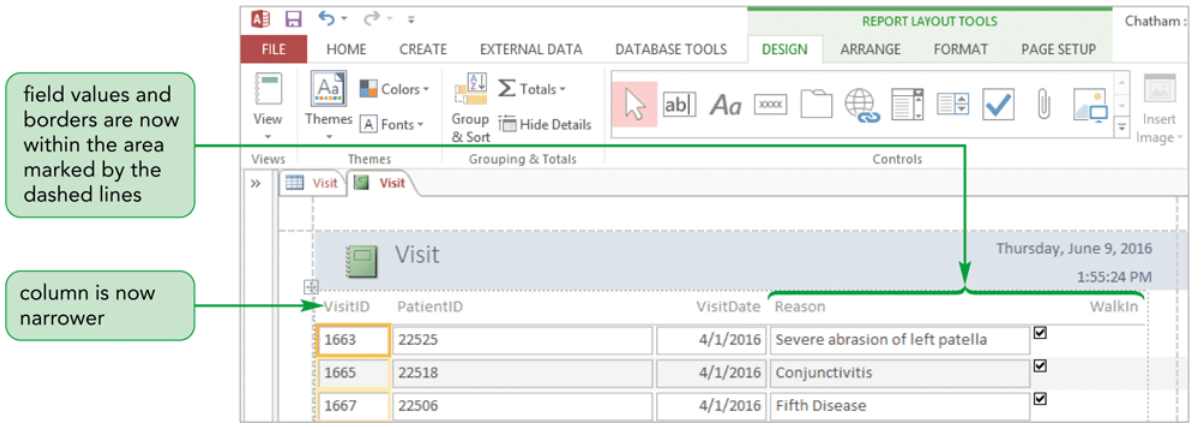
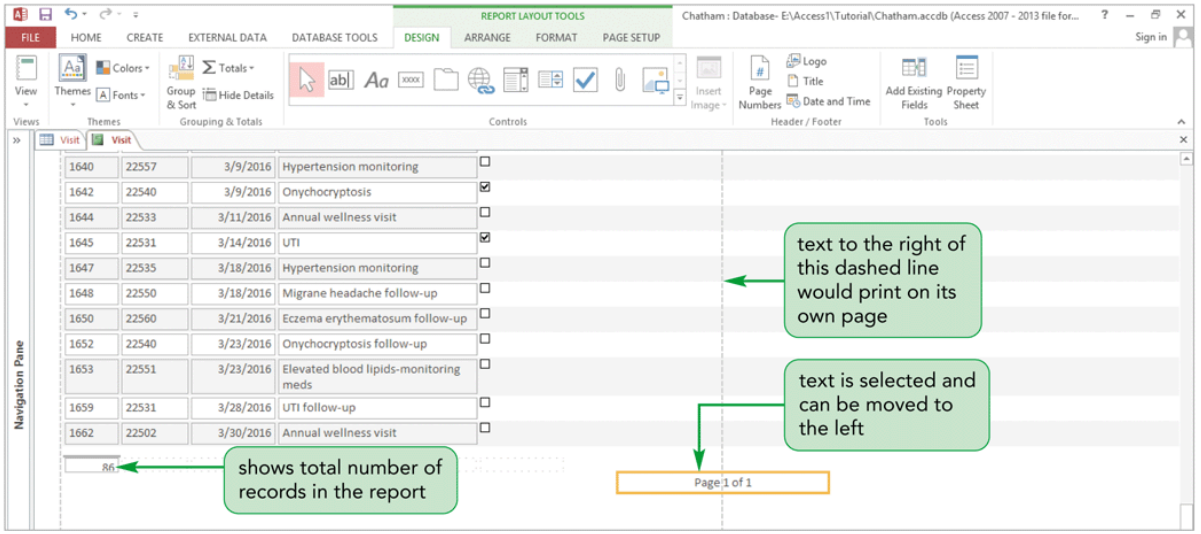
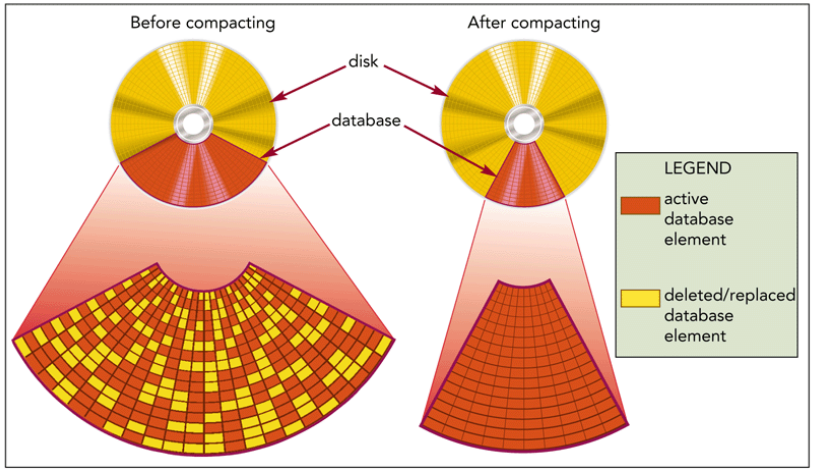


Figure 1-26 Report page number selected



Managing a Database

- Activities involved in database management include compacting and repairing a database and backing up and restoring a Database
- **Compacting and Repairing a Database**
  - Rearranges the data and objects in a database to decrease its file size, thereby making more storage space available and enhancing the performance of the database



Managing a Database (Cont.)

Backing Up and Restoring a Database

- The process of making a copy of the database file to protect your database against loss or damage
- The Back Up Database command enables you to back up your database file from within the Access program, while you are working
- Steps:
  - Click the FILE tab to display the Info screen in Backstage view
  - Click Save As in the navigation bar
  - Click Back Up Database in the Advanced section of the Save Database As pane
  - Click the Save As button

Excel or Access?

Ask the following questions

1. Do you need to store data in separate tables that are related to each other?
2. Do you have a very large amount of data to store?
3. Will more than one person need to access the data at the same time?
    - If you answer “yes” to any of these questions, an Access database is most likely the appropriate application to use

Guidelines for Designing Databases

- Case – Riverview Veterinary Care Center
  - Database currently contains one table, named the Visit table
  - User wants to track information about the clinic's animals, their owners, and the invoices sent to them for services provided
    - This information includes such items as each owner's name and address, animal information, and the amount and billing date for each invoice
  - Create three new tables—named Billing, Owner, and Animal—to contain the additional data Kimberly wants to track
  - After adding records to the tables, you will define the necessary relationships between the tables, and learn how to modify the fields

Guidelines for Database Design (Cont.)

- Identify all the fields needed to produce the required information
- Organize each piece of data into its smallest useful part
- Group related fields into tables
- Determine each table's primary key
- Include a common field in related tables
- Avoid data redundancy

VisitID	AnimalBreed	Zip
VisitDate	OwnerID	Email
Reason	FirstName	InvoiceNum
OffSite	LastName	InvoiceDate
AnimalID	Phone	InvoiceAmt
AnimalName	Address	InvoiceItem
AnimalBirthDate	City	InvoicePaid
AnimalType	State	

Creating a Table in Design View

- Creating a table in Design view involves entering the field names and defining the properties for the fields, specifying a primary key for the table, and then saving the table structure
- **Defining Fields**
  - When you first create a table in Design view, the insertion point is located in the first row's Field Name box, ready for you to begin defining the first field in the table
  - Enter values for the Field Name, Data Type, and Description field properties, and then select values for all other field properties in the Field Properties pane
  - These other properties will appear when you move to the first row's Data Type box

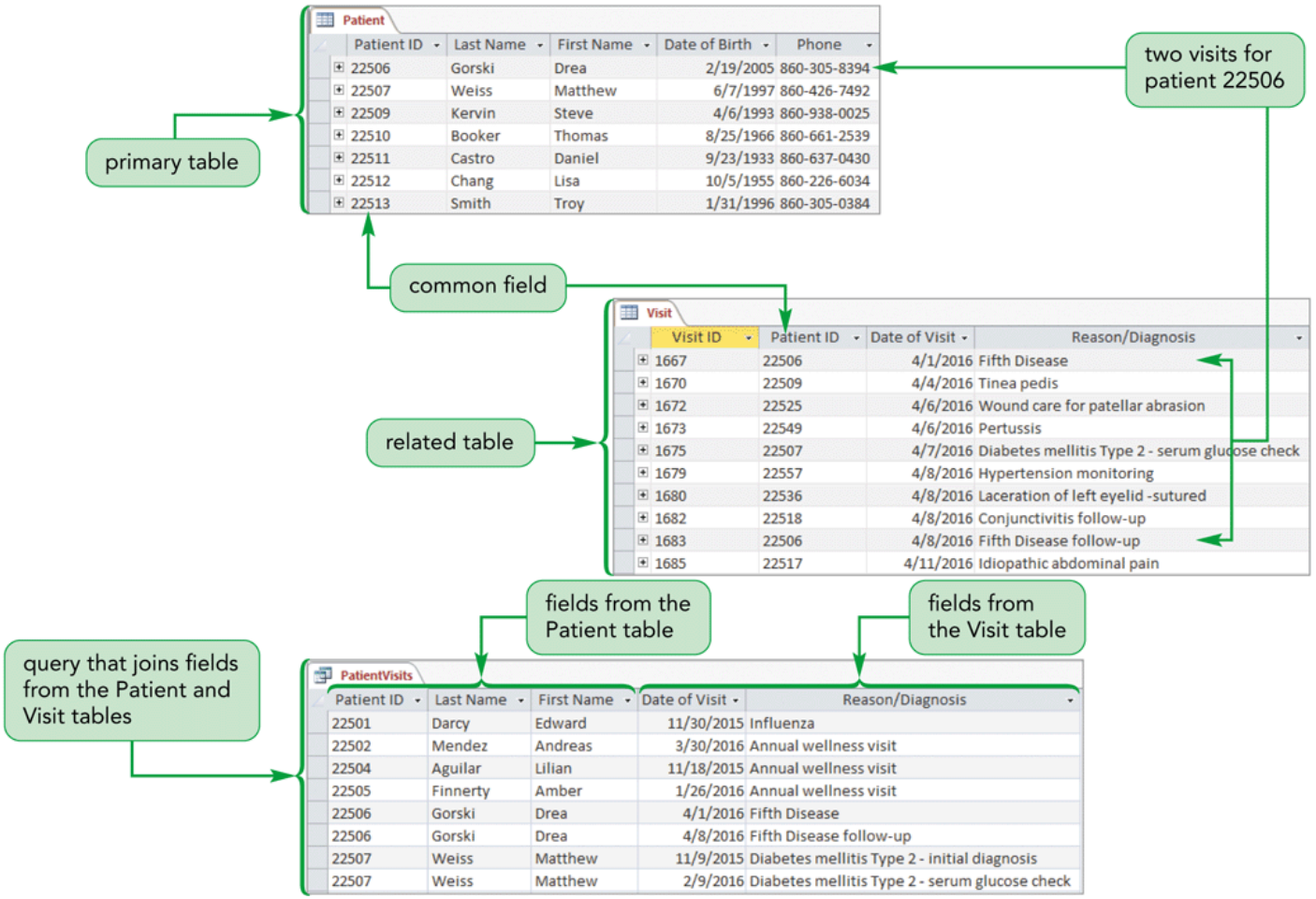
Field Name	Data Type	Field Size	Description	Other
InvoiceNum	Short Text	5	Primary key	Caption = Invoice Num
VisitID	Short Text	4	Foreign key	Caption = Visit ID
InvoiceAmt	Currency			Format = Currency Decimal Places = 2 Caption = Invoice Amt
InvoiceDate	Date/Time			Format = mm/dd/yyyy Caption = Invoice Date
InvoicePaid	Yes/No			Caption = Invoice Paid

Creating a Table in Design View (Cont.)

- **Specifying the Primary Key**
  - A primary key uniquely identifies each record in a table
  - Access does not allow duplicate values in the primary key field
  - When a primary key has been specified, Access forces you to enter a value for the primary key field in every record in the table (entity integrity)
  - You can enter records in any order, but Access displays them by default in order of the primary key's field values
  - Access responds faster to your requests for specific records based on the primary key
- **Saving the Table Structure**
  - The last step in creating a table is to name the tableand save the table's structure

Defining Table Relationships

Figure 2-40 One-to-many relationship and sample query



Defining Table Relationships (Cont.)

- **One-to-Many Relationships**
  - A one-to-many relationship exists between two tables when one record in the first table matches zero, one, or many records in the second table, and when one record in the second table matches at most one record in the first table
- **Referential Integrity**
  - A set of rules that Access enforces to maintain consistency between related tables when you update data in a database

Defining Table Relationships (Cont.)

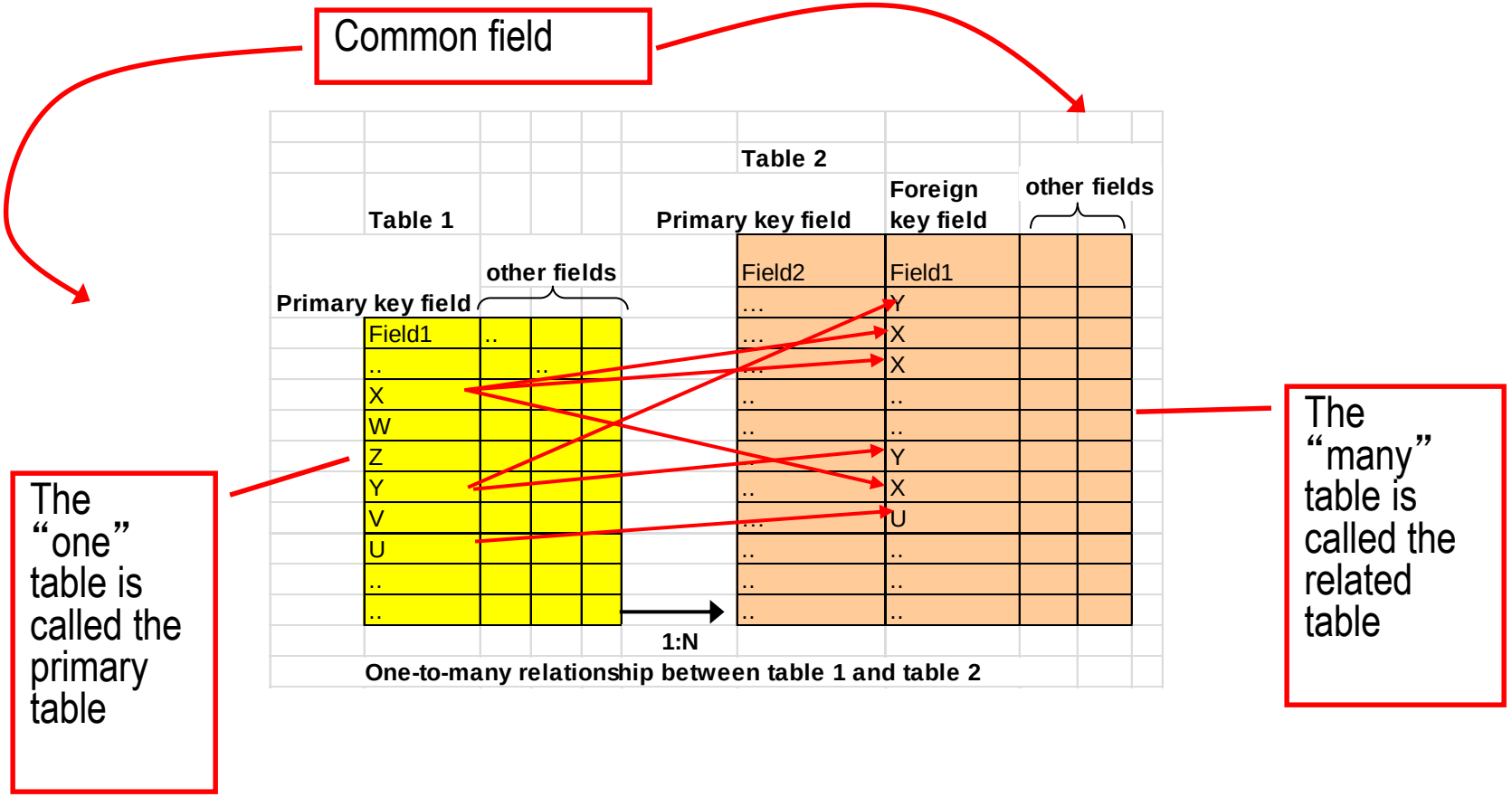
- **Referential Integrity**
  - When you add a record to a related table, a matching record must already exist in the primary table, preventing the possibility of orphaned records
  - If you attempt to change the value of the primary key in the primary table, Access prevents this change if matching records exist in a related table
  - With the **Cascade Update Related Fields** option, Access permits the change in value to the primary key and changes the appropriate foreign key values in the related table
  - If you attempt to delete a record in the primary table, Access prevents the deletion if matching records exist in a related table. However, if you choose the **Cascade Delete Related Records** option, Access deletes the record in the primary table and also deletes all records in related tables that have matching foreign key values

Table relationships

- One-to-many
- One to one
- Many-to-many

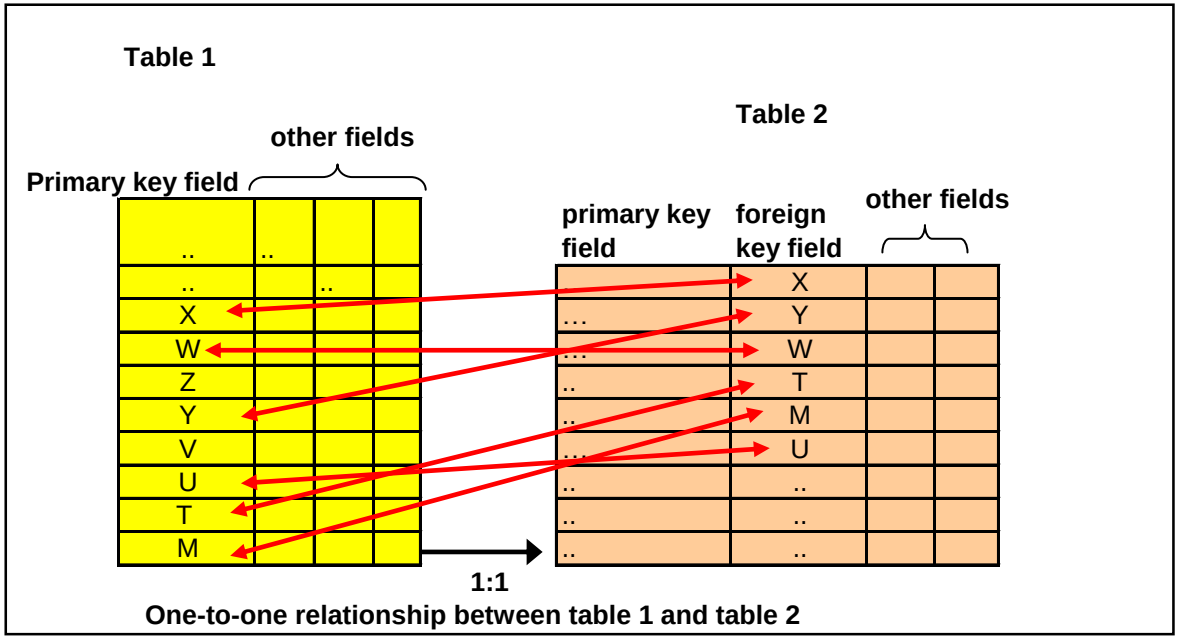
- **one-to-many relationship (1:N):**
  - a relationship that exists between two tables when one record in the first table matches **zero, one, or many** records in the second table, and when one record in the second table matches exactly one record in the first table.

One-to-many: the most common type of relationship



One-to-one relationships (1:1)

Two tables with a one-to-one relationship. Each record in the first table matches at most 1 record in the second table and conversely each record in the second table matches at most 1 record in the first table.

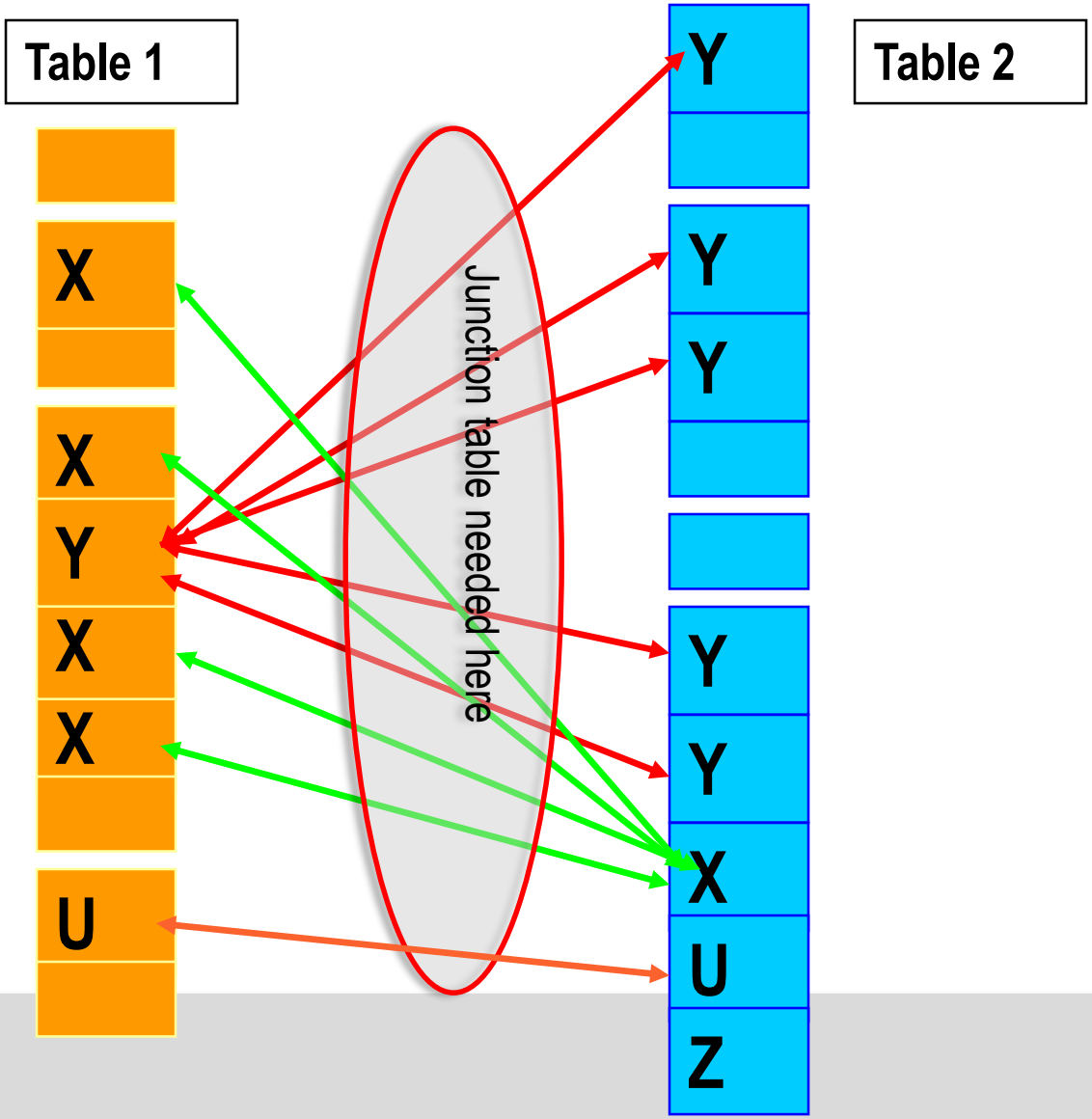


Many to many relationships (M:N)

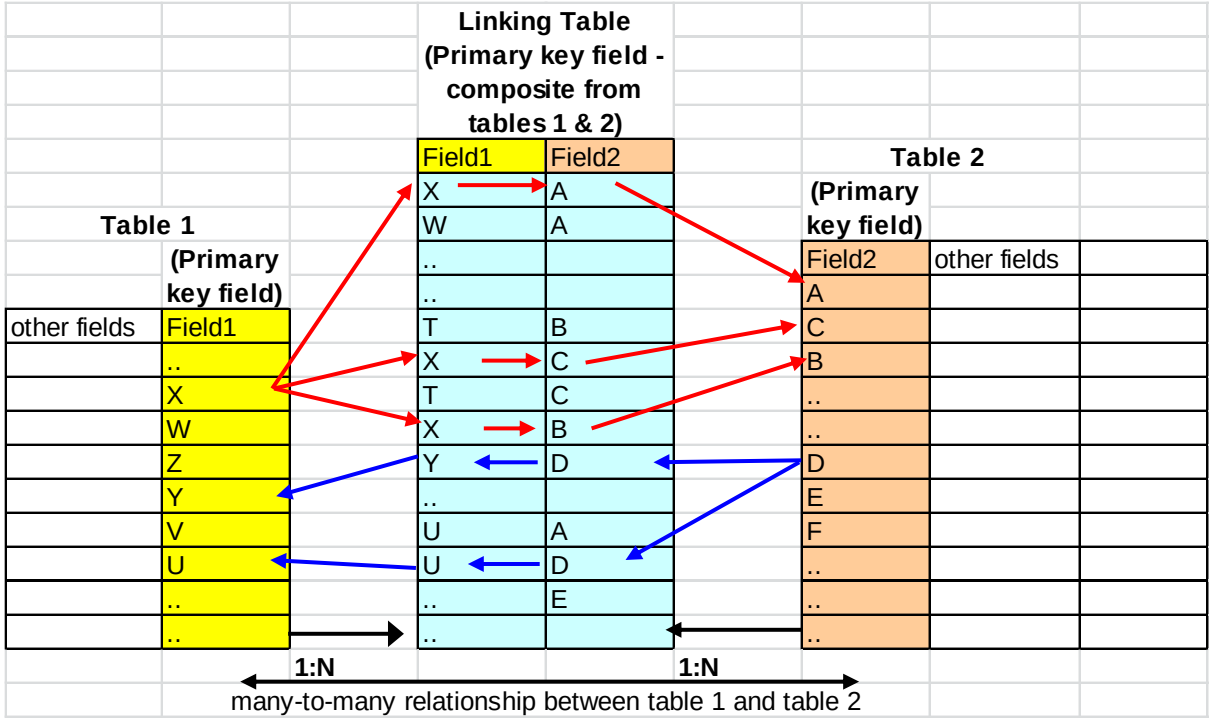
- In a many-to-many relationship, a record in the first table can have 0, 1 or many matching records in the second table, and a record in the second table can have 0,1 or many matching records in the first table

- In **Access**, this type of relationship is only possible by creating a third table (called a ***junction table or linking table***) whose primary key contains two fields - the primary keys from both tables. (I.e., the linking table has a **composite** primary key).

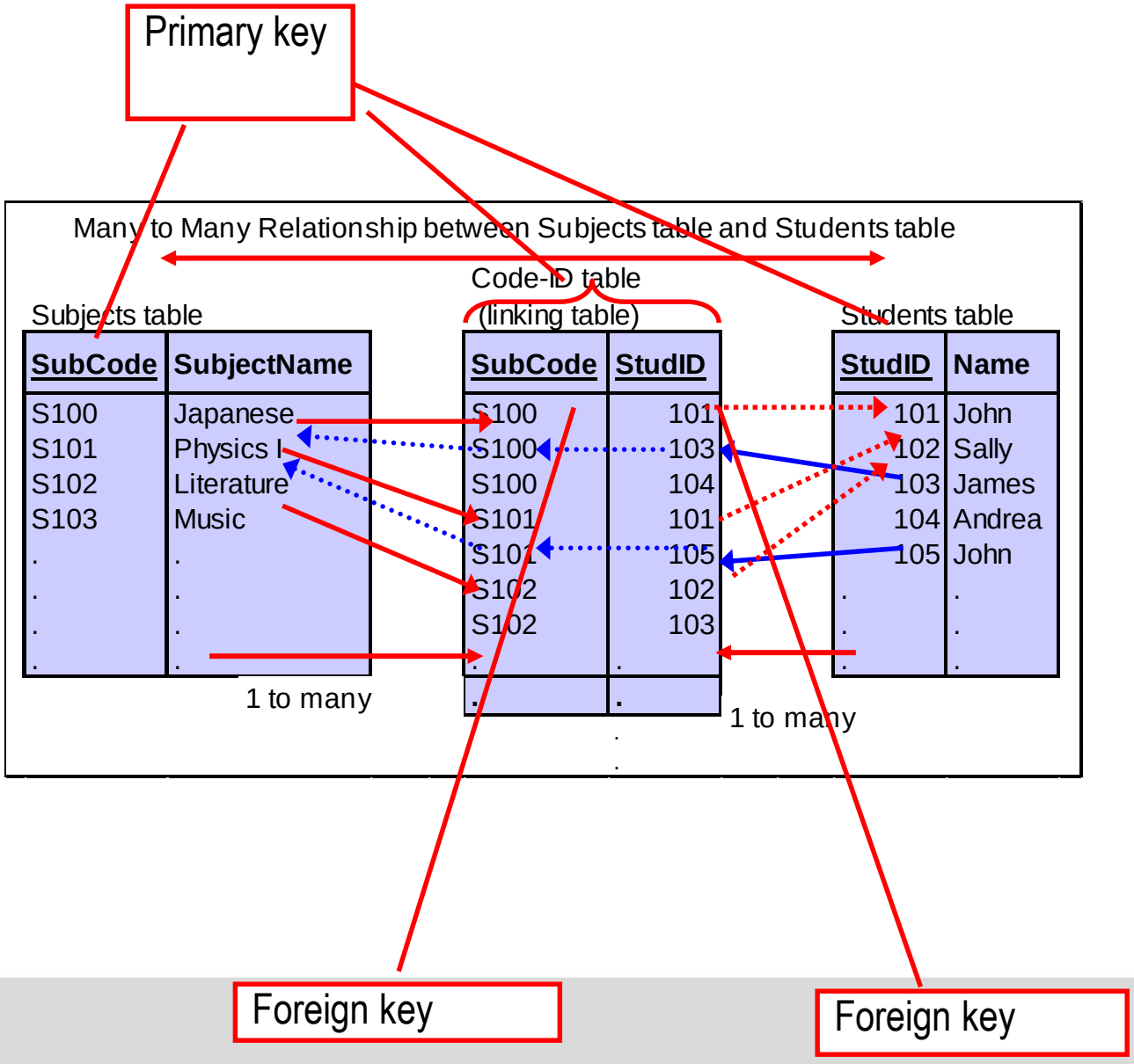
Many to many relationship – between table 1 and table 2 – requires a **linking table** also called a **junction table**



Many to many relationship in Access



Example - Many to Many Relationships



Parent-child relationships

- A **one to many relationship** is also known as a **parent-child relationship**.
- The **Parent** record comes from the primary table (“one” table).
- The **Child** record comes from the related table (“many” table).
- An **Orphaned** record occurs when a record from the related table has no corresponding parent in the primary table. Orphaned records are one type of inconsistency that can occur in databases.

Database inconsistencies

Access has features which you can select to try and reduce the incidence of inconsistencies caused by orphaned records in databases

These features are

- **Enforce referential integrity**
- **Cascade update**
- **Cascade delete**

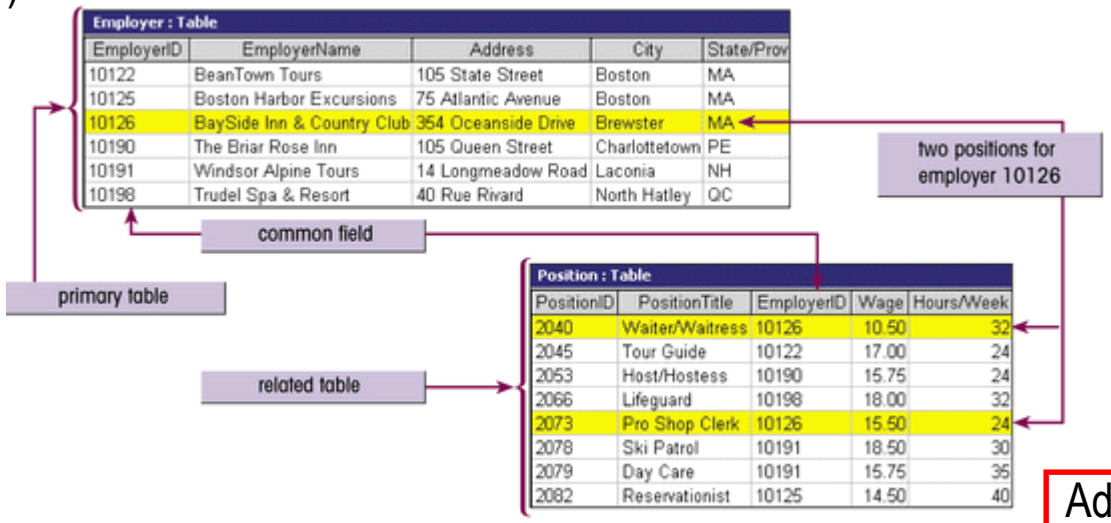
First: examples of inconsistency problems that can occur in related tables.

Inconsistencies that can occur in related tables

The Employer table and the Position table have a 1 to many relationship with EmployerID as the common field

Problem 1:if we add a new record to the Position table with EmployerID = 10132

Results in: orphaned record (because a record with EmployerID = 10132 does not appear in the Employer table)



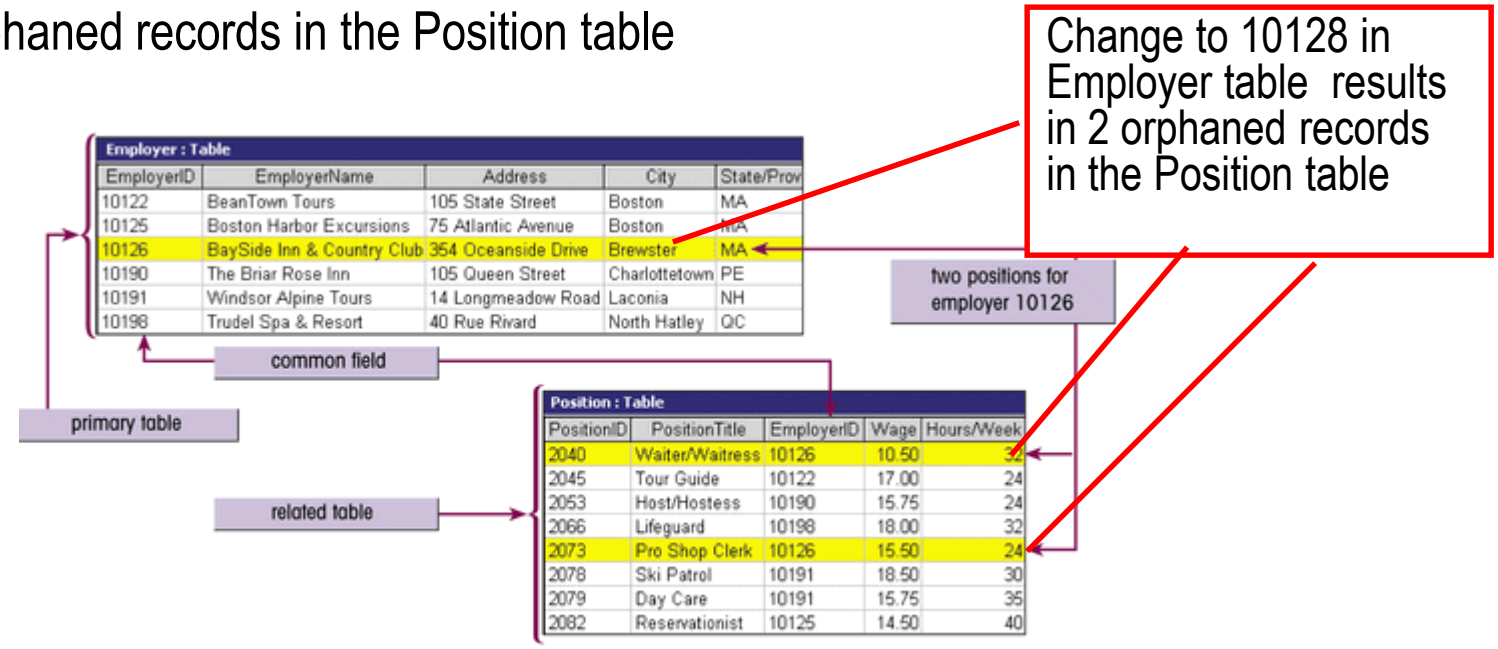
Add a record with EmployerID = 10132 to Position table.

This EmployerID does not appear in the Employer table

Inconsistencies that can occur in related tables

Problem 2: if we change the EmployerId for BaySide Inn & Country Club from 10126 to 10128 in table Employer

Results in: 2 orphaned records in the Position table



Employer : Table

EmployerID	EmployerName	Address	City	State/Prov
10122	BeanTown Tours	105 State Street	Boston	MA
10125	Boston Harbor Excursions	75 Atlantic Avenue	Boston	MA
10126	BaySide Inn & Country Club	354 Oceanside Drive	Brewster	MA
10190	The Briar Rose Inn	105 Queen Street	Charlottetown	PE
10191	Windsor Alpine Tours	14 Longmeadow Road	Laconia	NH
10198	Trudel Spa & Resort	40 Rue Rivard	North Hatley	QC

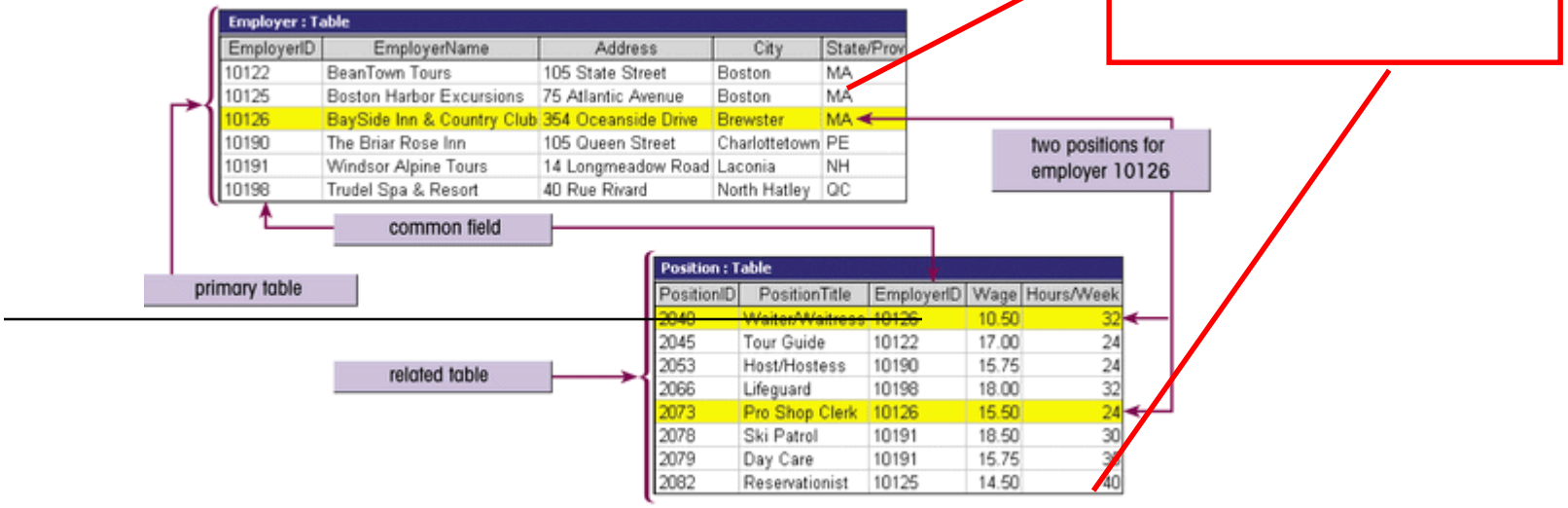
Position : Table

PositionID	PositionTitle	EmployerID	Wage	Hours/Week
2040	Waiter/Waitress	10126	10.50	32
2045	Tour Guide	10122	17.00	24
2053	Host/Hostess	10190	15.75	24
2066	Lifeguard	10198	18.00	32
2073	Pro Shop Clerk	10126	15.50	24
2078	Ski Patrol	10191	18.50	30
2079	Day Care	10191	15.75	35
2082	Reservationist	10125	14.50	40

Inconsistencies that can occur in related tables

Problem 3: if in table Employer we delete the record for Boston Harbour Excursions (EmployerID 10125) as they are no longer a client

Results in: 1 orphaned records in the Position table



Employer : Table

EmployerID	EmployerName	Address	City	State/Prov
10122	BeanTown Tours	105 State Street	Boston	MA
10125	Boston Harbor Excursions	75 Atlantic Avenue	Boston	MA
10126	BaySide Inn & Country Club	354 Oceanside Drive	Brewster	MA
10190	The Briar Rose Inn	105 Queen Street	Charlottetown	PE
10191	Windsor Alpine Tours	14 Longmeadow Road	Laconia	NH
10198	Trudel Spa & Resort	40 Rue Rivard	North Hatley	QC

Position : Table

PositionID	PositionTitle	EmployerID	Wage	Hours/Week
2040	Waiter/Waitress	10126	10.50	32
2045	Tour Guide	10122	17.00	24
2053	Host/Hostess	10190	15.75	24
2066	Lifeguard	10198	18.00	32
2073	Pro Shop Clerk	10126	15.50	24
2078	Ski Patrol	10191	18.50	30
2079	Day Care	10191	15.75	36
2082	Reservationist	10125	14.50	40

How can we avoid problems of inconsistency?

Answer:

Referential integrity -

the set of rules that Access enforces to maintain consistency between related tables when you update the data in a database.

There are three referential integrity rules that Access enforces

Referential Integrity Rules

Rule 1

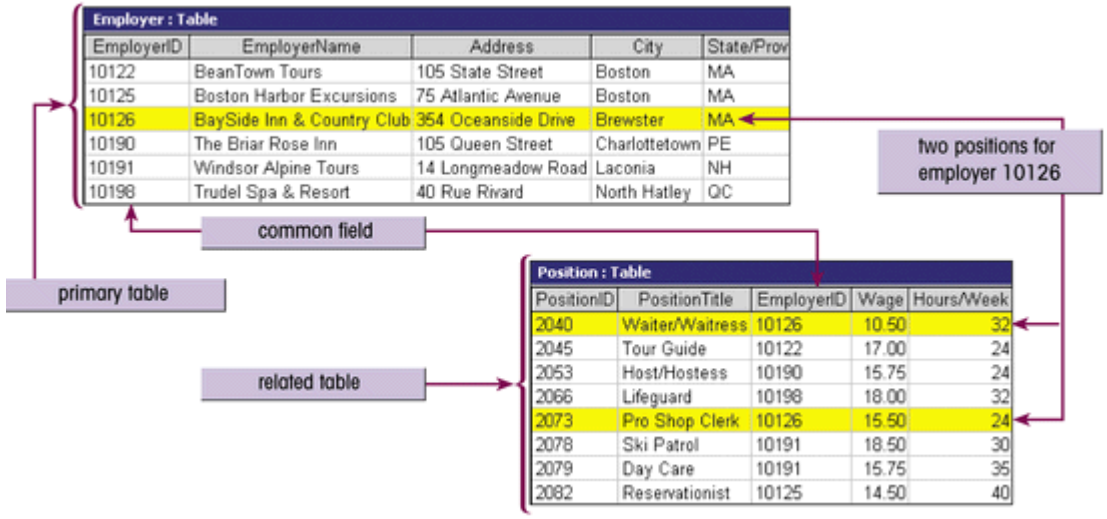
If you add a record to a related table a matching record must exist in the primary table or Access will not accept the entry.

Inconsistencies that can occur in related tables:

Problem 1:

Add a new record to the Position table with EmployerID = 10132 which does not appear in Employer Table

With Referential Integrity enforced, Access prevents this happening!



.....10132.....

Referential Integrity Rules

Rule 2

If you try to change the value of a primary key in a primary table thus leaving an orphaned foreign key, Access prevents you from doing so.

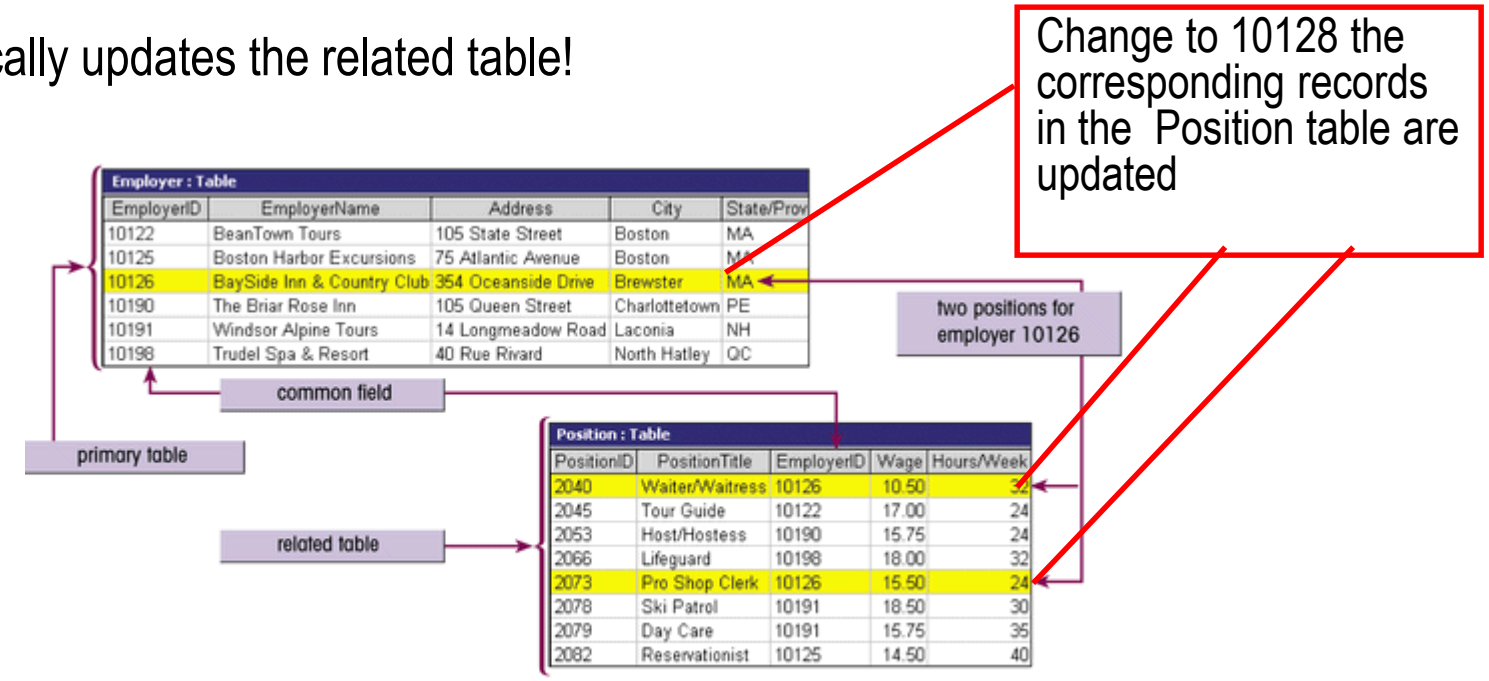
- Alternatively:
  - **cascade update** allows you to change the value of the primary key and automatically changes the appropriate foreign key values in the related table.

Problem 2:

Change the EmployerId for BaySide Inn & Country Club from 10126 to 10128

With Referential Integrity enforced, Access prevents you from doing this unless the **Cascade update** option is enforced then:

Access automatically updates the related table!



Referential Integrity Rules

Rule 3

If you try to delete a record in a primary table, Access will prevent you from doing so if there are matching records in a related table.

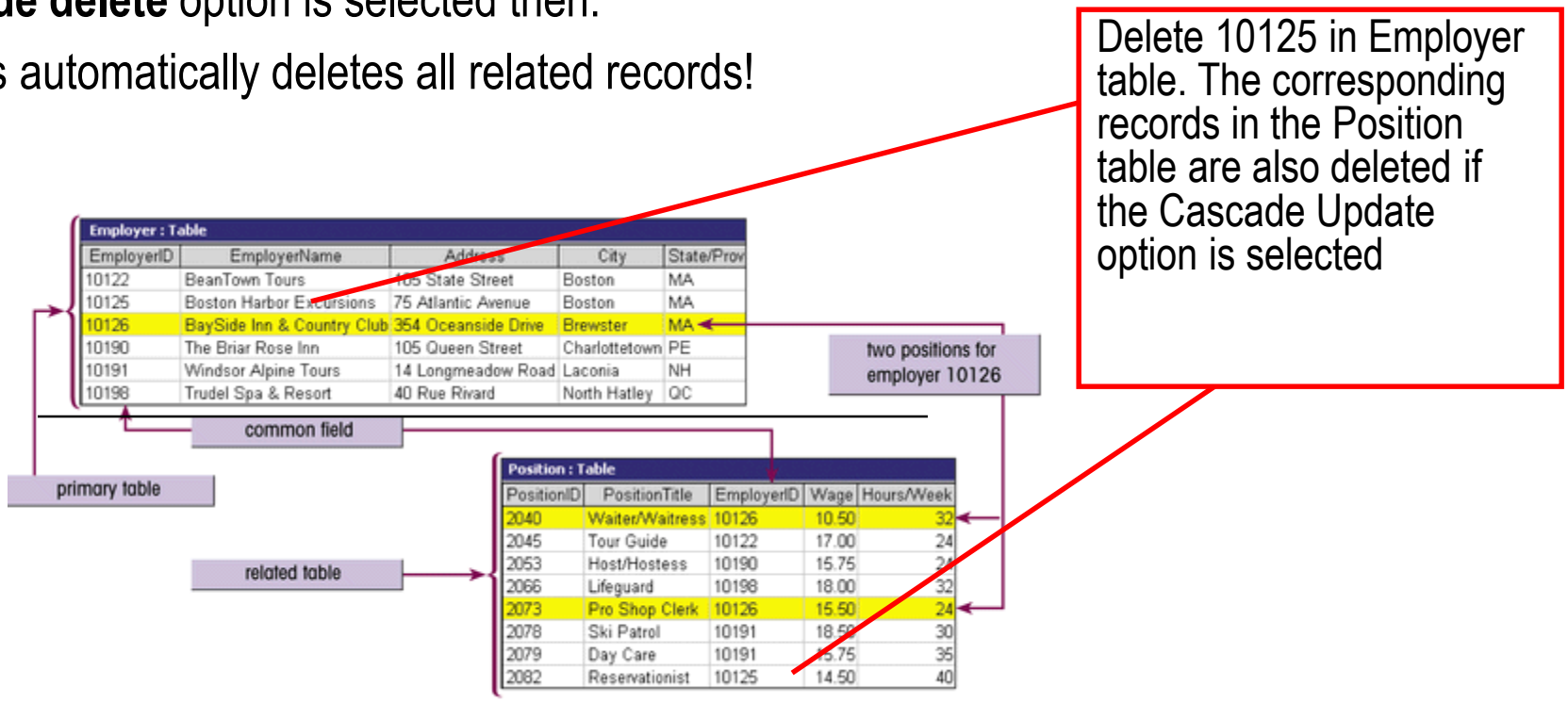
- Alternatively :
  - **cascade delete** allows you to delete a record in the primary table and then deletes all corresponding records in related tables.

Problem 3:

Delete the record for Boston Harbour Excursions (EmployerID 10125) as they are no longer required

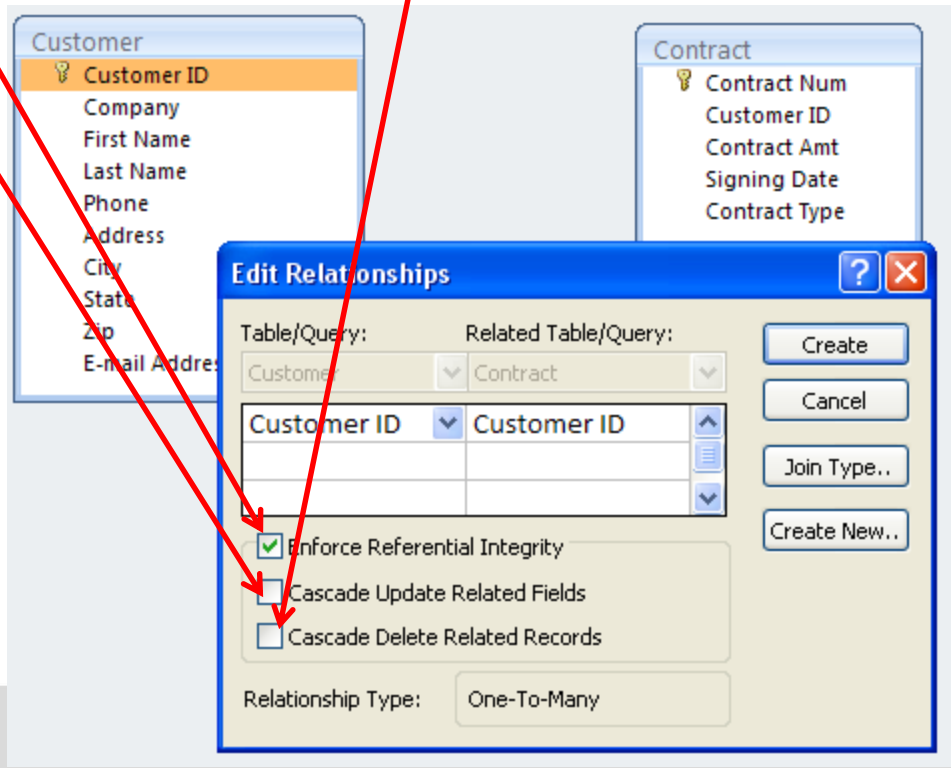
With Referential Integrity enforced, Access prevents you from doing this unless the **cascade delete** option is selected then:

Access automatically deletes all related records!



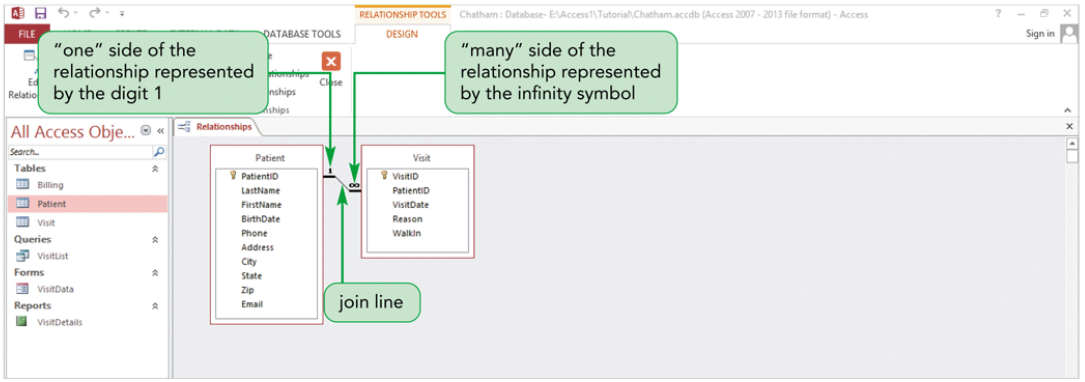
Enforcing Referential Integrity, cascade update and cascade delete

- Referential Integrity can be selected when defining the table relationships.
- Cascades Update and Cascade Deletes can also be selected when defining the table relationships.

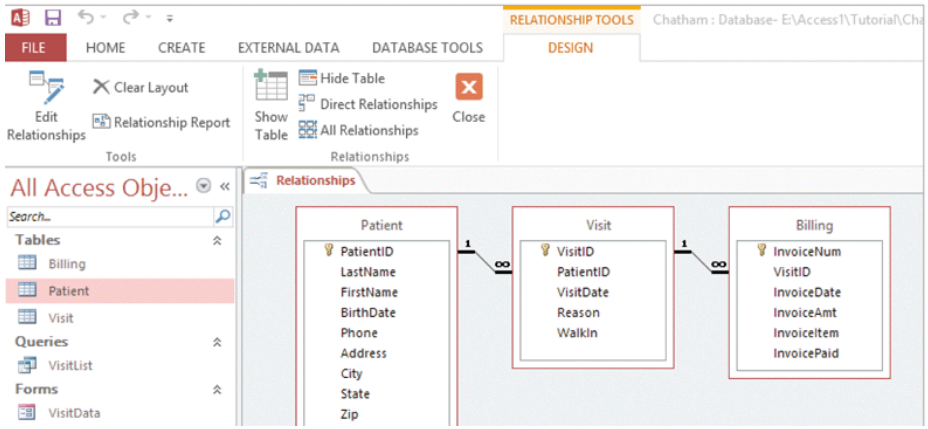


Defining Table Relationships (Cont.)

Figure 2-43 Defined relationship in the Relationships window



Both relationships defined



Summary

- Database query, form and report
- Guidelines for designing databases and setting field properties
- A table in Design view
- Types of relationships between two tables

Homework

- Go through example files and concepts
- NP Access 2016, Module 2