Week 19

A-Level Computer Science

Homework bundle for this week

本周作业合集

exercises.lol

Contents 目录

8. Databases

Starter Quiz · 课前小测

A-Level Computer Science Name: _____ Score: _____

1. Storing a customer's address in several separate files leads to:

   ☐ data redundancy, which can cause inconsistency

   ☐ faster queries

   ☐ automatic backups

   ☐ better security

2. In an E-R diagram, the number of one entity that can relate to one of the other, marked at each end of the line, is the _____.

   _____

3. A data dictionary in a DBMS holds:

   ☐ a description of every table, field, type and key

   ☐ the actual customer records

   ☐ the user's passwords

   ☐ the program source code

4. Which is part of the Data Definition Language (DDL)?

   ☐ CREATE TABLE

   ☐ SELECT

   ☐ UPDATE

   ☐ INSERT

5. In a flat-file system the same data is often duplicated across files, which can become inconsistent when only one copy is updated.

   ☐ True ☐ False

6. Each customer can place many orders, but each order belongs to one customer. This relationship is:

   ☐ one-to-many (1:M)

   ☐ one-to-one (1:1)

   ☐ many-to-many (M:N)

   ☐ no relationship

7. The logical schema of a database describes:

   ☐ the tables, attributes and relationships, endently of how the data is physically stored

☐ where each record is stored on the disk

☐ the passwords of the users

☐ the SQL queries most often run

8. DDL defines the structure (e.g. CREATE TABLE), while DML works with the data inside it (SELECT, INSERT, UPDATE, DELETE).

☐ True ☐ False

9. Which are limitations of the file-based approach? Select **all** that apply.

☐ the same data is duplicated in several files

☐ copies of the data become inconsistent

☐ programs depend on the file layout, so a change means rewriting them

☐ there are too many tables to join

Tick every correct option.

10. Which are data-security features of a DBMS? Select **all** that apply.

☐ access rights given to individual users or groups

☐ backup procedures with a log of changes

☐ encryption of the stored data

☐ normalising the tables to 3NF

Tick every correct option.

8. Databases

Answers · 答案

A-Level Computer Science

1. **data redundancy, which can cause inconsistency**

   The same data held in many places (redundancy) can be updated separately and become inconsistent.

2. **cardinality**

   Cardinality is one or many at each end; crow's-foot notation draws a bar for one and a foot for many.

3. **a description of every table, field, type and key**

   The data dictionary describes the database structure, so programs query it rather than hard-coding the layout.

4. **CREATE TABLE**

   DDL changes the structure (CREATE, ALTER, DROP). SELECT/INSERT/UPDATE/DELETE are DML (working with data).

5. **True**

   That redundancy and the resulting inconsistency is the core problem the relational model solves.

6. **one-to-many (1:M)**

   One customer → many orders, each order → one customer: a one-to-many relationship.

7. **the tables, attributes and relationships, independently of how the data is physically stored**

   The logical design is separate from the physical storage, which is what lets the storage change without the programs noticing.

8. **True**

   Definition vs Manipulation: DDL shapes the tables; DML reads and changes the rows.

9. **the same data is duplicated in several files; copies of the data become inconsistent; programs depend on the file layout, so a change means rewriting them**

   Redundancy, inconsistency and data dependence are the three named limitations. Tables and joins belong to the relational approach that replaces it.

10. **access rights given to individual users or groups; backup procedures with a log of changes; encryption of the stored data**

    Who may access, and recovery after loss, are security. Normalisation is a design step against redundancy.

A-LEVEL COMPUTER SCIENCE • EXERCISE SHEET

8.1 Database Concepts

Name: _____ Class: _____ Date: _____ **Total: 22 marks**

KEY WORDS relational database 关系数据库 • entity 实体 • primary key 主键 • foreign key 外键 • integrity 完整性

Objective

Build the skills to answer exam questions on **database concepts** — why relational databases beat flat files, the relational terms, keys, relationships and **normalisation** 规范化.

You must be able to:

- state the limitations of a **file-based** approach and how a **relational database** 关系数据库 fixes them
- use the terms **table**, **record**, **field**, **primary key** 主键 and **foreign key** 外键
- read an **entity-relationship (E-R) diagram** and explain whether tables are in **3NF**

1 Worked examples

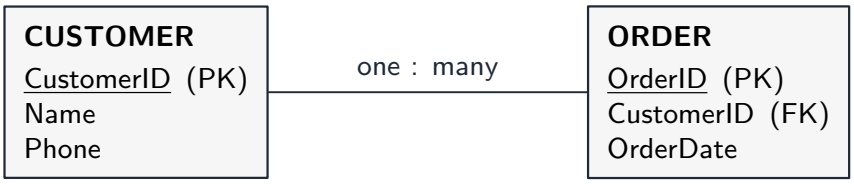
Study these first. Each one shows the method for a question type used later — follow the steps and you can do the Practice and Exam-style questions yourself.

■ File-based limits → relational database

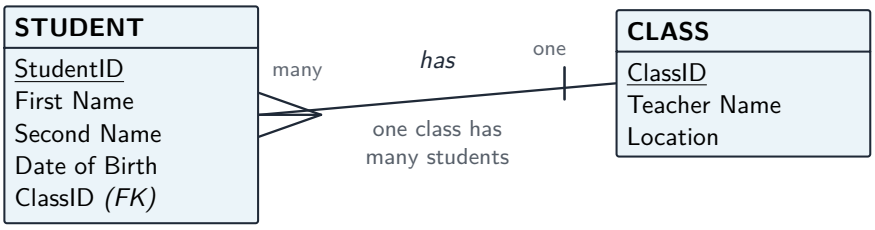
Storing data in separate **flat files** causes **data redundancy** (the same fact stored in several files), **data inconsistency** (copies updated separately disagree), and **data dependence** (programs tied to the file format). A relational database stores data once in linked **tables** managed by one DBMS, fixing all three.

■ Relational terms and keys

Term	Meaning
table (relation)	a grid for one entity, e.g. CUSTOMER
record (row)	one instance of the entity
field (column)	one piece of data about each record
primary key	field(s) that **uniquely identify** each record
foreign key	a field that refers to the primary key of another table



The **foreign key** CustomerID in ORDER refers to the **primary key** CustomerID in CUSTOMER, linking each order to its customer.



An entity-relationship diagram models the data

■ Normalisation (up to 3NF)

Normalisation removes redundancy. A table is in **Third Normal Form (3NF)** when every non-key field depends on **the key, the whole key, and nothing but the key** — i.e. atomic values (1NF), no partial dependency on part of a composite key (2NF), and no field depending on another non-key field (3NF). A **many-to-many** relationship is split using a **link (junction) table** holding two foreign keys.

■ Relationships and referential integrity

Two tables are linked by a **relationship** — **one-to-one**, **one-to-many** or **many-to-many**. The foreign key is placed on the **"many" side**. An **entity-relationship (E-R) diagram** shows these links.

Referential integrity 参照完整性 means every **foreign key** value must match an existing **primary key** in the other table (or be empty). It stops you recording, for example, an order for a customer who does not exist — the DBMS rejects a foreign key that has no matching record.

2 Practice

Now apply the methods above.

2.1 State **two** limitations of a file-based approach to storing data. [2]

2.2 Define **primary key** and **foreign key**. [2]

2.3 State what an **entity** is in a relational database. [1]

2.4 State what **First Normal Form (1NF)** requires. [1]

3 Exam-style questions

3.1 (a) State **two** problems caused by storing data in separate flat files. [2]

(b) State how a relational database removes one of them. [1]

3.2 Define each term: (a) record (b) field (c) primary key. [3]

3.3 Explain what it means for a table to be in **Third Normal Form (3NF)**. [3]

3.4 A school needs to record which students belong to which clubs; a student can join many clubs and a club has many students. Describe how to represent this **many-to-many** relationship using tables. [3]

3.5 (a) State what is meant by **referential integrity**. [2]

(b) A database has a `CUSTOMER` table and an `ORDER` table; one customer can place many orders. State the **relationship type** and which table holds the **foreign key**. [2]

Solutions

Each answer shows the steps, then the **result**.

2.1

- any two of: data redundancy (same data in several files); data inconsistency (copies disagree); data dependence (programs tied to the file format); weak security/sharing/querying

2.2

- primary key = field(s) that **uniquely identify** each record
- foreign key = a field that refers to the **primary key of another table**

2.3

- a **thing** the database stores data about (e.g. a customer, a book) — one table per entity

2.4

- all values are **atomic** (single values) — no repeating groups in a field

3.1

(a) any two: redundancy; inconsistency; data dependence; poor sharing/security

(b) e.g. each fact is stored **once** in a table, so there is no redundancy/inconsistency

3.2

- record = one **row** (one instance of the entity)
- field = one **column** (one piece of data)
- primary key = field(s) that **uniquely identify** a record

3.3

- every non-key field depends on **the whole primary key** and on **nothing but the key**
- values are atomic (1NF) and there is no partial dependency (2NF)
- no non-key field depends on another non-key field (no transitive dependency)

3.4

- create a **link (junction) table** (e.g. `STUDENT_CLUB`)
- it holds a foreign key to `STUDENT` and a foreign key to `CLUB`
- each row records one student-in-one-club, so many-to-many becomes two one-to-many relationships

3.5

(a) every foreign key value must match an existing primary key in the related table (or be null)

- so the database cannot hold a reference to a record that does not exist

(b) a **one-to-many** relationship

- the foreign key (`CustomerID`) is held in the `ORDER` table — the ”many” side

2 A relational database, WORKEXPERIENCE, stores data about students and the companies where they complete work experience.

Students can complete multiple work experience placements but can only complete one placement at a time.

Students can complete more than one placement at the same company.

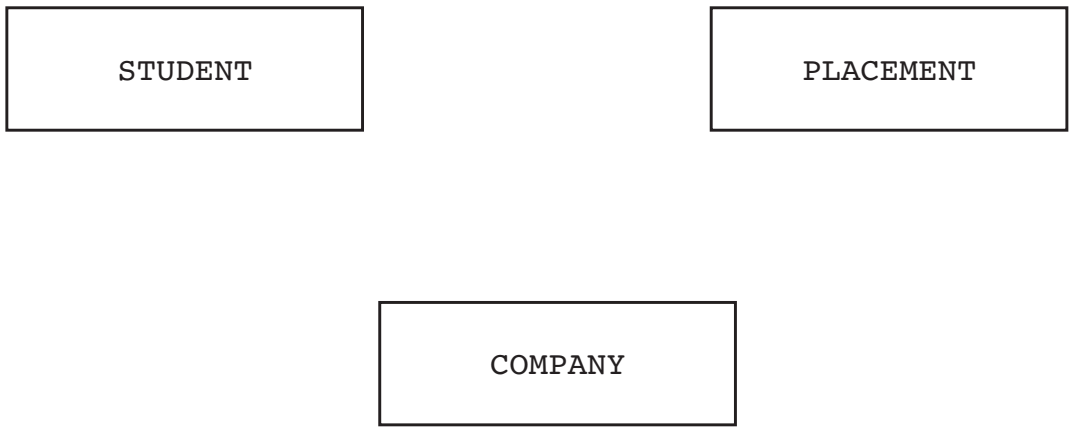
Part of the database is shown:

STUDENT(StudentID, FirstName, TelephoneNumber, UniversityName)

PLACEMENT(PlacementID, StudentID, CompanyID, StartDate, EndDate, Complete)

COMPANY(CompanyID, CompanyName, MaxStudentsPerPlacement)

(a) Complete the entity-relationship (E-R) diagram for the database WORKEXPERIENCE.



[2]

(b) Explain why the database WORKEXPERIENCE is in Third Normal Form (3NF).

.....

.....

.....

..... [2]

(c) Some example data from the `PLACEMENT` table is shown:

PlacementID	StudentID	CompanyID	StartDate	EndDate	Complete
PC001	LDEA01	MSCM	01/01/2023	12/07/2023	TRUE
PC002	LDEA01	MSCM	01/04/2024	08/05/2024	FALSE
PC003	ALAU02	NEAM	09/03/2020	10/03/2021	FALSE
PC004	LOLI75	GOUZ	07/06/2018	11/09/2018	TRUE

(i) Write a Structured Query Language (SQL) script to delete all placements that have been completed.

.....

.....

.....

..... [2]

(ii) Write an SQL script to return the total number of placements completed by the student with ID LDEA01 at the company with ID NEAM. The total should be given an appropriate name.

.....

.....

.....

.....

.....

..... [4]

(d) The database tables are repeated here:

```
STUDENT(StudentID, FirstName, TelephoneNumber, UniversityName)

PLACEMENT(PlacementID, StudentID, CompanyID, StartDate, EndDate,
          Complete)

COMPANY(CompanyID, CompanyName, MaxStudentsPerPlacement)
```

Explain what is meant by referential integrity, and how it applies to this database.

.....

.....

.....

.....

.....

..... [3]

4 A relational database, SHIPPING, stores data about the ships in a company and the containers that are carried on the ships.

The database has the following tables:

CONTAINER(ContainerID, Type, Weight, OwnerName, ShipID)

SHIP(ShipID, Type, Capacity, ShipName)

(a) Describe the relationship between the two tables. Refer to the primary and foreign keys in your answer.

.....

.....

.....

..... [2]

(b) The table CONTAINER needs an additional field to store the data for the last inspection date.

Write a Structured Query Language (SQL) script to add **one** field to the table CONTAINER to store the date of last inspection of the container, for example 08/07/2019.

.....

.....

.....

..... [2]

(c) Write an SQL script to return the number of containers stored in the database for the ship with the name Caledonia.

.....

.....

.....

.....

.....

..... [4]

(d) Describe the purpose of a developer interface in a Database Management System (DBMS).

.....

.....

.....

..... [2]

5 A company creates a relational database to store data about its customers.

The database, REVIEWS, stores data about the customers, products, complaints and staff.

The database has four tables:

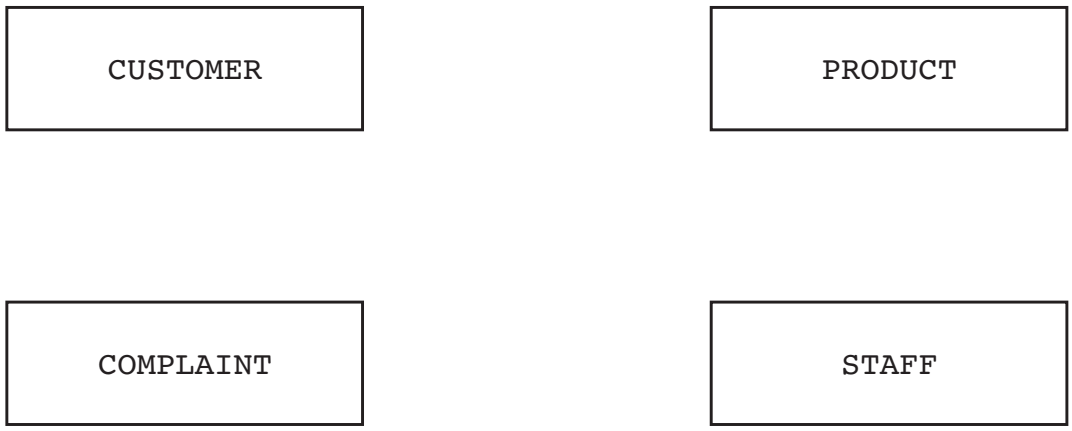
CUSTOMER(CustomerID, CustomerFirstName, CustomerLastName, CustomerEmail)

PRODUCT(ProductID, ProductName, ProductDetail, Price, Rating)

COMPLAINT(ComplaintID, ProductID, CustomerID, ComplaintDetails, StaffID)

STAFF(StaffID, StaffFirstName, StaffLastName, Department, RemoteWorker)

(a) Complete the entity-relationship (E-R) diagram for the database REVIEWS.



[3]

(b) Some example data from the `STAFF` table is shown.

StaffID	StaffFirstName	StaffLastName	Department	RemoteWorker
1	Ralph	Jura	E	Yes
2	Luca	Emcee	A	Yes
3	Darwin	Acula	F	No

Write a Structured Query Language (SQL) script to define the table `STAFF`.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(c) Products are given a rating between 1 and 10 inclusive.

Write an SQL script to return only the `ProductID`, `ProductName` and `ComplaintDetails` for all products with a rating of 5 or less. The results need to be displayed in descending order of rating.

.....

.....

.....

.....

.....

.....

.....

.....

[5]

(d) A Database Management System (DBMS) is used to maintain and manage the database.

Describe **two** ways in which the DBMS can be used to ensure the security of the customer data.

1 .....

.....

.....

.....

2 .....

.....

.....

.....

[4]

Question 6 starts on the next page.

5 A company builds and sells furniture to customers. The company stores data about customers, their payment cards and their furniture orders in a database.

The database, FURNITURE, has the following tables:

CUSTOMER(CustomerID, Name, Phone)

CUSTOMER_CARD_DATA(CardID, CustomerID, CardType, CardNumber, EndDate)

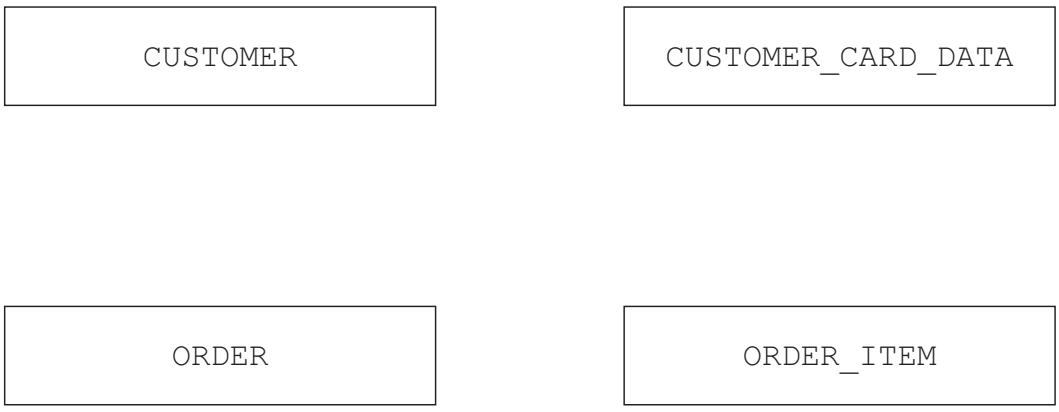
ORDER(OrderID, CustomerID, TotalCost, Paid, OrderDate, Complete)

ORDER_ITEM(OrderItemID, OrderID, Type, Height, Width, Depth, Details)

The primary keys are underlined in each table.

The attribute Complete in the table ORDER stores the Boolean value TRUE if the order has been built and FALSE if the order has not been built.

(a) Complete the entity-relationship (E-R) diagram for the database.



[3]

(b) Identify **one** attribute in the table CUSTOMER_CARD_DATA that could be a candidate key.

..... [1]

(c) Identify **two** tables in the database that contain one or more foreign keys.
Give **one** attribute that is a foreign key in each table.

	Table	Foreign key
1		
2		

[2]

(d) Explain the reasons why the data in the table `ORDER_ITEM` cannot be stored in the table `ORDER`

.....

.....

.....

.....

.....

..... [3]

(e) Write an Structured Query Language (SQL) script to output the customer ID, the customer's name and the total cost of the customer's orders that have **not** been paid.

The output of the total cost must have an appropriate title.

.....

.....

.....

.....

.....

.....

.....

..... [4]

2 A shop repairs electronic devices, for example mobile phones and tablet computers. The shop owner stores the data about the repairs using a file-based approach.

(a) Give **one** limitation of using a file-based approach to store the data **and** explain how a relational database addresses this limitation.

Limitation .....

.....

Explanation .....

.....

.....

.....

[3]

(b) The shop owner creates a relational database called `FIXIT`.

The database stores data about the customers and the devices for repair.

Some devices need new parts that are ordered from suppliers.

The database `FIXIT` is designed to include the following tables:

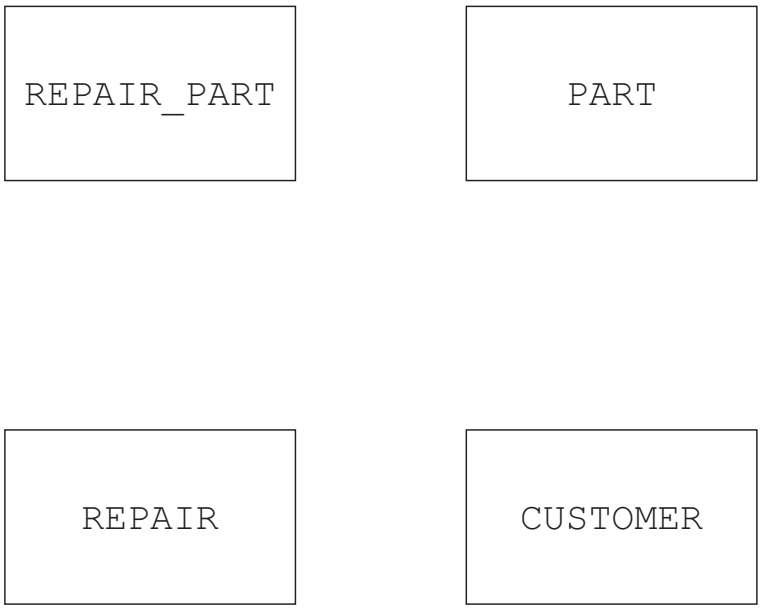
`PART(PartID, Description, Price, SupplierID)`

`CUSTOMER(CustomerID, FirstName, LastName, ContactNumber)`

`REPAIR(RepairNumber, StartDate, EndDate, CustomerID, Device)`

`REPAIR_PART(PartID, RepairNumber, Quantity)`

(i) Complete the entity-relationship (E-R) diagram for the given tables.



[3]

(ii) The table shows sample data for the table `REPAIR_PART`.

PartID	**RepairNumber**	**Quantity**
ACD128SA	0022	3
PPOR543DWW	0022	1
TR453	0023	1
PPOR543DWW	0023	2
WED5	0024	5

Write a Structured Query Language (SQL) script to define the table `REPAIR_PART`.

Include constraints (restrictions) on the data that can be entered into each field where appropriate.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(iii) Suppliers send invoices to the company for the parts that are used. A new table, `INVOICE`, stores the data about each invoice and whether it has been paid or not.

The design for the table `INVOICE` is shown:

`INVOICE(InvoiceID, SupplierID, AmountDue, Paid, DatePaid)`

The table shows sample data for the table `INVOICE`.

InvoiceID	SupplierID	AmountDue	Paid	DatePaid
000001	JK675	22.50	TRUE	01/01/2024
000002	WR443	358.99	FALSE	
000003	JK675	10.21	FALSE	

Write an SQL script to return the total amount due to the supplier with the ID of JK675 for all the invoices that have **not** currently been paid.

.....

.....

.....

.....

.....

..... [3]

(c) Complete the table by writing a definition for each of the database terms.

Term	Definition
Referential integrity	....
Candidate key	....
Tuple	....

[3]

6 A company uses a relational database to store data about its customers, employees and the individual repair jobs that customers have booked.

(a) Explain the benefits of using a relational database instead of a file-based approach.

.....

.....

.....

.....

.....

..... [3]

(b) The company decides which employees will work on each repair job. An employee can log into the database to access information about their repair jobs.

The database is normalised and includes these tables:

- `CUSTOMER` stores personal data about each customer
- `EMPLOYEE` stores personal data about each employee
- `LOGIN_DATA` stores the username and password for each employee
- `JOB` stores the data about each repair job
- `JOB_EMPLOYEE` stores the employees that are working on each repair job.

(i) Identify each relationship between the database tables **and** explain how each relationship can be implemented in the normalised database.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

(ii) The database also has the table `INVOICE` that stores data about each invoice that is sent to a customer.

Example data from the table `INVOICE` is given.

InvoiceID	**DateSent**	**Amount**	**Paid**	**JobID**
29262	12/12/2023	105.20	Y	221
26765	11/11/2023	200.00	Y	315
13290	02/01/2024	50.00	Y	315
34090	05/02/2024	25.95	N	569

Write a Structured Query Language (SQL) script to return the total amount of all the invoices sent in the year 2023 that have been paid.

.....

.....

.....

.....

.....

..... [3]

6 A company is developing a website that will allow users to create an account and then play a quiz every day. The data about the users and the quizzes are stored in a database.

A user must select a unique username and enter a valid email address to create an account. All users must be over the age of 16. A new quiz is given to the users every day. Each quiz is stored in its own text file.

The database stores the filename of each quiz and the date it can be played. The user gets a score for each quiz they complete, which is stored in the database. The scores are used to give each user a rating, for example Gold.

(a) Create a 3-table design for this database normalised to Third Normal Form (3NF).

Give your table design in the format:

`TableName(PrimaryKey, Field1, Field2, …)`

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

(b) The company is using a Database Management System (DBMS) to set up the database.

Describe what is meant by the following DBMS features:

Data dictionary .....

.....

.....

.....

Logical schema .....

.....

.....

.....

[4]

(c) The company has another database, FARMING, for a different game.

The database FARMING has a table named EVENT which is shown with some sample data.

PlayerID	EventID	Category	Points
000123	3	Build	100
000124	1	Grow	36
000123	4	Grow	22
000123	7	Create	158
000125	3	Grow	85
000125	4	Build	69

(i) The database FARMING has a second table created named PLAYER that has the primary key PlayerID.

The field PlayerID in EVENT needs to be set up as a foreign key to link to PlayerID in PLAYER.

Write a Structured Query Language (SQL) script to change the table definition for EVENT to link the foreign key to PLAYER.

.....

.....

.....

..... [2]

(ii) Write an SQL script to return the number of events that each player has completed.

.....

.....

.....

.....

.....

..... [3]

4 An assessment board wants to store the marks students achieved in exams in a database named RECORDS.

Part of the database design includes these two tables:

EXAM(ExamID, Subject, Level, TotalMarks)

EXAM_QUESTION(ExamQuestionID, ExamID, QuestionNumber, Question, MaxMark)

(a) Identify the relationship between EXAM and EXAM_QUESTION.

.....

..... [1]

(b) Sample data for the table EXAM is shown:

ExamID	Subject	Level	TotalMarks
00956124	Computer Science	2	75
00956125	Computer Science	3	120
00956126	Mathematics	2	100
00956127	Mathematics	3	150
00956128	Physics	2	70
00956129	Physics	3	80

Write a Structured Query Language (SQL) script to define the table EXAM.

.....

.....

.....

.....

.....

..... [3]

(c) The table `EXAM_QUESTION` has been created but the foreign key has not been linked.

Write an SQL script to update `EXAM_QUESTION` **and** link the foreign key to `EXAM`.

.....

.....

.....

..... [2]

(d) The database also needs to store data about the students, the exams the students have taken and the marks the students achieved in each question of each exam.

Describe the additional tables that will need to be included in the database **and** explain how all the tables in the database will be linked.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

8.2 Database Management Systems (DBMS

Name: _____ Class: _____ Date: _____ **Total: 15 marks**

KEY WORDS database management system (dbms) 数据库管理系统 • data dictionary 数据字典 • logical schema 逻辑模式 • integrity 完整性 • query 查询

Objective

Build the skills to answer exam questions on the **Database Management System (DBMS)** 数据库管理系统 — the features it provides and the tools it gives users.

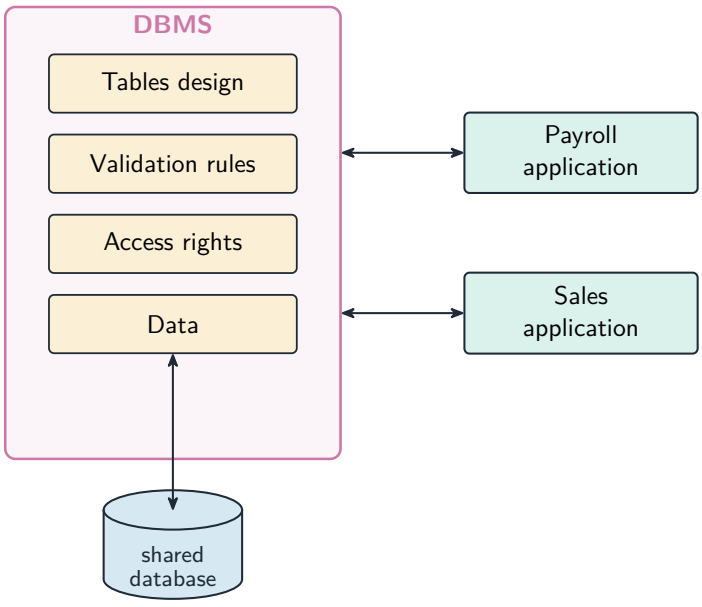
You must be able to:

- describe the features of a DBMS that fix the problems of a file-based approach
- describe how the software tools in a DBMS are used in practice

1 Worked examples

Study these first. Each one shows the method for a question type used later — follow the steps and you can do the Practice and Exam-style questions yourself.

■ Why a DBMS



A DBMS replaces the older file-based approach to data

A **DBMS** is the software that manages the database centrally for all programs, so the

structure, rules and security live in one place — not hard-coded into each program.

■ **Features that fix file-based problems**

Feature	What it does
data dictionary	describes every table, field, type and key
logical schema 逻辑模式	the overall logical design: the tables, fields, keys and relationships
redundancy control	each fact stored **once** (no inconsistency)
concurrent access control	locks/transactions let many users work safely
integrity rules	keys, unique and range constraints enforced centrally
transactions	a group of operations that all succeed or all fail
views	virtual tables showing each user only their slice
security & **backup**	per-user permissions; backup and recovery

■ **The DBMS tools**

A query builder, a forms designer, a report generator, a data-dictionary editor, user management, and an **SQL** editor — so designers, programmers and end-users each work at the right level.

2 Practice

Now apply the methods above.

2.1 State what a **data dictionary** holds. [1]

2.2 State what a **transaction** is. [1]

2.3 State **two** features a DBMS provides that a file-based system does not. [2]

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

3 Exam-style questions

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

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

3.3 State **two** software tools found in a DBMS and what each is used for. [2]

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

Solutions

Each answer shows the steps, then the **result**.

2.1

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

2.2

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

2.3

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

2.4

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

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

3.2

- the DBMS uses **locks** and **transactions**
- two users cannot change the same data at once and each transaction completes fully or not at all

3.3

- any two with a use, e.g. query builder (build queries without writing SQL); forms designer (build data-entry screens); report generator (produce formatted output)

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

5 A company creates a relational database to store data about its customers.

The database, REVIEWS, stores data about the customers, products, complaints and staff.

The database has four tables:

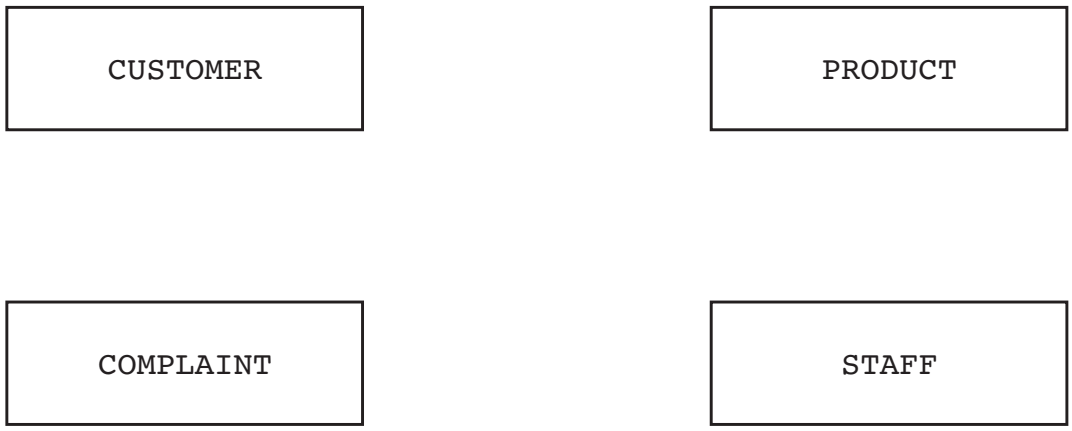
CUSTOMER(CustomerID, CustomerFirstName, CustomerLastName, CustomerEmail)

PRODUCT(ProductID, ProductName, ProductDetail, Price, Rating)

COMPLAINT(ComplaintID, ProductID, CustomerID, ComplaintDetails, StaffID)

STAFF(StaffID, StaffFirstName, StaffLastName, Department, RemoteWorker)

(a) Complete the entity-relationship (E-R) diagram for the database REVIEWS.



[3]

(b) Some example data from the `STAFF` table is shown.

StaffID	StaffFirstName	StaffLastName	Department	RemoteWorker
1	Ralph	Jura	E	Yes
2	Luca	Emcee	A	Yes
3	Darwin	Acula	F	No

Write a Structured Query Language (SQL) script to define the table `STAFF`.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(c) Products are given a rating between 1 and 10 inclusive.

Write an SQL script to return only the `ProductID`, `ProductName` and `ComplaintDetails` for all products with a rating of 5 or less. The results need to be displayed in descending order of rating.

.....

.....

.....

.....

.....

.....

.....

[5]

(d) A Database Management System (DBMS) is used to maintain and manage the database.

Describe **two** ways in which the DBMS can be used to ensure the security of the customer data.

1 .....

.....

.....

.....

2 .....

.....

.....

.....

[4]

Question 6 starts on the next page.

6 A shop sells pens to customers. Customers place an order with the shop and collect the items the next day. The shop uses a database to store the information about the orders.

The database contains the following tables:

CUSTOMER(CustomerID, CustomerName, Email)

ORDER(OrderID, CustomerID, Date, Collected)

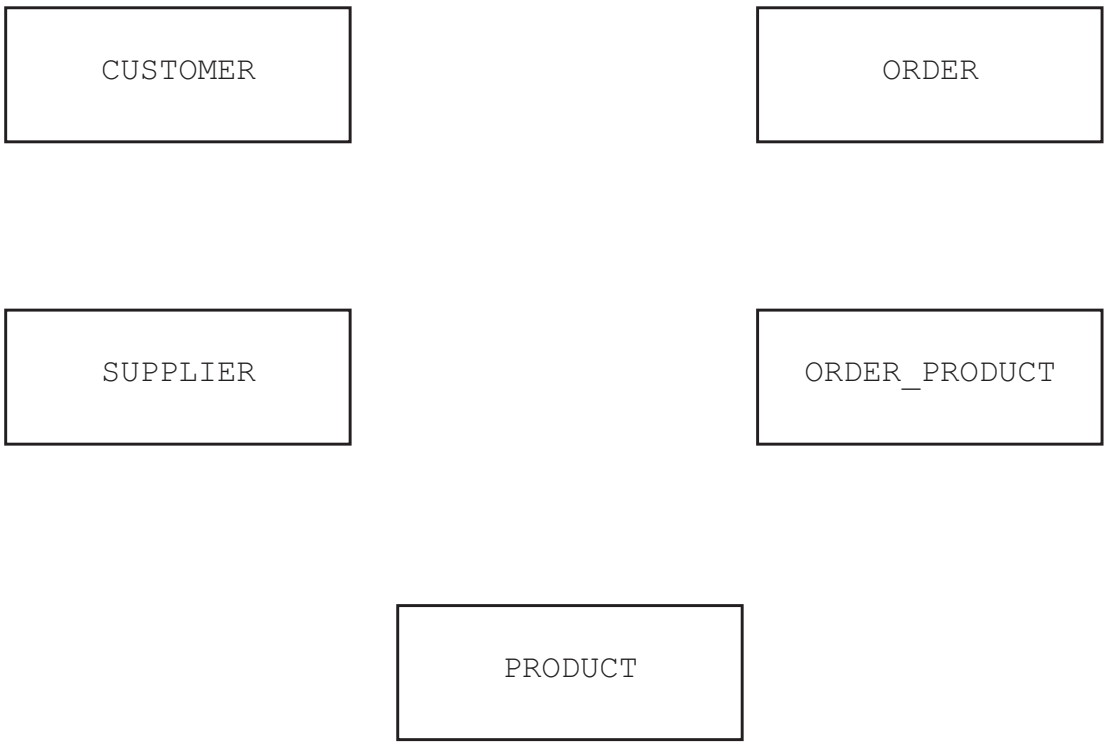
ORDER_PRODUCT(OrderID, ProductID, Quantity)

PRODUCT(ProductID, ProductName, QuantityInBox, Cost, SupplierID)

SUPPLIER(SupplierID, SupplierName, SupplierEmail)

The primary keys are underlined.

(a) Complete the entity-relationship (E-R) diagram for the database.



[4]

(b) A new product needs to be entered into the database. The product has the ID 002323, the product name 'Blue ball point 2mm', there are 50 in a box, the product costs $5.00 and the supplier has the ID SFX223.

Write a Structured Query Language (SQL) script to enter the new product into the table `PRODUCT`.

.....

.....

.....

.....

.....

.....

.....

..... [4]

(c) The attribute `Collected` in the table `ORDER` stores the Boolean value `TRUE` if the order has been collected and `FALSE` if the order has not been collected.

Write an SQL script to return the customer name for each customer that has orders they have not collected. Include the number of orders each customer has not collected with an appropriate title.

An example output might be:

CustomerName	**NotCollected**
Jack Wright	2
Lin Cho	1
Santaya Yui	1

.....

.....

.....

.....

.....

.....

.....

[4]

(d) The shop uses a Database Management System (DBMS).

(i) One feature provided by a DBMS is a data dictionary.

The data dictionary stores the attribute names, table names, foreign keys and primary keys in the database.

Identify **three** other items stored in a data dictionary.

1 .....

2 .....

3 .....

[3]

(ii) The DBMS provides a developer interface.

Explain how a database designer can make use of the developer interface.

.....

.....

.....

.....

.....

..... [3]

4 A company makes ice cream and sells it to shops.

The ice cream is made in batches: a large quantity of one type and flavour of ice cream that is then split into smaller quantities for sale.

The company's owner has designed a relational database, `ICECREAM`, to store data about their ice cream and customers.

Some of the tables in the database are given. The database is not normalised.

`BATCH(BatchID, Type, Flavour, Size, SellingPrice, EndDate)`

`CUSTOMER(CustomerID, CompanyName, EmailAddress, TelephoneNumber)`

`SALE(SaleID, BatchID, CustomerID, Quantity, Date)`

(a) Identify **two** foreign keys in the table `SALE` **and** the table that each foreign key references.

Foreign key 1 .....

Table name 1 .....

Foreign key 2 .....

Table name 2 .....

[2]

(b) Write an SQL script to return the total quantity of ice cream sold to the customer with the ID of 0034E in the year 2023.

.....

.....

.....

.....

.....

..... [3]

(c) The table definition for `BATCH` is repeated here:

`BATCH(BatchID, Type, Flavour, Size, SellingPrice, EndDate)`

Sample data for the table `BATCH` is given:

BatchID	Type	Flavour	Size	SellingPrice	EndDate
KIV12	Plain	Vanilla	1	2.20	12/12/2024
RIC14	Plain	Chocolate	0.5	1.80	12/12/2024
TYL1	Non-dairy	Lemon	0.5	2.10	01/02/2025
FYV2	Non-dairy	Vanilla	0.25	1.50	02/02/2025
BIV13	Plain	Vanilla	1	2.20	02/02/2024

(i) Write an SQL script to define the table `BATCH`.

Include constraints (restrictions) on the data that can be entered into each field where appropriate.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(ii) The table `BATCH` is not normalised.

Normalise the database table `BATCH`.

Write the table definitions for your new tables.

Identify any primary and foreign keys in your tables.

Do **not** change or include the tables `CUSTOMER` and `SALE`.

.....

.....

.....

.....

.....

.....

.....

..... [4]

(d) Complete the following table by defining each database term.

Database term	Definition
Entity	....
Attribute	....

[2]

(e) A Database Management System (DBMS) supports data integrity.

Explain how a DBMS supports data integrity.

.....

.....

.....

.....

.....

..... [3]

A-LEVEL COMPUTER SCIENCE • EXERCISE SHEET

8.3 Data Definition Language (DDL) and Data Manipulation Language (DML)

Name: _____ Class: _____ Date: _____ **Total: 39 marks**

KEY WORDS data definition language 数据定义语言 • data manipulation language 数据操纵语言 • query 查询 • dbms 数据库管理系统 • join 连接

Objective

Build the skills to answer exam questions on **SQL** 结构化查询语言 — the difference between DDL and DML, and reading and writing simple SQL statements.

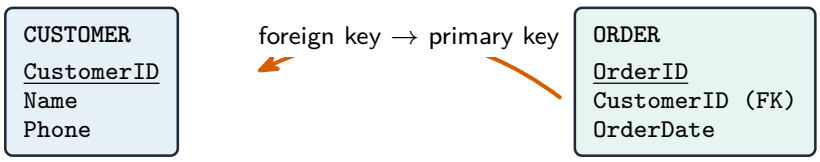
You must be able to:

- explain that the DBMS uses **DDL** to build the structure and **DML** to work with the data
- read and write simple **SQL (DDL)** statements (CREATE, ALTER, DROP)
- write **SQL (DML)** to query or change data in up to two tables

1 Worked examples

Study these first. Each one shows the method for a question type used later — follow the steps and you can do the Practice and Exam-style questions yourself.

■ DDL or DML?



DDL builds the tables and keys that DML statements then query

SQL has two halves: **Data Definition Language (DDL)** changes the **structure** (tables, keys); **Data Manipulation Language (DML)** works with the **data** (query, insert, update, delete).

DDL — building the structure

```
CREATE TABLE Student (
  StudentID INTEGER PRIMARY KEY,
  Name      VARCHAR(50) NOT NULL,
  Year      INTEGER
);

ALTER TABLE Student ADD Email VARCHAR(100);
```

A **foreign key** is declared with `FOREIGN KEY … REFERENCES`:

```
CREATE TABLE Grade (
  StudentID INTEGER,
  Mark      INTEGER,
  FOREIGN KEY (StudentID) REFERENCES Student(StudentID)
);
```

DML — working with the data

```
INSERT INTO Student (StudentID, Name, Year) VALUES (7, 'Lee', 12);

SELECT Name FROM Student WHERE Year = 12 ORDER BY Name;

UPDATE Student SET Year = 13 WHERE StudentID = 7;

DELETE FROM Student WHERE Year = 13;
```

To query **two tables**, match the foreign key to the primary key:

```
SELECT Student.Name, Grade.Mark
FROM   Student, Grade
WHERE  Student.StudentID = Grade.StudentID;
```

2 Practice

Now apply the methods above.

2.1 State the difference between **DDL** and **DML**. [2]

2.2 Write an SQL statement to list the **Name** of every student in **Year 12**. [2]

2.3 Write an SQL statement to **insert** a new student (StudentID 9, name "Sam", Year 11). [2]

2.4 Name the SQL statement used to **change a table's structure**. [1]

---

3 Exam-style questions

3.1 Write SQL **DDL** to create a table `Book` with: `BookID` (integer, primary key), `Title` (text, must not be empty), `Author` (text). [3]

3.2 Write SQL to change the `Year` of the student with `StudentID` 5 to 13. [2]

3.3 Tables `Student(StudentID, Name)` and `Grade(StudentID, Mark)` are linked by `StudentID`. Write SQL to list each student's **Name** with their **Mark**. [3]

3.4 Write SQL to **delete** every student in Year 13. [2]

3.5 Write SQL **DDL** to create a table `Grade` with `StudentID` (integer) as a **foreign key** referencing `Student(StudentID)`, and `Mark` (integer). [3]

3.6 Write SQL to list the **Name** of every Year-12 student, sorted into **alphabetical order** of name. [3]

3.7 A shipping company's database has two tables:

`SHIP(ShipID, ShipName, Capacity)` and `CONTAINER(ContainerID, ShipID, Contents, Weight)`

(a) **Describe** the relationship between the two tables, referring to the **primary** and **foreign** keys. [3]

(b) The company must record the date each container was last inspected. **Write** the SQL to add a suitable field to the `CONTAINER` table. [2]

(c) **Write** an SQL script to return the **number** of containers stored for the ship whose `ShipName` is `'Aurora'`. [4]

(d) **Write** an SQL script to list the `ShipName` and the total `Weight` carried, for every ship carrying more than 500 tonnes, heaviest first. [4]

(e) **Describe** the purpose of the **developer interface** in a DBMS, and **give** one feature it provides that the query processor does not. [3]

Solutions

Each answer shows the steps, then the **result**.

2.1

- DDL defines/changes the **structure** (tables, keys)
- DML works with the **data** (insert, update, delete, query)

2.2

- `SELECT … FROM` the student table, then filter Year 12
- `SELECT Name FROM Student WHERE Year = 12;`

2.3

- `INSERT INTO … VALUES`
- `INSERT INTO Student (StudentID, Name, Year) VALUES (9, 'Sam', 11);`

2.4

- `ALTER` (TABLE) — a DDL statement

3.1

```
CREATE TABLE Book (
  BookID INTEGER PRIMARY KEY,
  Title  VARCHAR(100) NOT NULL,
  Author VARCHAR(50)
);
```

- `CREATE TABLE` + fields; `PRIMARY KEY`; `NOT NULL` on Title

3.2

- `UPDATE … SET`
- `UPDATE Student SET Year = 13 WHERE StudentID = 5;`

3.3

- both tables, joined on `StudentID`
- `SELECT Student.Name, Grade.Mark FROM Student, Grade WHERE Student.StudentID = Grade.StudentID;`

3.4

- `DELETE FROM`
- `DELETE FROM Student WHERE Year = 13;`

3.5

```
CREATE TABLE Grade (
  StudentID INTEGER,
  Mark      INTEGER,
  FOREIGN KEY (StudentID) REFERENCES Student(StudentID)
);
```

- `CREATE TABLE` + fields; `FOREIGN KEY`; `REFERENCES` the correct table/field

3.6

- `SELECT … WHERE` and `ORDER BY`
- `SELECT Name FROM Student WHERE Year = 12 ORDER BY Name;`

3.7

(a) one ship has **many** containers, so it is a one-to-many relationship

- `ShipID` is the **primary key** of `SHIP`, uniquely identifying each ship
- `ShipID` in `CONTAINER` is a **foreign key** referring to it, which links each container to its ship

(b) `ALTER TABLE CONTAINER`

- `ADD LastInspected DATE;`

(c)

```
SELECT COUNT(*)
FROM CONTAINER, SHIP
WHERE CONTAINER.ShipID = SHIP.ShipID
  AND SHIP.ShipName = 'Aurora';
```

- `SELECT COUNT`; both tables; the join condition; the name condition, quoted

(d)

```
SELECT ShipName, SUM(Weight)
FROM SHIP, CONTAINER
WHERE SHIP.ShipID = CONTAINER.ShipID
GROUP BY ShipName
HAVING SUM(Weight) > 500
ORDER BY SUM(Weight) DESC;
```

- `SUM` with `GROUP BY`; the join; `HAVING` — not `WHERE`, because the condition is on an aggregate; `ORDER BY … DESC`

(e) the developer interface is the part of the DBMS used to **create and maintain** the database

- defining tables and data types, setting keys and relationships, and managing users and access rights
- the query processor only **runs** queries against a structure that already exists — it cannot create it

4 A relational database, SHIPPING, stores data about the ships in a company and the containers that are carried on the ships.

The database has the following tables:

CONTAINER(ContainerID, Type, Weight, OwnerName, ShipID)

SHIP(ShipID, Type, Capacity, ShipName)

(a) Describe the relationship between the two tables. Refer to the primary and foreign keys in your answer.

.....

.....

.....

..... [2]

(b) The table CONTAINER needs an additional field to store the data for the last inspection date.

Write a Structured Query Language (SQL) script to add **one** field to the table CONTAINER to store the date of last inspection of the container, for example 08/07/2019.

.....

.....

.....

..... [2]

(c) Write an SQL script to return the number of containers stored in the database for the ship with the name Caledonia.

.....

.....

.....

.....

.....

..... [4]

(d) Describe the purpose of a developer interface in a Database Management System (DBMS).

.....

.....

.....

..... [2]

5 A company creates a relational database to store data about its customers.

The database, REVIEWS, stores data about the customers, products, complaints and staff.

The database has four tables:

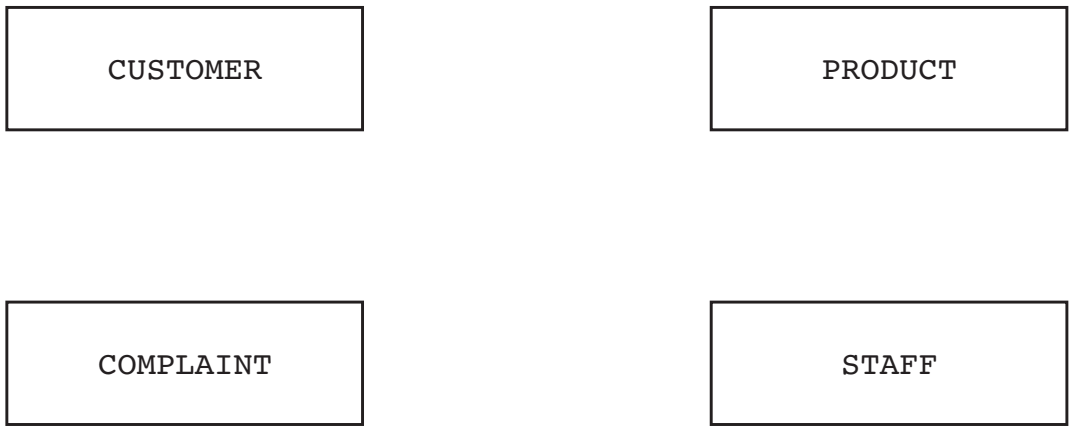
CUSTOMER(CustomerID, CustomerFirstName, CustomerLastName, CustomerEmail)

PRODUCT(ProductID, ProductName, ProductDetail, Price, Rating)

COMPLAINT(ComplaintID, ProductID, CustomerID, ComplaintDetails, StaffID)

STAFF(StaffID, StaffFirstName, StaffLastName, Department, RemoteWorker)

(a) Complete the entity-relationship (E-R) diagram for the database REVIEWS.



[3]

(b) Some example data from the `STAFF` table is shown.

StaffID	StaffFirstName	StaffLastName	Department	RemoteWorker
1	Ralph	Jura	E	Yes
2	Luca	Emcee	A	Yes
3	Darwin	Acula	F	No

Write a Structured Query Language (SQL) script to define the table `STAFF`.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(c) Products are given a rating between 1 and 10 inclusive.

Write an SQL script to return only the `ProductID`, `ProductName` and `ComplaintDetails` for all products with a rating of 5 or less. The results need to be displayed in descending order of rating.

.....

.....

.....

.....

.....

.....

.....

.....

[5]

(d) A Database Management System (DBMS) is used to maintain and manage the database.

Describe **two** ways in which the DBMS can be used to ensure the security of the customer data.

1 .....

.....

.....

.....

2 .....

.....

.....

.....

[4]

Question 6 starts on the next page.

5 A company builds and sells furniture to customers. The company stores data about customers, their payment cards and their furniture orders in a database.

The database, FURNITURE, has the following tables:

CUSTOMER(CustomerID, Name, Phone)

CUSTOMER_CARD_DATA(CardID, CustomerID, CardType, CardNumber, EndDate)

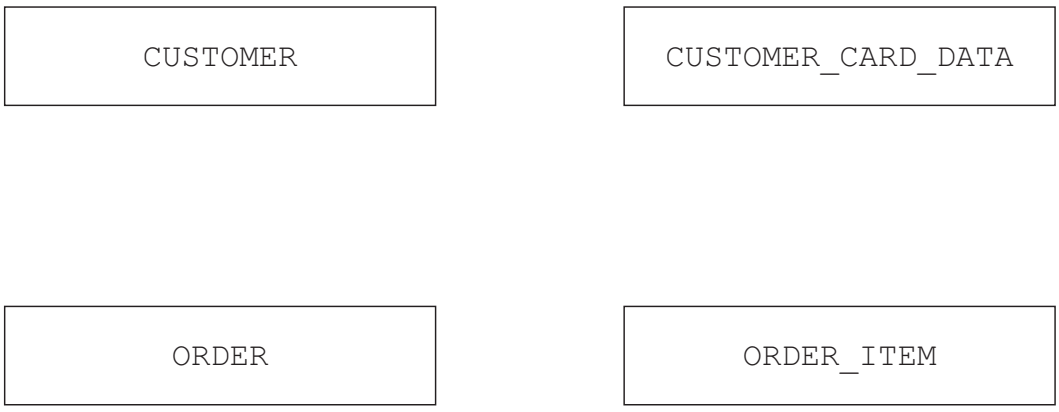
ORDER(OrderID, CustomerID, TotalCost, Paid, OrderDate, Complete)

ORDER_ITEM(OrderItemID, OrderID, Type, Height, Width, Depth, Details)

The primary keys are underlined in each table.

The attribute Complete in the table ORDER stores the Boolean value TRUE if the order has been built and FALSE if the order has not been built.

(a) Complete the entity-relationship (E-R) diagram for the database.



[3]

(b) Identify **one** attribute in the table CUSTOMER_CARD_DATA that could be a candidate key.

..... [1]

(c) Identify **two** tables in the database that contain one or more foreign keys.
Give **one** attribute that is a foreign key in each table.

	Table	Foreign key
1		
2		

[2]

(d) Explain the reasons why the data in the table `ORDER_ITEM` cannot be stored in the table `ORDER`

.....

.....

.....

.....

.....

..... [3]

(e) Write an Structured Query Language (SQL) script to output the customer ID, the customer's name and the total cost of the customer's orders that have **not** been paid.

The output of the total cost must have an appropriate title.

.....

.....

.....

.....

.....

.....

.....

..... [4]

6 A shop sells pens to customers. Customers place an order with the shop and collect the items the next day. The shop uses a database to store the information about the orders.

The database contains the following tables:

CUSTOMER(CustomerID, CustomerName, Email)

ORDER(OrderID, CustomerID, Date, Collected)

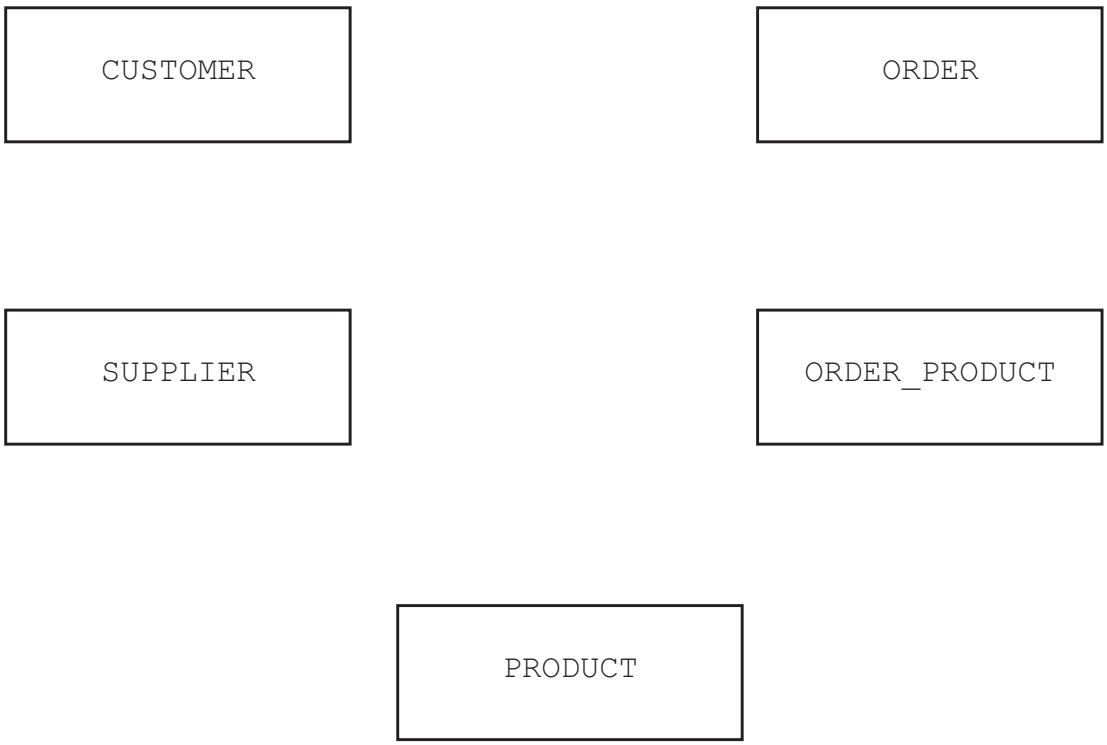
ORDER_PRODUCT(OrderID, ProductID, Quantity)

PRODUCT(ProductID, ProductName, QuantityInBox, Cost, SupplierID)

SUPPLIER(SupplierID, SupplierName, SupplierEmail)

The primary keys are underlined.

(a) Complete the entity-relationship (E-R) diagram for the database.



[4]

(b) A new product needs to be entered into the database. The product has the ID 002323, the product name 'Blue ball point 2mm', there are 50 in a box, the product costs $5.00 and the supplier has the ID SFX223.

Write a Structured Query Language (SQL) script to enter the new product into the table `PRODUCT`.

.....

.....

.....

.....

.....

.....

.....

..... [4]

(c) The attribute `Collected` in the table `ORDER` stores the Boolean value `TRUE` if the order has been collected and `FALSE` if the order has not been collected.

Write an SQL script to return the customer name for each customer that has orders they have not collected. Include the number of orders each customer has not collected with an appropriate title.

An example output might be:

CustomerName	**NotCollected**
Jack Wright	2
Lin Cho	1
Santaya Yui	1

.....

.....

.....

.....

.....

.....

.....

[4]

(d) The shop uses a Database Management System (DBMS).

(i) One feature provided by a DBMS is a data dictionary.

The data dictionary stores the attribute names, table names, foreign keys and primary keys in the database.

Identify **three** other items stored in a data dictionary.

1 .....

2 .....

3 .....

[3]

(ii) The DBMS provides a developer interface.

Explain how a database designer can make use of the developer interface.

.....

.....

.....

.....

.....

..... [3]

2 A shop repairs electronic devices, for example mobile phones and tablet computers. The shop owner stores the data about the repairs using a file-based approach.

(a) Give **one** limitation of using a file-based approach to store the data **and** explain how a relational database addresses this limitation.

Limitation .....

.....

Explanation .....

.....

.....

.....

[3]

(b) The shop owner creates a relational database called `FIXIT`.

The database stores data about the customers and the devices for repair.

Some devices need new parts that are ordered from suppliers.

The database `FIXIT` is designed to include the following tables:

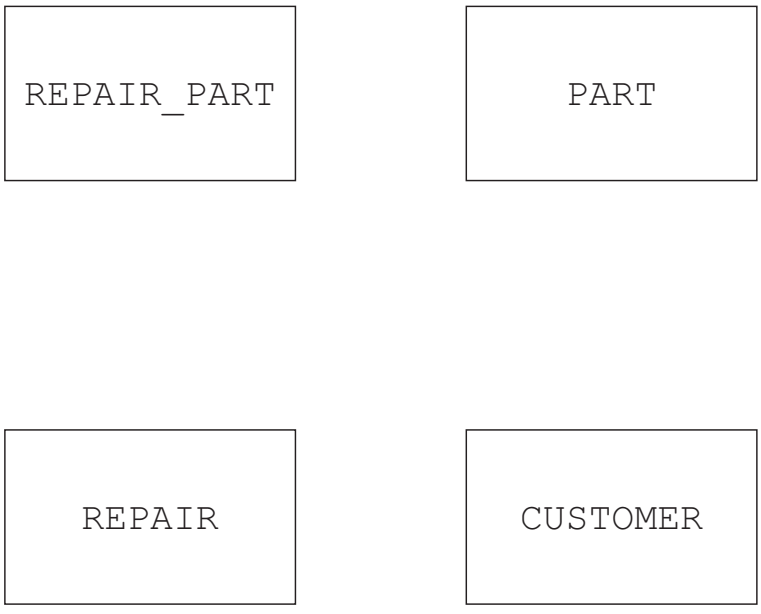
`PART(PartID, Description, Price, SupplierID)`

`CUSTOMER(CustomerID, FirstName, LastName, ContactNumber)`

`REPAIR(RepairNumber, StartDate, EndDate, CustomerID, Device)`

`REPAIR_PART(PartID, RepairNumber, Quantity)`

(i) Complete the entity-relationship (E-R) diagram for the given tables.



[3]

(ii) The table shows sample data for the table `REPAIR_PART`.

PartID	**RepairNumber**	**Quantity**
ACD128SA	0022	3
PPOR543DWW	0022	1
TR453	0023	1
PPOR543DWW	0023	2
WED5	0024	5

Write a Structured Query Language (SQL) script to define the table `REPAIR_PART`.

Include constraints (restrictions) on the data that can be entered into each field where appropriate.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(iii) Suppliers send invoices to the company for the parts that are used. A new table, `INVOICE`, stores the data about each invoice and whether it has been paid or not.

The design for the table `INVOICE` is shown:

`INVOICE(InvoiceID, SupplierID, AmountDue, Paid, DatePaid)`

The table shows sample data for the table `INVOICE`.

InvoiceID	SupplierID	AmountDue	Paid	DatePaid
000001	JK675	22.50	TRUE	01/01/2024
000002	WR443	358.99	FALSE	
000003	JK675	10.21	FALSE	

Write an SQL script to return the total amount due to the supplier with the ID of JK675 for all the invoices that have **not** currently been paid.

.....

.....

.....

.....

.....

..... [3]

(c) Complete the table by writing a definition for each of the database terms.

Term	Definition
Referential integrity	....
Candidate key	....
Tuple	....

[3]

6 A company uses a relational database to store data about its customers, employees and the individual repair jobs that customers have booked.

(a) Explain the benefits of using a relational database instead of a file-based approach.

.....

.....

.....

.....

.....

..... [3]

(b) The company decides which employees will work on each repair job. An employee can log into the database to access information about their repair jobs.

The database is normalised and includes these tables:

- `CUSTOMER` stores personal data about each customer
- `EMPLOYEE` stores personal data about each employee
- `LOGIN_DATA` stores the username and password for each employee
- `JOB` stores the data about each repair job
- `JOB_EMPLOYEE` stores the employees that are working on each repair job.

(i) Identify each relationship between the database tables **and** explain how each relationship can be implemented in the normalised database.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

(ii) The database also has the table `INVOICE` that stores data about each invoice that is sent to a customer.

Example data from the table `INVOICE` is given.

InvoiceID	**DateSent**	**Amount**	**Paid**	**JobID**
29262	12/12/2023	105.20	Y	221
26765	11/11/2023	200.00	Y	315
13290	02/01/2024	50.00	Y	315
34090	05/02/2024	25.95	N	569

Write a Structured Query Language (SQL) script to return the total amount of all the invoices sent in the year 2023 that have been paid.

.....

.....

.....

.....

.....

..... [3]

4 A company makes ice cream and sells it to shops.

The ice cream is made in batches: a large quantity of one type and flavour of ice cream that is then split into smaller quantities for sale.

The company's owner has designed a relational database, `ICECREAM`, to store data about their ice cream and customers.

Some of the tables in the database are given. The database is not normalised.

`BATCH(BatchID, Type, Flavour, Size, SellingPrice, EndDate)`

`CUSTOMER(CustomerID, CompanyName, EmailAddress, TelephoneNumber)`

`SALE(SaleID, BatchID, CustomerID, Quantity, Date)`

(a) Identify **two** foreign keys in the table `SALE` **and** the table that each foreign key references.

Foreign key 1 .....

Table name 1 .....

Foreign key 2 .....

Table name 2 .....

[2]

(b) Write an SQL script to return the total quantity of ice cream sold to the customer with the ID of 0034E in the year 2023.

.....

.....

.....

.....

.....

..... [3]

(c) The table definition for `BATCH` is repeated here:

`BATCH(BatchID, Type, Flavour, Size, SellingPrice, EndDate)`

Sample data for the table `BATCH` is given:

BatchID	Type	Flavour	Size	SellingPrice	EndDate
KIV12	Plain	Vanilla	1	2.20	12/12/2024
RIC14	Plain	Chocolate	0.5	1.80	12/12/2024
TYL1	Non-dairy	Lemon	0.5	2.10	01/02/2025
FYV2	Non-dairy	Vanilla	0.25	1.50	02/02/2025
BIV13	Plain	Vanilla	1	2.20	02/02/2024

(i) Write an SQL script to define the table `BATCH`.

Include constraints (restrictions) on the data that can be entered into each field where appropriate.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(ii) The table `BATCH` is not normalised.

Normalise the database table `BATCH`.

Write the table definitions for your new tables.

Identify any primary and foreign keys in your tables.

Do **not** change or include the tables `CUSTOMER` and `SALE`.

.....

.....

.....

.....

.....

.....

.....

..... [4]

(d) Complete the following table by defining each database term.

Database term	Definition
Entity	....
Attribute	....

[2]

(e) A Database Management System (DBMS) supports data integrity.

Explain how a DBMS supports data integrity.

.....

.....

.....

.....

.....

..... [3]

4 An assessment board wants to store the marks students achieved in exams in a database named RECORDS.

Part of the database design includes these two tables:

EXAM(ExamID, Subject, Level, TotalMarks)

EXAM_QUESTION(ExamQuestionID, ExamID, QuestionNumber, Question, MaxMark)

(a) Identify the relationship between EXAM and EXAM_QUESTION.

.....

..... [1]

(b) Sample data for the table EXAM is shown:

ExamID	Subject	Level	TotalMarks
00956124	Computer Science	2	75
00956125	Computer Science	3	120
00956126	Mathematics	2	100
00956127	Mathematics	3	150
00956128	Physics	2	70
00956129	Physics	3	80

Write a Structured Query Language (SQL) script to define the table EXAM.

.....

.....

.....

.....

.....

..... [3]

(c) The table `EXAM_QUESTION` has been created but the foreign key has not been linked.

Write an SQL script to update `EXAM_QUESTION` **and** link the foreign key to `EXAM`.

.....

.....

.....

..... [2]

(d) The database also needs to store data about the students, the exams the students have taken and the marks the students achieved in each question of each exam.

Describe the additional tables that will need to be included in the database **and** explain how all the tables in the database will be linked.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]