

Exercise walk-through

8.2

Database Management Systems (DBMS)

A-Level Computer Science

The DBMS

A **DBMS** manages the database centrally – **data** management and **data modelling** define the **logical schema**, and it also handles **backup** and recovery, per-user security, **transactions** that wholly succeed or wholly fail, a **query processor** to run queries, and a **developer interface** of tools and APIs:

data dictionary

concurrent access

transactions

views

backup & security

query processor

What a DBMS provides, and the problems it removes

the tools

a data-dictionary editor, a query builder, a forms builder, a report generator, user management, and an SQL editor

data redundancy

file-based storage repeats the same fact in several files; a database stores it once

data inconsistency

which is what happens when one copy is updated and another is not

data integrity

constraints and keys enforced by the DBMS itself, not by each program

data security

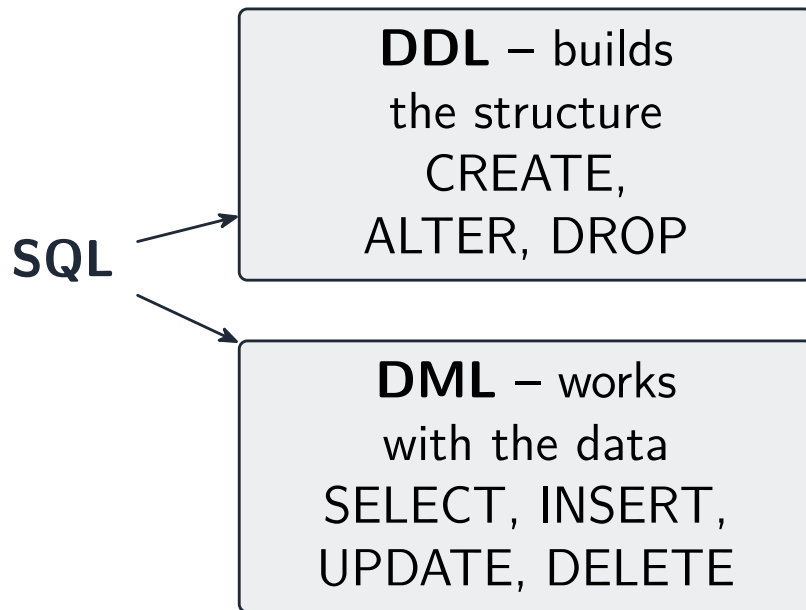
one place to set who may read and who may write

data independence

programs are written against the logical view, so the storage can change without rewriting them

Redundancy causes inconsistency, and inconsistency destroys integrity. The three arguments are one chain, and the exam expects it stated as one.

SQL – DDL and DML



DDL

```
CREATE TABLE CUSTOMER  
(CustomerID INTEGER PRIMARY KEY,  
Name VARCHAR(50) NOT NULL);
```

- **DDL** : structure
- **DML** : the data

SQL – querying data

SELECT

```
SELECT Name, Phone  
FROM CUSTOMER  
WHERE City = 'London'  
ORDER BY Name ASC;
```

A **join** combines tables on a foreign key; **aggregate functions** (COUNT, SUM) pair with GROUP BY.

```
SELECT name, grade  
FROM students  
WHERE grade > 80;
```

result

name	grade
Ann	88
Cara	91

Always put a WHERE on UPDATE/DELETE, or it hits every row.

SQL's two halves, and the exam's SQL rules

DDL

Data Definition Language – creates or changes the **structure**: CREATE TABLE, ALTER TABLE, keys and constraints

DML

Data Manipulation Language – works on the **data**: SELECT, INSERT, UPDATE, DELETE

the types

INTEGER, REAL, VARCHAR(n), CHAR(n), DATE, TIME, BOOLEAN, DECIMAL(p,s)

exact names

use the **table and field names from the question**, spelled as given

quoting

single quotes round strings ('Smith'); **do not quote numbers**

the operators

LIKE 'A%' matches anything starting with A (% any string, _ one character); IN (1,2,3); BETWEEN 10 AND 20

Combine conditions with AND / OR / NOT, and **end every statement with a semicolon**. Marks are lost on those details, not on the logic.

Exercises 2.1-2.2

2.1 State what a **data dictionary** holds.

[1]

data dictionary

2.2 State what a **transaction** is.

[1]

transaction

Exercises 2.1-2.2

2.1 State what a **data dictionary** holds.

[1]

data dictionary

2.1

- a description of every table, field, data type and key in the database (its structure)
-

2.2 State what a **transaction** is.

[1]

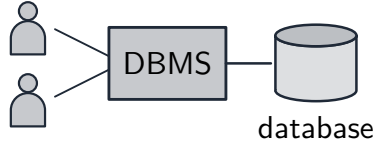
transaction

2.2

- a group of operations treated as one unit – they **all succeed or all fail**

Exercises 2.3·2.4

2.3 State **two** features a DBMS manages centrally for a shared database. [2]

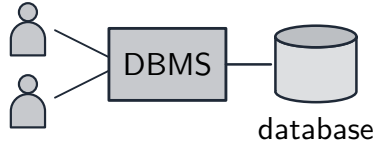


2.4 State what a **view** is. [1]

view

Exercises 2.3-2.4

2.3 State **two** features a DBMS manages centrally for a shared database. [2]



2.3

- any two of: data dictionary; concurrency control; transactions; central integrity rules; views; per-user security; backup/recovery

2.4 State what a **view** is. [1]

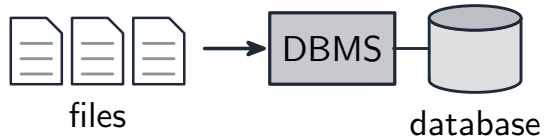
view

2.4

- a **virtual table** that shows a user a chosen slice of the data (selected rows/columns)

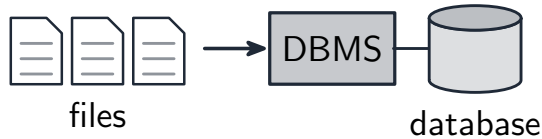
Exercise 3.1

3.1 Describe **two** features of a DBMS that solve problems of the file-based approach. [4]



Exercise 3.1

3.1 Describe **two** features of a DBMS that solve problems of the file-based approach. [4]

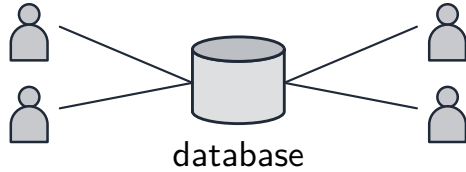


3.1

- any two, described, e.g. **central integrity rules** – keys and constraints enforced for all programs, so data stays valid
- **concurrency control** – locks/transactions let many users edit without corrupting data

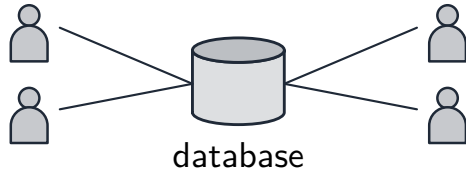
Exercise 3.2

3.2 Explain how a DBMS allows many users to work on the data at the same time safely. [2]



Exercise 3.2

3.2 Explain how a DBMS allows many users to work on the data at the same time safely. [2]



3.2

- the DBMS uses **locks** and **transactions**
- conflicting updates are controlled so they do not overwrite each other; a transaction completes fully or not at all

Exercises 3.3-3.4

3.3 Describe the roles of the **developer interface** and **query processor** when a programmer writes and runs a query. [4]

DBMS

3.4 Describe what is meant by the DBMS feature **data dictionary** and the DBMS feature **logical schema**. [2]

data dictionary

logical schema

Exercises 3.3-3.4

3.3 Describe the roles of the **developer interface** and **query processor** when a programmer writes and runs a query. [4]

DBMS

3.3

- Developer interface: facilities through which the developer creates or changes database objects, for example an SQL editor used to enter a query.
- Query processor: interprets/checks and plans the query, then executes it to retrieve matching data.

3.4 Describe what is meant by the DBMS feature **data dictionary** and the DBMS feature **logical schema**. [2]

data dictionary

logical schema

3.4

- **data dictionary:** data about the data – a description of every table, field, data type and key
- **logical schema:** the overall logical design of the database – its tables, fields, keys and relationships