

Knowledge organiser: Database development *Build and interrogate your own software system*

Summary

A **database** is a way of storing information in an organised, logical way. **Validation and verification** are two ways to check that the data entered into a computer is correct. Data entered incorrectly is of little use.

There are two main methods of verification:

Double entry - entering the data twice and comparing the two copies. This effectively doubles the workload, and as most people are paid by the hour, it costs more too.

Proofreading data - this method involves someone checking the data entered against the original document. This is also time-consuming and costly.

Validation is an automatic computer check to ensure that the data entered is sensible and reasonable. It does not check the accuracy of data.

Key Vocabulary

Criteria	A set of rules or conditions that must be met. Often used in searches.
Database	A data store designed in an organised way, making it easier to search for the information you need.
Field	An element of a database record in which one piece of information is stored. For example 'name' in an electronic address book.
Front-end	The part of an application seen and used by the end user.
Flat-file database	A database in which all the data is stored in a single table is known as a flat file database.
Key Field	A unique identifier for a database record or table entry.
Multi-Access	A system that can be used by several users simultaneously via a local area network (LAN).
Query	A search or question performed inside a database.
Record	All of the data relating to one entity in a database.
Validation	Checking input data is sensible and in the right format.
Verification	Verification is performed to ensure that the data entered exactly matches the original source.

Relational databases

A relational **database** has more than one table and the tables are linked using **key fields**. For example, a library database could have three tables:

Customer - when a customer joins the library a **record** is created. It stores their details such as their first name and surname and includes a unique Customer ID.

Book - each book in the library has a record. It stores details about the book, such as the author and title and includes a unique book ID.

Lending - when a customer borrows a book, the lending table stores the customer's unique ID and the book's unique ID in a record. The record could also include additional information such as when the book was borrowed and when it's due back.

Why use a database?

- ◆ Databases can store very large numbers of records efficiently (they take up little space).
- ◆ It is very quick and easy to find information.
- ◆ It is easy to add new data and to edit or delete old data.
- ◆ Data can be searched easily, e.g. 'find all Ford cars'.
- ◆ Data can be sorted easily, for example into 'date first registered' order.
- ◆ Data can be imported into other applications, for example a mail-merge letter to a customer saying that an MOT test is due.
- ◆ More than one person can access the same database at the same time - multi-access.

Validation

For example, a secondary school student is likely to be aged between 11 and 16. The computer can be programmed only to accept numbers between 11 and 16. This is a **range check**.

Types of validation

There are a number of validation types that can be used to check the data that is being entered.

- ◆ Spell check

Data capture

Before setting up a database the data must be collected. This can be done using a data capture form.

A data capture form is designed to collect specific data.

Boxes

Set amount of spaces

Data capture forms often use **boxes** or a **set amount of spaces** and occasionally provide examples too.

This is to make sure each field is completed correctly.

