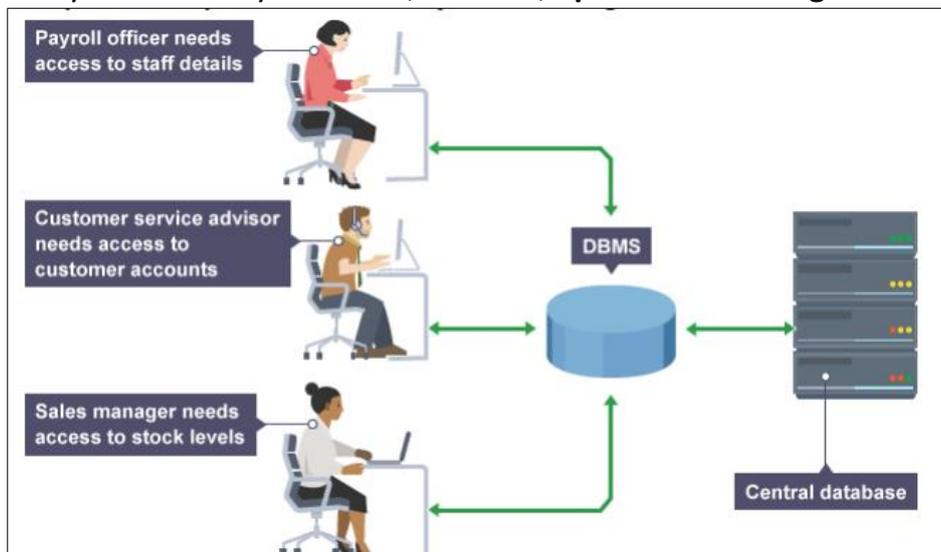


Objective:

- Show understanding of features provided by Database Management System (DBMS) that address issues of a file based approach;
Including: • data management, including maintaining a data dictionary • data modelling • logical schema • data integrity • data security, including backup procedures and the use of access rights to individuals / groups of users.
- Show understanding of how software tools found within DBMS are used in practice.
Including use and purpose of: • Developer interface • Query processor

What is DBMS?

Database management system (DBMS) is a **computer software application** that interacts with user, other applications, and database itself to **capture** and **analyze data**. DBMS provides users and programmers with a systematic way to **create, retrieve, update and manage data**.



How a DBMS addresses the limitations of a file based approach?

Data Redundancy Issue

This is solved by storing data in separate **linked tables**, which reduces **duplication** of data. Items of data used to link tables by the use of **foreign keys**.

DBMS will **flag** errors when any attempt is made to accidentally delete this type of item.

Data Inconsistency Issue

This is solved by storing most items of data only **once**, allowing **updated items** to be seen by all applications. As data is not inconsistent, **integrity** of data stored is improved. Consistent data is **easier** to maintain as an item of data will only be changed once, not multiple times, by different applications.

Data Dependency Issue

Data is **independent** of applications using database, so changes made to the **structure** of the data will be managed by **DBMS** and have no effect on the applications using the database.

Any **fields or tables** added to or removed from database will **not affect** applications that do not use those fields/tables, as each application only has **access** to the fields/tables it requires.

Information from a database is more **easily** available in a form that is required so it is **not dependent** on the structure of the data and the **application** used.

DBMS includes facilities to query data stored using a defined **query language** or a **query-by-example** facility.

DBMS Approach

DBMS uses **structured approach** to management, organisation and maintenance of data in a database.

- ☞ Entry of new data, storage of data, alteration and deletion of data are all managed by DBMS.
- ☞ DBMS uses **Data Dictionary** to store **metadata** for a database, including definition of tables, attributes name, data types, relationships between tables and any indexing. **Data dictionary** can define **validation rules** used for entry of data and contain data about physical storage of data.
Use of data dictionary improves **integrity** of data stored, helping to ensure that it is **accurate, complete** and **consistent**.
- ☞ **Data Modelling Tool** is used to show **data structure** of database. **E-R** diagram is an example of a data model. A **logical schema** is a **data model** for a specific database that is independent of DBMS used to build the database. **Logical Schema** describes how **relationships** will be implemented in logical structure of database.
- ☞ DBMS helps to provide **Data Security** to prevent unwanted alteration, corruption, deletion or sharing of data with others that have no right to access it.

ESQ: How DBMS help us to make data secure?

- Using usernames and passwords to prevent unauthorised access to database.
- Using access rights to manage actions authorised users can take. Users could read / write / delete, or read only, or append only.
- Using access rights to manage the parts of the database they have access to.
- Automatic creation and scheduling of regular back-ups
- Encryption of the data stored.
- Automatic creation of an audit trail or activity log to record actions taken by users of database

Use and Purpose of DBMS

Developer interface:

9608/13/O/N/19

- To create user friendly features e.g. forms to enter new bookings
- To create outputs e.g. report of bookings on a given date
- To create interactive features such as buttons and menus that users can click on or interact with to perform different actions within the database system.
- Developer interface allows a developer to write queries in structured query language (SQL) rather than using query-by-example. These queries are then processed and executed by **query processor**.

Query Processor:

9618/M/J/23/P12

- Query Processor allows construction of more complex queries to interrogate database.
- Query Processor allows the user to enter criteria, then finds and returns the appropriate result.
- Query processor takes a query written in **SQL** and processes it.
- Query processor includes a **DDL interpreter**, a **DML compiler** and a **query evaluation engine**. Any **DDL** statements are interpreted and recorded in the database's data dictionary. DML statements are compiled into low level instructions that are executed by the query evaluation engine. DML compiler will also optimise the query.

Purpose of Query Processor in DBMS:

9618/O/N/22/P13

- Enables users to input criteria or conditions for retrieving specific data from the database.
- Conducts searches within the database based on the provided criteria to identify and retrieve relevant data.
- Structures and organizes the retrieved data into a format suitable for presentation to the user.

DBMS provides **data management**. This includes development of a **data dictionary** that stores information about the data stored, such as **field names** and **primary keys**. **Logical schema** uses methods such as an **E-R diagram** to show structure of the database and its relationships. **Query processor** allows a user to perform searches to find specific data. DBMS also provides a **developer interface** that allows user to create tables, forms and reports. **9618 P12 May22**