

TOPIC 12

Database Management Systems (Microsoft Access)

Senior Three — ICT

By Emmanuel Gidudu

Connects to: Topics 3, 5 & 9

What You Should Already Know

Before starting this topic, confirm you have the following prior knowledge:

Spreadsheets (Topics 5 & 9)

Understanding data in tables: rows = records, columns = fields. Experience filtering and sorting data.

File Management (Topic 3)

Saving, opening, and organising data files on a computer.

ICT and Organisations (Topic 1)

Awareness of how organisations use ICT to store and manage records.

Basic Computer Skills

Familiarity with the Microsoft Office interface and navigating Windows.



If you are unsure about any of the above, review the earlier topics before proceeding.

What is Data? What is Information?



Data

- Raw, unprocessed facts entered into the computer
- Example: unarranged names, random numbers
- On its own, data has no meaning
- Data is the INPUT to the computer system



Information

- Data that has been processed, organised, and given meaning
- Example: sorted student list, calculated averages
- Information is meaningful to the final user
- Information is the OUTPUT from the computer system

Qualities of Valuable Information:

- Accurate — free from errors and correct | Timely — available when needed
- Organised — easy for the end user to understand | Verifiable — can be checked
- Useful — serves the purpose of the user | Cost-effective — worth the cost to produce
- Accessible — available to every authorised user

 *GIGO — Garbage In, Garbage Out: the quality of output depends entirely on the quality of input data.*

Hierarchical Organisation of Data

Data Hierarchy — From Smallest to Largest

Level	Description	Example
Character	The smallest unit — a single letter, digit or symbol	A, 3, @
Field	A combination of characters; smallest unit a user can access	STUDENT-NAME = 'EMMANUEL'
Record	A group of related fields about one entity	All details of one student
File / Table	A collection of related records stored together	All students in Senior Three
Database	A collection of logically related files/tables	The entire school database
Databank	An enormous collection of two or more	National education databank

 *Memory aid: Character → Field → Record → File → Database → Databank (smallest to largest)*

Key Field Concepts:

- Field Name: Uniquely identifies each field (e.g. STUDENT-ID, STUDENT-NAME)
- Field Size: Maximum number of characters a field can contain

Key Database Definitions

Essential Vocabulary

Term	Definition
Database	A collection of logically related data designed to meet information needs of an organisation
Primary Key	A unique identifier for each record in a table — no two records can share the same value
Foreign Key	A copy of the primary key from another table — used to create relationships
Record	A complete set of data items in one row of a table (all details of one entity)
Field	A single column of a table containing one type of data for all records
Attribute	A group of fields/columns in a table (used in relational database terminology)
View	A virtual table derived from one or more base tables — does not store data itself
Macro	An automated procedure/action in a database application
Data Integrity	The degree to which data is correct — both accurate (free from errors) and consistent

What is a Database Management System (DBMS)?



Definition

A Database Management System (DBMS) is software that allows users to create, store, organise, retrieve, and manage data in a database efficiently and securely.

Key Functions of a DBMS:

Data Storage

Stores large amounts of data in an organised, structured way

Data Retrieval

Allows quick, efficient access to needed information using queries

Data Security

Protects data from unauthorised access using passwords and privileges

Data Integrity

Ensures data is accurate, consistent, and free from conflicts

Backup & Recovery

Helps restore data in case of failure, corruption or accidental deletion

Advantages of Database Management Systems

Advantage	Explanation
Sharing of data	Data is easily shared among different users and applications simultaneously
Data persistence	Data exists beyond the scope of the process it was created for
Data security	Protected from unauthorised access using passwords, roles and recovery facilities
Data validity & integrity	Correctness and consistency: free from errors and no conflicts between related data
Large data storage	Capable of storing enormous amounts of data for personal and organisational use
Non-redundancy	Eliminates duplication of data — no two records represent the same real-world entity
Data independence	Data and user programs can be altered independently of each other
Consistency	The system always produces consistent values with respect to defined relationships

Types of Databases & Popular DBMS Software

Flat Database

- Consists of ONE table only
- Simple structure — like a spreadsheet
- Limited; cannot handle complex data relationships
- Example: A single contact list

Relational Database

- Consists of TWO OR MORE related tables
- Tables are linked through common fields (keys)
- Avoids data duplication through normalisation
- Example: Microsoft Access, MySQL, Oracle

Popular DBMS Software

Software	Key Features	Typical Use
Microsoft Access	User-friendly, forms/reports, part of MS Office	Schools, small businesses
MySQL	Free, open-source, very fast, web-based	Websites, large web apps
Oracle Database	Enterprise-level, high performance	Banks, governments
SQLite	Lightweight, embedded, no server needed	Mobile apps, small projects
PostgreSQL	Open-source, advanced features, reliable	Scientific apps, enterprises

Uses of DBMS in Real Life

Schools

Student records, library catalogue, staff information, exam marks

Hospitals

Patient details, appointment booking, medical history, prescriptions

Banks

Account records, transaction history, customer data, loan management

Airlines

Booking systems, passenger manifests, flight schedules, seat allocation

Government

National ID records, tax records, voter registration, land registry

E-Libraries

Book catalogue, borrowing records, member accounts, late fees

Personal

Family records, personal contacts, home inventory, budgets

Shops

Inventory management, sales records, supplier details, pricing

👉 Class Discussion (5 minutes in pairs)

1. If your school has 1,200 students and 5 teachers record marks on paper, what problems might arise?
2. How would a DBMS solve these problems? Name 3 specific benefits.
3. Name 3 databases you personally interact with every day (think apps, shops, transport).
4. What data do you think is stored when you register for a national exam?

Database Objects in Microsoft Access

Object	Purpose	Analogy
Tables	Store the actual data in rows (records) and columns (fields)	Like a spreadsheet grid
Queries	Retrieve specific data from tables based on conditions/criteria	Like asking a question
Forms	Provide a user-friendly interface for data entry — one record at a time	Like a paper form
Reports	Display data in a formatted, printable layout with totals/grouping	Like a printed document
Macros	Automate repetitive tasks without programming	Like a recorded keyboard shortcut

 **REMEMBER:** Tables store data. Queries ask questions. Forms enter data. Reports present data.

Views in Access:

- Datasheet View: Displays data in rows and columns — used to view, enter and edit records
- Design View: Used to CREATE and modify the structure of tables, forms, queries and reports

Creating a Database in Microsoft Access

Step-by-Step: Creating and Saving a New Database

- Step 1: Open Microsoft Access (Start Menu → Microsoft Office → Access)
- Step 2: On the Start screen, click 'Blank Desktop Database'
- Step 3: Type a file name (e.g. SchoolDB.accdb) and choose a save location
- Step 4: Click 'Create' — Access opens with a blank Table1 in Datasheet View

Alternative: Creating from a Template

- A template is a pre-built database with ready-made tables, forms, reports and queries
- Click File → New → search for a template (e.g. 'Students', 'Contacts', 'Northwind')
- Select a template, enter a file name → click Create
- The database opens ready to use — customise it to meet your specific needs

 *The default file format for Access is .accdb (Access Database). Older versions used .mdb format.*

Creating a Table in Design View

Step-by-Step: Table Design View

- Step 1: Click the Create tab → in the Tables group → click Table Design
- Step 2: In Design View, enter the Field Name, Data Type, and Description for each field
- Step 3: To set a Primary Key — right-click the field name → select Primary Key (a key icon appears)
- Step 4: Save the table (Ctrl+S) — a dialog box asks for the table name — type it and click OK
- Step 5: Switch to Datasheet View (Home → View → Datasheet View) to start entering records

Example: STUDENT-TABLE Design

Field Name	Data Type	Description
STUDENT-ID	AutoNumber	Primary Key — unique student identifier (set as primary key)
STUDENT-NAME	Short Text	Stores the full name of the student
STUDENT-AGE	Number	Stores the age of the student
REG-DATE	Date/Time	The date the student registered

Data Types in Microsoft Access

Data Type	Used For	Example Fields
Short Text	Text up to 255 characters	Name, City, Email, Gender
Long Text (Memo)	Long blocks of text — paragraphs	Comments, Descriptions, Notes
Number	Numeric values used for calculations	Score, Age, Quantity, Price
Date/Time	Dates and times	DateOfBirth, EnrollmentDate, RegDate
Currency	Monetary values — avoids rounding errors	Fees, Salary, Price
AutoNumber	Automatic unique numbers — ideal for Primary Key	StudentID, OrderID
Yes/No	Boolean: only two values (True/False)	IsActive, HasPaid, IsEnrolled
Attachment	File attachments (images, documents)	StudentPhoto, Certificate
Hyperlink	Web links / URLs	SchoolWebsite, EmailAddress



EXAM TIP: Use AutoNumber for Primary Key fields. Use Short Text for names (not Number). Use Date/Time for all dates.

Field Properties

Field properties are characteristics that define how data is stored, displayed, and validated in each field. Set them in the lower panel of Design View.

Property	Description	Example
Field Size	Maximum number of characters allowed in the field	50 for a name field
Format	How the field value is displayed/printed	Short Date → 01/01/2024
Decimal Places	Number of decimal places for numeric fields	2 → displays 45.75
Input Mask	Forces data entry into a specific pattern/format	99/99/9999 → DD/MM/YYYY
Caption	Label text shown on forms and reports instead of field name	Date of Birth
Default Value	Value automatically inserted before user types	=Date() for today's date
Validation Rule	Expression that must be true when entering data	>=5 And <=30 for age
Validation Text	Error message displayed when validation rule is broken	Age must be between 5 and 30
Required	Whether a value must be entered (Yes/No)	Yes for compulsory fields

Validation Rules, Validation Text & Input Masks

What is Validation?

- Validation is the process of checking input data against predefined rules before it is accepted
- It prevents incorrect/invalid data from being stored in the database (data integrity)
- To set: Design View → click the field → Field Properties panel → Validation Rule + Validation Text

Validation Rule Examples

Field	Validation Rule	Validation Text
STUDENT-AGE	≥ 5 And ≤ 30	Age must be between 5 and 30
Score	≥ 0 And ≤ 100	Score must be between 0 and 100
Gender	"Male" Or "Female"	Please enter Male or Female
EnrollmentYear	≥ 2000 And ≤ 2030	Please enter a valid year

Input Masks — Controlling Data Format:

- An Input Mask controls the FORMAT of data entered, guiding the user to type data correctly
- Phone: (999) 000-0000 → Forces (256) 701-234567 format

Types of Validity Checks

Validity Check	Description	Example
Presence/Completeness Check	Verifies that a required field contains data	Student name must not be blank
Range Check	Determines whether a number falls within a specified range	Age between 5 and 30
Alphabetic/Numeric Check	Ensures the correct type of data is entered	Name field contains only letters
Format Check	Checks that data matches a required pattern/format	Date in DD/MM/YYYY format
Consistency Check	Checks logical relationships between two or more fields	End date after start date
Check Digit	Appended digit/character to verify accuracy of a key value	Barcode verification digit

➡ Quick Check — Which validation check applies?

1. A student enters their age as '200'. Which check would catch this?
2. A teacher types 'abc' into the Score field. Which check applies?
3. A date field is left empty. Which check ensures it must be filled?

Entering Data into a Table

Method 1: Directly in Datasheet View

- Open the table in Datasheet View (double-click it in the Navigation Pane)
- Home tab → Records group → click New (blank record), or click the asterisk (*) row
- Click the first field and type your data — press Tab to move to the next field
- Press Enter or move to a new row — Access saves the record automatically

Method 2: Using a Form (Recommended for Data Entry)

- Forms display one record at a time in a clean, user-friendly layout
- Create a form: Create tab → Forms group → click Form (Access creates it automatically)
- Switch to Form View (bottom-right corner icon) to start entering data
- Click the New Record button (Ctrl++) to add a new record
- Fill in each field → press Tab to move forward → press Enter to save
- Navigate records using the arrow buttons at the bottom of the form
- Close the form and reopen the table to confirm new records were saved

Sample Table Design: Students Table

Well-Designed Students Table

Field Name	Data Type	Properties / Notes
StudentID	AutoNumber	Primary Key — unique identifier (set as primary key)
FirstName	Short Text	Field size: 50 Required: Yes
LastName	Short Text	Field size: 50 Required: Yes
DateOfBirth	Date/Time	Format: Short Date Input Mask: 99/99/9999
Gender	Short Text	Field size: 10 Validation: 'Male' Or 'Female'
Class	Short Text	Field size: 10 Example: '3A', '3B', 'S3'
EnrollmentDate	Date/Time	Default value: =Date() (today's date)
IsActive	Yes/No	Default: Yes — indicates current student



EXAM TIP: Every table must have a Primary Key. Use AutoNumber unless the data itself is unique (e.g. NIN, Student Number).

Queries — Introduction



What is a Query?

A query extracts specific records or fields from one or more tables based on criteria you specify. Queries are the primary way to 'ask questions' of your database.

Step-by-Step: Creating a Query in Design View

- Step 1: Create tab → Queries group → click Query Design
- Step 2: In the 'Show Table' dialog — select the table(s) to query → click Add → Close
- Step 3: Double-click (or drag) the fields you want from the table box to the query grid below
- Step 4: In the 'Criteria' row under a field, type your condition (e.g. >18 for age)
- Step 5: Click Run (the red exclamation mark '!') to see the results
- Step 6: Save the query (Ctrl+S) and give it a descriptive name (e.g. qryAbove18)



TIP: Use the 'Sort' row in the query grid to sort results (Ascending/Descending). Use the 'Show' checkbox to hide a field from results while still using it as a filter.

Query Operators & Criteria

Common Query Operators

Operator	Meaning	Example Criteria	Returns...
=	Equal to	= 'Form 3A' or = 80	Records exactly matching the value
<>	Not equal to	<> 'Male'	All records except 'Male'
< and <=	Less than / Less than or equal to	< 50 or <= 40	Records below/at the value
> and >=	Greater than / Greater than or equal to	> 70 or >= 60	Records above/at the value
Between	Between two values (inclusive)	Between 40 And 80	Records in the range 40–80
Like	Pattern match using wildcards (*)	Like 'A*'	Names starting with letter A
In	Value is in a specified list	In ('Male','Female')	Records matching any listed value
Is Null	Field is empty / no data entered	Is Null	Records where field is blank
Is Not Null	Field has a value	Is Not Null	Records where field is not blank



In Access, text values in criteria must be in quotes: 'Male'. Numbers are entered without quotes: >70. Dates use #: >#01/01/2024#.

Query Criteria — Detailed Examples

Text Field Criteria

To Find Records That...	Use This Criterion
Exactly match 'China'	"China"
Do NOT match 'Mexico'	Not "Mexico"
Start with letter 'U'	Like "U*"
Contain the word 'Korea'	Like "*Korea*"
End with 'ina' (e.g. China, Argentina)	Like "*ina"
Are blank / empty	Is Null
Are not blank	Is Not Null
Match USA or UK	"USA" Or "UK"
Match a list of values	In("France", "China", "Germany")

Date Field Criteria

To Find Records That...	Use This Criterion
-------------------------	--------------------

Exercise 19 — Queries Practice



Database Setup

Use a Students database with fields: StudentID, FirstName, LastName, Class, MathsScore, EnglishScore, Gender, DateOfBirth

Exercise 4.1 — Create These Queries

1. Create a query that shows ALL students in Class '3A'. Name it: qryClass3A
2. Create a query showing students with a MathsScore GREATER THAN 70. Name it: qryHighMaths
3. Create a query showing FEMALE students with an EnglishScore BETWEEN 50 and 80. Name it: qryFemaleEnglish
4. Create a query showing students whose FirstName BEGINS WITH 'A' (use Like 'A*'). Name it: qryNamesA
5. Create a query showing students where DateOfBirth IS NOT NULL (all with a recorded birth date). Name it: qryWithDOB
6. CHALLENGE: Create a query showing students where MathsScore + EnglishScore > 140 (combined score)

Reports in Microsoft Access

Reports display data from tables or queries in a formatted, printable layout. They can include grouping, sorting, totals, and calculated fields.

Step-by-Step: Creating a Report Using Report Wizard

- Step 1: In the Navigation Pane, click the table or query to base the report on
- Step 2: Create tab → Reports group → click Report Wizard (or click Report for a quick report)
- Step 3: Select the fields to include using the '>' button (one field) or '>>' (all fields) → click Next
- Step 4: Choose grouping levels (e.g., group by Class) → click Next
- Step 5: Choose a sort order (e.g., sort by STUDENT-NAME Ascending) → click Next
- Step 6: Choose a layout (Tabular/Columnar/Justified) and orientation (Portrait/Landscape) → Next
- Step 7: Enter a title for the report → click Finish
- Step 8: The report opens in Print Preview — switch to Design View for further adjustments

Automatic Calculations in Reports (Design View):

- Add a Text Box in the Report Footer section and type a formula:
- Count: =Count([STUDENT-ID]) Average: =Avg([MathsScore]) Sum: =Sum([MathsScore])

Relational Databases — Two Tables

A relational database links two or more tables through a common field — usually a Primary Key in one table and a Foreign Key in another. This avoids repeating data (redundancy) and ensures data integrity.

Data Terminology (Three Perspectives)

File Processing Environment	Relational Database Developer	Relational Database User
File	Relation	Table
Record	Tuple	Row
Field	Attribute	Column

Rules for a Good Relational Database:

- Each row (record) must have a Primary Key that uniquely identifies that record
- Each column (field) must have a unique, descriptive name
- Each fact should be stored only once — no duplication across tables
- Related tables are linked through common key fields (Primary Key ↔ Foreign Key)



Goal of database design: Remove data redundancy by dividing data into subject-based tables, then link them through

Types of Table Relationships

Three Types of Relationships

One-to-One (1:1)

Each record in Table A matches exactly ONE record in Table B.

Examples: One NIN → One citizen. One vehicle ID → One vehicle. One president → One country.

One-to-Many (1:M)

One record in Table A can match MANY records in Table B.

Examples: One customer → many orders. One teacher → many students. One class → many students.

Many-to-Many (M:M)

Many records in Table A match many records in Table B.

Requires a JUNCTION TABLE (third table) to break it into two 1:M relationships. Example: Students and Subjects.



One-to-Many is the most common relationship. Many-to-Many always needs a junction/linking table in between.

Creating Table Relationships in Access

Step-by-Step: Creating a Relationship

- Step 1: Go to Database Tools tab → Relationships group → click Relationships
- Step 2: In the 'Show Table' dialog — add both tables → click Add → click Close
- Step 3: Drag the Primary Key field from the 'one' table onto the Foreign Key field in the 'many' table
- Step 4: The 'Edit Relationships' dialog appears — check 'Enforce Referential Integrity'
- Step 5: Optionally check 'Cascade Update Related Fields' and 'Cascade Delete Related Records'
- Step 6: Click Create — a line appears between the two tables (with 1 and ∞ symbols)
- Step 7: Close the Relationships window → click Yes to save

Why Create Relationships?

- Enables queries across multiple tables
- Informs form and report designs
- Prevents 'orphan records'
- Ensures consistent data across tables

Referential Integrity

- Prevents orphan records (orders without customers)
- Cascade Update: updates all related records
- Cascade Delete: deletes all related records
- Access enforces integrity automatically

Relationship Design Example: Students & Marks

Table 1: tblStudents

Field Name	Data Type	Notes
StudentID	AutoNumber	Primary Key (PK)
FirstName	Short Text	Field size: 50
LastName	Short Text	Field size: 50
Class	Short Text	e.g. '3A', 'S3'
Gender	Short Text	Validation: 'Male' Or 'Female'

Table 2: tblMarks

Field Name	Data Type	Notes
MarkID	AutoNumber	Primary Key (PK)
StudentID	Number	Foreign Key (FK) — links to tblStudents.StudentID
Subject	Short Text	e.g. Mathematics, English
Score	Number	Validation: ≥ 0 And ≤ 100

Exercise 20 — Relational Database Practice



Task: Create a Library Database

Create a 2-table relational database called LibraryDB.accdb for a school library system.

Exercise 4.2 — Step-by-Step Tasks

1. Create Table 1: tblMembers — fields: MemberID (AutoNumber, PK), FirstName, LastName, Class, JoinDate
2. Create Table 2: tblBorrowed — fields: BorrowID (AutoNumber, PK), MemberID (FK), BookTitle, Author, BorrowDate, ReturnDate
3. Add validation rule: BorrowDate must be $\geq$ #01/01/2024# with appropriate validation text
4. Create a relationship between tblMembers and tblBorrowed via MemberID — enforce referential integrity
5. Create a form for data entry into tblMembers and enter at least 5 member records
6. Enter at least 8 borrowing records using the form
7. Create a query showing all books borrowed by a specific member (use criteria on MemberID)
8. Create a report listing all overdue books (ReturnDate < Date())



Save as: LibraryDB_YourName.accdb

Homework Assignment 5 — Final Database Project

Create a comprehensive database for a fictional scenario of your choice:

Option A — Hospital

Patient management system: patient records, doctors, appointments and diagnoses

Option B — Library

School library system: books, members, borrowing records and late fees

Option C — Shop

Small shop inventory and sales system: products, customers, orders and suppliers

All Options Must Include:

- At least 2 related tables with primary and foreign keys
- Appropriate data types for all fields | At least 2 validation rules with validation text
- At least 1 input mask | A form for data entry | At least 10 records in each main table
- At least 5 different queries using various operators (=, >, Between, Like, Is Null, etc.)
- A formatted report with at least one calculated field (Count, Sum, or Average)



Also write a 1-page report explaining your database design choices. Save as: FinalDB_YourName.accdb. Due: End of term.

Scenario Questions — Think & Discuss

Scenario 1

A school currently keeps student records in exercise books. They want to move to a digital database. What fields would you include in a Students table? What data types would you assign to each field?

Scenario 2

A query returns no results even though you are sure the data exists. What are three possible causes and how would you troubleshoot each?

Scenario 3

A hospital wants to store patient records and also track which doctor treated each patient. Design a 2-table relational database with appropriate fields and describe the relationship between the two tables.

Scenario 4

Why is it important to set validation rules in a database? Give two examples of problems that could occur if validation rules are NOT used.

Topic 12 — Summary

What We Covered in This Topic

Section	Key Learning Points
Database Basics	Data vs information, data hierarchy, GIGO, data integrity, key definitions
DBMS Concepts	Functions, advantages, types (flat vs relational), popular software
Real-Life Uses	Schools, hospitals, banks, airlines, government, libraries
MS Access Interface	Database objects: Tables, Queries, Forms, Reports, Macros, Modules
Creating Tables	Design View, field names, data types, primary key, field properties
Data Entry	Datasheet View, forms, navigation buttons, saving records
Validation	Validation rules, validation text, input masks, types of validity checks
Queries	Query Design, operators (=, <>, Between, Like, In, Is Null), running queries
Reports	Report Wizard, grouping, sorting, calculated fields
Relationships	1:1, 1:M, M:M, creating relationships, referential integrity

Final Revision Questions