People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University of Tissemsilt

Faculty of Economic, Commercial, and Management Sciences

Practical Workbook

Computer Sciences 2: Microsoft Access

Speciality: Marketing

Level: 2nd Year Undergraduate Students

Author: Redouane GUELAILIA

Department: Commercial Sciences

Academic year: 2025 / 2026

Table of contents

Preface ..... **IV**

Session I: Introduction to databases and Microsoft Access **1**

Session II Creating Tables **9**

Session III Establishing Relationships **16**

Session IV: Queries **23**

Session V: Forms (User Interface Design) **31**

Session VI: Reports (Data Presentation and Printing) **39**

Session VII: Introduction to Macros and VBA Automation **47**

Conclusion **54**

List of tables and figures

Table 1: Data types in Microsoft Access 5
Figure 1: Microsoft Access interface 7
Figure 2: Structure of a table in Microsoft Access 12
Figure 3: Accessing the relationships window 19
Figure 4: Creating Students–Enrollments relationship 20
Figure 5: Creating Courses–Enrollments relationship 21
Figure 6: Final relationships diagram 22
Figure 7: Creating a new query 26
Figure 8: Adding a table to the query 26
Figure 9: Selecting query fields 27
Figure 10: Query results 28
Figure 11: Applying query criteria 29
Figure 12: Multi-table query design 30
Figure 13: Creating a form from a table 34
Figure 14: Automatically generated form 35
Figure 15: Form navigation controls 36
Figure 16: Switching to design view 37
Figure 17: Creating a report from a table 42
Figure 18: Automatically generated report 43
Figure 19: Switching to design view 44
Figure 20: Print settings window 45
Figure 21: Creating a new macro 50
Figure 22: Macro design view 51
Figure 23: Executing a macro 51
Figure 24: VBA editor interface 52
Table 3: Macros vs VBA 53

Preface

The increasing reliance on data-driven decision-making in modern organizations has made the mastery of database systems an essential skill in the field of business and management sciences. Databases constitute the backbone of information systems, enabling the structured storage, organization, and retrieval of data in an efficient and reliable manner.

This document serves as practical course material for the subject Computer sciences 2: Introduction to Databases and Microsoft Access, delivered to second-year undergraduate students in commercial Sciences within the Faculty of Economic, Commercial, and Management Sciences.

Unlike theoretical modules, this course is primarily practice-oriented and is designed to introduce students to the fundamental concepts and operational skills required to design, create, and manipulate relational databases using Microsoft Access. It focuses on the progressive development of practical competencies through guided laboratory sessions rather than extensive theoretical analysis.

The main objective of this workbook is to enable students to acquire hands-on experience in database construction and management. By the end of the course, students are expected to be able to design a complete database system, including tables, relationships, queries, forms, and reports, as well as perform basic automation tasks. The emphasis is placed on learning by doing, ensuring that students develop operational skills applicable to real-world business environments, particularly in student management and administrative data processing contexts.

The content of this practical guide has been carefully structured into sequential sessions, each addressing a specific aspect of database development. The progression follows a logical workflow, starting from basic database creation and

table design, moving through relational modeling and data extraction, and culminating in user interface design and basic automation.

Given the practical nature of this module, only essential theoretical explanations are included. These are limited to what is strictly necessary for understanding the procedures and successfully completing the exercises. The priority is given to clarity, step-by-step instructions, and guided laboratory activities.

The pedagogical approach adopted relies primarily on direct practice using Microsoft Access in a computer laboratory environment. This ensures that students actively engage with the software and gradually build autonomy in handling database operations.

Continuous updates may be introduced to improve clarity, enhance exercises, or adapt to software evolution. Constructive feedback from students and instructors is highly encouraged in order to improve the quality and effectiveness of this practical guide.

For any suggestions, corrections, or contributions, please contact the author at the following email address: **guelailia.redouane@univ-tissemsilt.dz**

Session I:
Introduction to
Databases and
Microsoft Access

This session provides a foundational overview of database systems and their role in information management. It introduces the basic concepts of relational databases and familiarizes students with the environment of Microsoft Access. The objective is to establish a clear understanding of how data is structured and managed within a database system.

1. Objectives

This session aims to provide students with a fundamental understanding of database systems and their role in information management. By the end of this session, students should be able to:

- Define the concept of a database
- Identify the main components of a relational database
- Understand the role of a Database Management System (DBMS)
- Create a new database using Microsoft Access
- Navigate the main elements of the software interface

2. Theoretical Background

Consider a scenario in which an academic department is required to manage a significant volume of information, including:

- Approximately 500 students
- 20 professors
- 10 courses per semester
- Examination marks for each student

When using spreadsheet software such as a traditional tabular tool, several limitations may arise. Data redundancy is likely to occur, as the same student information may be repeated across multiple sheets. This duplication increases the risk of inconsistency; for instance, a student's name may be entered differently in separate instances (ex. "Karim KATBI" versus "K. KATBI"), leading to data integrity issues. Furthermore, performing complex data retrieval tasks—such as identifying all students who have passed a specific course in a given academic year—becomes cumbersome and inefficient.

In contrast, the use of Microsoft Access as a relational database system provides a more structured and reliable solution. Student information is stored in a single, centralized table, thereby eliminating redundancy. Academic results are linked to individual students through unique identifiers, ensuring accuracy and consistency. Additionally, the system enables efficient data retrieval through queries, allowing users to extract specific information quickly and effectively.

2.1. What is a database?

A **database** is an organized collection of data that can be easily stored, managed, and retrieved.

In business, databases are used to manage:

- Customer information
- Sales transactions
- Inventory
- Employee records

Example (Student Database)

In this course, we will build a database to manage:

- Student personal information
- Courses
- Marks
- Professors

2.2. Basic Database Concepts

Understanding basic database concepts is essential for the effective design and use of relational database systems. These concepts provide the foundational framework for organizing, storing, and managing data in a structured manner. Key elements include tables, which represent collections of related data; records, which correspond to individual entries; and fields, which define specific attributes of those entries. Mastery of these fundamental notions is necessary in order to ensure data consistency, facilitate efficient retrieval, and support accurate information management within database applications. Here are some key concepts

- **Table :** A table stores data in rows and columns.
- **Field:** A field is a column in a table (ex. StudentName).
- **Record:** A record is a row in a table (one student).
- **Primary Key:** A primary key uniquely identifies each record.

Example:

StudentID	Name	Email
1	Ali	ali@email.com

2.3. Data Types

Data types define the nature of the values that can be stored in a field within a database table. Each field must be assigned an appropriate data type in order to ensure data accuracy, consistency, and efficient storage. For example, textual information such as names is stored as text, numerical values as numbers, and dates as date/time fields. In Microsoft Access, selecting the correct data type is essential for enabling proper data validation, facilitating calculations, and preventing entry errors.

Table 1: Data types in Microsoft Access

Data Type	When to use	Example
Short Text	Names, cities, phone numbers, majors	"Marketing"
Long Text	Long comments, addresses	Frères Khattel Street, Mellakou, Tiaret
Number	Quantities, IDs (if not AutoNumber)	15
Currency	Prices, fees, salaries	1250,00 DZD
Date/Time	Birthdays, enrollment dates	15/03/2005

Data Type	When to use	Example
Yes/No	Boolean (True/False)	HasScholarship? Yes/No
AutoNumber	Automatic unique ID	1,2,3...

3. Introduction to Microsoft Access

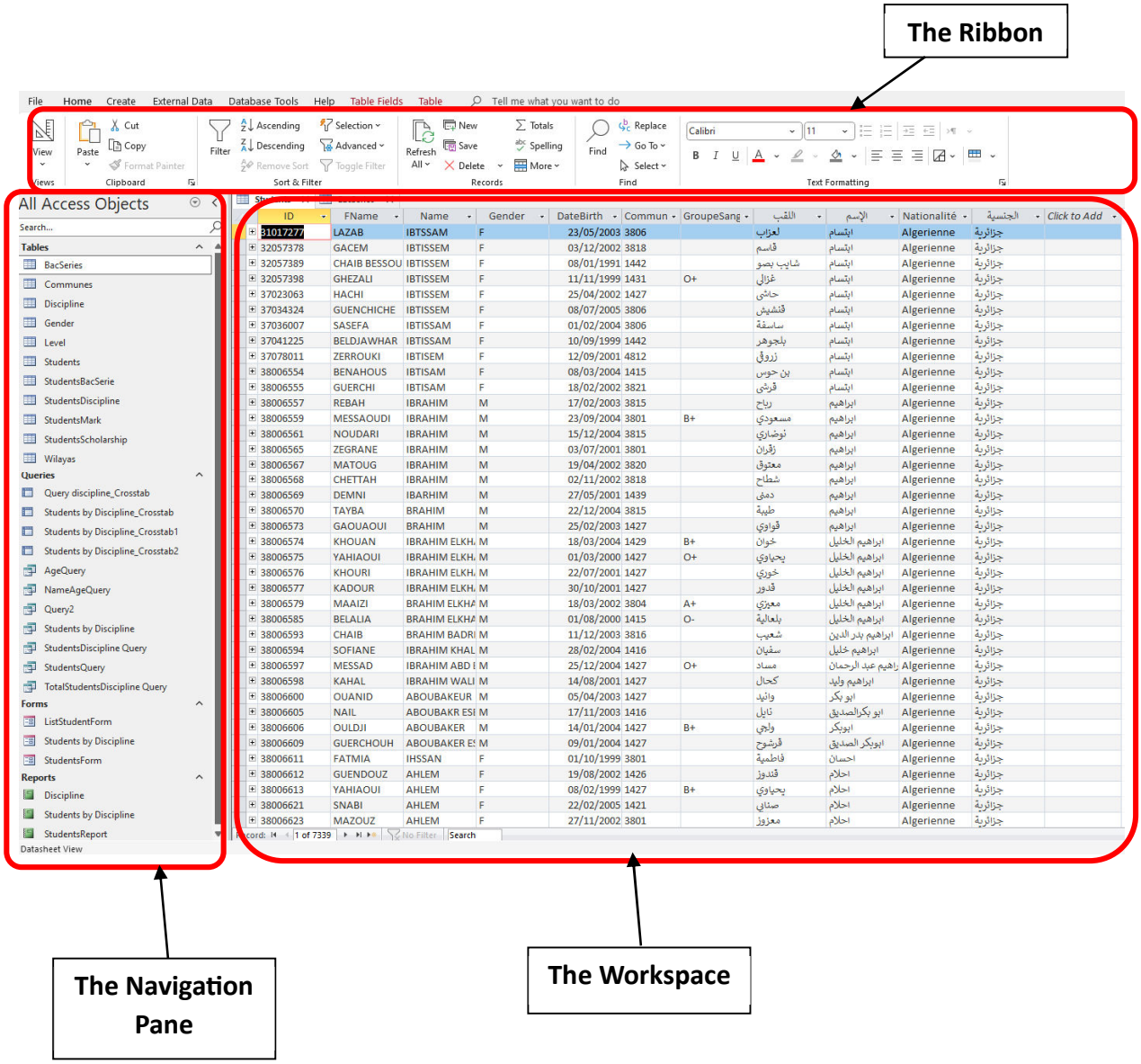
Microsoft Access is a relational Database Management System (DBMS) that enables users to design, create, and manage databases through a graphical user interface.

Exploring the Interface

The main components of the interface include:

- The Ribbon, which provides access to commands and tools
- The Navigation Pane, which displays database objects
- The Workspace, where objects are created and modified

Figure 1: Microsoft Access interface



1. Practical Activity

Create a database to store basic student information.

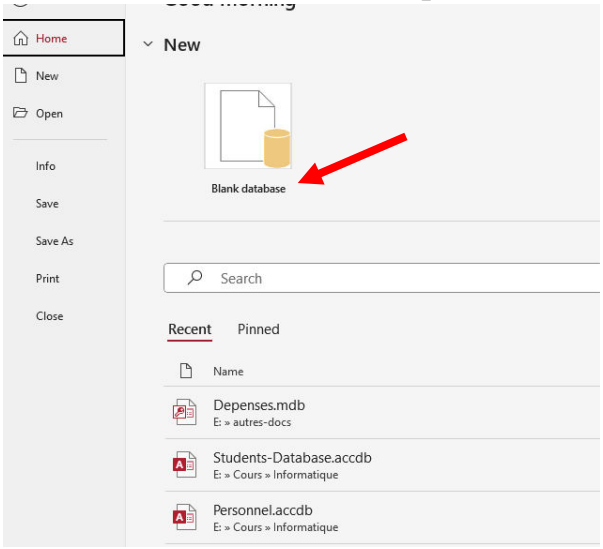
The following steps should be carried out:

Step 1: Launching the Application

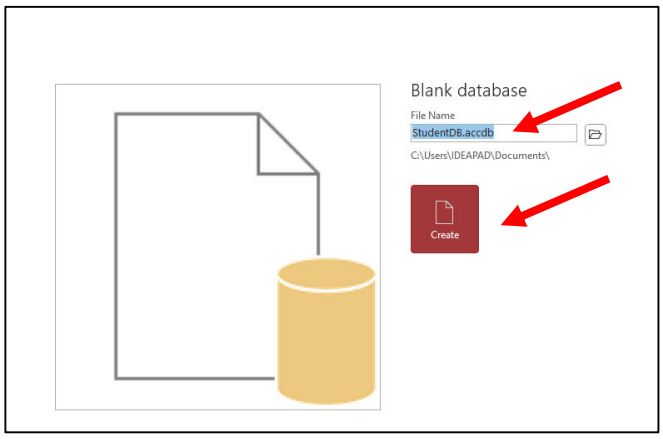
Open Microsoft Access from the operating system.

Step 2: Creating a New Database

- Select the Blank Database option



- Specify the file name: StudentDB
- Choose an appropriate storage location
- Click on Create



The database file will be saved with the extension .accdb.

6. Questions

1. What is the difference between a field and a record?
2. Why is the primary key important?
3. Give one example of database use in business
4. Why would it be a problem if a student's name was stored in 10 different Excel files? How does Access solve this?"

Session II

Creating Tables

This session focuses on the design and creation of tables, which constitute the core structure of any relational database. Students will learn how to define fields, select appropriate data types, and organize data efficiently using Microsoft Access. Emphasis is placed on proper table design principles to ensure data consistency and integrity.

1. Objectives

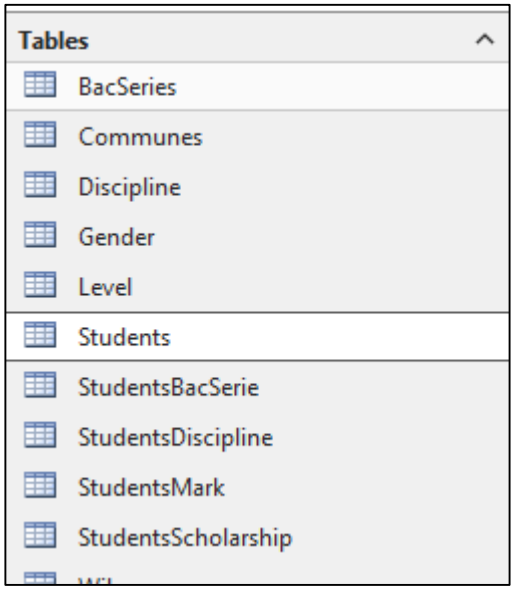
This session aims to develop students' ability to design well-structured tables within a relational database. By the end of this session, students should be able to:

- Apply principles of proper table design
- Select appropriate data types for each field
- Define primary keys correctly
- Create multiple related tables using Microsoft Access

2. Theoretical Background

In relational database systems, data is organized into tables that resemble a grid composed of rows and columns, similar in structure to an accountant's ledger or a spreadsheet.

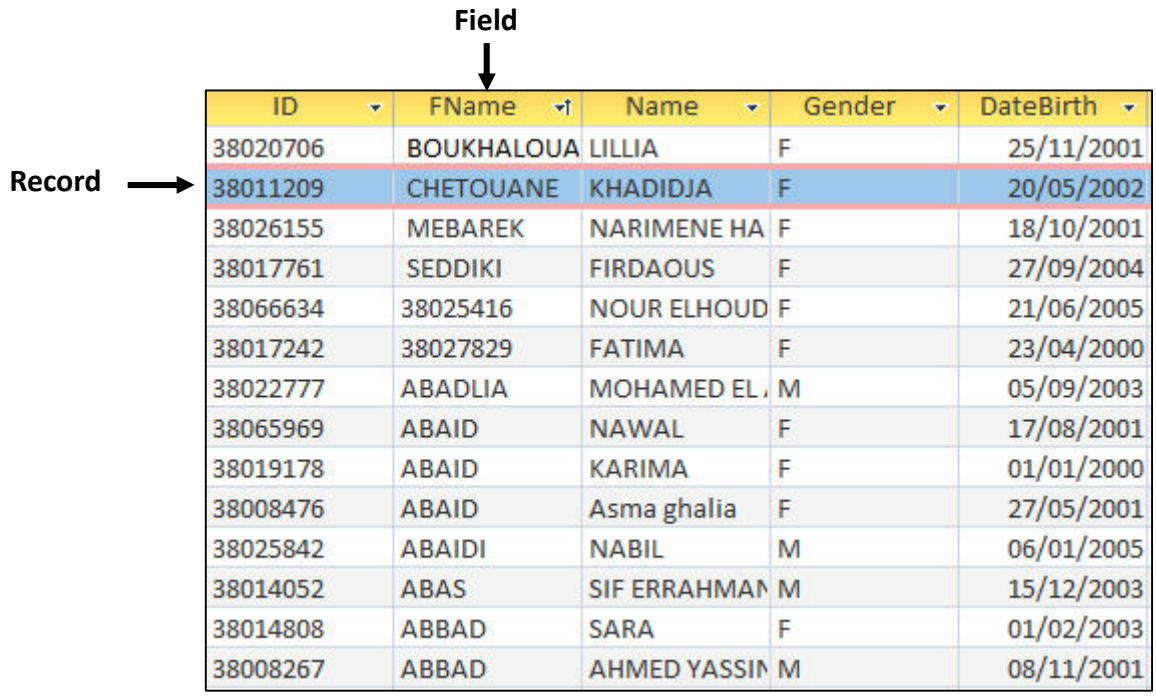
Microsoft Access organizes information into such tables, where each table represents a distinct entity or concept within the system. In simple database applications, a single table may be sufficient; however, most real-world databases require multiple tables to properly represent different types of data. For instance, a database may include a table dedicated to students, another to Professors, and a third to exams, each storing a specific category of information.



Within this structure, each row is more precisely referred to as a **record**, while each column is defined as a **field**. A record represents a coherent and meaningful collection of data describing a single entity. Conversely, a field corresponds to a single attribute that is consistently defined across all records within a table.

For example, in a *Students* table, each record contains all relevant information pertaining to one student. Each field within the table represents a specific characteristic of that student, such as its name, date birth, or field.

Figure 2: Structure of a table in Microsoft Access



ID	FName	Name	Gender	DateBirth
38020706	BOUKHALOUA	LILLIA	F	25/11/2001
38011209	CHETOUANE	KHADIDJA	F	20/05/2002
38026155	MEBAREK	NARIMENE HA	F	18/10/2001
38017761	SEDDIKI	FIRDAOUS	F	27/09/2004
38066634	38025416	NOUR ELHOUD	F	21/06/2005
38017242	38027829	FATIMA	F	23/04/2000
38022777	ABADLIA	MOHAMED EL	M	05/09/2003
38065969	ABAID	NAWAL	F	17/08/2001
38019178	ABAID	KARIMA	F	01/01/2000
38008476	ABAID	Asma ghalia	F	27/05/2001
38025842	ABAIDI	NABIL	M	06/01/2005
38014052	ABAS	SIF ERRAHMAN	M	15/12/2003
38014808	ABBAD	SARA	F	01/02/2003
38008267	ABBAD	AHMED YASSIN	M	08/11/2001

This structured approach ensures data consistency, reduces redundancy, and enhances the efficiency of data management operations within a relational database system.

3. Practical Procedure

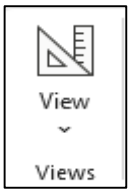
The following steps should be carried out:

Step 1: Opening the Existing Database

Open the previously created database *StudentDB* using Microsoft Access.

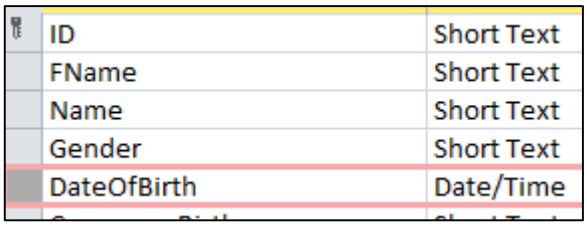
Step 2: Modifying the Students Table

- Open the *Students* table in *Design View*



- Add the following field:

Field Name	Data Type
DateOfBirth	Date/Time



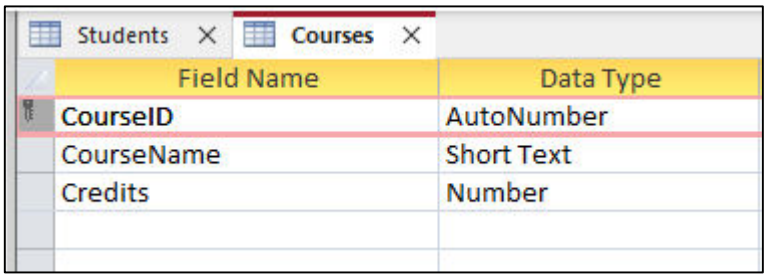
- Adjust field properties where necessary (ex. required fields)

Step 3: Creating the Courses Table

- Select the *Create* tab
- Choose *Table Design*
- Define the following structure:

Field Name	Data Type
CourseID	AutoNumber (Primary Key)
CourseName	Short Text
Credits	Number

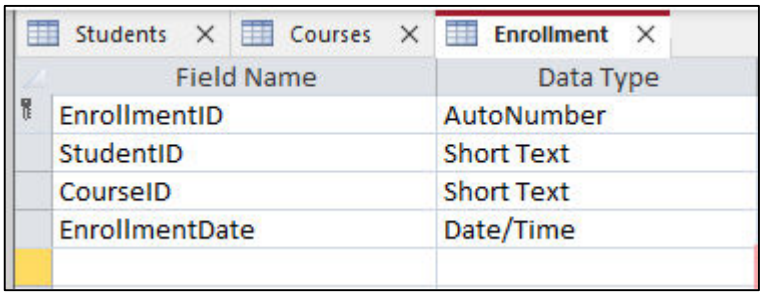
- Assign *CourseID* as the primary key
- Save the table as *Courses*



Step 4: Creating the Enrollments Table

- Create a new table in *Design View*
- Define the following fields:

Field Name	Data Type
EnrollmentID	AutoNumber (Primary Key)
StudentID	Short Text
CourseID	Short Text
EnrollmentDate	Date/Time



- Save the table as *Enrollments*

Particular attention should be paid to ensuring that **StudentID** and **CourseID** are defined using compatible data types.

Step 5 Setting the Primary Key

- Right-click on the gray box next to StudentID

- Choose **Primary Key** (a small key icon appears)

Why is this important? The Primary Key uniquely identifies each student. No two students can have the same StudentID.

Step 5: Entering Sample Data

- Insert sample records into the *Courses* table
- Insert corresponding records into the *Enrollments* table

This step prepares the database for the establishment of relationships in subsequent sessions.

4. Questions

- Explain why data should be distributed across multiple tables.
- Discuss the importance of selecting appropriate data types.
- Why must related fields share the same data type?

Session III

Establishing Relationships

This session introduces the concept of relationships between tables, a fundamental aspect of relational database design. Students will learn how to link tables using primary and foreign keys in Microsoft Access, ensuring logical data organization and referential integrity within the database.

1. Objectives

This session aims to introduce the fundamental concept of relationships between tables in a relational database system. By the end of this session, students should be able to:

- Understand the concept of relationships in relational databases
- Distinguish between different types of relationships (one-to-one, one-to-many)
- Link tables using primary and foreign keys
- Establish relationships using Microsoft Access
- Ensure referential integrity within a database

2. Theoretical Background

In relational database systems, data is distributed across multiple tables in order to reduce redundancy and improve data consistency. However, these tables are not independent; they are logically connected through **relationships**.

A relationship defines how data in one table is associated with data in another table. These associations are typically established through the use of **primary keys** and **foreign keys**. A primary key uniquely identifies each record in a table, while a foreign key is a field in another table that refers to that primary key.

The most common type of relationship is the **one-to-many relationship**, where a single record in one table is associated with multiple records in another table. For example, one student may be enrolled in multiple courses, and one course may include multiple students.

Microsoft Access provides a visual interface to define and manage relationships between tables, ensuring data integrity through referential constraints. These constraints prevent inconsistencies such as orphan records or invalid references.

Properly defined relationships are essential for building reliable and scalable database systems, as they allow data to be connected logically without duplication.

3. Practical Procedure

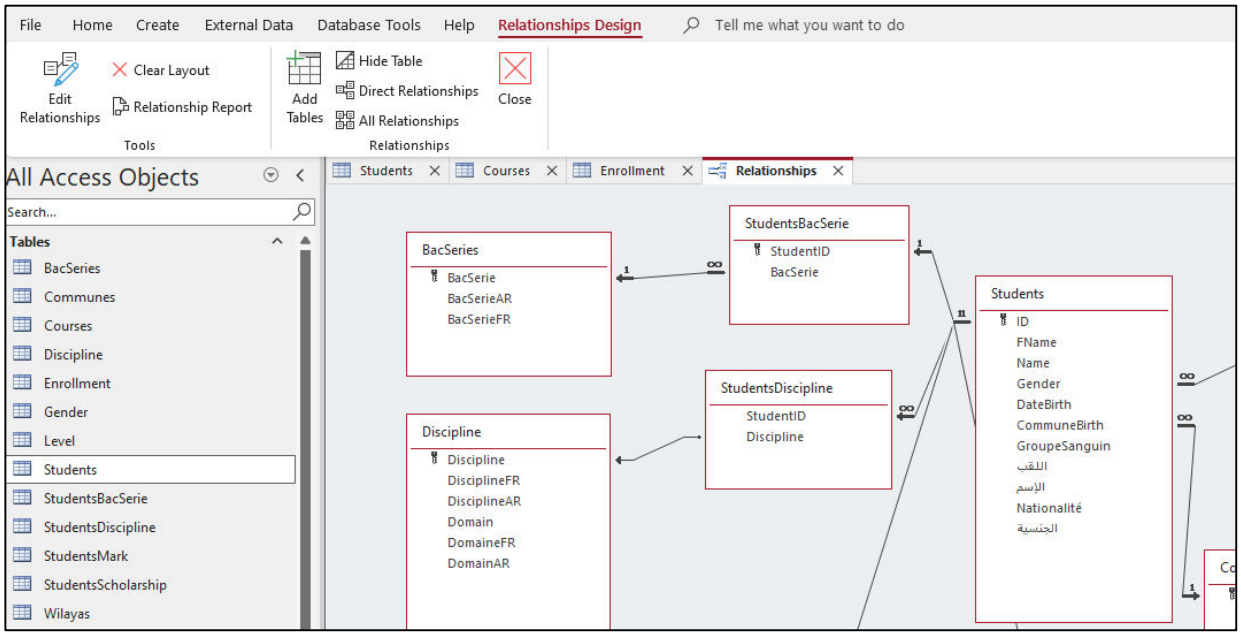
The following steps should be carried out using the existing *StudentDB* database.

Step 1: Opening the Relationships Tool

1. Open Microsoft Access
2. Open the database *StudentDB.accdb*
3. Navigate to the **Database Tools** tab

4. Click on **Relationships**

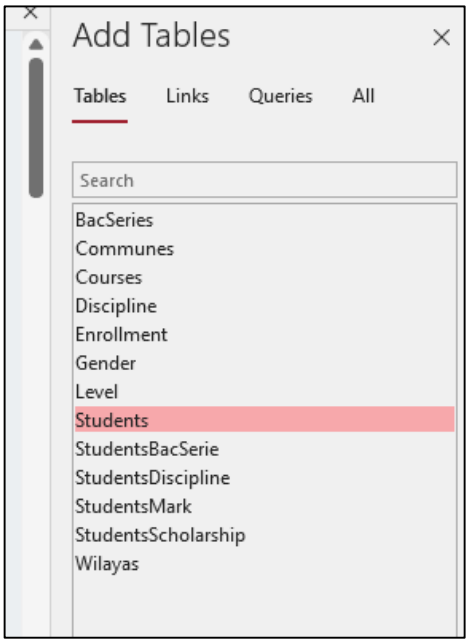
Figure 3: Accessing the relationships window



Step 2: Adding Tables

1. Click on **Show Table**
2. Add the following tables:
    - Students
    - Courses
    - Enrollments
3. Click **Close**

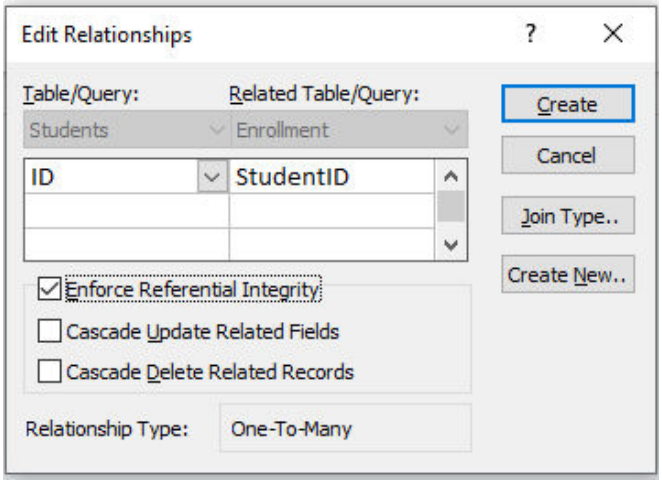
Figure 4: Adding tables to the Relationships window



Step 3: Creating a Relationship (Students → Enrollments)

1. Select the **StudentID** field in the *Students* table
2. Drag and drop it onto the **StudentID** field in the *Enrollments* table
3. A dialog box appears
4. Check the option **Enforce Referential Integrity**
5. Click **Create**

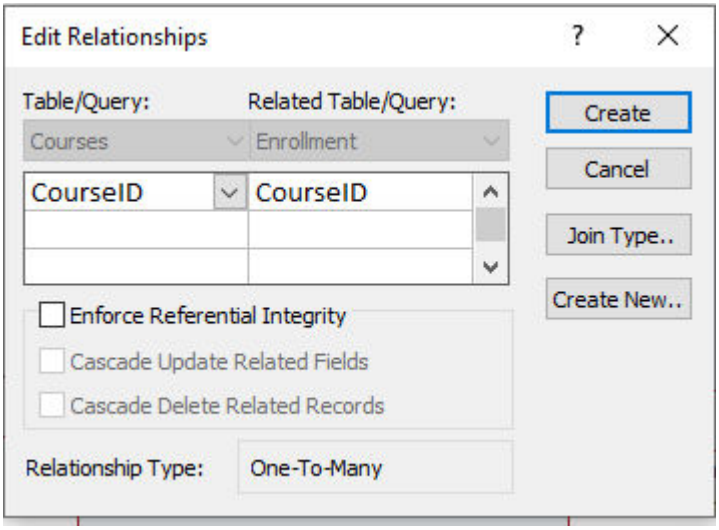
Figure 4: Creating Students–Enrollments relationship



Step 4: Creating a Relationship (Courses → Enrollments)

1. Select the **CourseID** field in the *Courses* table
2. Drag and drop it onto the **CourseID** field in the *Enrollments* table
3. Enable **Enforce Referential Integrity**
4. Click **Create**

Figure 5: Creating Courses–Enrollments relationship

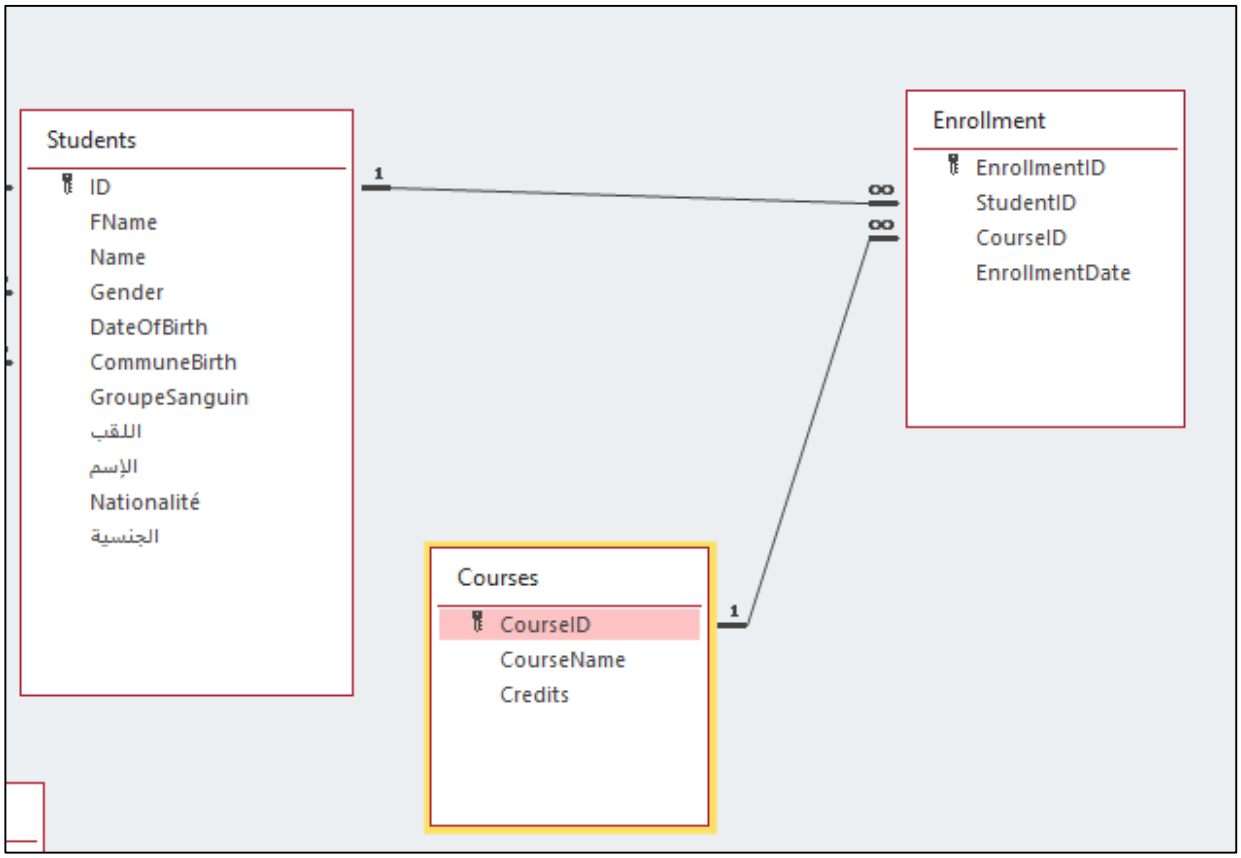


Step 5: Verifying Relationships

The Relationships window should now display:

- One-to-many relationship between Students and Enrollments
- One-to-many relationship between Courses and Enrollments

Figure 6: Final relationships diagram



4. Questions

- What is the role of a foreign key in a relational database?
- Why is referential integrity important?
- Explain why many-to-many relationships require a linking table.

Session IV:

Queries

This session focuses on data retrieval and manipulation through queries. Students will learn how to extract specific information from one or multiple tables using Microsoft Access. The session highlights the importance of queries in transforming raw data into meaningful information for analysis and decision-making.

1. Objectives

This session aims to introduce the concept of data retrieval through queries in a relational database system. By the end of this session, students should be able to:

- Understand the role of queries in data manipulation and analysis
- Create simple queries using Microsoft Access
- Apply filtering criteria to extract specific information
- Perform basic data aggregation operations
- Interpret query results for decision-making purposes

2. Theoretical Background

In relational database systems, data is stored in structured tables; however, effective use of this data requires the ability to retrieve specific subsets of information. This function is performed through **queries**.

A query is a database operation that allows users to extract, filter, and manipulate data stored in one or more tables. Queries are essential for transforming raw data into meaningful information that can support analysis and decision-making.

Queries may operate on a single table or on multiple related tables. They can be used to:

- Select specific records based on defined criteria
- Sort and organize data
- Perform calculations and aggregations (ex. sums, averages, counts)

Microsoft Access provides a visual tool known as **Query Design View**, which allows users to build queries without writing code. This interface simplifies the process of defining fields, selecting tables, and applying conditions.

In business contexts, queries are widely used to generate reports such as sales summaries, student performance lists, or customer segmentation results.

3. Practical Procedure

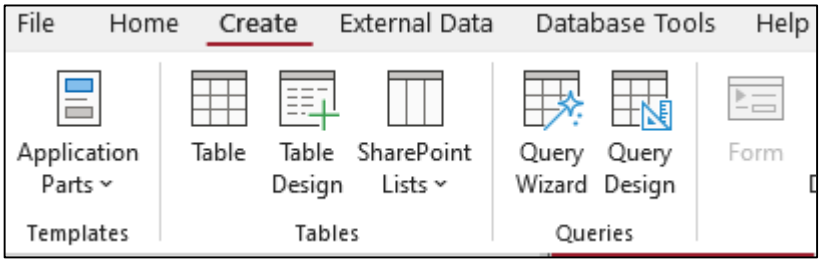
The following steps should be carried out using the existing *StudentDB* database.

Step 1: Creating a Query

1. Open Microsoft Access
2. Open the database *StudentDB.accdb*
3. Navigate to the **Create** tab

4. Click on **Query Design**

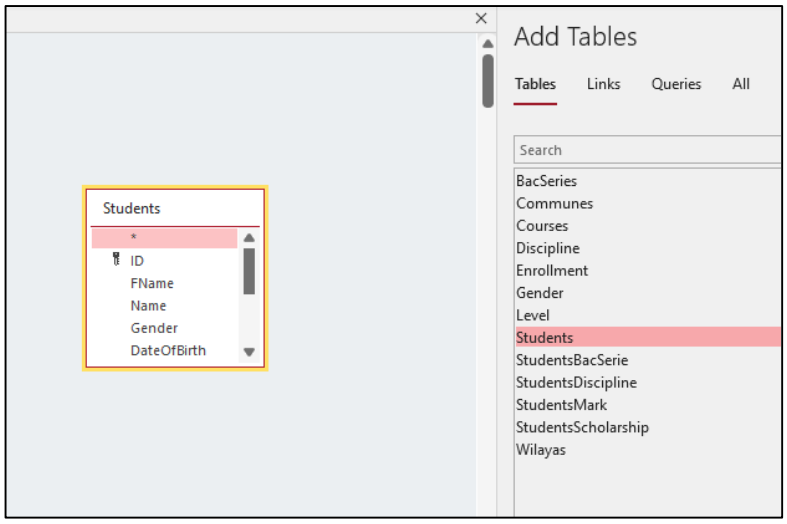
Figure 7: Creating a new query



Step 2: Adding Tables

1. In the "Show Table" window, add:
    - Students
2. Click **Close**

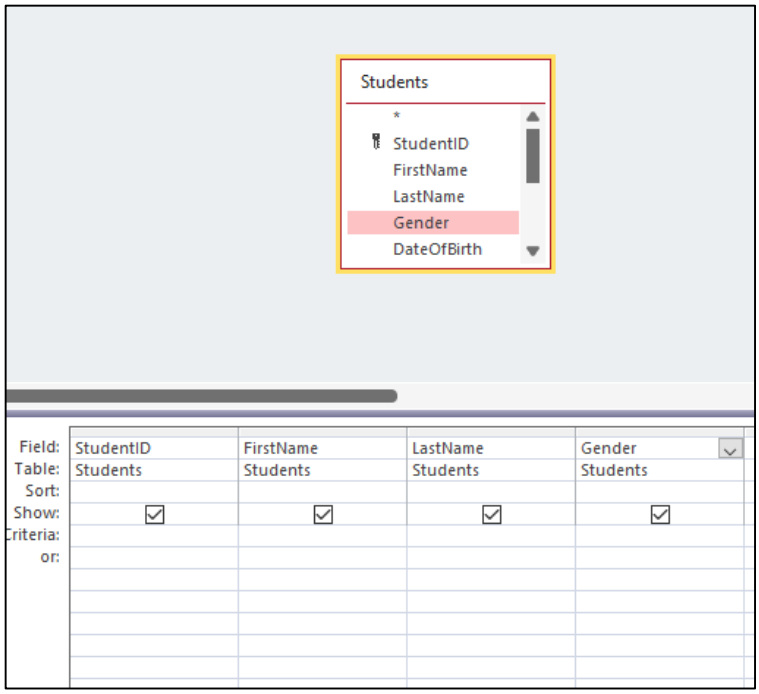
Figure 8: Adding a table to the query



Step 3: Selecting Fields

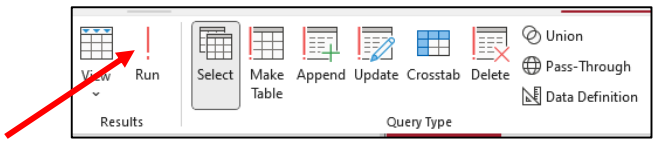
1. Double-click the following fields to include them in the query:
    - StudentID
    - FirstName
    - LastName
    - Gender

Figure 9: Selecting query fields



Step 4: Running the Query

1. Click on the **Run** button (red exclamation mark)



2. The system will display all selected records

Figure 10: Query results



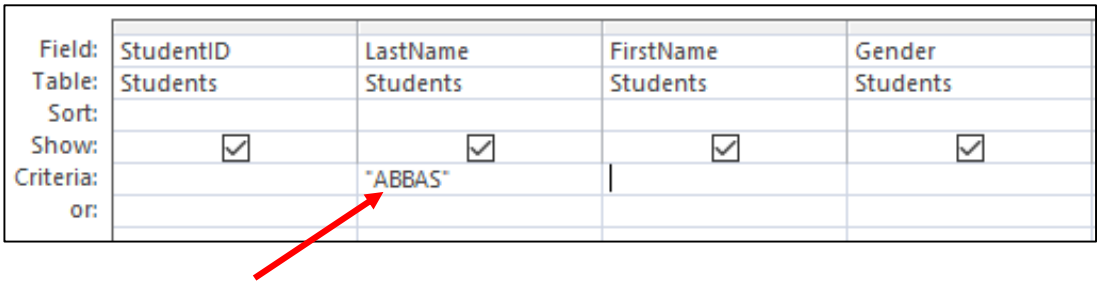
Query1

StudentID	LastName	FirstName	Gender
38020706	BOUKHALOUA	LILLIA	F
38011209	CHETOUANE	KHADIDJA	F
38026155	MEBAREK	NARIMENE HA	F
38017761	SEDDIKI	FIRDAOUS	F
38066634	38025416	NOUR ELHOUD	F
38017242	38027829	FATIMA	F
38022777	ABADLIA	MOHAMED EL	M
38065969	ABAID	NAWAL	F
38019178	ABAID	KARIMA	F
38008476	ABAID	Asma ghalia	F
38025842	ABAIDI	NABIL	M
38014052	ABAS	SIF ERRAHMAN	M
38014808	ABBAD	SARA	F
38008267	ABBAD	AHMED YASSIN	M
38066728	ABBAD	NOUR EL HOUD	F
38067896	ABBADI	HALA	F
38012001	ABBADI	RACHIDA	F
38019019	ABBAOUI	KARIM	M
38025299	ABBAS	MANEL	F
38021317	ABBAS	MOHAMMED	M
38008124	ABBAS	AHMED	M
38009262	ABBAS	BELAID	M
38007835	ABBAS	ILYES	M
38009295	ABBAS	BELKIS	F
38021159	ABBAS	MADJID	M
38010457	ABBAS	HOURIA	F
38012105	ABBAS	REKIA	F
38026388	ABBAS	NASSREDDINE	M

4. Applying Criteria (Filtering Data)

Step 5: Filtering Students by City or Condition

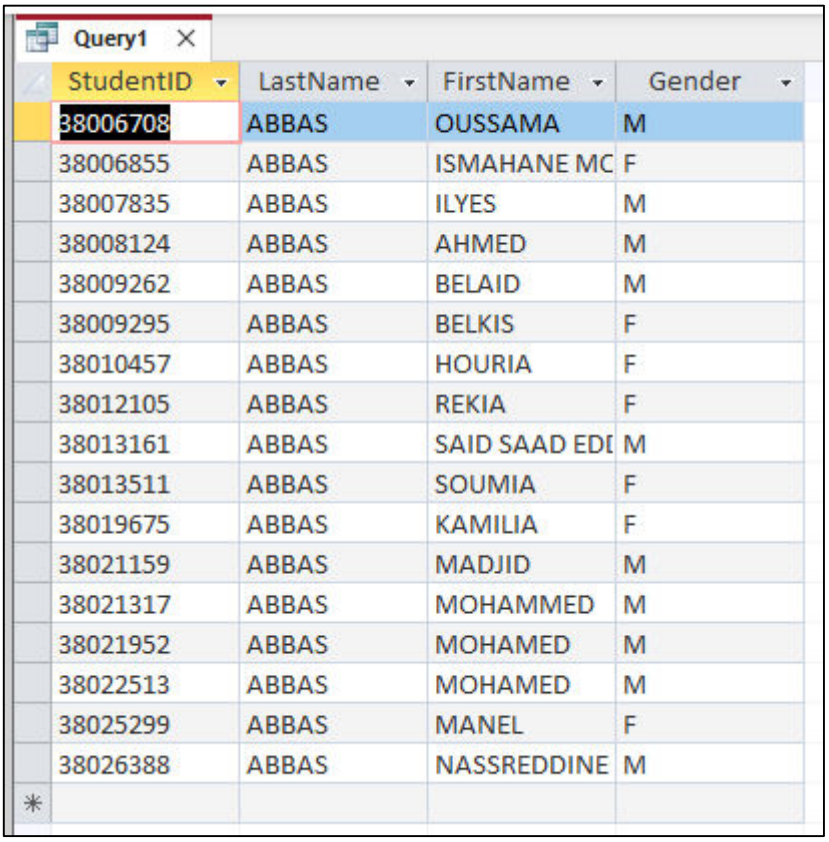
1. Return to **Design View**
2. In the **Criteria** row under a field (ex. LastName), enter a condition such as:
    - "Abbas" (or "A*")



Field:	StudentID	LastName	FirstName	Gender
Table:	Students	Students	Students	Students
Sort:				
Show:	☑	☑	☑	☑
Criteria:		"ABBAS"		
or:				

3. Run the query again

Figure 11: Applying query criteria



Query1

StudentID	LastName	FirstName	Gender
38006708	ABBAS	OUSSAMA	M
38006855	ABBAS	ISMAHANE MC	F
38007835	ABBAS	ILYES	M
38008124	ABBAS	AHMED	M
38009262	ABBAS	BELAID	M
38009295	ABBAS	BELKIS	F
38010457	ABBAS	HOURIA	F
38012105	ABBAS	REKIA	F
38013161	ABBAS	SAID SAAD EDI	M
38013511	ABBAS	SOUMIA	F
38019675	ABBAS	KAMILIA	F
38021159	ABBAS	MADJID	M
38021317	ABBAS	MOHAMMED	M
38021952	ABBAS	MOHAMED	M
38022513	ABBAS	MOHAMED	M
38025299	ABBAS	MANEL	F
38026388	ABBAS	NASSREDDINE	M

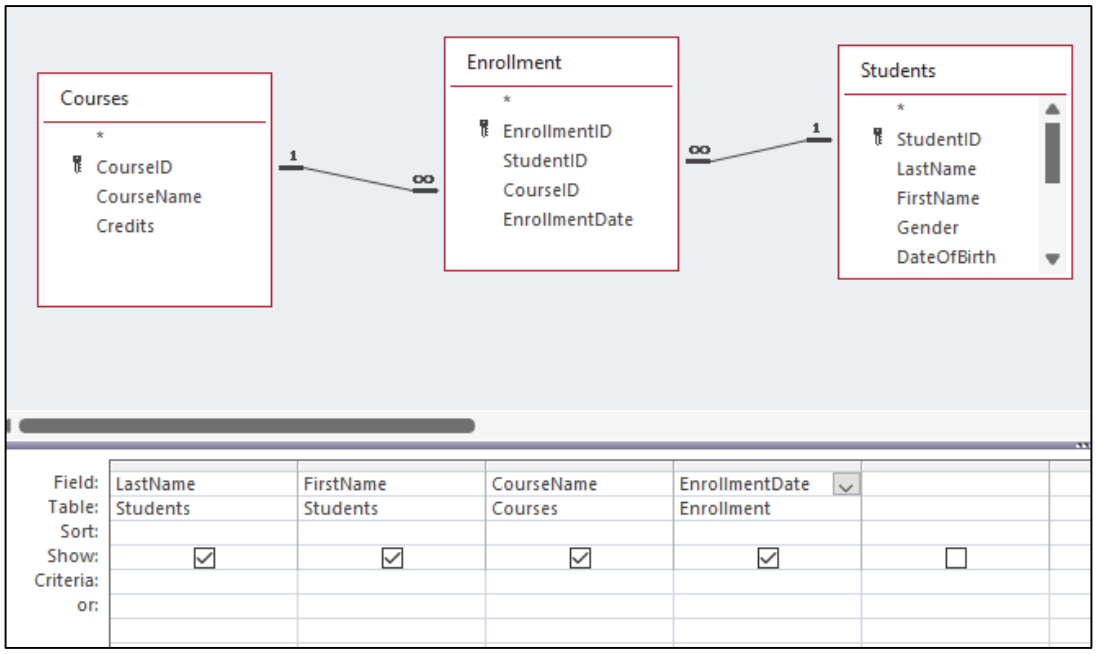
5. Query with Multiple Tables (Advanced Introduction)

Step 6: Using Enrollments Data

1. Create a new query
2. Add the following tables:
    - Students
    - Enrollments
    - Courses
3. Select:
    - FirstName

- LastName
- CourseName
- EnrollmentDate

Figure 12: Multi-table query design



4. Questions

- What is the role of a query in a database system?
- Why are queries essential for decision-making in business contexts?
- What is the advantage of using multiple tables in a single query?

Session V:

Forms (User Interface Design)

This session introduces forms as a user interface for interacting with database content. Students will learn how to create and customize forms using Microsoft Access in order to facilitate data entry, navigation, and modification in a structured and user-friendly manner.

1. Objectives

This session aims to introduce the concept of forms as a user-friendly interface for interacting with database data. By the end of this session, students should be able to:

- Understand the role of forms in database systems
- Create and customize forms using Microsoft Access
- Use forms for data entry, modification, and navigation
- Improve usability and presentation of database content

2. Theoretical Background

In relational database systems, direct manipulation of tables is not always efficient or user-friendly, particularly for end-users who are not familiar with database structures. For this reason, **forms** are used as an interface layer between users and data.

A form is a structured interface that allows users to:

- Enter new data into tables
- Modify existing records
- Navigate through stored information

Forms do not store data themselves; instead, they are directly linked to underlying tables. This ensures that any modification made through a form is immediately reflected in the database.

Microsoft Access provides tools to automatically generate forms based on existing tables or queries. These forms can then be customized to improve readability, usability, and visual structure.

In business applications, forms are widely used for:

- Customer registration systems
- Student management systems
- Order entry interfaces

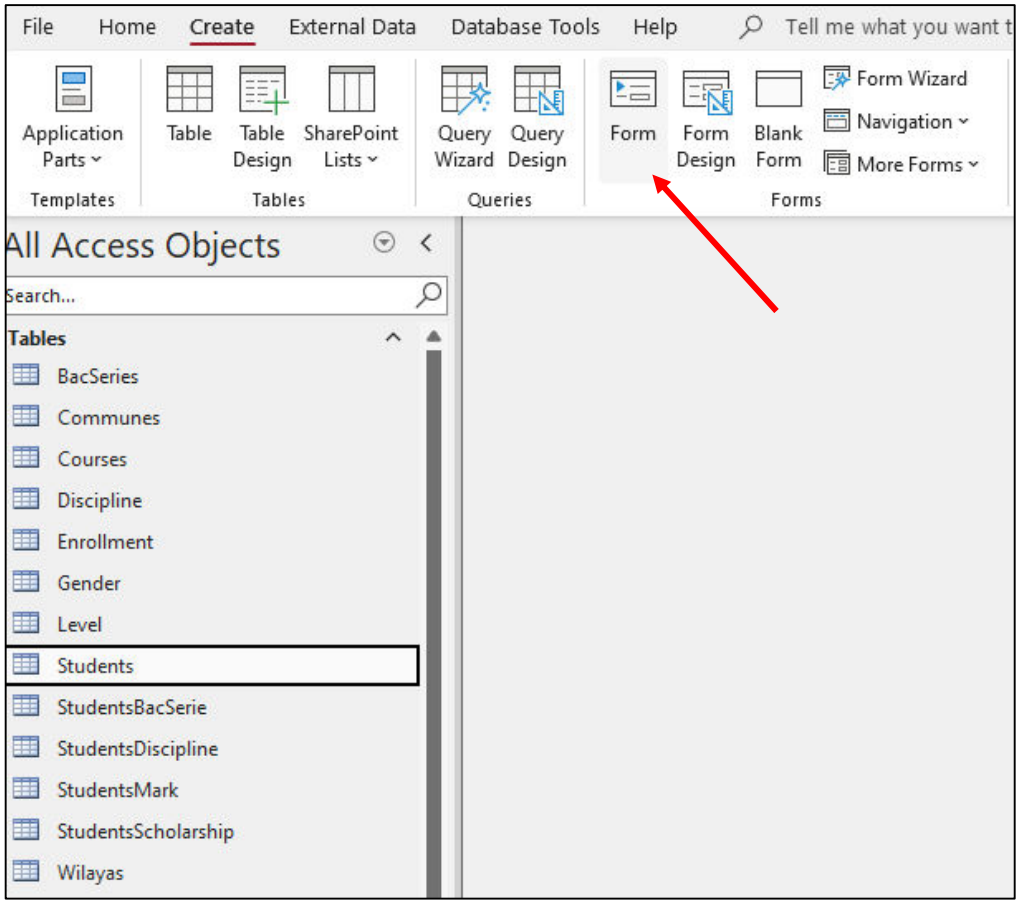
3. Practical Procedure

The following steps should be carried out using the existing *StudentDB* database.

Step 1: Creating a Form

1. Open Microsoft Access
2. Open the database *StudentDB.accdb*
3. In the Navigation Pane, select the *Students* table
4. Navigate to the **Create** tab
5. Click on **Form**

Figure 13: Creating a form from a table



Step 2: Automatic Form Generation

The system will automatically generate a form based on the selected table.

The form will display:

- One record at a time
- All fields from the *Students* table

Figure 14: Automatically generated form

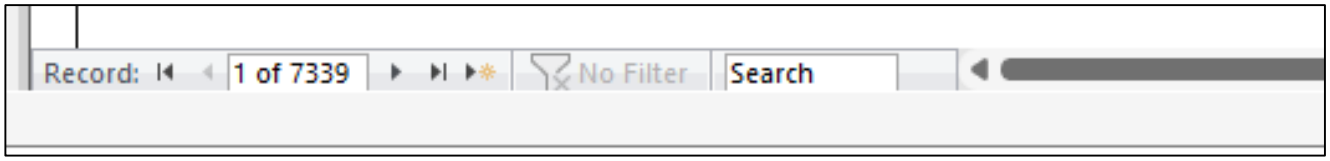


Step 3: Navigating Through Records

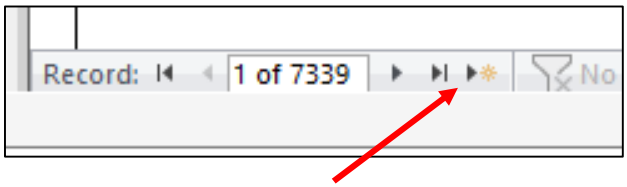
At the bottom of the form, use navigation buttons to:

- Move to the next record
- Move to the previous record
- Jump to the first or last record

Figure 15: Form navigation controls



Step 4: Adding a New Record



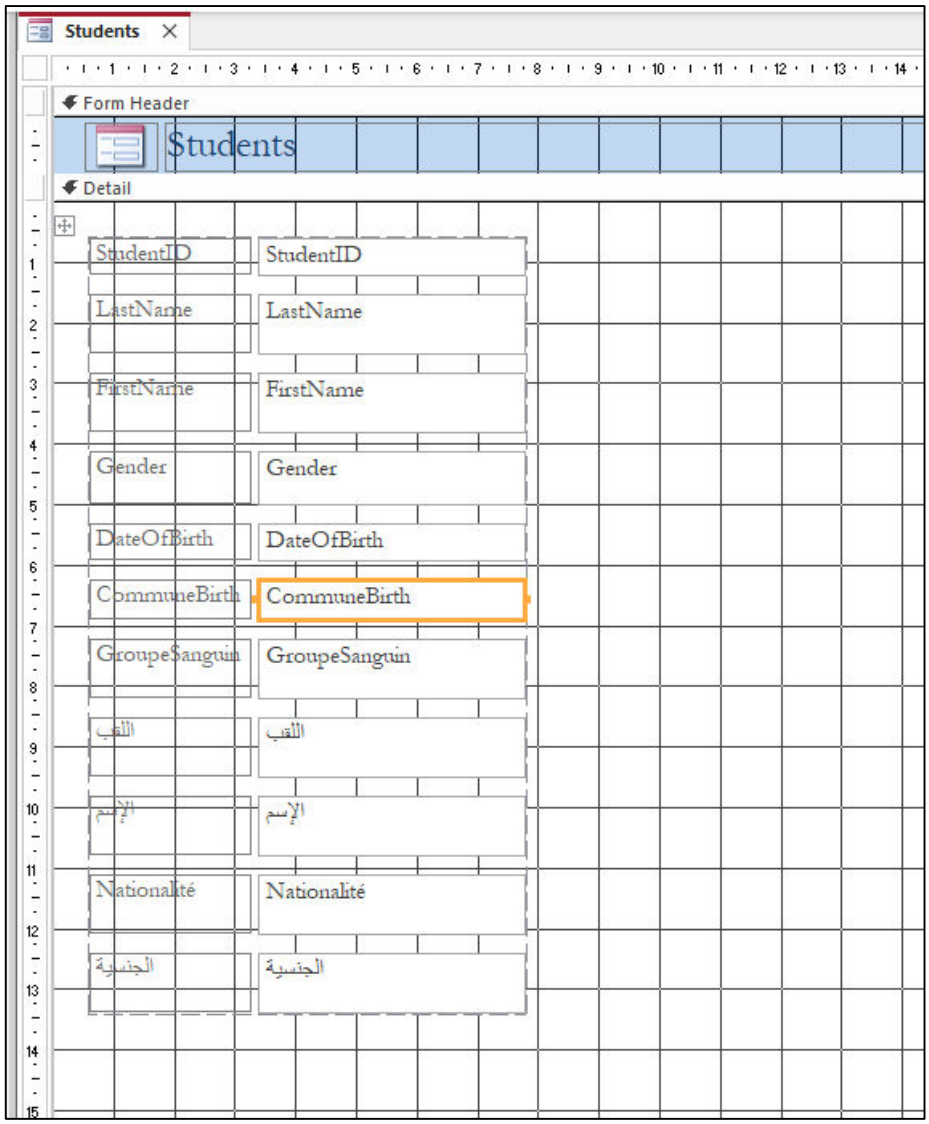
1. Click on **New (blank) record**
2. Enter the following data:
    - FirstName
    - LastName
    - DateOfBirth
    - Gender
3. Save the record automatically

- Customizing the Form

Step 5: Enter Design View

1. Right-click on the form name
2. Select **Design View**

Figure 16: Switching to design view



Step 6: Adding a Title

1. Insert a title at the top of the form
2. Example:
   Student Management Form

Step 7: Adjusting Layout

Students may:

- Resize fields
- Align labels and input boxes
- Improve readability

4. Questions

- What is the role of a form in a database system?
- Why are forms preferred over direct table manipulation for end-users?
- How does a form improve data usability and accessibility?

Session VI: Reports (Data Presentation and Printing)

This session focuses on the presentation and formatting of data through reports. Students will learn how to generate structured and printable outputs using Microsoft Access, allowing data to be communicated effectively for academic or administrative purposes.

1. Objectives

This session aims to introduce the concept of reports as a structured and formatted output of database information. By the end of this session, students should be able to:

- Understand the role of reports in database systems
- Generate reports using Microsoft Access
- Organize and format data for presentation purposes
- Group, sort, and summarize information
- Produce printable documents for decision-making and reporting

2. Theoretical Background

In relational database systems, direct manipulation of tables is not always efficient or user-friendly, particularly for end-users who are not familiar with database.

In relational database systems, data is often required not only for internal processing but also for presentation, analysis, and communication. This requirement is addressed through **reports**.

A report is a structured and formatted output that presents data in a readable and printable format. Unlike forms, which are primarily used for data entry and modification, reports are designed for **data visualization and dissemination**.

Reports may include:

- Tabular data presentation
- Grouped and sorted records
- Summary calculations (such as totals and averages)
- Headers and footers for documentation purposes

Microsoft Access provides tools to automatically generate reports based on tables or queries. These reports can then be customized to meet specific formatting and presentation requirements.

In business environments, reports are essential for:

- Academic performance summaries
- Financial and sales reporting
- Administrative documentation
- Decision-support systems

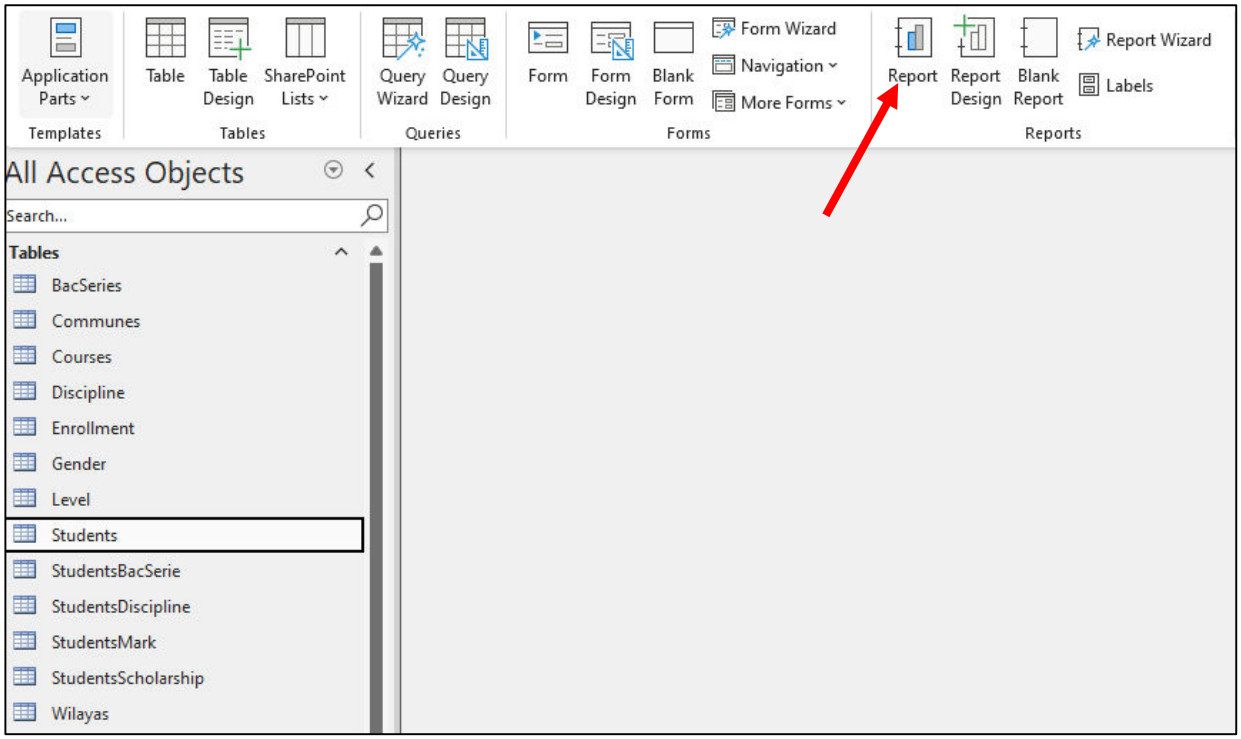
3. Practical Procedure

The following steps should be carried out using the existing *StudentDB* database.

Step 1: Creating a report from a table

1. Open Microsoft Access
2. Open the database *StudentDB.accdb*
3. In the Navigation Pane, select the *Students* table
4. Navigate to the **Create** tab
5. Click on **Report**

Figure 17: Creating a report from a table

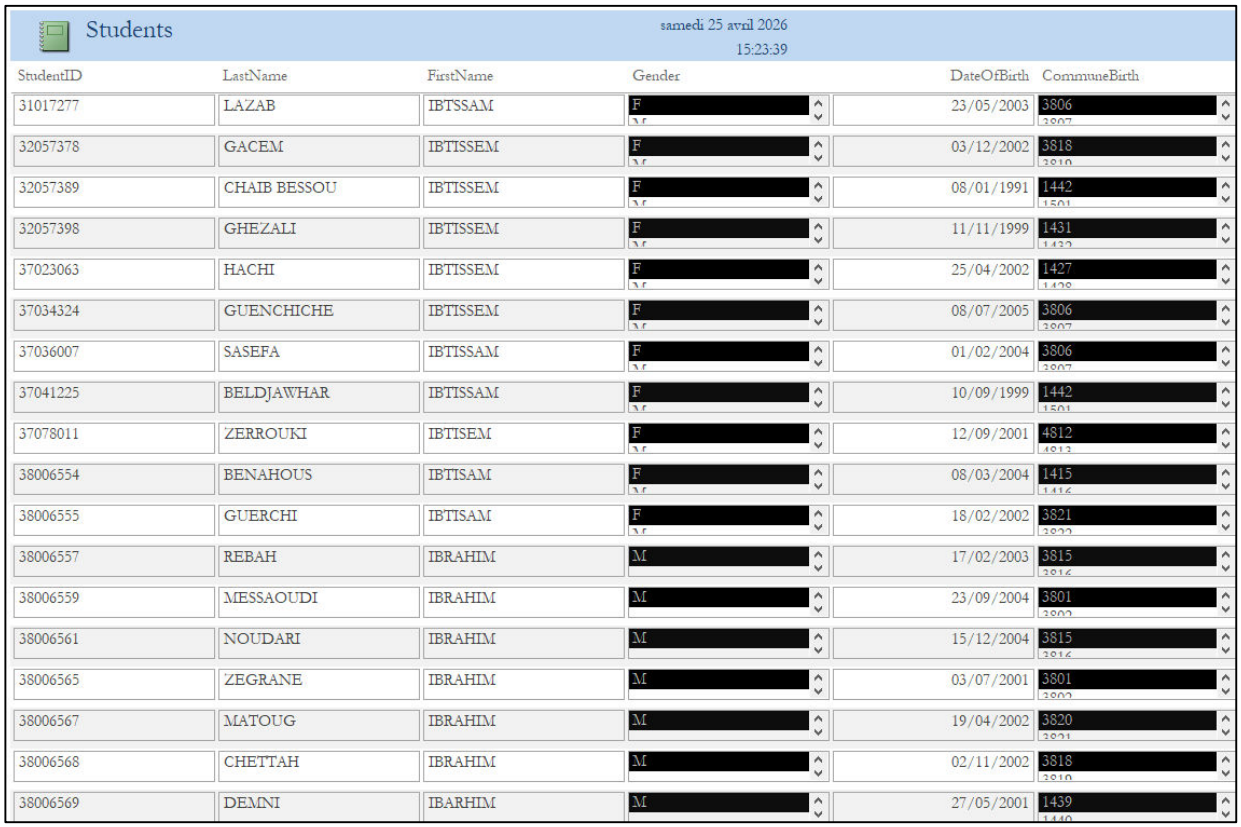


Step 2: Automatic Report Generation

The system will automatically generate a report containing:

- All records from the selected table
- All fields included in a structured layout

Figure 18: Automatically generated report



Students

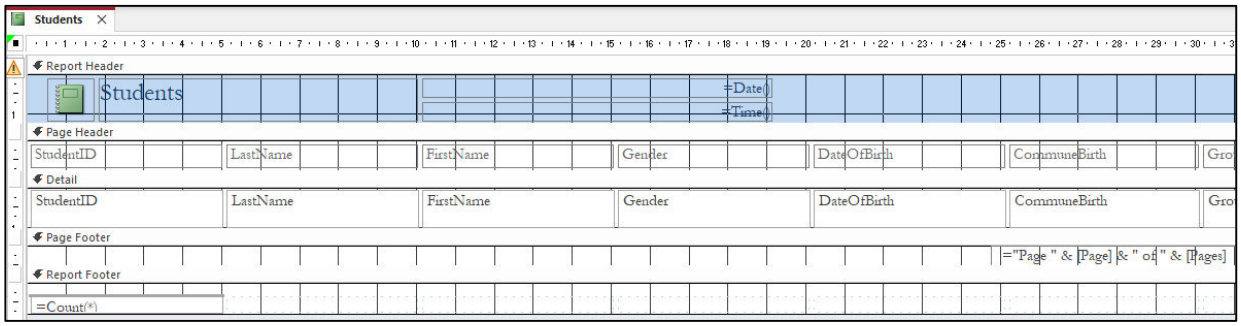
samedi 25 avril 2026
15:23:39

StudentID	LastName	FirstName	Gender	DateOfBirth	CommuneBirth
31017277	LAZAB	IBTSSAM	F	23/05/2003	3806
32057378	GACEM	IBTISSEM	F	03/12/2002	3818
32057389	CHAIB BESSOU	IBTISSEM	F	08/01/1991	1442
32057398	GHEZALI	IBTISSEM	F	11/11/1999	1431
37023063	HACHI	IBTISSEM	F	25/04/2002	1427
37034324	GUENCHICHE	IBTISSEM	F	08/07/2005	3806
37036007	SASEFA	IBTISSAM	F	01/02/2004	3806
37041225	BELDJAWHAR	IBTISSAM	F	10/09/1999	1442
37078011	ZERROUKI	IBTISEM	F	12/09/2001	4812
38006554	BENAHOUS	IBTISAM	F	08/03/2004	1415
38006555	GUERCHI	IBTISAM	F	18/02/2002	3821
38006557	REBAH	IBRAHIM	M	17/02/2003	3815
38006559	MESSAOUDI	IBRAHIM	M	23/09/2004	3801
38006561	NOUDARI	IBRAHIM	M	15/12/2004	3815
38006565	ZEGRANE	IBRAHIM	M	03/07/2001	3801
38006567	MATOUG	IBRAHIM	M	19/04/2002	3820
38006568	CHETTAH	IBRAHIM	M	02/11/2002	3818
38006569	DEMNI	IBARHIM	M	27/05/2001	1439

Step 3: Viewing and Navigating the Report

1. Use the **Print Preview** mode to view the report layout
2. Scroll through pages to verify content
3. Check alignment and readability

Figure 19: Switching to design view



- Customizing the Report

Step 4: Switching to Design View

1. Right-click the report name
2. Select **Design View**

Step 5: Adding a Title

1. Insert a title at the top of the report
2. Example:
   Student List Report

Step 6: Organizing Data

Students may:

- Adjust column widths
- Align text fields
- Modify font styles for readability

- Grouping and Sorting

Reports can be enhanced by organizing data logically.

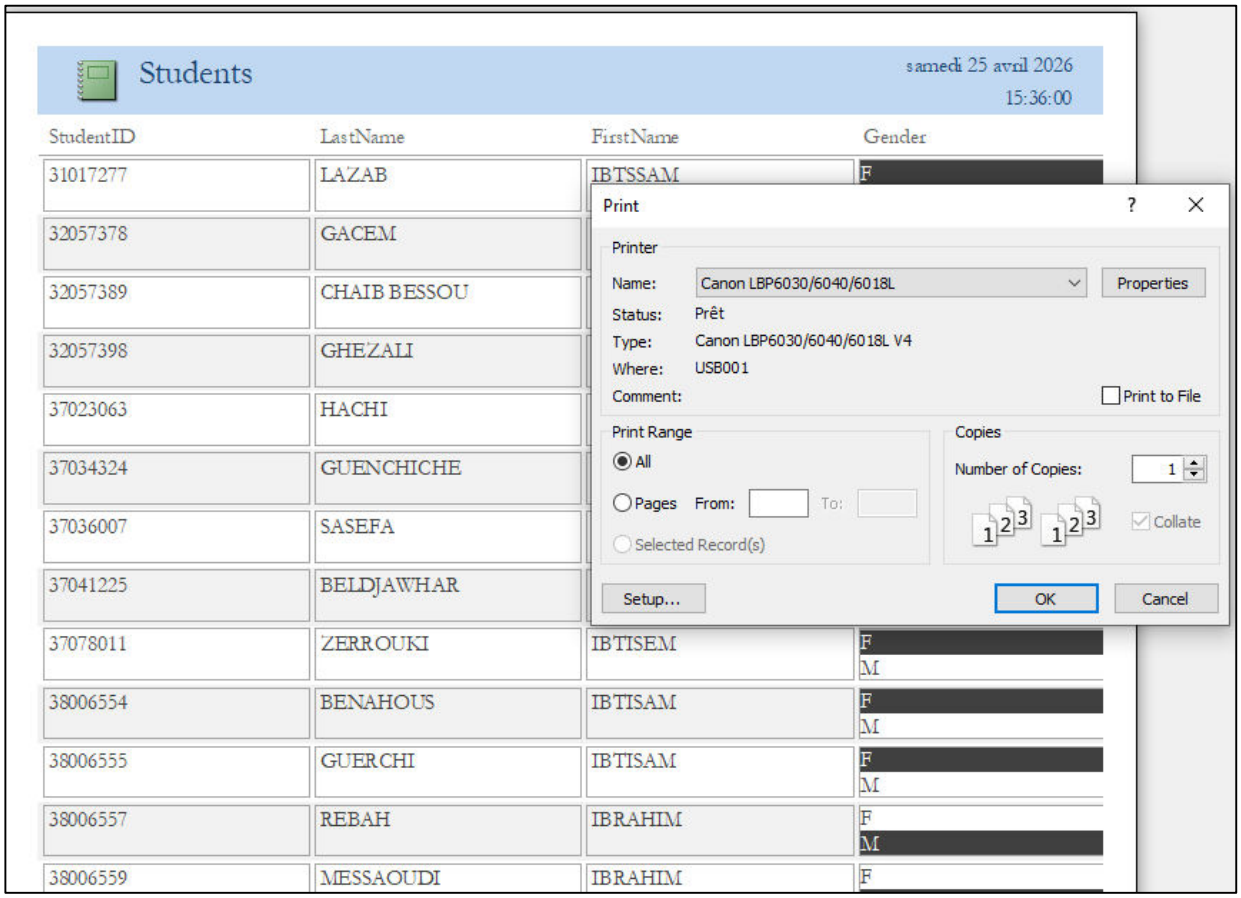
Example:

- Sort students by **LastName (Ascending)**
- Group data by **City or Class (if available)**

6. Printing the Report

1. Click **File → Print**
2. Select printer or PDF export option
3. Confirm print settings

Figure 20: Print settings window



4. Questions

- What is the main difference between a form and a report?
- Why are reports important in business decision-making?
- How does grouping data improve the readability of a report?

Session VII:
Introduction to Macros and VBA Automation

This session introduces the concept of automation in database systems. Students will learn how to use macros to automate routine tasks and will be exposed to the basic principles of VBA within Microsoft Access. The session highlights the role of automation in improving efficiency and enhancing database functionality.

1. Objectives

This session aims to introduce the concepts of automation in database systems using macros and basic Visual Basic for Applications (VBA). By the end of this session, students should be able to:

- Understand the role of automation in database applications
- Create simple macros using Microsoft Access
- Execute actions automatically using predefined commands
- Understand the basic purpose of VBA in database customization
- Differentiate between macros and VBA procedures

1. Theoretical Background

In relational database systems, repetitive tasks such as opening forms, running queries, or generating reports can be automated in order to improve efficiency and reduce user intervention. This automation is achieved through **macros** and **Visual Basic for Applications (VBA)**.

A **macro** is a sequence of predefined actions that are executed automatically in response to a specific event, such as clicking a button or opening a form. Macros are generally used for simple automation tasks and do not require programming knowledge.

VBA (Visual Basic for Applications) is a programming language integrated into Microsoft Access that allows users to create more advanced and flexible automation procedures. Unlike macros, VBA enables conditional logic, loops, and complex data manipulation.

In business applications, automation is essential for:

- Improving user efficiency
- Reducing manual errors
- Standardizing repetitive processes
- Enhancing application interactivity

3. Practical Procedure

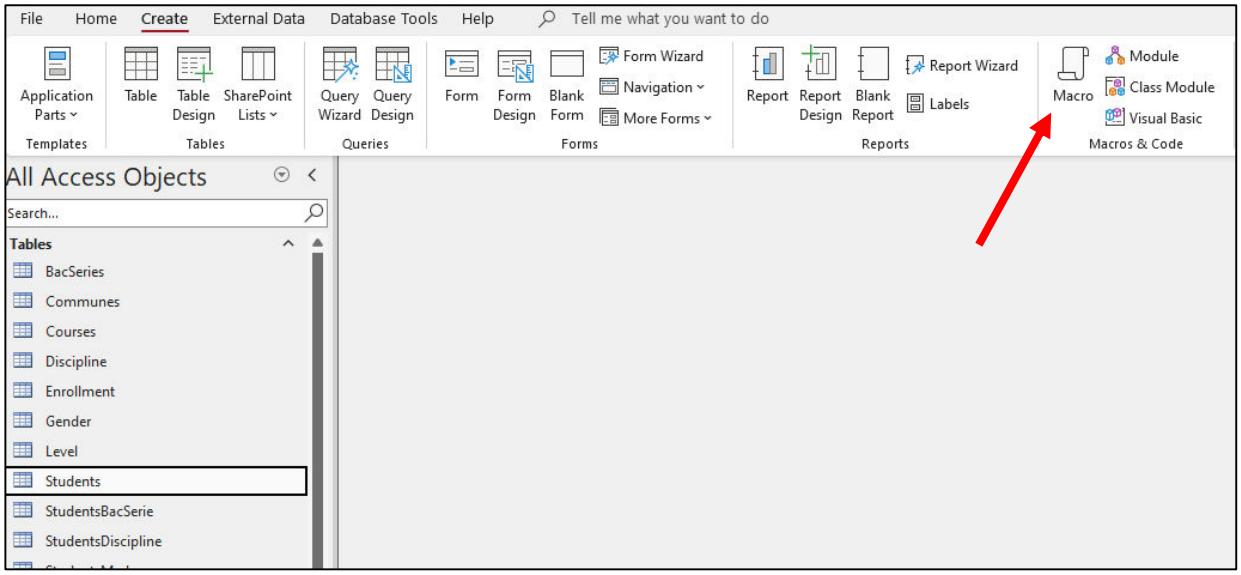
The following steps should be carried out using the existing *StudentDB* database.

Step 1: Creating a Simple Macro

1. Open Microsoft Access
2. Open the database *StudentDB.accdb*

3. Navigate to the **Create** tab
4. Click on **Macro**

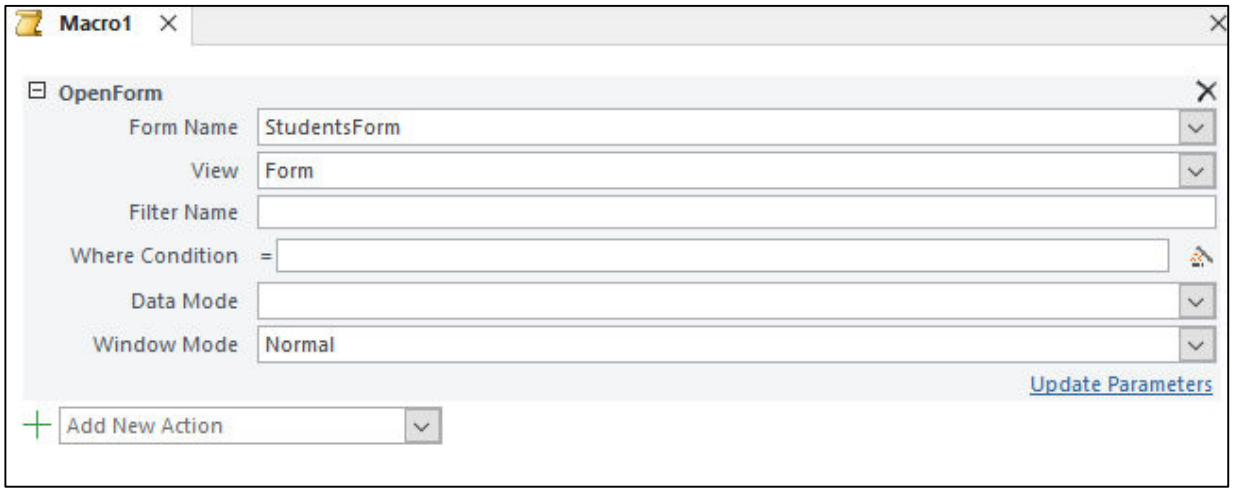
Figure 21: Creating a new macro



Step 2: Defining Macro Actions

1. In the macro design window, select an action such as:
    - **OpenForm**
2. In the action arguments:
    - Form Name: *Students*
    - View: Form

Figure 22: Macro design view



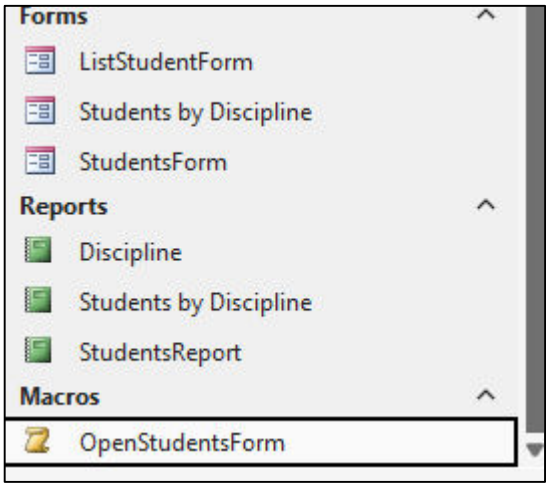
Step 3: Saving the Macro

1. Save the macro with the name:
   OpenStudentsForm
2. Close the macro editor

Step 4: Running the Macro

1. Double-click the macro in the Navigation Pane
2. The *Students* form should open automatically

Figure 23: Executing a macro

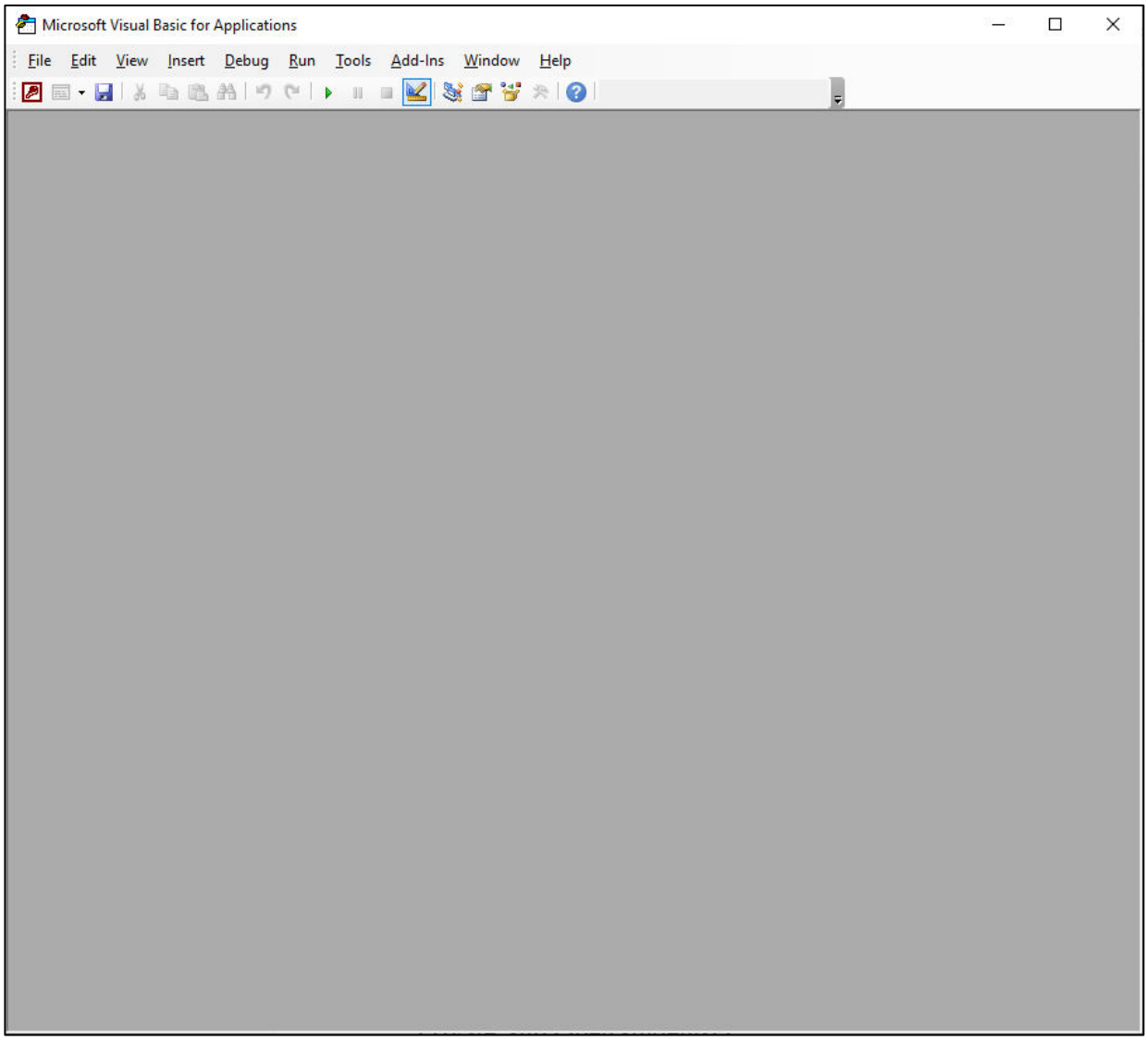


4. Introduction to VBA (Conceptual Overview)

Step 5: Opening the VBA Editor

1. Press **Alt + F11**
2. The VBA editor window opens

Figure 24: VBA editor interface



Step 6: Basic VBA Example (Conceptual)

A simple VBA procedure may look like:

```
Private Sub OpenStudents()
    DoCmd.OpenForm "Students"
End Sub
```

This code instructs the system to open the *Students* form automatically.

5. Comparison: Macros vs VBA

Table 2: Macros vs VBA

Macros	VBA
No programming required	Requires programming knowledge
Simple automation tasks	Complex logic and conditions
Limited flexibility	Highly flexible and powerful

6. Questions

- Why is automation important in database applications?
- What are the limitations of macros compared to VBA?
- In which situations would VBA be preferred over macros?

Conclusion

This practical workbook has provided a comprehensive introduction to relational database management systems through the use of Microsoft Access. By progressing through structured sessions, students have been guided from the fundamental concepts of databases to more advanced functionalities such as queries, forms, reports, and basic automation.

At the end of this training, students are expected to have acquired a solid understanding of how data is structured, managed, and exploited within a relational database environment. They should also be capable of designing a complete and functional database system adapted to basic business needs, particularly in the context of student or organizational data management.

Beyond technical skills, this workbook also aims to develop analytical thinking by encouraging students to understand the logic behind data organization and the importance of information systems in decision-making processes.

Finally, this introduction serves as a foundation for more advanced studies in database design, information systems, and business intelligence. Mastery of these concepts will enable students to effectively engage with modern digital tools and contribute to data-oriented environments in their future professional careers.