Database

Papers Dock

COMPUTER SCIENCE 9618 PAPER 1

Database

DATA STORAGE

File based approach	Relational Database
The data is stored in one or more separate computer files	Is a way of structuring information in tables, rows and columns.

Limitations In Using A File-Based Approach

- Data redundancy (data is repeated in more than one file)
- Data dependency (changes to data means changes to program accessing the data)
- Lack of data integrity (entries that should be same can be different in different places)
- Lack of data privacy (all users have access to all data if a single flat file)

Benefits Of Relational Database Instead Of Flat File

- Reduced data redundancy
- Reduced data dependency
- Improved data integrity
- Improved data privacy
- Program-data independence
- Ability to create ad hoc queries (queries designed for a particular purpose)

Describe the features of relational database that address the limitations of file based approach ?

Multiple tables are linked together
- which reduces data redundancy
- increases data integrity
- referential integrity can be enforced

Program-data independence means that
- structure of data can change and does not affect program
- structure of programs can change and does not affect data

Complex queries can be more easily written
- to find specific data

Different users can be given different access rights
- which improves security

Different users can be given different views of the data
- so they do not see confidential information
- and data privacy is maintained

Database Terms

Roll No.	Name	Course
CS08	Steive	Comp. Sci.
EE54	Jhoson	Electronics
B12	Eva	Biology
F32	Jhoson	Finance
M26	Erica	Maths

DBMS	Database Management System
Entity	The concept or object in the system that we want to model and store information about.
Attributes	A data item, represented as a field within a table.
Primary Key	A unique identifier for each tuple.
Foreign Key	A field in one table that links to a primary key in another table.
Candidate Key	A set of keys that could uniquely identify a record. One of them is chosen as the primary key.
Secondary Key	The remaining keys from the candidate keys that are not selected as the primary key.
Indexing	Indexing is a technique used in a relational database to speed up data retrieval operations. It works similarly to an index in a book, allowing the database to locate data faster without scanning the entire table.

The SOFTWARE_MANAGEMENT database has the following tables:

CUSTOMER_DETAILS(CustomerID, CompanyName, Address1, Address2, City)

SOFTWARE_PURCHASED(SoftwareName, SoftwareDescription, CustomerID, LicenceType, LicenceCost, RenewalDate)

Term	Example
Entity	CUSTOMER_DETAILS
Foreign key	CustomerID
Attribute	CustomerID

The bank uses a relational database, ACCOUNTS, to store the information about customers and their accounts.

The database stores the customer's first name, last name and date of birth.

The bank has several different types of account. Each account type has a unique ID number, name (for example, regular or saving) and bonus (for example, $5.00, $10.00 or $15.00).

A customer can have more than one account.

Each customer's account has its own ID number and stores the amount of money the customer has in that account.

The bank creates a normalised, relational database to store the required information. There are three tables:

- CUSTOMER
- ACCOUNT_TYPE
- CUSTOMER_ACCOUNT

(i) Write the attributes for each table to complete the database design for the bank.

CUSTOMER(CustomerID , Firstname, lastname, DateofBirth
.....
.....)

ACCOUNT_TYPE(AccountID, Name, Bonus
.....
.....)

CUSTOMER_ACCOUNT(CustomerAccountID, Amount, CustomerID, AccountID
.....
.....)

Identify the primary key for each table that you designed in **part (d)(i)**.

CUSTOMER CustomerID

ACCOUNT_TYPE AccountID

CUSTOMER_ACCOUNT CustomerAccountID

[2]

Identify **one** foreign key in one of the tables that you designed in **part (d)(i)**.

Table name CUSTOMER_ACCOUNT

Foreign key CustomerID

[1]

Definition	Term
All the data about one entity	Table
The data in one row of a table	Record/Tuple
A column or field in a table	Attribute

Raj owns houses that other people rent from him. He has a database that stores details about the people who rent houses, and the houses they rent. The database, HOUSE_RENTALS, has the following structure:

CUSTOMER(CustomerID, FirstName, LastName, DateOfBirth, Email)
HOUSE(HouseID, HouseNumber, Road, Town, Bedrooms, Bathrooms)
RENTAL(RentalID, CustomerID, HouseID, MonthlyCost, DepositPaid)

(a) Give the definition of the following database terms, using an example from the database HOUSE_RENTALS for each definition.

Term	Definition and example
Field	A column or attribute in a table FirstName in CUSTOMER table
Entity	Anything that data can be stored about E.g Customers, House
Foreign key	A field in one table that is linked to a primary key in another table e.g CustomerID in RENTAL Table

[6]

Term	Definition and example
Field	A column/attribute in a table e.g. `CustomerID` in the table `CUSTOMER`
Entity	Anything that data can be stored about e.g. A customer or a house
Foreign Key	A field in one table that is **linked** to a **Primary Key** in another table e.g. `CustomerID` / `HouseID` in table `RENTAL`

Concept Of Primary And Foreign Key

Customers Table

CustomerID (PK)	Name	Email	Phone
1	John Doe	john@example.com	1234567890
2	Alice Smith	alice@email.com	9876543210
3	Bob Johnson	bob@mail.com	5678901234

Orders Table

OrderID (PK)	CustomerID (FK)	OrderDate	TotalAmount
101	1	2025-03-01	100.50
102	2	2025-03-02	250.00
103	1	2025-03-03	75.20

STRUCTURED QUERY LANGUAGE	
Data Definition Language	Data Manipulation Language

Data Definition Language (DDL)

1) DDL Statement to create Database

```
CREATE DATABASE  <Name> ;
```

(i) Write a Data Definition Language (DDL) statement to create the EMPLOYEES database.

CREATE DATABASE EMPLOYEES ;

[1]

2) DDL Statement to create a table

CHARACTER
VARCHAR
VARCHAR(n) String Which Contains Maximum Of n Characters
BOOLEAN
INTEGER
REAL
DATE
TIME

```
CREATE TABLE  <Name>  (
      <Attribute Name>   <Datatype> ,
      <Attribute Name>   <Datatype> ,
       <Attribute Name>   <Datatype> ,
       <PRIMARY KEY (Name) NOT NULL
) ;
```

EmployeeID	FirstName	LastName	DateOfBirth	Gender	DepartmentNumber
156FJEK	Harvey	Kim	12/05/1984	Male	S1
558RRKL	Catriona	Moore	03/03/1978	Female	F2
388LMDV	Oscar	Ciao	01/01/1987	Male	F2

(ii) Write a DDL statement to define the table EMPLOYEE_DATA, and declare EmployeeID as the primary key.

```
CREATE TABLE Employee_Data (
    EmployeeID VARCHAR(7),
    FirstName VARCHAR,
    LastName VARCHAR,
    DateOfBirth DATE,
    Gender VARCHAR(6),
    DepartmentNumber VARCHAR(2),
    PRIMARY KEY (EmployeeID) NOT NULL
);
```

(c) Example data from the table RENTAL are given:

RentalID	CustomerID	HouseID	MonthlyCost	DepositPaid
1	22	15B5L	1000.00	Yes
2	13	3F	687.00	No
3	1	12AB	550.00	Yes
4	3	37	444.50	Yes

(i) Complete the following Data Definition Language (DDL) statement to define the table RENTAL.

```
CREATE ..... ..... (
        RentalID INTEGER NOT NULL,
        CustomerID INTEGER NOT NULL,
        HouseID ..... (5) NOT NULL,
        MonthlyCost ..... NOT NULL,
        DepositPaid BOOLEAN NOT NULL,
        ..... (RentalID)
);
```

[4]

```
CREATE TABLE RENTAL(
      RentalID INTEGER NOT NULL,
      CustomerID INTEGER NOT NULL,
      HouseID VARCHAR (5) NOT NULL,
      MonthlyCost REAL/CURRENCY NOT NULL,
      DepositPaid BOOLEAN NOT NULL,
      PRIMARY KEY (RentalID)
);
```

(c) (i) Sample data to be stored in the table `STUDENT_TEST` is shown.

StudentID	TestID	Mark
12	A1	50
12	P10	100
13	A1	75
14	P10	60

Write a Structured Query Language (SQL) script to create the table `STUDENT_TEST`.

```
CREATE TABLE STUDENT_TEST (
    StudentId INTEGER,
    TestID VARCHAR,
    Mark INTEGER,
    PRIMARY KEY(StudentID,TestID),
    FOREIGN KEY(TestID) REFERENCES TEST(TestID),
    FOREIGN KEY(StudentID) REFERENCES STUDENT(StudentID)
);
```

3) DDL Statement to add new field

ALTER TABLE <Table-Name> ;
ADD < field name > < Datatype> ;

(d) The table STUDENT needs an additional field to store the student's telephone number, for example 012-3456.

Write a Data Definition Language (DDL) statement to add the new field to the table STUDENT.

ALTER TABLE STUDENT

ADD TelNum VARCHAR ;

[2]

(ii) The table DEVICE needs a new attribute to store whether the device has been returned by the staff member, or not.

Write a Structured Query Language (SQL) script to insert the new attribute into the table DEVICE.

ALTER TABLE DEVICE

ADD Retuned BOOLEAN ;

[2]

- ALTER TABLE DEVICE
- ADD *appropriate field name, appropriate data type*

e.g.

```
ALTER TABLE DEVICE
ADD Returned Boolean;
```

Data Manipulation Language (DML)

1) Add the data into the table

INSERT INTO <Table-Name> ;
VALUES (<data in col 1>, <data in col 2>, ...) ;

RoomNumber	RoomType
1	Standard
2	Double
3	Executive
4	Standard

(ii) Room number **5** is a **Double** room.

Complete the Data Manipulation Language (DML) statement to add the details for room number 5 to the table ROOM.

INSERT ..... INTO ..... ROOM

VALUES (..... 5 , Double .....) ;

[2]

It is possible to write the `INSERT INTO` statement in two ways:

1. Specify both the column names and the values to be inserted:

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

2. If you are adding values for all the columns of the table, you do not need to specify the column names in the SQL query. However, make sure the order of the values is in the same order as the columns in the table. Here, the `INSERT INTO` syntax would be as follows:

```
INSERT INTO table_name
VALUES (value1, value2, value3, ...);
```

2) Update the data / record in a table.

UPDATE < table name >
SET <col1> = <value 1>, <col2> = <value 2>, ...
WHERE <condition>

table name = B-Nurse

(ii) Fatima Woo is an Area B nurse with the nurse ID of 076. She has recently married, and her new family name is Chi. Write an SQL command to update her record.

UPDATE B-Nurse

SET FamilyName = "Chi"

WHERE NurseID = "076" ; [3

3) Display / return a value

SELECT <col1>, <col2>, ...
FROM <table-name>
WHERE <condition>

B_NURSE (NurseID, Familyname, Specialism)

(i) Write an SQL query to display the Nurse ID and family name for all Area B nurses with a specialism of 'THEATRE'.

SELECT NurseID , FamilyName
FROM B - Nurse
WHERE Specialism = "THEATRE" ; [3]

LESSON(LessonID, StudentID, InstructorID, LessonDate, LessonTime)

(e) Write a Data Manipulation Language (DML) statement to return the date and time of all future lessons booked with the instructor whose InstructorID is Ins01.

```
SELECT LessonDate , LessonTime
FROM LESSON
WHERE InstructorID = "Ins01"
AND LessonDate > 04/03/2025 ;
```

4) COUNT () AVG () SUM ()

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```

```
SELECT AVG(column_name)
FROM table_name
WHERE condition;
```

```
SELECT SUM(column_name)
FROM table_name
WHERE condition;
```

ProductID	ProductName	SupplierID	CategoryID	Unit	Price
1	Chais	1	1	10 boxes x 20 bags	18
2	Chang	1	1	24 - 12 oz bottles	19
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22
5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.35

Write the SQL statement to find the number of products

```
SELECT COUNT(ProductID)
FROM Products;
```

Write the SQL statement to find the average price of all products

```
SELECT AVG(Price)
FROM Products;
```

Write the SQL statement to calculate the sum of all the prices

```
SELECT SUM(Price)
FROM Products;
```

Two Table Question

The table shows example data in GAME_DEVELOPMENT.

GameName	Genre	TeamNumber	DevelopmentStage	ManagerID
Bunny Hop	Platform	4	Analysis	23KP
Fried Eggs	Retro	2	Programming stage 1	9RTU
Create-a-game	Action	1	Acceptance testing	11TF

(ii) Another table, PRODUCT_MANAGER, is created.

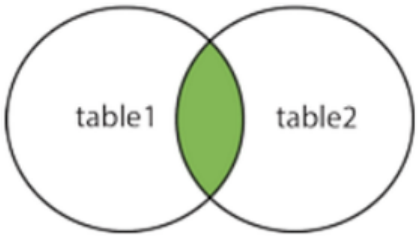
PRODUCT_MANAGER(ManagerID, FirstName, LastName)

Complete the Data Manipulation Language (DML) statement to return the game name, genre and team number of all games managed by the product manager with the first name 'James' and the last name 'Fitz'.

```
SELECT GameName, Genre, TeamNumber
FROM Game_Development, Product_Manager
WHERE Product_Manager.FirstName = "James"
AND Product_Manager.LastName = "Fitz"
AND Product_Manager.ManagerID = Game_Development.ManagerID;
```

Inner Join

The `INNER JOIN` keyword selects records that have matching values in both tables.




```
SELECT  < col name >
FROM    < table 1 >
INNER JOIN < table 2 >
ON   table1 . colname  =  table2 . colname
WHERE   <condition> ;
```

A database, FILMS, stores information about films and actors.

Part of the database is shown:

```
ACTOR(ActorID, FirstName, LastName, DateOfBirth)
FILM_FACT(FilmID, FilmTitle, ReleaseDate, Category)
FILM_ACTOR(ActorID, FilmID)
```

(c) Complete the SQL script to return the IDs of all the actors in the film with the title Cinderella.

```
SELECT ActorID
FROM FILM_ACTOR
INNER JOIN FILM_FACT
ON FILM_FACT.FilmID = FILM_ACTOR . FilmID
WHERE FILM_FACT.FilmTitle = "Cinderella" ;
```

[4]

GROUP BY ORDER BY

(c) The database has the following tables:

```
CUSTOMER(CustomerID, CompanyName)

SOFTWARE(SoftwareID, SoftwareName, OperatingSystem, Description)

LICENCE(LicenceID, CustomerID, SoftwareID, DateOfPurchase,
        LicenceType, Cost, ExpiryDate)
```

(iii) The company needs a list of all software licences that have an expiry date on or before 31/12/2019.

Write an SQL query to return the fields `CustomerID`, `SoftwareID`, `LicenceType`, `Cost` and `ExpiryDate` for all licences that expire on, or before 31/12/2019. Group the output by `CustomerID`, and in ascending order of cost.

```
SELECT CustomerID, SoftwareID, LicenseType, Cost, ExpiryDate
FROM Licence
WHERE ExpiryDate <= "31/12/2019"
GROUP BY CustomerID
ORDER BY Cost ASC;
```

(iv) The database tables are repeated here for reference:

```
BIRD_TYPE(BirdID, Name, Size)

BIRD_SEEN(SeenID, BirdID, Date, Location, PersonID)

PERSON(PersonID, FirstName, LastName, EmailAddress)
```

Complete the SQL script to return the number of birds of each size seen by the person with the ID of J_123.

```
SELECT BIRD_TYPE.Size, ..... (BIRD_TYPE.BirdID)
       AS NumberOfBirds
FROM BIRD_TYPE, .....
WHERE ..... = "J_123"
AND BIRD_TYPE.BirdID = .....
..... BIRD_TYPE.Size;
```

[5]

```
SELECT BIRD_TYPE.Size, COUNT(BIRD_TYPE.BirdID) AS
NumberOfBirds

FROM BIRD_TYPE, BIRD_SEEN

WHERE BIRD_SEEN.PersonID = "J_123"

AND BIRD_TYPE.BirdID = BIRD_SEEN.BirdID

GROUP BY BIRD_TYPE.Size;
```

Size	NumberOfBirds
Small	5
Medium	3
Large	7

DELETE

The DELETE statement is used to delete existing records in a table.

```
DELETE FROM <table name>
WHERE  <condition> ;
```

WHERE AND LIKE

```
SELECT COUNT(TelescopeID)
FROM TELESCOPE
WHERE CompanyID LIKE 'HW%';
```

This query counts how many TelescopeID entries belong to companies whose CompanyID starts with "HW".

- HW → The value must start with "HW".
- % → A wildcard that represents any sequence of characters (or nothing) after "HW"

Referential Integrity

Referential integrity ensures that relationships between database tables remain consistent. If one table references another, the referenced data must exist and follow specific rules to maintain data consistency.

I) Referential Integrity Ensures Data Exists

When a table references another table using a foreign key, the referenced data must already exist. This prevents records from pointing to non-existing data.
For example, consider two tables: Students and Courses. There is also an Enrollments table that tracks which students are enrolled in which courses.

Student_ID	Student_Name
101	Alice
102	Bob

Course_ID	Course_Name
301	Math
302	Science

Enrollment_ID	Student_ID	Course_ID
1	101	301
2	102	302
3	105	301 ❌ (Invalid)

In this case, Student_ID 105 does not exist in the Students table, making the reference invalid. The database will not allow adding this entry unless Student_ID 105 exists.

2) A Primary Key Cannot Be Deleted Unless All Dependent Records Are Deleted

If a primary key in one table is referenced in another table, it cannot be deleted unless all related records are removed first.

For example, consider a Customers table and an Orders table.

Customer_ID	Customer_Name
1	John
2	Sara

Order_ID	Customer_ID	Product
101	1	Laptop
102	2	Phone

If an attempt is made to delete Customer_ID 1, it will fail because Order_ID 101 still references it.

The solution is to first delete Order_ID 101 from the Orders table and then delete Customer_ID 1 from the Customers table.

3) Cascading Delete

If a primary key is deleted, all dependent records should also be deleted automatically.

For example, if cascading delete is enabled and Customer_ID 1 is deleted, Order_ID 101 from the Orders table will also be deleted automatically.

Customer_ID	Customer_Name
2	Sara

Order_ID	Customer_ID	Product
102	2	Phone

This approach prevents orphaned records, ensuring that no data is left without a valid reference.

4) A Primary Key Cannot Be Updated Unless All Dependent Records Are Updated

If a primary key value is changed, the same change must be made in all dependent records to maintain referential integrity.

For example, consider the Employees and Salaries tables.

Employee_ID	Employee_Name
201	Alice
202	Bob

Salary_ID	Employee_ID	Salary
501	201	5000
502	202	6000

If Employee_ID 201 is changed to 301, the reference in the Salaries table will break. The solution is to update both the Employees and Salaries tables simultaneously.

5) Cascading Update/Edit

If a primary key value is updated, all related records in other tables should be updated automatically.
For example, if Employee_ID 201 is changed to 301, the system will also update all instances of 201 in the Salaries table.

Employee_ID	Employee_Name
301	Alice
202	Bob

Salary_ID	Employee_ID	Salary
501	301	5000
502	202	6000

This prevents broken references and maintains consistency across tables.

6) Every Foreign Key Value Must Have a Matching Primary Key Value

A foreign key cannot have a value that does not exist in the referenced table.
For example, consider a Customers table and an Orders table.

Customer_ID	Customer_Name
1	John
2	Sara

Order_ID	Customer_ID	Product
101	1	Laptop
102	3 ❌ (Invalid)	Phone

Customer_ID 3 does not exist in the Customers table, making this reference invalid. The solution is to ensure that all Customer_ID values in the Orders table exist in the Customers table.

7) Foreign Keys Must Be the Same Data Type as the Corresponding Primary Key

A foreign key must have the same data type as the corresponding primary key to maintain consistency.

For example, consider the Users and Orders tables

Users (Primary Table)
User_ID (INTEGER)
Name (VARCHAR)

Orders (Foreign Key)
Order_ID (INTEGER)
User_ID (VARCHAR) ❌ (Mismatch)

In this case, User_ID in the Orders table is stored as VARCHAR, while in Users, it is INTEGER. This mismatch breaks referential integrity. The solution is to ensure that both User_ID columns have the same data type (INTEGER).

What is meant by Referential Integrity?

- Referential integrity is making sure tables do not try to reference data which does not exist.
- A primary key cannot be deleted unless all dependent records are already deleted.
- Cascading delete
- A primary key cannot be updated unless all dependent records are already updated.
- Cascading update/edit
- Every foreign key value has a matching value in the corresponding primary key.
- The foreign keys must be the same data type as the corresponding primary key.

Explain the reasons why referential integrity is important in a database?

- Referential Integrity makes sure data is consistent
- Referential Integrity makes sure all data is up-to-date
- Referential integrity ensures that every foreign key has a corresponding primary key
- Referential Integrity prevents records from being added / deleted / modified incorrectly
- Referential Integrity makes sure that if data is changed in one place, the change is reflected in all related records
- Referential Integrity makes sure any queries return accurate and complete results

Entity Relationship Diagram

An Entity-Relationship Diagram (ERD) is a visual representation of how different entities in a database are related to each other.

Relation Type	Example
One-to-One	Person ------ Passport
One-to-Many	Father ------< Children
Many-to-One	Children >------ Mother
Many-to-Many	Children >------< Teachers

- **One-to-One (1:1) → One person has one passport, and each passport belongs to one person.**
- **One-to-Many (1:M) → One father has many children, but each child has only one father.**
- **Many-to-One (M:1) → Many children have one mother.**
- **Many-to-Many (M:M) → Children have multiple teachers, and each teacher teaches multiple children.**

(a) Describe the relationship shown above.

Many To One

.....[1]

The relationship for Area B of the hospital is:

(i) Explain what the degree of relationship is between the entities B-NURSE and B-WARD.

Many To Many

.....[1]

6 A shop sells plants to customers. The shop manager has a relational database to keep track of the sales.

The database, PLANTSALES, has the following structure:

PLANT(PlantName, QuantityInStock, Cost)

CUSTOMER(CustomerID, FirstName, LastName, Address, Email)

PURCHASE(PurchaseID, CustomerID)

PURCHASE_ITEM(PurchaseID, PlantName, Quantity)

(ii) Draw an entity-relationship (E-R) diagram for the database PLANTSALES.

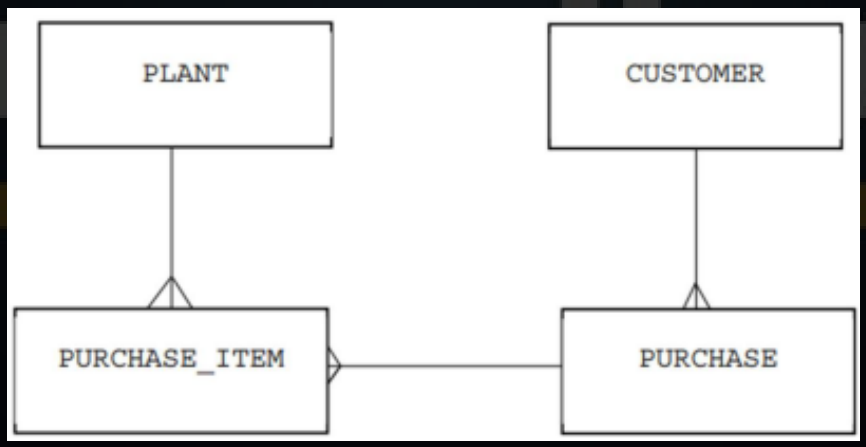
PLANT

CUSTOMER

PURCHASE_ITEM

PURCHASE

[3]

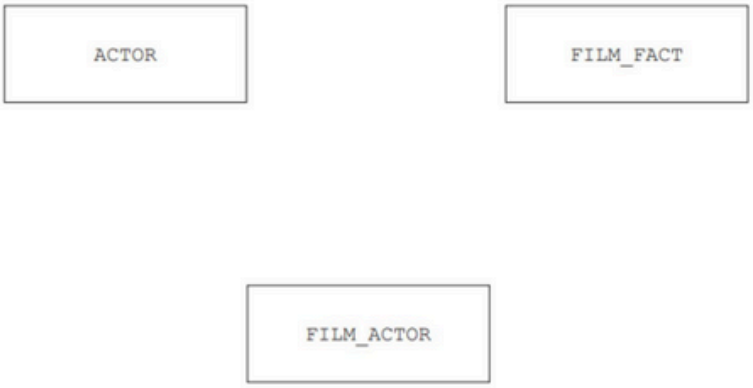


A database, FILMS, stores information about films and actors.

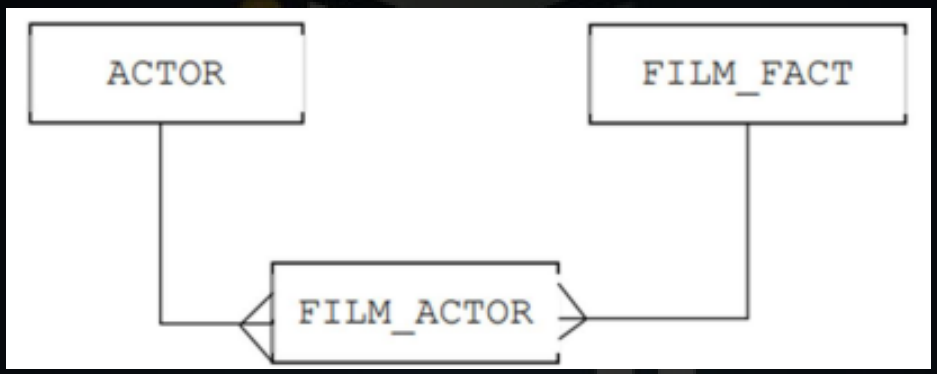
Part of the database is shown:

ACTOR(ActorID, FirstName, LastName, DateOfBirth)
FILM_FACT(FilmID, FilmTitle, ReleaseDate, Category)
FILM_ACTOR(ActorID, FilmID)

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



[2]



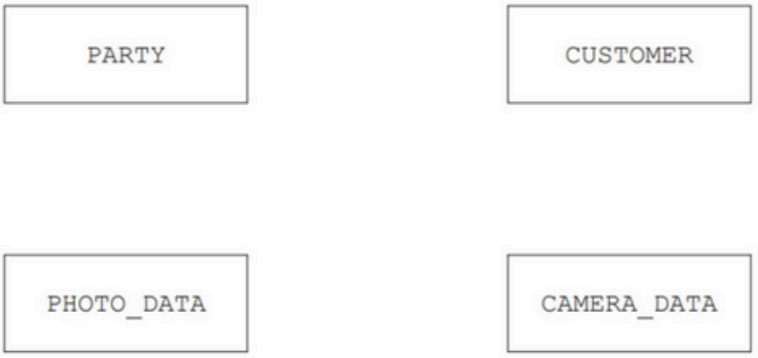
CUSTOMER(CustomerID, FirstName, LastName, Telephone)

PARTY(PartyID, CustomerID, PartyDate, StartTime)

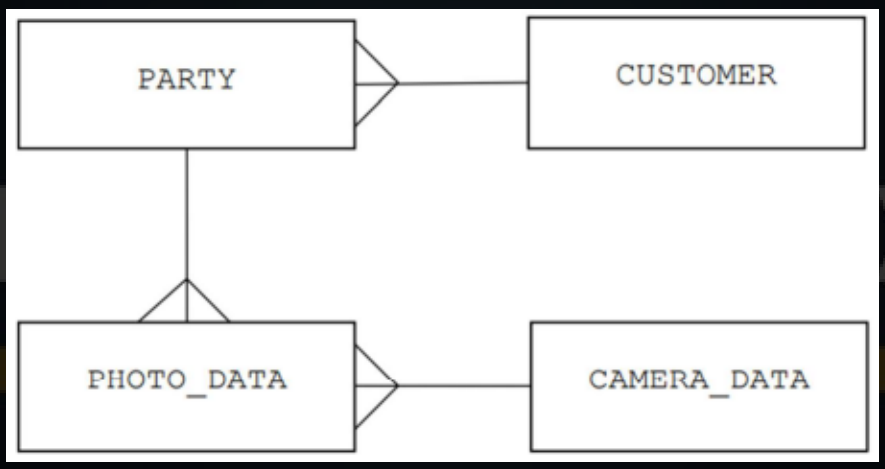
PHOTO_DATA(PhotoID, PartyID, TimeTaken, CameraID)

CAMERA_DATA(CameraID, LensType, LightingType)

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



[3]



Normalization

Imagine you have a messy bookshelf where books are scattered without order. Normalization is like organizing your bookshelf so that everything is properly categorized, easy to find, and avoids redundancy.

1NF (First Normal Form)

- No repeated group of attributes
- All attributes should be atomic (indivisible values)
- No duplicate rows

Example Table (Before 1NF - Unorganized Data)

Student_ID	Name	Subjects
101	Ali	Math, Science
102	Sara	English
103	Ahmed	Math, English

Problem

- The Subjects column contains multiple values (not atomic).
- Data is not structured properly.

After 1NF (Organized Data)

Student_ID	Name	Subject
101	Ali	Math
101	Ali	Science
102	Sara	English
103	Ahmed	Math
103	Ahmed	English

Now, every column has atomic values, and there are no repeated groups of attributes.

2NF (Second Normal Form)

- Should be in 1NF
- No partial dependency (Every non-key attribute should depend on the full primary key)

Partial Dependency: When an attribute depends only on part of a composite primary key.

Example Table (Before 2NF - Not in 2NF)

Order_ID	Product_ID	Product_Name	Supplier
1	P101	Laptop	Dell
1	P102	Mouse	Logitech
2	P101	Laptop	Dell
2	P103	Keyboard	HP

Problem

- The composite primary key here is (Order_ID, Product_ID).
- Product_Name and Supplier depend only on Product_ID, not on the full primary key (Order_ID, Product_ID).
- This is a partial dependency, violating 2NF

Solution

- Split the table to remove partial dependency, Products into another table.

Order Table (After 2NF)

Order_ID	Product_ID
1	P101
1	P102
2	P101
2	P103

Product Table (After 2NF)

Product_ID	Product_Name	Supplier
P101	Laptop	Dell
P102	Mouse	Logitech
P103	Keyboard	HP

3NF (Third Normal Form)

- Should be in 1NF and 2NF
- No non-key dependency
- No transitive dependency (An attribute should depend only on the primary key, not on another non key attribute)

A transitive dependency occurs when a non-key attribute depends on another non-key attribute, which in turn depends on the primary key. In other words, if A depends on B, and B depends on C, then A is transitively dependent on C.

Example Table (Before 3NF)

Student_ID	Subject	Teacher	Teacher_Age
101	Math	Mr. A	45
101	Science	Mr. B	40
102	English	Mr. C	50

Problem

- Teacher_Age depends on Teacher, which in turn depends on Student_ID.
- This means Teacher_Age is transitively dependent on Student_ID through Teacher.
- Solution: Split the table.

Student Table

Student_ID	Subject
101	Math
101	Science
102	English

Teacher Table

Teacher	Subject
Mr. A	Math
Mr. B	Science
Mr. C	English

Teacher Info Table

Teacher	Age
Mr. A	45
Mr. B	40
Mr. C	50

A teacher uses a relational database, MARKS, to store data about students and their test marks.

The database has the following structure:

STUDENT(StudentID, FirstName, LastName)

TEST(TestID, Description, TotalMarks)

STUDENT_TEST(StudentID, TestID, Mark)

Give the highest level of Normal Form (NF) the database MARKS is in **and** justify your choice.

Normal Form .....

Justification .....

.....

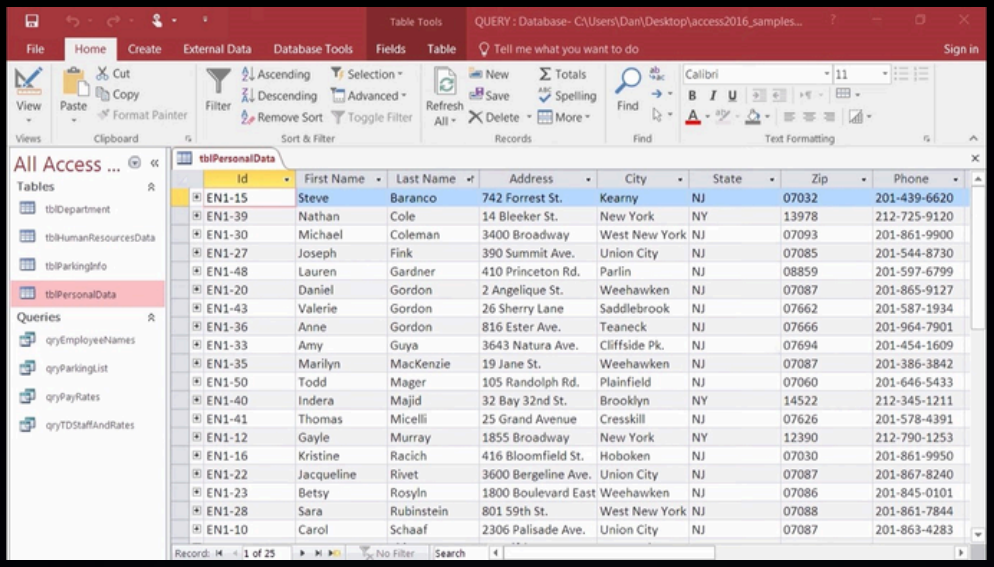
.....

1 mark for 3NF

1 mark per bullet for justification to **max 2**

- There are no repeated attributes // it is already in 2NF
- Each field is fully dependent on the corresponding primary key // no partial dependencies
- No transitive dependencies

Database Management System



DBMS Tools

Developer Interface

- To create user-friendly features, e.g., forms to enter the new booking.
- To create outputs, e.g., report of bookings on a given date.
- To create interactive features, e.g., buttons or menus.

Query Processor

- To create SQL queries.
- To search for data that meets set criteria, e.g., all bookings for next week.
- To perform calculation of extracted data, e.g., number of empty rooms.
- Organizes the results to be displayed.

Features Of DBMS

Describe the purpose and contents of the data dictionary in the DBMS

- stores all the information about the database (metadata) e.g
- field / attribute names
- table name
- validation rules
- data types
- primary keys and foreign keys
- relationships

Describe levels of schema of database ?

External Schema

- The individual's view of the database

Conceptual Schema

- Describes the views which user of the database might have.

Logical Schema

- Describes how the relationships will be implemented in the logic structure of the database.

Physical / Internal Schema

- Describes how the data will be stored on the physical media.

Describe what is meant by logical schema ?

- the overview of a database structure
- models the problem / situation
- by using methods such as an ER diagram
- independent of any particular DBMS

Logical Schema is the feature that describes the relationship between data and its structure.

Describe what is meant by Data Integrity and how it is implemented in a database ?

- Data integrity is a method of making sure the data is consistent
- This can be implemented by enforcing referential integrity
- if data in one table is deleted or updated all table are updated (Cascading Update/Delete)
- Validation and Verification rules can be followed

What is data redundancy?

- Repeated data

Explain how a relational database can help to reduce data redundancy?

- Because each record of data is stored once and is referenced by a primary key
- Because data is stored in individual tables
- And the tables are linked by relationships
- By the proper use of primary and foreign keys
- By enforcing referential integrity
- By going through the normalization process

Describe the methods that a DBMS can use to improve the security of the database ?

Backup / recovery procedures

- **Automatically takes copies of the database and stores off-site on a regular basis (weekly, etc.).**
- **So that the data can be recovered if lost.**

Use of access rights

- **Some users are given different access permissions to different tables.**
- **Read/write, read-only, full access, etc.**

Views

- **Different users are able to see different parts of the database.**
- **Only see what users need to see (by example).**

Record and table locking

- **Prevents simultaneous access to data.**
- **So updates are not lost / data is not overwritten.**

Encryption

- **The data is turned into ciphertext.**
- **So it cannot be understood without a decryption key.**

Authentication

- **Verifies user identity with credentials like passwords or biometrics.**
- **Uses OTP, security tokens, or multi-factor authentication for extra security.**

Tools In DBMS

Query Processor

- Allows the user to enter criteria
- Searches for the data that meets the entered criteria
- Organizes the results to be displayed to the user

Developer Interface

- To create user-friendly features, e.g., forms to enter the new booking.
- To create outputs, e.g., report of bookings on a given date.
- To create interactive features, e.g., buttons or menus.

Tasks performed by DBMS developer interface:

- Create a table
- Set up relationships between tables
- Create a form
- Create a report
- Create a query

Database

Question 1

7 Bobby and Kim are discussing databases.

(a) Bobby tells Kim that a file-based approach is usually better than a relational database.

Explain why Bobby is incorrect.

.....

.....

.....

.....

.....

..... [3]

(b) Bobby has a shop that sells products to customers. His database will store data about his customers, their payment details, orders and the products he sells. Customers will have login details to access their accounts. The database will update customers' payment and login details without keeping any historical records.

(i) Give **one** example of each of the following relationships from Bobby's database.

one-to-one

.....

.....

one-to-many

.....

.....

many-to-many

.....

.....

[3]

(ii) Tick (✓) **one** box to identify the relationship that cannot be directly implemented in a normalised relational database.

Relationship	Tick (✓)
one-to-one	
one-to-many	
many-to-many	

[1]

(iii) Bobby wants to name his database `SHOPORDERS`.

Write a Data Definition Language (DDL) statement to define a new database with the name `SHOPORDERS`.

.....

..... [1]

(c) A database has a data dictionary.

Give **three** items that are stored in a data dictionary.

1 .....

2 .....

3 .....

[3]

Question 2

1 Raj owns houses that other people rent from him. He has a database that stores details about the people who rent houses, and the houses they rent. The database, HOUSE_RENTALS, has the following structure:

CUSTOMER(CustomerID, FirstName, LastName, DateOfBirth, Email)

HOUSE(HouseID, HouseNumber, Road, Town, Bedrooms, Bathrooms)

RENTAL(RentalID, CustomerID, HouseID, MonthlyCost, DepositPaid)

(a) Give the definition of the following database terms, using an example from the database HOUSE_RENTALS for each definition.

Term	Definition and example
Field	....
Entity	....
Foreign key	....

[6]

(b) Tick (✓) **one** box to identify whether the database HOUSE_RENTALS is in Third Normal Form (3NF) or not in 3NF.
Justify your choice using one or more examples from the database HOUSE_RENTALS.

In 3NF	
Not in 3NF	

Justification .....

.....

.....

..... [2]

(c) Example data from the table RENTAL are given:

RentalID	CustomerID	HouseID	MonthlyCost	DepositPaid
1	22	15B5L	1000.00	Yes
2	13	3F	687.00	No
3	1	12AB	550.00	Yes
4	3	37	444.50	Yes

(i) Complete the following Data Definition Language (DDL) statement to define the table RENTAL.

```
CREATE ..... ..... (
      RentalID INTEGER NOT NULL,
      CustomerID INTEGER NOT NULL,
      HouseID ..... (5) NOT NULL,
      MonthlyCost ..... NOT NULL,
      DepositPaid BOOLEAN NOT NULL,
      ..... (RentalID)
);
```

[4]

(ii) Write a Data Manipulation Language (DML) script to return the first name and last name of all customers who have **not** paid their deposit.

.....

.....

.....

.....

.....

..... [4]

Question 3

5 Javier owns many shops that sell cars. He employs several managers who are each in charge of one or more shops. He uses the relational database `CARS` to store the data about his business.

Part of the database is shown:

```
SHOP(ShopID, ManagerID, Address, Town, TelephoneNumber)

MANAGER(ManagerID, FirstName, LastName, DateOfBirth, Wage)

CAR(RegistrationNumber, Make, Model, NumberOfMiles, ShopID)
```

(a) Tick (✓) **one** box in each row to identify whether each field is a primary key or a foreign key.

Table	Field name	Primary key	Foreign key
MANAGER	ManagerID		
SHOP	ManagerID		
CAR	RegistrationNumber		
CAR	ShopID		

[2]

(b) Describe the ways in which access rights can be used to protect the data in Javier's database from unauthorised access.

.....

.....

.....

.....

.....

..... [3]

(c) Javier uses Data Definition Language (DDL) and Data Manipulation Language (DML) statements in his database.

(i) Complete the following DML statements to return the number of cars for sale in each shop.

```
SELECT COUNT(.....)

FROM .....

..... ShopID
```

[3]

(ii) Complete the DML statement to include the following car in the table `CAR`.

Field	Data
RegistrationNumber	123AA
Make	Tiger
Model	Lioness
NumberOfMiles	10500
ShopID	12BSTREET

```
..... ..... CAR

..... ("123AA","Tiger","Lioness",10500,"12BSTREET")
```

[2]

Question 4

6 A shop sells plants to customers. The shop manager has a relational database to keep track of the sales.

The database, PLANTSALES, has the following structure:

PLANT(PlantName, QuantityInStock, Cost)

CUSTOMER(CustomerID, FirstName, LastName, Address, Email)

PURCHASE(PurchaseID, CustomerID)

PURCHASE_ITEM(PurchaseID, PlantName, Quantity)

(a) The database is normalised.

(i) The table lists the following three stages of normalisation:

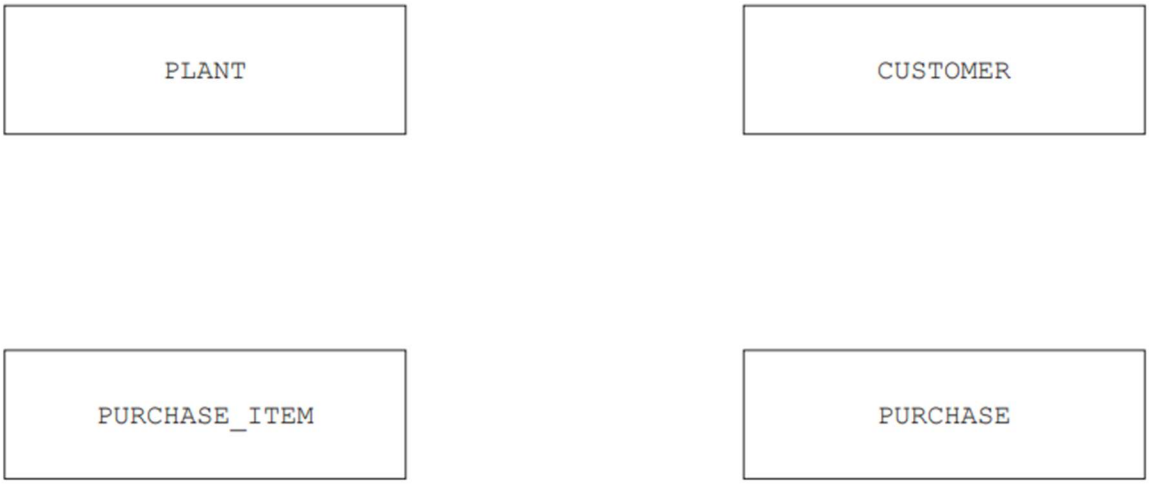
- The first stage is from a database that is not normalised (0NF) to First Normal Form (1NF).
- The second stage is from 1NF to Second Normal Form (2NF).
- The third stage is from 2NF to Third Normal Form (3NF).

Tick (✓) **one** box in each row to identify the appropriate stage for each task.

Task	Normalisation stage		
	0NF to 1NF	1NF to 2NF	2NF to 3NF
Remove any partial key dependencies			
Remove any repeating groups of attributes			
Remove any non-key dependencies			

[2]

(ii) Draw an entity-relationship (E-R) diagram for the database PLANTSALES.



[3]

(b) The shop manager uses a Database Management System (DBMS).

Describe the purpose **and** contents of the data dictionary in the DBMS.

.....

.....

.....

.....

.....

..... [3]

(c) The shop manager uses both Data Definition Language (DDL) and Data Manipulation Language (DML) statements to create and search the database.

(i) Complete the DML statements to return the total number of items purchased with the purchase ID of 3011A.

```
SELECT SUM(.....)

FROM .....

WHERE ..... = .....;
```

[4]

(ii) Write DDL statements to include a field in the table `PURCHASE` to store the date of the order.

.....

.....

.....

..... [3]

Question 5

4 A teacher uses a relational database, `MARKS`, to store data about students and their test marks.

The database has the following structure:

`STUDENT(StudentID, FirstName, LastName)`

`TEST(TestID, Description, TotalMarks)`

`STUDENT_TEST(StudentID, TestID, Mark)`

(a) Describe the advantages of using a relational database compared to a file-based approach.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(b) Give the highest level of Normal Form (NF) the database MARKS is in **and** justify your choice.

Normal Form .....

Justification .....

.....

.....

[3]

(c) **(i)** Sample data to be stored in the table STUDENT_TEST is shown.

StudentID	TestID	Mark
12	A1	50
12	P10	100
13	A1	75
14	P10	60

Write a Structured Query Language (SQL) script to create the table STUDENT_TEST.

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(ii) Write a Structured Query Language (SQL) script to find the average mark of students in test A7.

.....

.....

.....

.....

.....

..... [3]

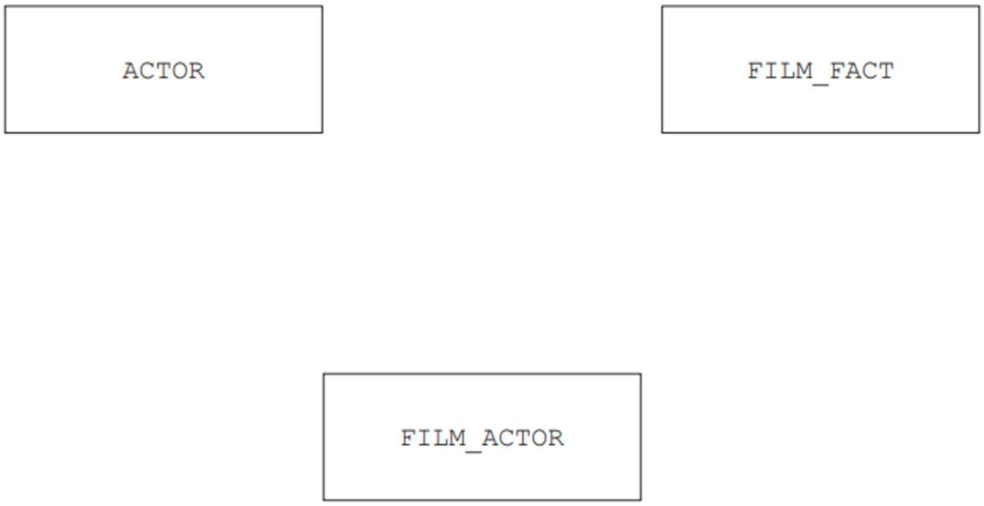
Question 6

5 A database, FILMS, stores information about films and actors.

Part of the database is shown:

ACTOR(ActorID, FirstName, LastName, DateOfBirth)
FILM_FACT(FilmID, FilmTitle, ReleaseDate, Category)
FILM_ACTOR(ActorID, FilmID)

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



[2]

(b) A composite primary key consists of two or more attributes that together form the primary key.

Explain why the table `FILM_ACTOR` has a composite primary key.

.....

.....

.....

..... [2]

(c) Complete the SQL script to return the IDs of all the actors in the film with the title Cinderella.

```
SELECT .....
FROM FILM_ACTOR
INNER JOIN .....
ON FILM_FACT.FilmID = .....
WHERE FILM_FACT.FilmTitle = .....;
```

[4]

(d) Write an SQL script to count the number of films that were released in January 2022.

.....

.....

.....

.....

.....

..... [3]

(e) A Database Management System (DBMS) is used to create and manipulate the database.

Complete the descriptions of the features and tools found in a DBMS using the given terms. Not all terms will be used.

Boolean	**data dictionary**	**data redundancy**	**field names**
input	**interface**	**logical schema**	**normalisation**
operating system	**output**	**primary keys**	**query**
structure			

A DBMS provides data management. This includes the development of a

..... that stores information about the data stored, such as

..... and ..... .

The ..... uses methods, such as an E-R diagram, to show the structure of the database and its relationships.

The ..... processor allows a user to perform searches to find specific data. The DBMS also provides a developer ..... that allows the user to create tables, forms and reports.

[6]

Question 7

6 A relational database, TECHNOLOGY, stores data about the staff in a company and the computer devices used by the staff.

The database has the following tables:

STAFF(StaffID, FirstName, LastName, DateOfBirth, JobTitle)

DEVICE(DeviceID, Type, DatePurchased, StaffID)

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

.....

.....

.....

.....

.....

.....

.....

..... [4]

(b) The database uses a Data Definition Language (DDL) and Data Manipulation Language (DML).

(i) Complete the SQL script to return the number of devices stored in the database for the staff member with the first name 'Ali' and last name 'Khan'.

```
SELECT ..... (STAFF.StaffID)

FROM .....

INNER JOIN DEVICE

..... STAFF.StaffID = DEVICE.StaffID

WHERE STAFF.FirstName = 'Ali'

..... STAFF.LastName = 'Khan';
```

[4]

(ii) The table `DEVICE` needs a new attribute to store whether the device has been returned by the staff member, or not.

Write a Structured Query Language (SQL) script to insert the new attribute into the table `DEVICE`.

.....

.....

.....

..... [2]

(c) The database is in Third Normal Form (3NF).

Complete the table by describing the three normal forms.

Normal Form	Description
First Normal Form (1NF)	....
Second Normal Form (2NF)	....
Third Normal Form (3NF)	....

[3]

Question 8

4 A photographer creates a relational database to store data about photographs taken at birthday parties.

The database, PHOTOGRAPHS, stores details of the customer, the party, the photographs taken and the cameras used.

The photographer has several cameras that are used for taking the photographs at the parties.

Each camera has a specific lens type (for example, XY32Z) and lighting type (for example, F1672).

Data about each photograph is stored in the database including the party at which it was taken, the time it was taken and the camera used.

The database has these four tables:

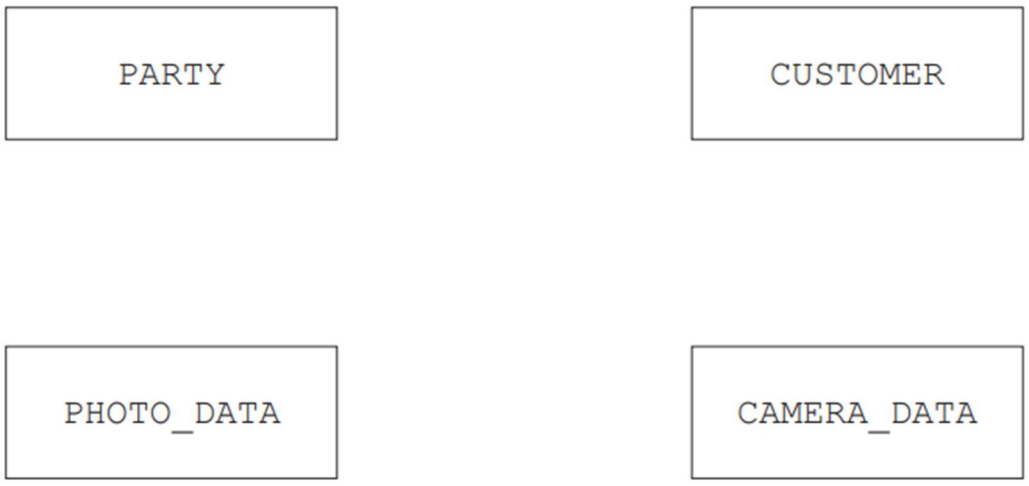
CUSTOMER(CustomerID, FirstName, LastName, Telephone)

PARTY(PartyID, CustomerID, PartyDate, StartTime)

PHOTO_DATA(PhotoID, PartyID, TimeTaken, CameraID)

CAMERA_DATA(CameraID, LensType, LightingType)

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



[3]

(b) The database is normalised and is in Third Normal Form (3NF).

Describe the characteristics of a database that is in Third Normal Form (3NF).

.....

.....

.....

.....

.....

..... [3]

(c) The table shows some sample data for the table `PHOTO_DATA`.

PhotoID	PartyID	TimeTaken	CameraID
ST23-56	BD987	08:34	NIK-02
ST23-57	BD987	08:55	NIK-02
ST23-60	BC08	09:01	CAN-01
ST23-61	BC08	10:23	CAN-12
ST23-62	BC08	10:56	NIK-01

(i) State what is meant by a **tuple**. Give an example of a tuple from `PHOTO_DATA`.

Tuple .....

.....

Example .....

.....

[2]

(ii) Complete the Structured Query Language (SQL) script to display the total number of photographs that have been taken using a camera with a camera ID starting with `CAN`.

```
SELECT ..... (.....)
FROM .....
WHERE CameraID LIKE ..... ;
```

[4]

(d) Write an SQL script to include two new fields in `CAMERA_DATA` to store the number of photographs currently on the camera **and** the date the camera was last used.

.....

.....

.....

.....

.....

..... [3]

Question 9

5 A relational database, GARDEN, has the following tables:

OWNER(OwnerID, FirstName, TelephoneNo, TreeID, TreePosition)

TREE(TreeID, ScientificName, MaxHeight, FastGrowing)

(a) The database is **not** in Third Normal Form (3NF).

Explain how the database can be normalised to 3NF.

.....

.....

.....

.....

.....

..... [3]

(b) Write the Structured Query Language (SQL) script to add a new record in the table TREE to store the following data:

Attribute	Value
TreeID	LOW_1276
ScientificName	Salix_Alba
MaxHeight	30.00
FastGrowing	TRUE

.....

.....

.....

..... [3]

(c) State what is meant by a **candidate key** in a relational database.

.....

..... [1]

(d) (i) Describe, using an example, what is meant by a **data dictionary**.

.....

.....

.....

..... [2]

(ii) Describe what is meant by a **logical schema**.

.....

.....

.....

..... [2]

Question 10

2 The relational database ASTRONOMY is used to store data about telescopes, the companies that own the telescopes and the photographs taken by the telescopes.

The database has these three tables:

```
COMPANY(TelephoneNumber, CompanyID, CompanyName)

PHOTOGRAPH(PhotoID, TelescopeID, DateTaken, TimeTaken, Elevation)

TELESCOPE(TelescopeID, CompanyID, SerialNumber)
```

(a) Complete the following table by writing the correct answer for each item.

Item	Answer
a suitable field for the primary key in COMPANY	
a candidate key in TELESCOPE	
the degree of relationship between TELESCOPE and PHOTOGRAPH	

[3]

(b) A Database Management System (DBMS) has several features.

Identify the feature that describes the relationship between data and its structure.

..... [1]

(c) Complete the SQL script to return the total number of telescopes owned by the company whose ID begins with HW.

```
SELECT ..... (.....)

FROM TELESCOPE

WHERE ..... LIKE .....;
```

[4]

(d) Write the SQL script to add **one** field to the table `PHOTOGRAPH` to store the resolution of the photograph, e.g. 1920 × 1068.

.....

.....

.....

..... [2]

(f) Describe the purpose of a query processor in a DBMS.

.....

.....

.....

..... [2]

Question 11

2 An organisation uses a database to store data about the types of bird that people have seen.

(a) The database is managed using a Database Management System (DBMS).

(i) State what is meant by a data dictionary **and** give **one** example of an item typically found in a data dictionary.

Definition .....

.....

Example .....

.....

[2]

(ii) State what is meant by data integrity **and** give **one** example of how this is implemented in a database.

Definition .....

.....

Example .....

.....

[2]

(b) The database, `Birds`, stores information about the types of bird and the people who have seen them.

Data about each bird seen is stored with its location and data about the person who saw the bird.

Database `Birds` has the following tables:

```
BIRD_TYPE(BirdID, Name, Size)

BIRD_SEEN(SeenID, BirdID, Date, Location, PersonID)

PERSON(PersonID, FirstName, LastName, EmailAddress)
```

(i) Complete the table by identifying **two** foreign keys and the database table where each is found.

Foreign key	Database table

[2]

(ii) The database `Birds` has been normalised.

Draw **one** line from each Normal Form to the most appropriate definition.

Normal Form	Definition
First Normal Form (1NF)	All fields are fully dependent on the primary key.
Second Normal Form (2NF)	There are no repeating groups of attributes.
Third Normal Form (3NF)	There are no partial dependencies.

[1]

(iii) Part of the database table `BIRD_TYPE` is shown:

BirdID	Name	Size
0123	Blackbird	Medium
0035	Jay	Large
0004	Raven	Large
0085	Robin	Small

The database only supports these data types:

- character
- varchar
- Boolean
- integer
- real
- date
- time

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

.....

.....

.....

.....

.....

.....

.....

..... [4]

(iv) The database tables are repeated here for reference:

```
BIRD_TYPE(BirdID, Name, Size)

BIRD_SEEN(SeenID, BirdID, Date, Location, PersonID)

PERSON(PersonID, FirstName, LastName, EmailAddress)
```

Complete the SQL script to return the number of birds of each size seen by the person with the ID of J_123.

```
SELECT BIRD_TYPE.Size, ..... (BIRD_TYPE.BirdID)
       AS NumberOfBirds
FROM BIRD_TYPE, .....
WHERE ..... = "J_123"
AND BIRD_TYPE.BirdID = .....
..... BIRD_TYPE.Size;
```

[5]

Question 12

2 A horse riding school uses a database, `Lessons`, to store data about lesson bookings.

This database is created and managed using a Database Management System (DBMS).

(a) The table contains names and descriptions of DBMS features and tools.

Complete the table by writing down the missing names and descriptions.

Name	Description
Data dictionary	....
Query processor	....
....	A model of a database that is not specific to one DBMS.
....	A software tool that allows the user to create items such as tables, forms and reports.

[4]

(b) Explain the reasons why referential integrity is important in a database.

.....

.....

.....

.....

.....

..... [3]

(c) The database `Lessons` has the following tables:

`HORSE(HorseID, Name, Height, Age, HorseLevel)`

`STUDENT(StudentID, FirstName, LastName, RiderLevel, PreferredHorseID)`

`LESSON(LessonID, Date, Time, StudentID, HorseID, LessonContent)`

Dates in this database are stored in the format `#DD/MM/YYYY#`.

The fields `RiderLevel` and `HorseLevel` can only have the values: `Beginner`, `Intermediate` or `Advanced`.

(i) Describe **two** methods of validating the field `RiderLevel`.

1 .....

.....

.....

2 .....

.....

.....

[2]

(ii) Write a Structured Query Language (SQL) script to return the names of all the horses that have the horse level intermediate or beginner.

.....

.....

.....

.....

.....

.....

.....

..... [4]

(iii) The following SQL script should return the number of riders that have the rider level beginner and have a lesson booked on 09/09/2023.

```
SELECT SUM(STUDENT.RiderLevel) AS NumberOfRiders

FROM STUDENT, LESSON

WHERE StudentID = StudentID

OR Date = #09/09/2023#

AND STUDENT.RiderLevel = Beginner;
```

There are **four** errors in the script.

Identify **and** correct each error.

1 .....

.....

2 .....

.....

3 .....

.....

4 .....

.....

[4]

Question 13

4 A shop rents cars to customers. The shop uses a relational database to store information about the rentals.

(a) Describe **two** ways in which a relational database addresses the limitations of a file-based approach.

1 .....

.....

.....

.....

2 .....

.....

.....

.....

[4]

(b) Complete the table by writing the missing term or description for each database feature.

Term	Description
....	An object that data is stored about.
Tuple	....
Secondary key	....
....	A field in one table that is linked to a primary key in another table.

[4]

(c) The car rental database is not normalised. The current database design is:

```
BOOKING(CarRegistration, StartDate, EndDate,
        CarModel, CarColour, CustomerFirstName)

CUSTOMER(CustomerFirstName, CustomerLastName, EmailAddress,
         TelephoneNumber)
```

Write a normalised database design for this database.

All tables must be in Third Normal Form (3NF).

Use the field names given **and** underline the primary key fields.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

Question 14

3 A shop manager has designed a relational database to store customer orders.

The database will have the following tables:

CUSTOMER(CustomerID, FirstName, LastName, Town)

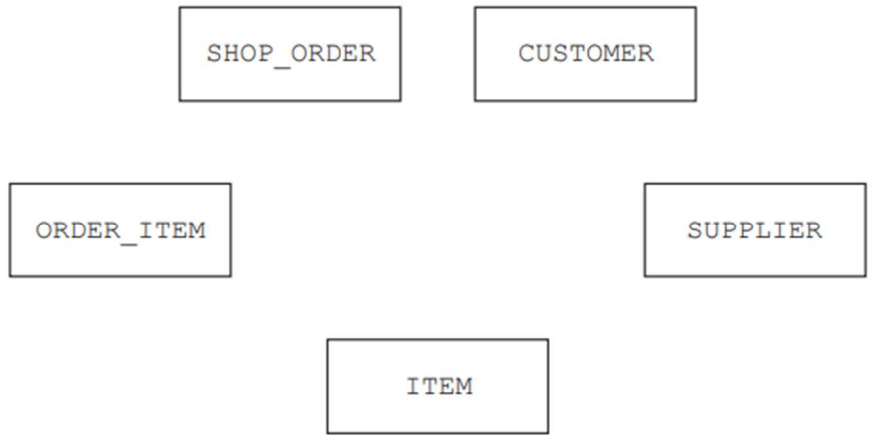
SHOP_ORDER(OrderNo, CustomerID, OrderDate)

SUPPLIER(SupplierID, EmailAddress, TelephoneNumber)

ITEM(ItemNumber, SupplierID, Description, Price)

ORDER_ITEM(ItemNumber, OrderNo, Quantity)

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



[3]

(b) Identify **three** advantages of a relational database compared to a file-based approach.

1 .....

.....

2 .....

.....

3 .....

.....

[3]

(c) (i) Write a Structured Query Language (SQL) script to define the database called `SHOP`.

.....

..... [1]

(ii) Write the SQL script to return the total quantity of items that the customer with the ID of HJ231 has ordered.

.....

.....

.....

.....

.....

..... [4]

Question 15

2 (a) State what is meant by the following terms in a relational database model.

Entity .....

.....

.....

Primary key .....

.....

.....

Referential integrity .....

.....

.....

[3]

(b) Authentication is one method a Database Management System (DBMS) can use to improve the security of a database.

Describe **other** methods that a DBMS can use to improve the security of a database.

.....

.....

.....

.....

.....

.....

.....

..... [4]

(c) The following database table is not normalised.

StudentName	DateOfBirth	TutorGroup	Subject	SubjectCode
Yuwei Chen	01/09/2004	SMH	English, Maths, Computer Science	EN, MA, CS
Claudia Raj	23/02/2005	JMB	Maths, Physics, Art	MA, PY, AR
Aamil Akram	24/01/2005	KMB	Art, Design, English language	AR, DE, EN
Areeba Faraz	21/12/2004	SMH	English language, Chemistry, Design	EN, CH, DE

Explain how to modify the table to put it into First Normal Form (1NF).

.....

.....

.....

.....

.....

.....

.....

..... [4]

Question 16

(b) The company uses a database, COURSES, to store data about the courses and their tutors.

Each course starts at different times of the year and may have a different tutor.

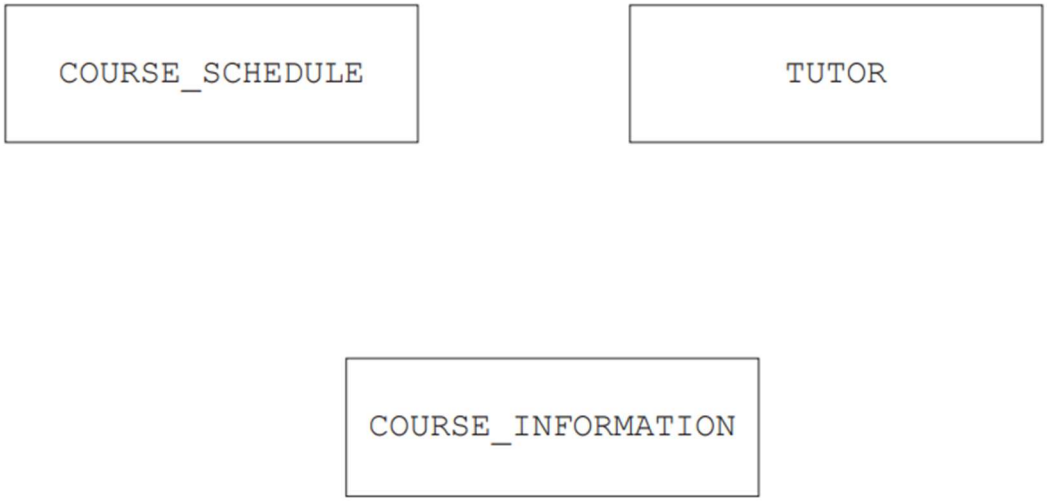
The database has the following structure:

COURSE_INFORMATION(CourseID, Description, Cost)

TUTOR(TutorID, TelephoneNumber, EmailAddress, TutorName)

COURSE_SCHEDULE(CourseID, DateStarted, TutorID)

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



[1]

(ii) Write the Structured Query Language (SQL) script to return the total number of courses that have started after 9 September 2023.

The value returned must have an appropriate field name.

.....

.....

.....

.....

.....

..... [4]

(c) An example of a tutor ID is NK16C6.

An administrative officer enters the tutor ID into the `TUTOR` table.

Explain how data verification can be used when the tutor ID is entered.

.....

.....

.....

.....

.....

.....

.....

..... [4]

Question 17

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]

Question 18

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]

Question 19

4 A theatre wants to use a database to store data about the shows that are scheduled, their customers and the seats that the customers have booked.

In the theatre:

- Each show can take place on multiple dates.
- Some dates can have more than one performance.
- There are multiple rows of seats.
- Each seat can be individually booked by its row letter and seat number, for example row E seat 2.

Part of the database design includes these tables:

```
SHOW(ShowID, Title, Duration)
SEAT(SeatID, RowLetter, SeatNumber)
PERFORMANCE(PerformanceID, ShowID, ShowDate, StartTime)
```

(a) Identify the relationship between the tables `PERFORMANCE` and `SHOW`.

.....

..... [1]

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

PerformanceID	ShowID	ShowDate	StartTime
0001	MK12	5/5/2025	13:00
0002	MK12	5/5/2025	19:30
0003	MK12	6/5/2025	19:00
0004	OP3	7/5/2025	18:30
0005	OP3	8/5/2025	18:30
0006	OP3	9/5/2025	13:00

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

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(c) Write an SQL script to return the number of times each show is scheduled. For example, in the sample data in part **(b)**, the show MK12 is scheduled three times.

The result needs to include the show name and a suitable field name for the number of times it is scheduled.

.....

.....

.....

.....

.....

.....

.....

..... [4]

(d) Customers give their first name, last name and email address when they are making a booking. One booking can include multiple seats.

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

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

Question 20

A driving school teaches people how to drive cars. The school has a relational database, DRIVING_SCHOOL, to store information about instructors, students, lessons and the cars used by instructors.

INSTRUCTOR(InstructorID, FirstName, LastName, DateOfBirth, Level)

CAR(Registration, Make, Model, EngineSize)

INSTRUCTOR_CAR(InstructorID, Registration)

STUDENT(StudentID, FirstName, LastName, DateOfBirth, Address1)

LESSON(LessonID, StudentID, InstructorID, LessonDate, LessonTime)

(a) Give **two** benefits to the **driving school** of using a relational database instead of a flat file.

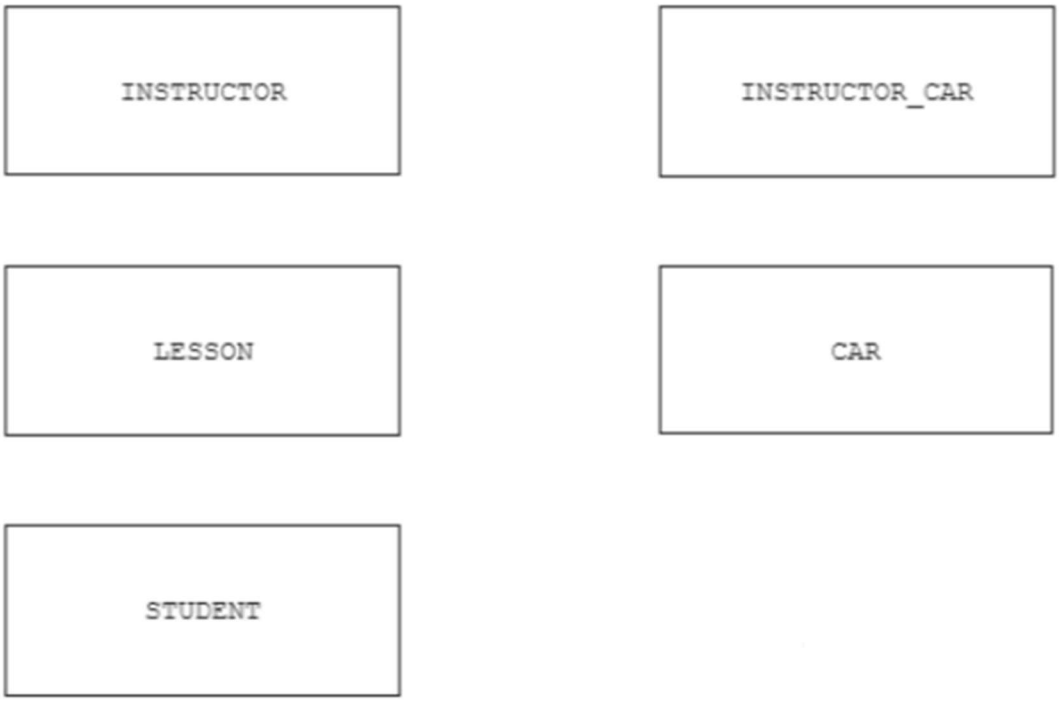
1 .....

.....

2 .....

..... [2]

(b) Complete the entity-relationship diagram for the database DRIVING_SCHOOL.



(c) The table shows some sample data for the table `INSTRUCTOR`.

InstructorID	FirstName	LastName	DateOfBirth	Level
Ins01	Jayden	Han	05/06/1974	1
Ins02	Freda	Choi	06/02/1978	2
Ins03	Kelly	Kim	01/12/1966	1
Ins04	Santana	Thompson	09/09/1985	3

Complete the Data Definition Language (DDL) statement to create the table `INSTRUCTOR`.

```
..... TABLE INSTRUCTOR(
    InstructorID VARCHAR(5),
    FirstName VARCHAR(15),
    LastName VARCHAR(15),
    DateOfBirth DATE,
    Level ..... ,
    ..... (InstructorID)
);
```

[3]

(d) The table `STUDENT` needs an additional field to store the student's telephone number, for example 012-3456.

Write a Data Definition Language (DDL) statement to add the new field to the table `STUDENT`.

.....

.....

.....

..... [2]

(e) Write a Data Manipulation Language (DML) statement to return the date and time of all future lessons booked with the instructor whose InstructorID is Ins01.

.....

.....

.....

.....

.....

.....

.....[4]

Question 21

A software development company has a relational database, SOFTWARE_MANAGEMENT. The database stores details of the customers who have purchased software, as well as the software and licences that customers have purchased.

The SOFTWARE_MANAGEMENT database has the following tables:

```
CUSTOMER_DETAILS(CustomerID, CompanyName, Address1, Address2, City)

SOFTWARE_PURCHASED(SoftwareName, SoftwareDescription, CustomerID,
                   LicenceType, LicenceCost, RenewalDate)
```

(a) Explain why this database is **not** in Third Normal Form (3NF). Refer to the tables in your answer.

Do **not** attempt to normalise the tables.

.....

.....

.....

.....

.....

..... [2]

(b) Give an example from the database SOFTWARE_MANAGEMENT for each of the following database terms.

Term	Example
Entity	
Foreign key	
Attribute	

[3]

(c) The company also develops computer games. They extend the relational database SOFTWARE_MANAGEMENT by adding a new table. The new table, GAME_DEVELOPMENT, stores details about the games and the software development teams creating them.

The table shows example data in GAME_DEVELOPMENT.

GameName	Genre	TeamNumber	DevelopmentStage	ManagerID
Bunny Hop	Platform	4	Analysis	23KP
Fried Eggs	Retro	2	Programming stage 1	9RTU
Create-a-game	Action	1	Acceptance testing	11TF

(i) Complete the Data Definition Language (DDL) statement to create the table GAME_DEVELOPMENT.

```
CREATE ..... ..... (
  GameName VarChar,
  Genre VarChar,
  ..... ..... ,
  DevelopmentStage VarChar,
  ManagerID VarChar,
  ..... (GameName)
);
```

[5]

(ii) Another table, PRODUCT_MANAGER, is created.

PRODUCT_MANAGER(ManagerID, FirstName, LastName)

Complete the Data Manipulation Language (DML) statement to return the game name, genre and team number of all games managed by the product manager with the first name 'James' and the last name 'Fitz'.

```
..... GameName, Genre, TeamNumber
FROM GAME_DEVELOPMENT, PRODUCT_MANAGER
WHERE PRODUCT_MANAGER.FirstName = "James"
AND PRODUCT_MANAGER.LastName = "Fitz"
AND .....
    = ..... ;
```

[3]

Question 22

6 Sheila creates a relational database for her hotel using a Database Management System (DBMS).

(a) Draw **one** line from each database term to its most appropriate description.

Database Term	Description
Primary key	A field in one table that links to a primary key in another table
Attribute	A collection of records and fields
Foreign key	The type of data that is being stored
Entity	A unique identifier for each tuple
	A data item, represented as a field within a table
	The concept or object in the system that we want to model and store information about

[4]

(b) Identify three tasks that Sheila can perform using the DBMS developer interface.

1 .....

2 .....

3 .....

(c) Sheila creates the database HOTEL with the following table structure:

ROOM(RoomNumber, RoomType)

BOOKING(BookingID, RoomNumber, CustomerID, StartDate)

CUSTOMER(CustomerID, FirstName, LastName, Address, Tel_Num)

(i) The following table shows some sample data for the table ROOM.

RoomNumber	RoomType
1	Standard
2	Double
3	Executive
4	Standard

Complete the Data Definition Language (DDL) statement to create the table ROOM.

```
..... TABLE ROOM(
  RoomNumber Integer,
  RoomType .....,
  ..... (RoomNumber)
);
```

[3]

(ii) Room number **5** is a **Double** room.

Complete the Data Manipulation Language (DML) statement to add the details for room number 5 to the table ROOM.

```
INSERT ..... ROOM
VALUES(.....);
```

[2]

(iii) The table BOOKING needs an additional field to store the number of nights (for example, 3) a customer is staying.

Write a Data Definition Language (DDL) statement to add the new field to the table BOOKING.

.....

.....

.....

.....

..... [2]

Question 23

(b) The software company stores information about customers and the software licences they have purchased. The company considers a file-based approach for the storage and retrieval of data.

(i) Give three limitations of a file-based approach to store the data.

1 .....

.....

2 .....

.....

3 .....

..... [3

(ii) The software company decides to use a database to overcome the limitations of a file-based system. Some of these limitations are addressed through the logical schema. Name and describe two levels of the schema of a database.

Name 1 .....

Description .....

.....

Name 2 .....

Description .....

..... [4]

(c) The database has the following tables:

```
CUSTOMER(CustomerID, CompanyName)

SOFTWARE(SoftwareID, SoftwareName, OperatingSystem, Description)

LICENCE(LicenceID, CustomerID, SoftwareID, DateOfPurchase,
        LicenceType, Cost, ExpiryDate)
```

(i) Identify the type of relationship that exists between the tables `CUSTOMER` and `LICENCE`.

.....

..... [1]

(ii) Describe how the relationship is created between the tables CUSTOMER and LICENCE.

.....

.. ..

.....

..... [2

(iii) The company needs a list of all software licences that have an expiry date on or before 31/12/2019.

Write an SQL query to return the fields `CustomerID`, `SoftwareID`, `LicenceType`, `Cost` and `ExpiryDate` for all licences that expire on, or before 31/12/2019. Group the output by `CustomerID`, and in ascending order of cost.

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

Question 24

A company uses a relational database, EMPLOYEES, to store data about its employees and departments. (a) The company uses a Database Management System (DBMS).

(i) The DBMS has a data dictionary. Describe what the data dictionary stores.

.....

.....

.....

..... [2]

(ii) The DBMS has a query processor. Describe the purpose of a query processor.

.....

.....

.....

..... [2]

(b) Relationships are created between tables using primary and foreign keys. Describe the role of a primary and a foreign key in database relationships.

.....

.....

.....

..... [2]

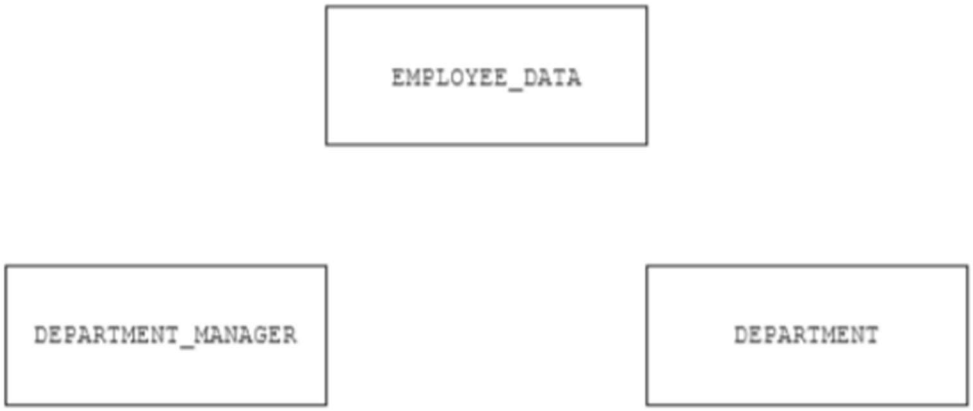
(c) In the company:

- An employee can be a manager.
- A department can have several managers and several employees.
- An employee can only belong to one department.

The EMPLOYEES database has three tables:

```
EMPLOYEE_DATA(EmployeeID, FirstName, LastName, DateOfBirth, Gender,
              DepartmentNumber)
DEPARTMENT(DepartmentNumber, DepartmentName)
DEPARTMENT_MANAGER(DepartmentNumber, EmployeeID, role)
```

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



[3]

(d) Give **three** reasons why the EMPLOYEES database is fully normalised.

1 .....

.....

2 .....

.....

3 .....

..... [3]

(e) Part of the EMPLOYEE_DATA table is shown.

EmployeeID	FirstName	LastName	DateOfBirth	Gender	DepartmentNumber
156FJEK	Harvey	Kim	12/05/1984	Male	S1
558RRKL	Catriona	Moore	03/03/1978	Female	F2
388LMDV	Oscar	Ciao	01/01/1987	Male	F2

(i) Write a Data Definition Language (DDL) statement to create the EMPLOYEES database.

.....

..... [1]

(ii) Write a DDL statement to define the table EMPLOYEE_DATA, and declare EmployeeID as the primary key.

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(iii) Write a Data Manipulation Language (DML) statement to return the first name and last name of all female employees in the department named Finance.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

Question 25

Moheem is creating a relational database to store data about his customers.

(a) Moheem has been told a relational database addresses some of the limitations of a file-based approach by reducing data redundancy.

(i) State what is meant by the term data redundancy.

.....

..... [1]

(ii) Explain how a relational database can help to reduce data redundancy.

.....

.....

.....

.. ..

.....

..... [3]

(b) Moheem uses a Database Management System (DBMS) to ensure the security and integrity of the data.

(i) Explain the difference between security and integrity.

.....

.....

. .....

..... [2]

(ii) Name and describe two security features provided by a DBMS.

Feature 1 .....

.....

.....

.....

Feature 2 .....

.....

.....

.....[4]

(iii) The DBMS provides software tools for the database developer.

Fill in the names of the missing software tools in the following statements.

A ..... allows a developer to extract data from a database.

A ..... enables a developer to create user-friendly forms and reports.

Question 26

(d) The bank uses a relational database, ACCOUNTS, to store the information about customers and their accounts.

The database stores the customer's first name, last name and date of birth.

The bank has several different types of account. Each account type has a unique ID number, name (for example, regular or saving) and bonus (for example, $5.00, $10.00 or $15.00).

A customer can have more than one account.

Each customer's account has its own ID number and stores the amount of money the customer has in that account.

The bank creates a normalised, relational database to store the required information. There are three tables:

- CUSTOMER
- ACCOUNT_TYPE
- CUSTOMER_ACCOUNT

(i) Write the attributes for each table to complete the database design for the bank.

CUSTOMER (.....

.....

.....)

ACCOUNT_TYPE (.....

.....

.....)

CUSTOMER_ACCOUNT (.....

.....

.....) [3]

(ii) Identify the primary key for each table that you designed in part (d)(i).

CUSTOMER .....

ACCOUNT_TYPE .....

CUSTOMER_ACCOUNT ..... [2]

(iii) Identify one foreign key in one of the tables that you designed in part (d)(i).

Table name .....

Foreign key ..... [1]

(iv) The following table has definitions of database terms.

Write the correct database term in the table for each definition.

Definition	Term
All the data about one entity	
The data in one row of a table	
A column or field in a table	

[3]

Question 27

Anushka needs to store information about bookings at a sports club.

(a) Anushka has a file-based storage system. She wants a relational database.

(i) Describe the features of a relational database that address the limitations of Anushka's file-based system. .....

.....

.....

... .

.....

.....

.....

..... [4]

(ii) The relational database design needs to be normalised. The following statements describe the three stages of database normalisation.

Complete the statements by filling in the missing words.

For a database to be in First Normal Form (1NF) there must be no ..... groups of attributes. For a database to be in Second Normal Form (2NF), it must be in 1NF, and contain no ..... key dependencies. For a database to be in Third Normal Form (3NF), it must be in 2NF, and all attributes must be fully dependent on the .....

..... . [4]

(b) The **normalised** relational database, SPORTS_CLUB, has the following table design.

MEMBER(MemberID, FirstName, LastName, MembershipType)

SESSION(SessionID, Description, SessionDate, SessionTime, NumberMembers)

TRAINER(TrainerID, TrainerFirstName, TrainerLastName)

MEMBER_SESSION(MemberID, SessionID)

SESSION_TRAINER(SessionID, TrainerID)

(i) Anushka has designed an entity-relationship (E-R) diagram for SPORTS_CLUB.

Complete the entity-relationship (E-R) diagram.

[2]

(ii) Anushka first needs to create the database that she has designed. Write a Data Definition Language (DDL) statement to create the SPORTS_CLUB database.

.....

..... [1

(iii) The table shows some sample data for the table SESSION.

SessionID	Description	SessionDate	SessionTime	NumberMembers
21PL	Pilates junior	04/04/2020	18:00	15
13AE	Aerobics senior	04/04/2020	19:00	20
33WG	Weightlifting advanced	04/04/2020	10:00	10

Write a DDL script to create the table SESSION.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(iv) Write a Data Manipulation Language (DML) script to return the first name and last name of all members who have Peak membership type.

.....

.....

.....

.....

..... [3

Question 28

A hotel needs to record information about customers and their bookings.

(a) The hotel has two types of room: double and family. Each room has a unique room number. The hotel stores information about the customers including their name, address and contact details. When a customer books a room, they give the start date and the number of nights they want to stay. If a customer wants more than one room, each room must have a separate booking. Each booking has an ID number.

The hotel creates a normalised, relational database to store the required information. There are three tables:

- CUSTOMER
- ROOM
- BOOKING

(i) Complete the database design for the hotel by writing the attributes for each table.

CUSTOMER (.....

.....

.....)

ROOM(.....

.....

.....)

BOOKING(.....

.....

.....) [3

(ii) Identify the primary key for each table that you designed in part (a)(i).

CUSTOMER .....

ROOM .....

BOOKING ..... [2]

(iii) Identify one foreign key in the tables that you designed in part (a)(i).

Table name .....

Foreign key ..... [1]

(b) The hotel wants to use a Database Management System (DBMS) to set up and manage the database. Describe, using examples, how the hotel can use the following DBMS tools:

Developer interface .....

.....

.....

.....

.....

Query processor .....

.....

.....

.....

..... [5

(c) The following table has four SQL scripts.

Tick (✓) **one** box in each row to identify whether the script is an example of a Data Definition Language (DDL) statement or a Data Manipulation Language (DML) statement.

Script	DDL	DML
`CREATE TABLE FILMS`		
`SELECT FilmID FROM FILMS`		
`ALTER TABLE FILMS ADD PRIMARY KEY (FilmID)`		
`CREATE DATABASE MYDATA`		

[2]

Question 29

7 A social media website has a relational database, WEBDATA, that stores the site's information.

The database has three tables to store users' details, and details of the images and text that they post.

```
USER(UserName, FirstName, SecondName, DateOfBirth)
PHOTO(PhotoID, UserName, Comment, UploadDate)
TEXTPOST(PostID, UserName, DateOfPost, TheText)
```

(a) (i) Explain how the relationship between the tables USER and PHOTO has been implemented.

.....

.. ..

.....

.....[2]

(ii) Draw the entity-relationship (E-R) diagram to show the relationships between the three tables.

[2]

(b) A database administrator decides to enforce referential integrity. Use an example from the database WEBDATA to explain what is meant by referential integrity.

.....

.....

.....

.....

.....

.....[3]

(c) The database has been normalised to Third Normal Form (3NF). Define the three stages of database normalisation.

1NF .....

.....

2NF .....

.....

3NF .....

..... [3

(d) The following shows sample data from the `USER` table.

UserName	FirstName	SecondName	DateOfBirth
gem123	John	Smith	01/01/1995
purpleSky	Muhammed	Ali	23/02/1956
OpenWindow	Sunny	Amir	03/03/1997
bluebird127	Raziya	Bello	04/03/1982

(i) Write an SQL script to create the USER table.

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(ii) The database administrator needs to alter the USER table. A new field, Country,needs to be added.

Write an SQL script to add the field Country to the USER table.

.....

.. ..

.....

.....[2]

Question 30

A company writes applications (apps) for smartphones. The company has a relational database, PURPLEGAME, which stores the information for one of its online game apps.

The database has three tables to store player's details, dates when they have logged into the app and in-app purchase details.

LOGIN(LoginID, PlayerID, Date)

PURCHASE(PurchaseID, PlayerID, PurchaseDate, Cost)

PLAYER(PlayerID, PlayerName, SkillLevel)

(a) Draw the entity-relationship (E-R) diagram to show the relationships between the three tables.

[2]

(b) The database manager is concerned about data integrity.

State what is meant by data integrity. Give an example of how the manager can ensure data integrity in the PURPLEGAME database.

.....

.....

.....

.....[2]

(c) The database designer states that the PURPLEGAME database is in Third Normal Form (3NF).

Tick (✓) **one** box to indicate whether this statement is true or false.

True	False

Justify your choice.

.....

.....

.....

.....

.....

.....[3]

(d) (i) The following table shows some sample data for the PLAYER table.

PlayerID	PlayerName	SkillLevel
fly918	Kylie	3
elephant11	Mehrdad	9
candy22	Suzi	15
greenGrass	Jason	22

Write an SQL script to create the PLAYER table.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(ii) The table, PLAYER, needs to be altered. A new field, DateOfBirth, needs to be added.

Write an SQL script to add the DateOfBirth field to the PLAYER table.

.....

.....

.....

.....[2]

Question 31

A movie theatre has a relational database that stores the movie schedule, and information about the movies. The theatre has several screens that play movies at the same time.

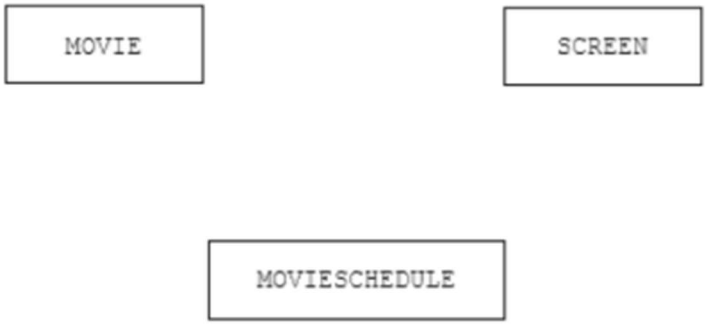
The database has three tables to store information about the movies, the screens and the movie schedule.

MOVIE(MovieID, Title, Length, Rating)

SCREEN(ScreenNumber, NumberSeats)

MOVIESCHEDULE(ScheduleID, MovieID, ScreenNumber, Time)

(a) Complete the entity-relationship (E-R) diagram to show the relationships between these tables.



[2]

(b) Explain how primary and foreign keys are used to link the tables in the movie theatre database.

.....

.....

.....

.....

.....

.....

.....[4]

(c) The database needs to store the name of the company that produced each movie, for example, Rocking Movies. Write an SQL script to add the attribute Production Company to the MOVIE table.

.....

.....

.....

.....

.....[2]

(d) Write an SQL script to display the title and rating of all movies scheduled to play on screen number 3.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[4]

Question 32

1 A hospital is divided into two areas, Area A and Area B. Each area has several wards. All the ward names are different.

A number of nurses are based in Area A. These nurses always work on the same ward. Each nurse has a unique Nurse ID of `STRING` data type.

(a) Describe the relationship shown above.

.....

.....[1]

(b) A relational database is created to store the ward and nurse data. The two table designs for Area A are:

A-WARD(WardName, NumberOfBeds)

A-NURSE(NurseID, FirstName, FamilyName, ...)

(i) Complete the design for the A-NURSE table. [1]

(ii) Explain how the relationship in part (a) is implemented.

.....

.....

.....

.....[2]

(c) In Area B of the hospital, there are a number of wards and a number of nurses.

Each Area B ward has a specialism.

Each Area B nurse has a specialism.

A nurse can be asked to work in any of the Area B wards where their specialism matches with the ward specialism.

The relationship for Area B of the hospital is:

(i) Explain what the degree of relationship is between the entities B-NURSE and B-WARD.

.....

.....[1]

ii) The design for the Area B data is as follows:

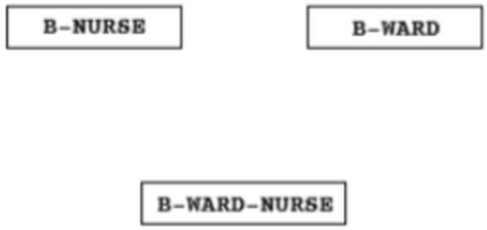
B-NURSE(NurseID, FirstName, FamilyName, Specialism)

B-WARD(WardName, NumberOfBeds, Specialism)

B-WARD-NURSE(.....)

Complete the attributes for the third table. Underline its primary key. [2]

(iii) Draw the relationships on the entity-relationship (E-R) diagram.



[2]

(d) Use the table designs in part (c)(ii).

(i) Write an SQL query to display the Nurse ID and family name for all Area B nurses with a specialism of 'THEATRE'. .....

.....

.....[3]

(ii) Fatima Woo is an Area B nurse with the nurse ID of 076. She has recently married, and her new family name is Chi. Write an SQL command to update her record.

UPDATE .....

SET .....

WHERE ..... [3

Question 33

1 Some shops belong to the Rainbow Retail buying group. They buy their goods from one or more suppliers.

Each shop has:

- a unique shop ID
- a single retail specialism (for example, food, electrical, garden).

Each supplier has:

- a unique supplier ID
- a similar single specialism recorded.

Rainbow Retail creates a relational database to record data about the shops and their suppliers.

The entity-relationship (E-R) diagram for the relationship between the `SHOP` and `SUPPLIER` tables is shown.

(a) Explain what the degree of relationship is between the entities SHOP and SUPPLIER.

.....

.....[1]

The database design is as follows:

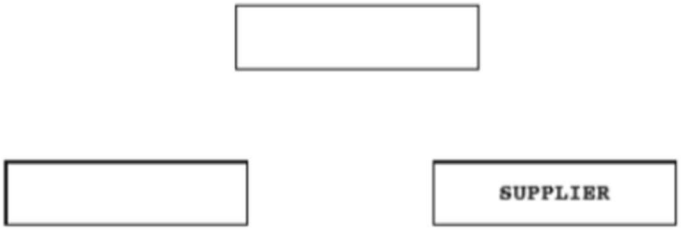
SHOP(ShopID, ShopName, Location, RetailSpecialism)

SUPPLIER(SupplierID, SupplierName, ContactPerson, RetailSpecialism)

SHOP-SUPPLIER(ShopID, SupplierID)

The SHOP–SUPPLIER table stores the suppliers that each shop has previously used. Primary keys are not shown.

b) (i) Label the entities and draw the relationships to complete the revised E-R diagram



[3]

(ii) Complete the following table to show for each database table:

- the primary key
- the foreign key(s) (if any):
  - Each table may contain none, one or more foreign key(s).
  - For a table with no foreign key, write 'None'.
- an explanation for the use of any foreign key.

Table	Primary key	Foreign key(s) (if any)	Explanation
SHOP			
SUPPLIER			
SHOP-SUPPLIER			

[5]

(iii) The database designer has implemented SUPPLIER.ContactPerson as a secondary key.

Describe the reason for this.

.....

.....

.....

.....[2]

(c) (i) Write an SQL query to display the shop ID and location of all shops with a 'GROCERY' specialism.

.....

.....

.....[3]

(ii) The existing shop with ID 8765 has just used the existing supplier SUP89 for the first time.

Write an SQL script to add this data to the database.

.....

.....

.....[3]

Question 34

7 A clinic is staffed by several doctors. The clinic serves thousands of patients. Each day and at any one time, there is only one doctor in the clinic available for appointments.

The clinic stores patient, doctor and appointment data in a relational database.

(a) (i) Underline the primary key for each table in the following suggested table designs.

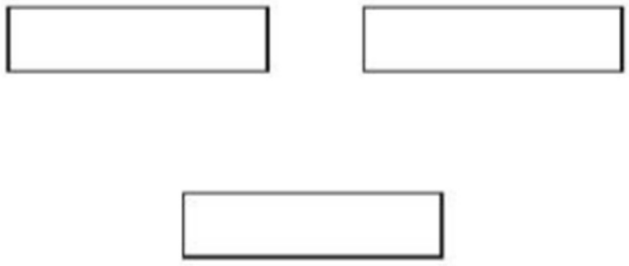
```
PATIENT(PatientID, PatientName, Address, Gender)

DOCTOR(DoctorID, Gender, Qualification)

APPOINTMENT(AppointmentDate, AppointmentTime, DoctorID, PatientID)
```

[2]

(ii) Complete the following entity-relationship (E-R) diagram for this design.



[2]

b) The doctors are concerned that many patients make appointments but do not attend them. Describe the changes to the table designs that could be made to store this information.

.....

.....[2]

(c) The doctors are about to set up a new clinic in the neighbouring village, SITE-B.

The original location is identified as SITE-A.

A new table is designed to store the ID of the doctor who is able to work at each site.

DOCTOR-AVAILABILITY(DoctorID, Site)

Five entries stored in the table are:

DoctorID	Site
098	SITE-A
074	SITE-A
117	SITE-B
098	SITE-B
033	SITE-B

(i) State what this data shows about the availability of the doctor with the ID of 098.

.....

.....[1]

(ii) Opening a new clinic in the neighbouring village will not require any additional table for storing appointments. It will need a change to the existing appointment table design. Show the revised APPOINTMENT table.

APPOINTMENT(.....

.....) [1]

(d) The doctor with the ID of 117 has recently been allocated a new DoctorID of 017.

(i) Write an SQL script to update this doctor's record in the database.

UPDATE .....

SET .....

WHERE ..... [3]

(ii) Describe why this update could cause problems with the existing data stored.

.....

.....

.....[2]

(e) Write an SQL script to display the date and time of all appointments made by the patient with the PatientID of 556.

.....

.....

.....[3

Question 35

A company takes customer service for its clients very seriously. The client • The client names are unique.

A visit

• The company arranges a date for a visit to gather feedback from a client.

• A visit to a client never takes more than one day.

• Over time, the client receives many visits. Staff (Interviewers)

• One or more staff attend the visit.

• If there is more than one staff member visiting, each performs a separate interview. Interviews

• Each interview is classified as either 'general' or by some specialism, for example, marketing, customer service or sales.

• A report is produced for each interview, InterviewText.

• Each interview is conducted by a single staff member. The client, visit, staff and interview data will be stored in a relational database.

a) (i) Underline the primary key for each table in the following suggested table designs.

STAFF(StaffID, StaffName, Department)

CLIENT(ClientName, Address, Town)

VISIT(ClientName, VisitDate)

INTERVIEW(ClientName, VisitDate, StaffID, SpecialistFocus, InterviewText)

[3]

(ii) For each of the pairs of entities, A, B and C, draw the relationship between the two entities.

[3]

(b) The company decides to produce a visit report, VisitReportText, for each visit made. This text will be produced from the one or more interview texts obtained at the visit. State how one or more of the given table designs can be changed to add this attribute.

.....

.....[1]

(c) Client ABC Holdings are now trading under the name of Albright Holdings.

(i) Write an SQL script to update this client's record in the database.

UPDATE .....

SET .....

WHERE ..... [3]

(ii) Describe why this update could cause problems with the existing data stored.

.....

.....

.....[2]

(d) Write an SQL script to display the Staff ID of each member of staff who performed an interview when they visited New Age Toys on 13/10/2016.

.....

.....

.....[3]

(e) At present, all interviews are performed in the UK. Many clients now operate in other countries in Europe. The company wants to perform interviews with the client's staff in other countries. Not all interview staff are willing to travel outside of the UK. State how one or more of the table designs should be revised to store this information.

.....

.....[1]

Question 36

A school stores a large amount of data. This includes student attendance, qualification, and contact details. The school's software uses a file-based approach to store this data.

(a) The school is considering changing to a DBMS.

(i) State what DBMS stands for.

.....[1]

(ii) Describe two ways in which the Database Administrator (DBA) could use the DBMS software to ensure the security of the student data.

1 .....

.....

.....

.....

2 .....

.....

.....

.....[4]

(iii) A feature of the DBMS software is a query processor.

Describe how the school secretary could use this software.

.....

.....

.....

.....[2]

(iv) The DBMS has replaced software that used a file-based approach with a relational database.

Describe how using a relational database has overcome the previous problems associated with a file-based approach.

.....

.....

.....

.....[3]

(b) The database design has three tables to store the classes that students attend.

STUDENT(StudentID, FirstName, LastName, Year, TutorGroup)

CLASS(ClassID, Subject)

CLASS-GROUP(StudentID, ClassID)

Primary keys are not shown. There is a one-to-many relationship between CLASS and CLASS–GROUP.

(i) Describe how this relationship is implemented.

.....

.....

.....[2]

(ii) Describe the relationship between CLASS-GROUP and STUDENT.

.....[1]

(iii) Write an SQL script to display the StudentID and FirstName of all students who are in the tutor group 10B. Display the list in alphabetical order of LastName.

.....

.....

.....

.....[4]

(iv) Write an SQL script to display the LastName of all students who attend the class whose ClassID is CS1.

.....

.....

.....

.....

.....

.....

.....[4]

Question 37

(c) The database design has three tables to store the qualifications and grades each student has attained. The following is a sample of the data from each table.

STUDENT

StudentID	FirstName	LastName	Tutor
001AT	Ahmad	Tan	11A
003JL	Jane	Li	11B
011HJ	Heather	Jones	10A

QUALIFICATION

QualCode	Level	Subject
CS1	IGCSE	Computer Science
MT9	IGCSE	Maths
SC12	IGCSE	Science

STUDENT-QUALIFICATION

QualCode	StudentID	Grade	DateOfAward
SC12	011HJ	A	31/8/2014
SC12	003JL	C	31/8/2014
CS1	003JL	B	31/8/2014

(i) Draw an Entity-Relationship (E-R) diagram to show the relationships between these three tables.

[2]

(ii) State the type of relationship that exists between STUDENT and STUDENT-QUALIFICATION.

.....[1]

(iii) Describe how the relationship between QUALIFICATION and STUDENT-QUALIFICATION is implemented.

.....

.....

.....

.....[2]

(d) (i) The database will store each student's date of birth.

Write an SQL script to add a date of birth attribute to the appropriate table.

.....

.....

.....

.....[2]

(ii) Write an SQL script to display the StudentID, Grade and DateOfAward for the QualCode value of SC12. .....

.....

.....

.....

.....

.....[3]

(iii) Write an SQL script to display the FirstName and LastName and QualCode for all STUDENT-QUALIFICATIONs for which the Grade value is A.

.....

.....

.....

.....

.....

.....

.....

.....

.....[4]

Question 38

(a) Five descriptions and seven relational database terms are shown below.

Draw a line to link each description to its correct database term.

Description	Database term
Any object, person or thing about which it is possible to store data	Secondary key
Dataset organised in rows and columns; the columns form the structure and the rows form the content	Candidate key
Any attribute or combination of attributes that can act as a unique key	Entity
Attribute(s) in a table that link to the primary key in another table to form a relationship	Foreign key
Attribute or combination of attributes that is used to uniquely identify a record	Primary key
	Table
	Tuple

[5]

(b) Explain what is meant by referential integrity.

.....

.....

.....

.....

.....

.....

.....[3]

Question 39

A health club offers classes to its members. A member needs to book into each class in advance.

(a) The health club employs a programmer to update the class booking system. The programmer has to decide how to store the records. The choice is between using a relational database or a file-based approach.

Give three reasons why the programmer should use a relational database.

1 .....

.....

.....

.....

2 .....

.....

.....

.....

3 .....

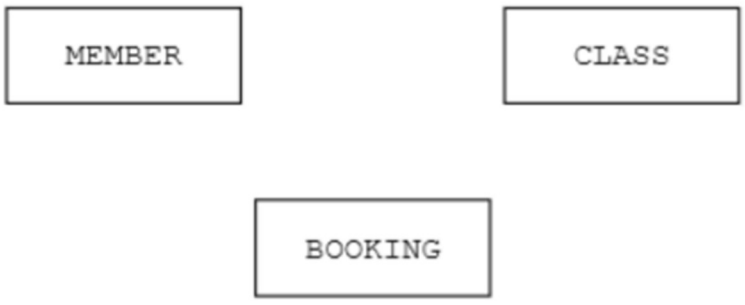
.....

.....

.....[6]

(b) The programmer decides to use three tables: `MEMBER`, `BOOKING` and `CLASS`.

Complete the Entity-Relationship (E-R) diagram to show the relationships between these tables.



[2]

(c) The `CLASS` table has primary key `ClassID` and stores the following data:

ClassID	Description	StartDate	ClassTime	NoOfSessions	AdultsOnly
DAY01	Yoga beginners	12/01/2016	11:00	5	TRUE
EVE02	Yoga beginners	12/01/2016	19:00	5	FALSE
DAY16	Circuits	30/06/2016	10:30	4	FALSE

Write an SQL script to create the CLASS table.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[6

Question 40

9 A database has been designed to store data about salespersons and the products they have sold.

The following facts help to define the structure of the database:

- each salesperson works in a particular shop
- each salesperson has a unique first name
- each shop has one or more salespersons
- each product which is sold is manufactured by one company only
- each salesperson can sell any of the products
- the number of products that each salesperson has sold is recorded

The table `ShopSales` was the first attempt at designing the database.

FirstName	Shop	ProductName	NoOfProducts	Manufacturer
Nick	TX	television set refrigerator digital camera	3 2 6	SKC WP HKC
Sean	BH	hair dryer electric shaver	1 8	WG BG
John	TX	television set mobile phone digital camera toaster	2 8 4 3	SKC ARC HKC GK

(a) State why the table is **not** in First Normal Form (1NF).

.....

.....[1]

(b) The database design is changed to:

`SalesPerson (FirstName, Shop)`

`SalesProducts (FirstName, ProductName, NoOfProducts, Manufacturer)`

Using the data given in the first attempt table (`ShopSales`), show how these data are now stored in the revised table designs.

Table: `SalesPerson`

FirstName	Shop

Table: SalesProducts

FirstName	ProductName	NoOfProducts	Manufacturer

[3]

(c) (i) A relationship between the two tables has been implemented.

Explain how this has been done.

.....

.....

.....

.....

.....[2]

(ii) Explain why the SalesProducts table is not in Third Normal Form (3NF).

.....

.....

.....

.....

.....[2]

(iii) Write the table definitions to give the database in 3NF.

.....

.....

.....

.....

.....[2]

Answer

Answer 1

7(a)	**1 mark** per bullet point to **max 3** • Flat-file has more data redundancy • … because the same data is stored many times // data is stored in different tables which are linked • There is program-data dependence with flat-files • ... because any changes to the structure of the data means the programs that access that data have to be re-written • Flat-file has more data inconsistency // worse data integrity • ... because duplicated data might be stored differently //…because when data is updated in one place, it is not updated everywhere • It is not easy to perform **complex** searches /queries • … because a new program has to be written each time • Flat files could have a lack of privacy • … as user views cannot easily be implemented	**3**
7(b)(i)	**1 mark** for each correct example one-to-one • e.g. customer to payment details // customer to login details one-to-many • e.g. customer to order many-to-many • e.g. order to product // customer to product	**3**

7(b)(ii)	**1 mark** **Relationship** / **Tick (✓)** one-to-one / one-to-many / many-to-many / ✓	**1**

7(b)(iii)	**1 mark** `CREATE DATABASE SHOPORDERS;`	**1**
7(c)	**1 mark** per item to **max 3** • table name • field name // attribute • data type • type of validation • Primary Key • Foreign Key • relationships	**3**

Answer 2

<table>
<tr><td>1(a)</td><td>1 mark for definition, 1 mark for appropriate example in each

<table>
<tr><th>Term</th><th>Definition and example</th></tr>
<tr><td>Field</td><td>A column/attribute in a table
e.g. <code>CustomerID</code> in the table <code>CUSTOMER</code></td></tr>
<tr><td>Entity</td><td>Anything that data can be stored about
e.g. A customer or a house</td></tr>
<tr><td>Foreign Key</td><td>A field in one table that is linked to a Primary Key in another table e.g. <code>CustomerID</code> / <code>HouseID</code> in table <code>RENTAL</code></td></tr>
</table>
</td><td>6</td></tr>
<tr><td>1(b)</td><td>1 mark per bullet point to max 2

• All fields in all tables are dependant fully on the PK and on no other fields
• for example all fields in <code>Customer</code> table are fully dependent on <code>CustomerID</code></td><td>2</td></tr>
<tr><td>1(c)(i)</td><td>1 mark for each correctly completed line

<pre>CREATE TABLE RENTAL(
    RentalID INTEGER NOT NULL,
    CustomerID INTEGER NOT NULL,
    HouseID VARCHAR (5) NOT NULL,
    MonthlyCost REAL/CURRENCY NOT NULL,
    DepositPaid BOOLEAN NOT NULL,
    PRIMARY KEY (RentalID)
);</pre></td><td>4</td></tr>
</table>

1(c)(ii)	**1 mark** per bullet point
	
	• Select `FirstName` and `LastName`
	• From both tables
	• Where `DepositPaid = No`
	• Joining tables (either `AND`, or `INNER JOIN`)
	
	Example script:
	`SELECT FirstName, LastName`
	`FROM CUSTOMER, RENTAL`
	`WHERE DepositPaid = No`
	`AND RENTAL.CustomerID = CUSTOMER.CustomerID;`
		4

Answer 3

5(a)	**1 mark** for 2 or 3 correct ticks, **2 marks** for 4 correct ticks **Table** / **Field name** / **Primary Key (PK)** / **Foreign Key (FK)** MANAGER / ManagerID / ✓ / SHOP / ManagerID / / ✓ CAR / RegistrationNumber / ✓ / CAR / ShopID / / ✓	**2**
5(b)	**1 mark** per bullet point • Access rights give managers / himself access to different elements • … by having different accounts / logins • … which have different access rights e.g. read only // no access / read / write • Specific views can be assigned to himself and to the managers • … e.g. managers can only see the data for their own shop(s)	**3**

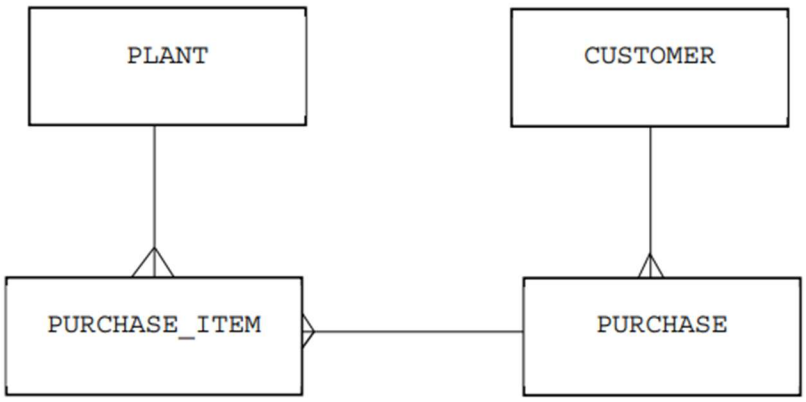
5(c)(i)	**1 mark** per correctly completed statement
	
	`SELECT COUNT(RegistrationNumber)`
	`FROM CAR`
	`GROUP BY ShopID`
		3

5(c)(ii)	**1 mark** for each correct statement
	
	`INSERT INTO CAR`
	`VALUES ("123AA","Tiger","Lioness",10500,"12BSTREET")`
		2

Answer 4

<table>
<tr>
<td>6(a)(i)</td>
<td>1 mark for 1 tick in the correct place
2 marks for all 3 ticks correct

<table>
<tr><th rowspan="2">Task</th><th colspan="3">Normalisation stage</th></tr>
<tr><th>0NF to 1NF</th><th>1NF to 2NF</th><th>2NF to 3NF</th></tr>
<tr><td>Remove any partial key dependencies</td><td></td><td>✓</td><td></td></tr>
<tr><td>Remove any repeating groups of attributes</td><td>✓</td><td></td><td></td></tr>
<tr><td>Remove any non-key dependencies</td><td></td><td></td><td>✓</td></tr>
</table>
</td>
<td>2</td>
</tr>
<tr>
<td>6(a)(ii)</td>
<td>1 mark for each correct relationship

 
</td>
<td>3</td>
</tr>
<tr>
<td>6(b)</td>
<td>1 mark for description of purpose
• Stores metadata about the database

1 mark for each example of contents to max 2
e.g.
• field / attribute names
• table name
• validation rules
• data types
• primary keys // foreign keys
• relationships</td>
<td>3</td>
</tr>
</table>

6(c)(i)	**1 mark** for each correctly completed space
	
	`SELECT SUM(Quantity)`
	`FROM PURCHASE_ITEM`
	`WHERE PurchaseID = "3011A";`
		4

6(c)(ii)	**1 mark** per bullet point
	
	• ALTER TABLE PURCHASE
	• ADD OrderDate
	• Suitable data type, e.g. DATE
	
	`ALTER TABLE PURCHASE`
	`ADD OrderDate DATE;`
		3

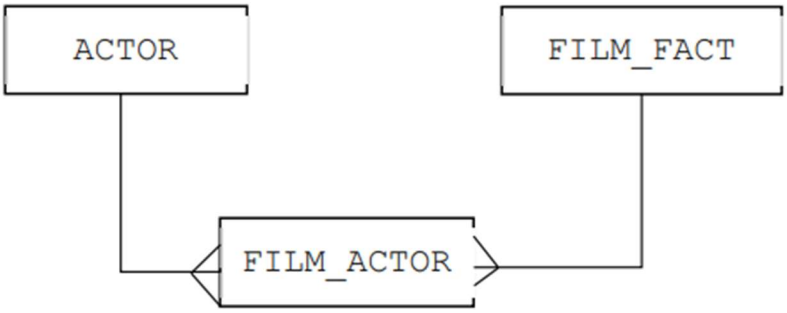
Answer 5

4(a)	**1 mark** per bullet point to **max 4** e.g. • Reduced data redundancy // less repeated data • …because each item of data is only stored once • Maintains data consistency // improves data integrity • …changes in one table will automatically update in another • ... linked data cannot be entered differently in two tables • Program-data independence • …changes to the data do not require programs to be re-written • Complex queries are easier to run • Can provide different views • ….so users can only see specific aspects of the database	**4**
4(b)	**1 mark** for 3NF **1 mark** per bullet for justification to **max 2** • There are no repeated attributes // it is already in 2NF • Each field is fully dependent on the corresponding primary key // no partial dependencies • No transitive dependencies	**3**

4(c)(i)	**1 mark** per bullet point
	
	• Create table, table name, opening and closing brackets
	• `StudentID` and `Mark` as integer
	• `TestID` as Varchar
	• Primary key correctly set up
	• Foreign keys correctly set up
	e.g.
	`CREATE TABLE STUDENT_TEST (`
	`StudentId INTEGER,`
	`TestID VARCHAR,`
	`Mark INTEGER,`
	`PRIMARY KEY(StudentID,TestID),`
	`FOREIGN KEY(TestID) REFERENCES TEST(TestID),`
	`FOREIGN KEY(StudentID) REFERENCES STUDENT(StudentID)`
	`);`
		5

4(c)(ii)	**1 mark** for each point
	
	• AVG(Mark)
	• SELECT and FROM STUDENT_TEST
	• WHERE clause
	
	e.g.
	`SELECT AVG(Mark)`
	`FROM STUDENT_TEST`
	`WHERE TestID = "A7";`
		3

Answer 6

5(a)	**1 mark** for each correct relationship 	**2**
5(b)	**1 mark** per point • Neither key uniquely identifies each tuple by itself • One actor cannot appear in the same film twice so together they are unique	**2**

5(c)	**1 mark** per correct entry
	`SELECT FILM_ACTOR.ActorID / ActorID`
	`FROM FILM_ACTOR`
	`INNER JOIN FILM_FACT`
	`ON FILM_FACT.FilmID = FILM_ACTOR.FilmID`
	`WHERE FILM_FACT.FilmTitle = "Cinderella" ;`
		4

5(d)	**1 mark** per point
	• COUNT and correct fieldname
	• SELECT and FROM statements, including the table name in FROM
	• WHERE statement
	e.g.
	`SELECT COUNT(FilmID)`
	`FROM FILM_FACT`
	`WHERE ReleaseDate >= #01/01/2022# AND ReleaseDate <= #31/01/2022#;`
	`// WHERE ReleaseDate BETWEEN #01/01/2022# AND #31/01/2022#;`
	`// WHERE ReleaseDate = "January 2022";`
		3

5(e)	**1 mark** for each correctly completed term • data dictionary • field names // primary keys • primary keys //field names • logical schema • query • interface A DBMS provides data management. This includes the development of a **data dictionary** that stores information about the data stored, such as **field names** and **primary keys**. The **logical schema** uses methods such as an E-R diagram to show the structure of the database and its relationships. The **query** processor allows a user to perform searches to find specific data. The DBMS also provides a developer **interface** that allows the user to create tables, forms and reports.	**6**

Answer 7

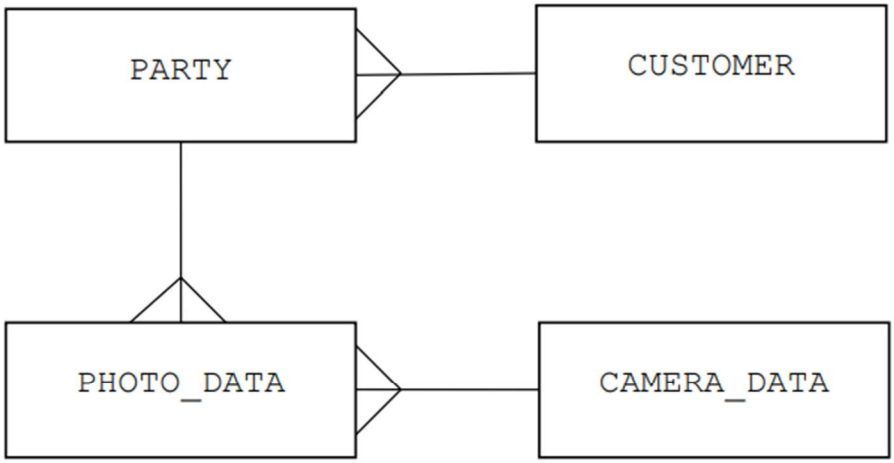
6(a)	**1 mark** per point • Primary key `StaffID` in STAFF… • …**links** to foreign key **`StaffID`** in DEVICE • One staff member can have many devices • Each device can only be with one member of staff	**4**

6(b)(i)	**1 mark** for each correctly completed statement
	
	`SELECT COUNT(STAFF.StaffID)`
	`FROM STAFF`
	`INNER JOIN DEVICE`
	`ON STAFF.StaffId = DEVICE.StaffID`
	`WHERE STAFF.FirstName = "Ali"`
	`AND STAFF.LastName = "Khan";`
		4

6(b)(ii)	**1 mark** per bullet point
	
	• `ALTER TABLE DEVICE`
	• `ADD` *appropriate field name*, *appropriate data type*
	
	e.g.
	`ALTER TABLE DEVICE`
	`ADD Returned Boolean;`
		2

6(c)	<table><tr><th>Normal Form</th><th>Description</th></tr><tr><td>First Normal Form (1NF)</td><td>No repeating groups or repeating attributes</td></tr><tr><td>Second Normal Form (2NF)</td><td>All attributes must be **fully** dependant on the (composite) primary key // No partial dependencies</td></tr><tr><td>Third Normal Form (3NF)</td><td>All attributes must be fully dependent on the primary key and no other attributes // no non-key dependencies // no transitive dependencies</td></tr></table> **1 mark** for each correct description	**3**

Answer 8

4(a)	**1 mark** for each correct relationship: • 1:M between CUSTOMER and PARTY • 1:M between PARTY and PHOTO_DATA • 1:M between CAMERA_DATA and PHOTO_DATA 	**3**
4(b)	**1 mark** for each bullet point (**max 3**): • no repeating groups of attributes // data is atomic • no partial key dependencies • no non-key dependencies // no transitive dependencies	**3**

4(c)(i)	**1 mark** for the definition, **1 mark** for the example: • definition: a single row in a table • example: from the `PHOTO_DATA` table	**2**

4(c)(ii)	**1 mark** for each correctly completed empty space:
	
	• COUNT
	• PhotoID
	• PHOTO_DATA
	• 'CAN*' // 'CAN%'
	
	`SELECT COUNT(PhotoID)`
	`FROM PHOTO_DATA`
	`WHERE CameraID LIKE 'CAN*'; // WHERE CameraID LIKE 'CAN%';`
		4

4(d)	**1 mark** for each bullet point:
	
	• ALTER TABLE CAMERA_DATA
	• ADD NumberStored INTEGER
	• , LastUsed DATE;
	
	`ALTER TABLE CAMERA_DATA`
	`ADD NumberStored INTEGER, LastUsed DATE;`
		3

Answer 9

5(a)	**1 mark** for each bullet point **(max 3)**: Solution 1: • removing the many-to-many relationship between Owner and Tree • ... by removing TreeID and TreePosition from the Owner table • ... and creating a linking table between Owner and Tree • ... that contains OwnerID, TreeID and TreePosition • ... (composite) primary key of the linking table should be OwnerID and TreeID // insert a named new primary key in the linking table Solution 2: • removing the many-to-many relationship between Owner and Tree • move TreePosition into TREE table • ... put OwnerID into TREE table • create a new table with suitable name (for the species of tree) • ... containing ScientificName, MaxHeight and FastGrowing • ... with ScientificName as primary key // or another suitable primary key	**3**

5(b)	**1 mark** for each bullet point:
	
	• INSERT INTO TREE
	• VALUES ( ) and correct values
	• Values in correct order
	
	Option 1:
	`INSERT INTO`
	`TREE(TreeID, ScientificName, MaxHeight, FastGrowing)`
	`VALUES('LOW_1276', 'Salix_Alba', 30.00, TRUE);`
	
	Option 2:
	`INSERT INTO`
	`TREE`
	`VALUES('LOW_1276', 'Salix_Alba', 30.00, TRUE);`
		3

5(c)	**1 mark** for: An attribute / field (or set of attributes / fields) that **could** be a primary key	**1**
5(d)(i)	**1 mark** for description • stores metadata about the database **1 mark** for a correct example For example: • field / attribute names • table name • validation rules • data types • primary keys // foreign keys • relationships	**2**
5(d)(ii)	**1 mark** for each bullet point **(max 2)**: • the overview of a database structure • models the problem / situation • ... by using methods such as an ER diagram • independent of any particular DBMS	**2**

Answer 10

<table>
<tr><td>2(a)</td><td>1 mark for each correct answer.

<table><tr><th>Item</th><th>Answer</th></tr><tr><td>a suitable field for the primary key in <code>COMPANY</code></td><td><code>CompanyID</code></td></tr><tr><td>a candidate key in <code>TELESCOPE</code></td><td><code>SerialNumber</code> // <code>TelescopeID</code></td></tr><tr><td>the degree of relationship between <code>TELESCOPE</code> and <code>PHOTOGRAPH</code></td><td>1:M / 1 to many</td></tr></table></td><td>3</td></tr>
<tr><td>2(b)</td><td>Logical schema</td><td>1</td></tr>
<tr><td>2(c)</td><td>1 mark for each correctly completed missing part:

<code>SELECT COUNT (TelescopeID)
FROM TELESCOPE
WHERE CompanyID LIKE 'HW%';</code></td><td>4</td></tr>
<tr><td>2(d)</td><td>1 mark for each bullet point:

• ALTER TABLE PHOTOGRAPH
• ADD Resolution TEXT;

<code>ALTER TABLE PHOTOGRAPH
ADD Resolution TEXT / VARCHAR(11);</code></td><td>2</td></tr>
<tr><td>2(f)</td><td>1 mark for each bullet point (max 2):

• allows the user to enter criteria
• searches for the data that meets the entered criteria
• organises the results to be displayed to the user</td><td>2</td></tr>
</table>

Answer 11

<table>
<tr><td>2(a)(i)</td><td>1 mark for definition:

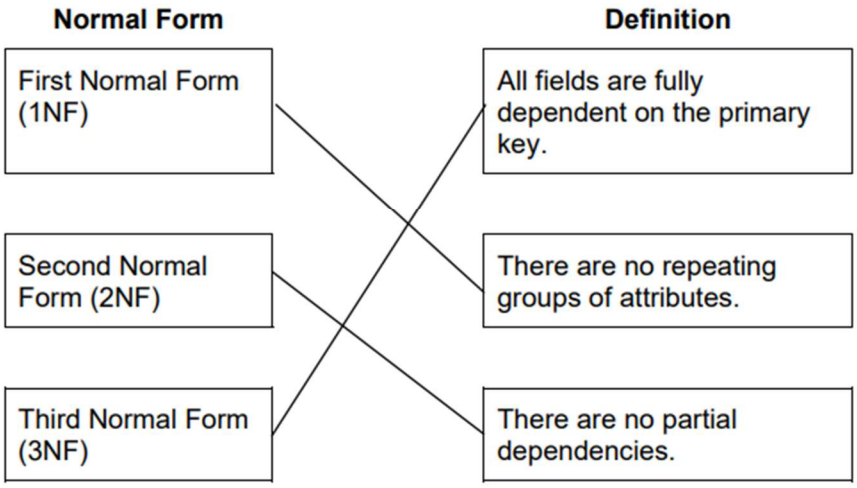
• Data about the data in the database // data about the structure of the database // metadata for a database

1 mark for a suitable example
Examples:
• table names
• data types
• field names</td><td>2</td></tr>
<tr><td>2(a)(ii)</td><td>1 mark for definition

• Methods of making sure the data is consistent

1 mark for example
Examples:
• Enforcing referential integrity
• If data in one table is deleted/edited all tables are updated // cascading update/delete
• Validation/verification rules</td><td>2</td></tr>
<tr><td>2(b)(i)</td><td>1 mark for each field name and table

<table><tr><th>Foreign key</th><th>Database table</th></tr><tr><td>BirdID</td><td>BIRD_SEEN</td></tr><tr><td>PersonID</td><td>BIRD_SEEN</td></tr></table></td><td>2</td></tr>
<tr><td>2(b)(ii)</td><td>1 mark for all 3 correct lines

 
</td><td>1</td></tr>
</table>

2(b)(iii)	**1 mark** each
	
	• CREATE TABLE start and end bracket
	• Bird ID as CHAR/VARCHAR
	• Name **and** size as VARCHAR/CHAR
	• Bird ID as primary key
	
	Example answer:
	`CREATE TABLE BIRD_TYPE(`
	`  BirdID CHAR(4) NOT NULL,`
	`  Name VARCHAR(9),`
	`  Size VARCHAR(6),`
	`  PRIMARY KEY (BirdID)`
	`);`
		4

2(b)(iv)	**1 mark** for each correctly completed space
	
	`SELECT BIRD_TYPE.Size, COUNT(BIRD_TYPE.BirdID) AS NumberOfBirds`
	
	`FROM BIRD_TYPE, BIRD_SEEN`
	
	`WHERE BIRD_SEEN.PersonID = "J_123"`
	
	`AND BIRD_TYPE.BirdID = BIRD_SEEN.BirdID`
	
	`GROUP BY BIRD_TYPE.Size;`
		5

Answer 12

2(a)	**1 mark** for each correct feature or description **Feature** / **Description** Data dictionary — Data about the data in the database // data about the structure of the database // metadata for a database Query processor — Software that allows the user to enter **criteria,** then finds and returns the appropriate result // software that processes and executes queries **written in SQL** **Logical schema** — A model of a database that is not specific to one DBMS **Developer interface** — A software tool that allows the user to create items such as tables, forms and reports	**4**

2(b)	**1 mark** each to **max 3** • Referential Integrity makes sure data is consistent • Referential Integrity makes sure all data is up-to-date • Referential integrity ensures that every foreign key has a **corresponding** primary key • Referential Integrity prevents records from being added / deleted / modified incorrectly • Referential Integrity makes sure that if data is changed in one place the change is reflected in all related records • Referential Integrity makes sure any queries return accurate and complete results	**3**
2(c)(i)	**1 mark** each to **max 2** • Presence check to make sure that the (rider level) is entered • Look-up / Existence check to make sure the rider level is only Beginner, Intermediate or Advanced • Length check to make sure the rider level entered is either 8 or 12 characters • Type check to make sure the rider level is alphanumeric	**2**

2(c)(ii)	**1 mark** each
	
	• SELECT field Name
	• FROM table HORSE
	• WHERE with Intermediate / Beginner
	• OR with Beginner / Intermediate
	
	Example answer:
	`SELECT Name`
	`FROM HORSE`
	`WHERE HorseLevel = "Intermediate"`
	`OR HorseLevel = "Beginner";`
		4

2(c)(iii)	**1 mark** each • SUM should be COUNT // SELECT COUNT(STUDENT.RiderLevel) • The WHERE statement needs the table names before each field name // WHERE STUDENT.StudentID = LESSON.StudentID • The OR should be AND // AND Date = #09/09/2023# • Beginner is missing the speech marks // STUDENT.RiderLevel = "Beginner";	**4**

Answer 13

4(a) — **4**

1 mark for each bullet point. Mark in pairs. **Max 2** for each description.

- Reduces data redundancy
- ... because linked tables mean that each data item is stored only once
- Reduces program-data dependency
- ... because the data is separate from the software so changes to the data do not require programs to be re-written
- Reduces data inconsistency // improves data integrity
- ... because by only storing data once it only needs to be updated once // changes in one table will automatically update in another // linked data cannot be entered differently in two tables
- Complex queries are easier to run
- Can provide different views
- ….so users can only see specific aspects of the database

4(b) — **4**

1 mark each

Term	Description
Entity	An object that data is stored about.
Tuple	**A row of data in a table about one instance of an object.**
Secondary key	**An additional/alternative key used as well as the primary key to locate specific data // a candidate key that has not been chosen as a primary key.**
Foreign key	A field in one table that is linked to a primary key in another table.

4(c) — **4**

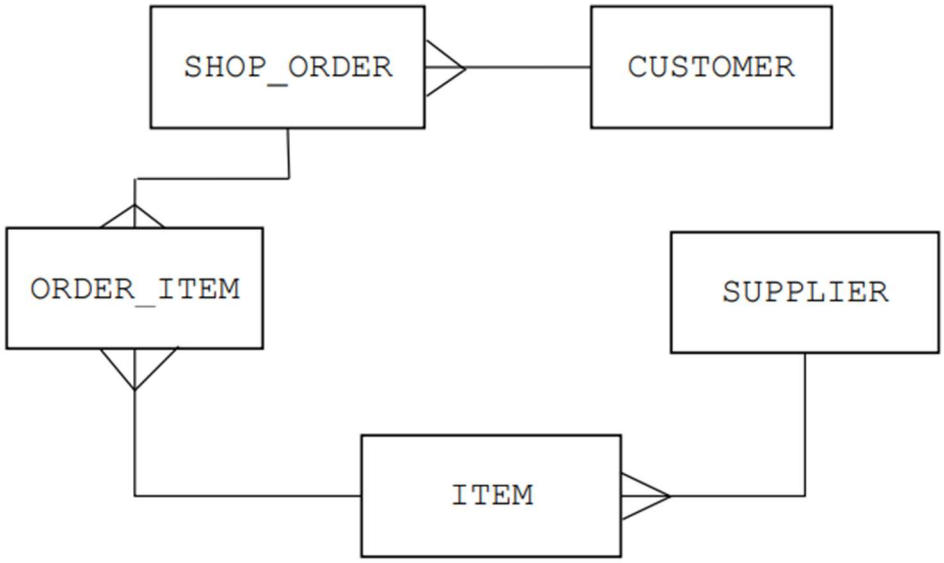
1 mark each

- Only 3 tables with appropriate identifiers (i.e. one table for customer, one for booking and one for car)
- Appropriate Primary key in each table underlined
- Booking table includes Primary key from car and Primary key from customer as Foreign keys
- All original fields are in correct tables

Example answer:

```
BOOKING(BookingID, CarRegistration, CustomerID,
StartDate, EndDate)
CAR(CarRegistration, CarModel, CarColour)
CUSTOMER(CustomerID, CustomerFirstName,
CustomerLastName, EmailAddress, TelephoneNumber)
```

Answer 14

3(a)	**1 mark** for each correct relationship or relationships • 1:M between CUSTOMER and SHOP_ORDER • 1:M between SUPPLIER and ITEM • 1:M between SHOP_ORDER and ORDER_ITEM **and** M:1 between ORDER_ITEM and ITEM  	**3**
3(b)	**1 mark** for each bullet point (**max 3**) • Reduced data **redundancy** • Improved data integrity / consistency / referential integrity • Allows for views / improved privacy • Allows for program-data independence • Complex queries can be executed	**3**
3(c)(i)	**1 mark** for `CREATE DATABASE SHOP;`	**1**

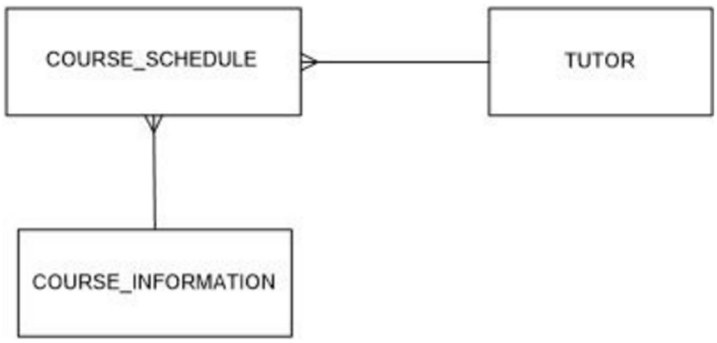
3(c)(ii)	**1 mark** for each line (**max 4**)
	
	Either:
	`SELECT SUM(Quantity)`
	`FROM ORDER_ITEM, SHOP_ORDER`
	`WHERE ORDER_ITEM.OrderNo = SHOP_ORDER.OrderNo`
	`AND SHOP_ORDER.CustomerID = 'HJ231';`
	OR
	`SELECT SUM(Quantity)`
	`FROM ORDER_ITEM (INNER) JOIN SHOP_ORDER`
	`ON ORDER_ITEM.OrderNo = SHOP_ORDER.OrderNo`
	`WHERE SHOP_ORDER.CustomerID = 'HJ231';`
		4

Answer 15

2(a)	**1 mark** for each term (**max 3**) Entity: • An object about which data can be stored Primary key: • The **unique** attribute / combination of attributes used to identify the **record / tuple** Referential integrity: • Makes sure that if data is changed in one place the change is reflected in all related records - cascading update/delete • Makes sure that data that does not exist cannot be referenced • Ensures that every foreign key has a **corresponding** primary key // A logical dependency of a foreign key on a primary key • Ensures that the data in the database is consistent / up to date • Prevents records from being added/deleted/modified incorrectly • Makes sure any queries return accurate and complete results	**3**

2(b)	**1 mark** for each bullet point (**max 4**) **Max 2** if no descriptions • Backup / recovery procedures • … automatically takes copies of the database and store off site on a regular basis / weekly, etc. • ... so that the data can be recovered if lost • Use of access rights • … some users are given different access permissions to different tables • ... read/write, read only, full access, etc. • Views • … different users are able to see different parts of the database • ... only see what users need to see // by example • Record and table locking • … prevents simultaneous access to data • ... so updates are not lost // data is not overwritten • Encryption • ... the data is turned into ciphertext • ... so it cannot be understood **without a decryption key**	**4**
2(c)	**1 mark** for each bullet point (**max 4**) • Identify **repeating** groups of attributes ... • ... Subject **and** SubjectCode • Ensure each field is atomic • ... StudentName should be split into e.g. FirstName and LastName • Identify the primary key for the table	**4**

Answer 16

3(b)(i)	**1 mark** for both 1:M relationships as follows:  	**1**

3(b)(ii)	**1 mark** for each bullet point. • SELECT Count(CourseID) • AS NumOfCourses • FROM COURSE_SCHEDULE • WHERE DateStarted > "09/09/23"; SELECT Count(CourseID) AS NumOfCourses FROM COURSE_SCHEDULE WHERE DateStarted > "09/09/23";	**4**
3(c)	**1 mark** for each bullet point. • The administrator completes a visual check / checks by eye • …that the tutor identifier input matches the tutor identifier on the **original** document • Double entry check // the administrator (or a second person) enters the number a second time • …and the **system** compares it with the first entry	**4**

Answer 17

6(a)	**1 mark** each: • User table with the username as the Primary Key • … containing at least email address, date of birth / age and rating • Quiz table with Quiz ID or date or file name as the Primary Key. • … containing at least the other field(s) not used as the PK • A joining table with an appropriate name including at least fields for user identification, quiz identification and score • … with an appropriate Primary Key • ... and Foreign Keys matching the Primary Keys of the other two tables USER(Username, Email, DateOfBirth, Rating) QUIZ(QuizID, Date, Filename) USER_QUIZ(Username, QuizID, Score)	**6**

6(b)	**1 mark** each to **max 2** for data dictionary and **max 2** for logical schema: Data dictionary: • Data about the data in the database // metadata • Identifies the characteristics of the data that will be stored • Appropriate example e.g. field names, table name, validation rules, data types, primary / foreign keys, relationships etc. Logical schema: • Conceptual design • Platform/database independent overview of the database • Is used to design the physical structure • Appropriate example e.g. Design of entities / E-R diagram / views	**4**

6(c)(i)	**1 mark** for each correct clause:
	
	• Alter table EVENT
	• Adding foreign key as PlayerID referencing correct table
	
	`ALTER TABLE EVENT`
	`ADD FOREIGN KEY(PlayerID)REFERENCES PLAYER(PlayerID);`
		2

6(c)(ii)	**1 mark** each:
	
	• Selecting PlayerID from EVENT
	• Counting EventID
	• Grouping by the PlayerID
	
	Example:
	`SELECT PlayerID, COUNT(EventID)`
	`FROM EVENT`
	`GROUP BY PlayerID;`
		3

Answer 18

4(a)	**1 mark** for: 1-to-many	**1**

4(b)	**1 mark** each:
	
	• Creating table EXAM with opening and closing brackets
	• All fields with appropriate data types and commas at end of lines
	• ExamID as primary key
	
	Example:
	`CREATE TABLE EXAM(`
	`ExamID varchar NOT NULL,`
	`Subject varchar,`
	`Level int,`
	`TotalMarks int,`
	`PRIMARY KEY(ExamID)`
	`);`
		3

4(c)	**1 mark** each:
	
	• Altering table EXAM_QUESTION
	• Linking ExamID to ExamID in EXAM
	
	Example.
	`ALTER TABLE EXAM_QUESTION`
	`ADD FOREIGN KEY (ExamID) REFERENCES EXAM(ExamID);`
		2

4(d)	**1 mark** each to **max 5**: • STUDENT table identified with suitable Primary Key • A linking table between STUDENT and EXAM with suitable Primary Key **and** appropriate name • … that includes the Primary Key of the STUDENT table as a Foreign Key to join with STUDENT • ... and includes the Primary Key of the EXAM table as a Foreign Key to join with EXAM • A linking table between STUDENT and EXAM_QUESTION with suitable Primary Key **and** appropriate name • … that includes the Primary Key of Table 2 as a Foreign Key to join with Table 2 • … that stores the ExamQuestionID and the mark for that question	**5**

Answer 19

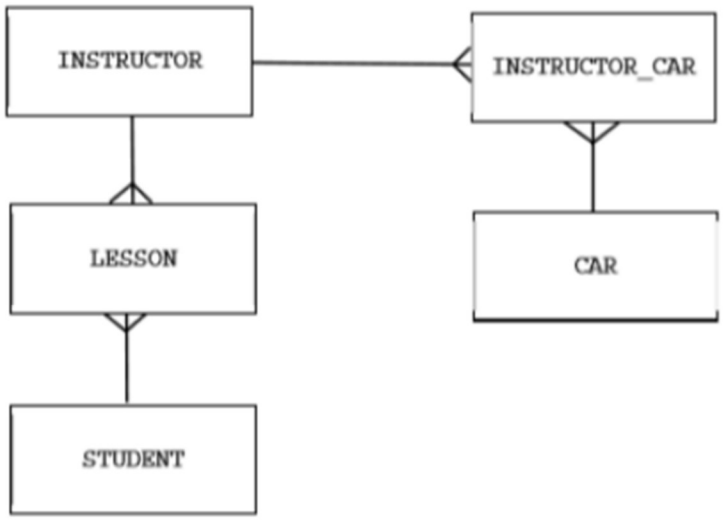
4(a)	**1 mark** for: many to 1 // there are many performances of each show	**1**

4(b)	**1 mark** each for:
	
	• Creating table `PERFORMANCE` with opening and closing brackets
	• Setting all four attributes with appropriate data types
	• Setting `PerformanceID` as primary key
	• Setting `ShowID` as foreign key referencing `SHOW` table
	
	Example:
	`CREATE TABLE PERFORMANCE(`
	`PerformanceID varchar NOT NULL,`
	`ShowID varchar,`
	`ShowDate Date,`
	`StartTime Time,`
	`PRIMARY KEY(PerformanceID),`
	`FOREIGN KEY(ShowID) REFERENCES SHOW(ShowID)`
	`);`
		4

4(c)	**1 mark** each for:
	
	• Selecting `COUNT` of an attribute in `PERFORMANCE` table with suitable name
	• `FROM` clause
	• Joining tables
	• Grouping by the title and selecting the title
	
	Example 1:
	`SELECT SHOW.Title, Count(PERFORMANCE.PerformanceID)`
	`AS NumberOfShowings`
	`FROM PERFORMANCE, SHOW`
	`WHERE PERFORMANCE.ShowID = SHOW.ShowID`
	`GROUP BY SHOW.Title;`
	
	Example 2:
	`SELECT SHOW.Title, Count(PERFORMANCE.PerformanceID)`
	`AS NumberOfShowings`
	`FROM PERFORMANCE INNER JOIN SHOW`
	`ON PERFORMANCE.ShowID = SHOW.ShowID`
	`GROUP BY SHOW.Title;`
		4

4(d)	**1 mark** each to **max 5**: e.g. • CUSTOMER table identified with suitable Primary Key and appropriate name • … and other suitable fields including name and email • BOOKING TABLE identified with suitable Primary Key and appropriate name • … that stores the Primary Key of the CUSTOMER table as a Foreign Key to join with CUSTOMER table • … and stores the Primary Key of the PERFORMANCE table as a Foreign Key to join with PERFORMANCE table • A linking table between Table 2 and SEAT with suitable Primary Key and appropriate name • … that includes the Primary Key of Table 2 as a Foreign Key to join with Table 2 • … that stores the SeatID.	**5**

Answer 20

7(a)	**1 mark** per bullet point to **max 2** • Reduced data redundancy • Reduced data dependency • Improved data integrity • Improved data privacy • Program-data independence • Ability to create ad hoc queries	2
7(b)	**1 mark** for each correct link 	4

7(c)	**1 mark** for each correctly completed statement
	
	• CREATE (line 1)
	• INTEGER (line 6)
	• PRIMARY KEY (line 7)
	
	`CREATE TABLE INSTRUCTOR(`
	`  InstructorID VARCHAR(5),`
	`  FirstName VARCHAR(15),`
	`  LastName VARCHAR(15),`
	`  DateOfBirth DATE,`
	`  Level INTEGER,`
	`  PRIMARY KEY (InstructorID)`
	`);`
		3

7(d)	**1 mark** per bullet point
	
	• Alter table student
	• Add an appropriate identifier with suitable data type
	
	`ALTER TABLE STUDENT`
	`ADD TelNum VARCHAR;`
		2

7(e)	**1 mark** per bullet point
	
	• Select lesson date and lesson time
	• From table LESSON
	• Where InstructorID = "Ins01"
	• And lesson date is greater than today's date
	
	`SELECT LessonDate, LessonTime`
	`FROM LESSON`
	`WHERE InstructorID = "Ins01"`
	`AND LessonDate > ########;`
		4

Answer 21

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>6(a)</td><td>1 mark per bullet point to max 2

• There are partial dependencies in the SOFTWARE_PURCHASED table // SoftwareDescription is dependent only on SoftwareName and not both SoftwareName and CustomerID

• There is a non-key dependency in the SOFTWARE_PURCHASED table // LicenceCost is dependent on LicenceType</td><td>2</td></tr>
<tr><td>6(b)</td><td>1 mark for a suitable example for each

<table>
<tr><th>Term</th><th>Example</th></tr>
<tr><td>Entity</td><td>SOFTWARE_PURCHASED //CUSTOMER_DETAILS</td></tr>
<tr><td>Foreign Key</td><td>CustomerID
(in SOFTWARE_PURCHASED table)</td></tr>
<tr><td>Attribute</td><td>Any valid example of an attribute from the tables</td></tr>
</table>
</td><td>3</td></tr>
<tr><td>6(c)(i)</td><td>1 mark for each correct entry (in bold)

<pre>
CREATE TABLE GAME_DEVELOPMENT (
  GameName VarChar,
  Genre VarChar,
  TeamNumber Integer,
  DevelopmentStage VarChar,
  ManagerID VarChar,
  PRIMARY KEY (GameName)
);
</pre>
</td><td>5</td></tr>
<tr><td>6(c)(ii)</td><td>1 mark for each correct entry (shown in brackets)

<pre>
SELECT (1) GameName, Genre, TeamNumber
FROM GAME_DEVELOPMENT, PRODUCT_MANAGER
WHERE PRODUCT_MANAGER.FirstName = "James"
AND PRODUCT_MANAGER.SecondName = "Fitz"
AND PRODUCT_MANAGER.ManagerID (1)
    = GAME_DEVELOPMENT.ManagerID (1) ;
</pre>
</td><td>3</td></tr>
</table>

Answer 22

Question	Answer	Marks
6(a)	**1 mark** for each correct line **Database Term** / **Description** Primary key → A unique identifier for each tuple Attribute → A data item, represented as a field within a table Foreign key → A field in one table that links to a primary key in another table Entity → The concept or object in the system that we want to model and store information about A collection of records and tables The type of data that is being stored	4
6(b)	**1 mark** per task to **max 3** • Create a table • Set up relationships between tables • Create / design a form • Create / design a report • Create / design a query (NOT run a query)	3

6(c)(i)	**1 mark** for each completed line
	
	`CREATE TABLE ROOM(`
	`  RoomNumber Integer,`
	`  RoomType Varchar,`
	`  PRIMARY KEY (RoomNumber)`
	`);`
		3

6(c)(ii)	**1 mark** for each completed line
	
	`INSERT INTO ROOM`
	`VALUES (5,"Double");`
		2

Question	Answer	Marks

6(c)(iii)	**1 mark** per bullet point
	
	• Alter table booking
	• Add number of nights with appropriate field name and data type
	
	`ALTER TABLE BOOKING`
	`ADD NumberNights Integer;`
		2

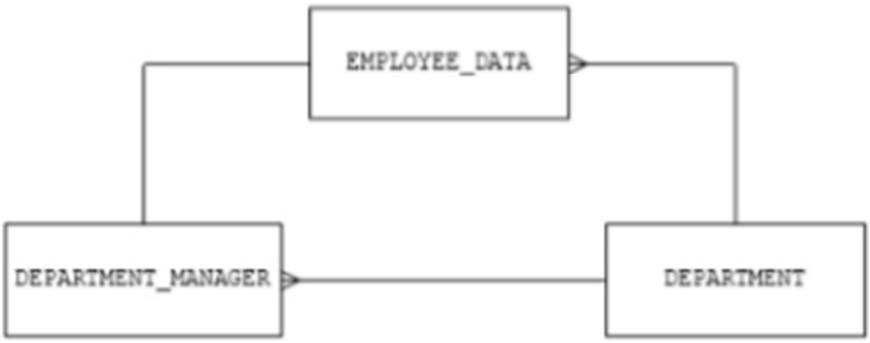
Answer 23

2(b)(i)	**1 mark** per bullet point to **max 3** ▯ Data redundancy // data is repeated in more than one file ▯ Data dependency // changes to data means changes to programs accessing that data ▯ Lack of data integrity // entries that should be the same can be different in different places ▯ Lack of data privacy // all users have access to all data if a single flat file	**3**
2(b)(ii)	**1 mark** for each correct name, **1 mark** for each matching description, max **2** marks per level ▯ External ▯ The individual's view(s) of the database ▯ Conceptual ▯ Describes the data as seen by the applications making use of the DBMS ▯ Describes the 'views' which users of the database might have ▯ Physical / Internal ▯ Describes how the data will be stored on the physical media ▯ Logical ▯ Describes how the relationships will be implemented in the logical structure of the database	**4**
2(c)(i)	1-to-many // 1 customer to/has many licences	**1**
2(c)(ii)	**1 mark** per bullet point ▯ `CustomerID` is the Primary key in `CUSTOMER` table ▯ Links to `CustomerID` as a Foreign key in `LICENCE` table	**2**

2(c)(iii)	**1 mark** per bullet point
	
	▯ Select with correct 5 fields
	▯ From `LICENCE`
	▯ Where `ExpiryDate <= '31/12/2019'` (any appropriate date type)
	▯ Group by `CustomerID`
	▯ Order by `Cost` (with or without ASC, but not DESC)
	
	`SELECT CustomerID, SoftwareID, LicenceType,`
	`  Cost, ExpiryDate`
	`FROM LICENCE`
	`WHERE ExpiryDate <= '31/12/2019'`
	`GROUP BY CustomerID ORDER BY Cost;`
		5

Answer 24

3(a)(i)	**1 mark** per bullet point • Stores all the information about the database // data about the data // metadata about the data • For example, fields, data types, validation, keys	**2**
3(a)(ii)	**1 mark** per bullet point to **max 2** • Allows the user to enter criteria • Searches for data which meets the entered criteria • Organises the results to be displayed to the user	**2**
3(b)	**1 mark** per bullet point to **max 2** • Primary key uniquely identifies each tuple // Each tuple in the table is unique • Primary key can be used as a foreign key in another table • ..... to form a link/relationship between the tables By example: • Identification of a primary key in a table • Describing <u>that</u> primary key in another table as a foreign key	**2**

3(c)	**1 mark** for each correct link 	**3**
3(d)	**1 mark** per bullet point to **max 3** • There are no repeating groups (1NF) • There are no partial dependencies (2NF) • There are no non-key dependencies // There are no transitive dependencies (3NF)	**3**
3(e)(i)	**1 mark** for correct answer `CREATE DATABASE EMPLOYEES;`	**1**

3(e)(ii)	**1 mark** per bullet
	
	- Create table `EMPLOYEE_DATA` with open and close brackets
	- `EmployeeID` as `VarChar` restricted to max 7, `Gender` as a `VarChar` restricted to max 6, `DepartmentNumber` as a `VarChar` restricted to max 2
	- `FirstName` and `LastName` as `VarChar` (any max lengths must be reasonable)
	- Date of birth as `Date`
	- Declaring `EmployeeID` as PK
	
	`CREATE TABLE EMPLOYEE_DATA (`
	`EmployeeID VarChar(7),`
	`FirstName VarChar,`
	`LastName VarChar,`
	`DateOfBirth Date,`
	`Gender VarChar(6),`
	`DepartmentNumber VarChar(2),`
	`PRIMARY KEY (EmployeeID)`
	`);`
		5

Question	Answer	Marks

3(e)(iii)	**1 mark** per bullet
	
	- Select FirstName and LastName only
	- From both tables
	- Where DepartmentName = "Finance"
	- AND Gender = "Female"
	- Joining tables (either AND, or inner join)
	
	`SELECT FirstName, LastName`
	`FROM EMPLOYEE_DATA, DEPARTMENT`
	`WHERE DepartmentName = "Finance"`
	`AND Gender = "Female"`
	`AND DEPARTMENT.DepartmentNumber =`
	`EMPLOYEE_DATA.DepartmentNumber;`
		5

Answer 25

5(a)(i)	**1 mark** for correct answer Repeated / duplicated data	**1**
5(a)(ii)	**1 mark** per bullet point • Because each record/piece of data is stored once and is referenced by a (primary) key • Because data is stored in individual tables • ...and the tables are linked by relationships • By the proper use of Primary and Foreign keys • By enforcing referential integrity • By going through the normalisation process	**3**
5(b)(i)	**1 mark** per bullet point • Security ensures that data is safe from unauthorised access // safe from loss • Integrity ensures that data is accurate / consistent / up to date	**2**

5(b)(ii)	**1 mark** for naming, **1 mark** for description For example: • Access rights // User accounts • Restrict actions (e.g. read / read-write) of specific users // unauthorised users cannot access the database • Views • Restrict which parts of the database specific users can see • Password // Biometrics // PIN code • Prevents unauthorised access • Automatic Backup • Create regular copies of data in case of loss • Encryption • Data is incomprehensible to unauthorised users	**4**
5(b)(iii)	**1 mark** per bullet • Query Processor • Developer Interface	**2**

Answer 26

<table>
<tr><td>4(d)(i)</td><td>1 mark per table

▯ Table CUSTOMER with fields FirstName, LastName, DateOfBirth, CustomerID
▯ Table ACCOUNT_TYPE with fields AccountID, Name, Bonus
▯ Table CUSTOMER_ACCOUNT with fields ID, CustomerID, AccountID, Amount

CUSTOMER (CustomerID, FirstName, LastName, DateOfBirth)

ACCOUNT_TYPE (AccountID, Name, Bonus)

CUSTOMER_ACCOUNT (ID, CustomerID, AccountID, Amount)</td><td>3</td></tr>
<tr><td>4(d)(ii)</td><td>1 mark for 1 or 2 correct Primary Keys, 2 marks for 3 correct Primary Keys

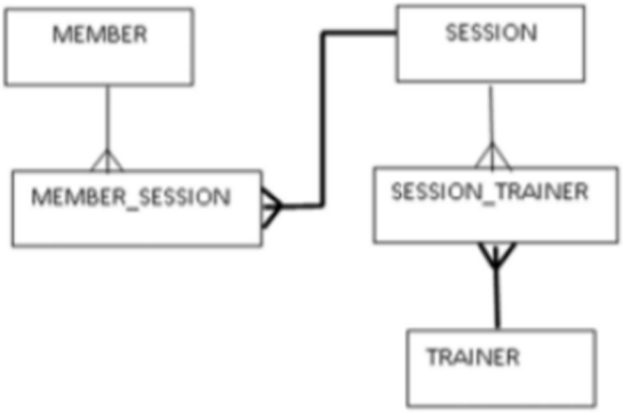
CUSTOMER: CustomerID
ACCOUNT_TYPE: AccountID
CUSTOMER_ACCOUNT: ID</td><td>2</td></tr>
<tr><td>4(d)(iii)</td><td>1 mark for both table name and Foreign Key

Table: CUSTOMER_ACCOUNT
Foreign Key: CustomerID / AccountID</td><td>1</td></tr>
<tr><td>4(d)(iv)</td><td>1 mark for each correct term

<table><tr><th>Definition</th><th>Term</th></tr><tr><td>All the data about one entity</td><td>Table / Relation</td></tr><tr><td>The data in one row of a table</td><td>Tuple / Record</td></tr><tr><td>A column or field in a table</td><td>Attribute</td></tr></table></td><td>3</td></tr>
</table>

Answer 27

4(a)(i)	**1 mark** per bullet point, **max 3 marks** from any group to **max 4** Multiple tables are linked together ... which eliminates / reduces data redundancy / duplication ... and increases data integrity / consistency ... which reduces compatibility issues ... so data need only be updated once ... and associated data will be automatically updated // referential integrity can be enforced ... which eliminates unproductive maintenance // which makes it easier to maintain the data Program-data independence means that ... the structure of data can change and does not affect program ... the structure of programs can change and does not affect data ... the data can be accessed by any appropriate program Allows concurrent access to data ... by the use of record locking ... by restricting over-writing changes Complex queries can be more easily written ... to search / find **specific** data // specific example related to the sports club Different users can be given different access rights ... which improves security Different users can be given different views of the data ... so they do not see confidential information ... and data privacy is maintained ... accept a valid example related to the sports club	**4**
4(a)(ii)	**1 mark** for each word in the correct position For a database to be in First Normal Form (1NF) there must be no **repeating** groups of attributes. For a database to be in Second Normal Form (2NF), it must be in 1NF, and contain no **partial** key dependencies. For a database to be in Third Normal Form (3NF), it must be in 2NF, and all attributes must be fully dependent on the **primary key**.	**4**

4(b)(i)	**1 mark** for SESSION to MEMBER_SESSION (one-to-many) **1 mark** for TRAINER to SESSION_TRAINER (one-to-many) 	**2**
4(b)(ii)	`CREATE DATABASE SPORTS_CLUB;`	**1**

4(b)(iii)	**1 mark** per bullet point
	▯ `CREATE TABLE SESSION` and `( );`
	▯ `SessionID` as `char(4) / varchar / varchar(4) / text` and `Description` as `varchar / varchar(x) / char(x) / text`
	▯ `SessionDate` as `date` and `SessionTime` as `time`
	▯ `NumberMembers` as `integer / integer(2) / int / int(2)`
	▯ `SessionID` set as a Primary Key
	
	Example 1:
	`CREATE TABLE SESSION(`
	`SessionID Char(4),`
	`Description Varchar,`
	`SessionDate Date,`
	`SessionTime Time,`
	`NumberMembers Integer,`
	`PRIMARY KEY (SessionID)`
	`);`
	
	Example 2:
	`CREATE TABLE SESSION(`
	`SessionID Char(4) NOT NULL PRIMARY KEY,`
	`Description Varchar,`
	`SessionDate Date,`
	`SessionTime Time,`
	`NumberMembers Integer);`
		5

4(b)(iv)	**1 mark** per bullet point
	
	▯ Select `FirstName` and comma and `LastName`
	▯ From `MEMBER`
	▯ Where `MembershipType = "Peak"`
	
	`SELECT FirstName, LastName`
	`FROM MEMBER`
	`WHERE MembershipType = "Peak";`
		3

Answer 28

3(a)(i)	**1 mark** per table
	
	▯ CUSTOMER table has at least customer ID, customer name, address and contact details
	▯ ROOM has at least room number, room type,
	▯ BOOKING has at least booking ID, room number, customer ID, start date, number of nights
	
	`CUSTOMER (CustomerID, Name, Address, ContactDetails)`
	
	`ROOM (RoomNumber, RoomType)`
	
	`BOOKING (BookingID, RoomNumber, CustomerID, StartDate, NumberNights)`
		3

3(a)(ii)	**1 mark** for 1 or 2 correct Primary Keys, **2 marks** for 3 correct Primary Keys
	
	`CUSTOMER: CustomerID`
	
	`ROOM: RoomNumber`
	
	`BOOKING: BookingID`
		2

3(a)(iii)	**1 mark** for both table name and Foreign Key Table: `BOOKING` Foreign Key: `CustomerID / RoomNumber`	1
3(b)	**1 mark** per bullet point to **max 2** plus **1 mark** for suitable example for each DBMS tool **Developer Interface** ▯ 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 e.g. buttons and menus **Query processor** ▯ To create SQL/QBE queries ▯ To search for data that meets **set criteria**, e.g. all bookings for next week ▯ To perform calculations on extracted data, e.g. number of empty rooms tomorrow	5
3c	**1 mark** for at least two correct rows, **2 marks** for all four correct rows	2

Script	DDL	DML
`CREATE TABLE FILMS`	✓	
`SELECT FilmID FROM FILMS`		✓
`ALTER TABLE FILMS ADD PRIMARY KEY (FilmID)`	✓	
`CREATE DATABASE MYDATA`	✓	

Answer 29

7(a)(i)	**1 mark** per bullet • UserName is the primary key in USER • UserName is (included as) a foreign key in PHOTO	2
7(a)(ii)	**1 mark** for each correct relationship 	2

7(b)	**1 mark** per bullet to **max 2** for explanation • Referential integrity is making sure tables do not try to reference data which does not exist // A value of one attribute of a table exists as a value of another attribute in a different table • A primary key cannot be deleted unless all dependent records are already deleted • Cascading delete • A primary key cannot be updated unless all dependent records are already updated • Cascading update / edit • Every foreign key value has a matching value in the corresponding primary key • The foreign keys must be the same data type as the corresponding primary key **1 mark** for a suitable example e.g. • A UserName cannot be deleted from the USER table if they have a related photo/textpost • If UserName is updated in USER table, it must also be updated in PHOTO and TEXTPOST tables • Cannot create/edit a record in TEXTPOST / PHOTO without a matching entry in USER table	3
7(c)	**Max 1 mark** from each bulleted group **1NF** • No repeated groups of attributes • All attributes should be atomic • No duplicate rows **2NF** (in 1NF and) • No partial dependencies **3NF** (in 2NF and) • No non-key dependencies • No transitive dependencies	3

7(d)(i)	**1 mark** per bullet ▯ CREATE TABLE USER and (); ▯ UserName, FirstName and SecondName as VARCHAR and commas ▯ DateOfBirth as DATE and comma ▯ PRIMARY KEY(UserName) ▯ An appropriate NOT NULL CREATE TABLE USER(UserName: varchar(15) NOT NULL, FirstName: varchar(25), SecondName: varchar(25), DateOfBirth: Date, PRIMARY KEY(UserName));	5

Question	Answer	Marks
7(d)(ii)	**1 mark** per bullet ▯ ALTER TABLE USER ▯ ADD COUNTRY varchar; ALTER TABLE USER ADD Country varchar;	2

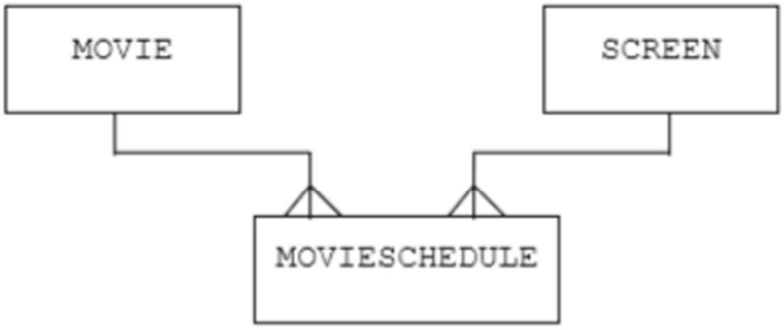
Answer 30

2(a)	**1 mark** for each correct relationship  	**2**
2(b)	**1 mark** for description - Ensure data is consistent / accurate // keep data consistent / accurate **1 mark** for example from: e.g. - Validation rules - Referential integrity - Verification - Input masks - Setting data types - Removing redundant data - Backup data - Access controls - Audit trail	**2**
2(c)	**1 mark** for the correct box ticked **True**: ✓ / **False**: (blank) **1 mark** per bullet for justification, **max 2** - No repeated attributes // data is atomic // No partial dependencies (no dual keys) - No non-key / transitive dependencies	**3**

2(d)(i)	**1 mark** per bullet
	
	- `CREATE TABLE PLAYER` and `();`
	- `PlayerID` and `PlayerName` as `VARCHAR` and commas
	- `SkillLevel` as `INT` and comma
	- `PRIMARY KEY(PlayerID)`
	- An appropriate `NOT NULL`
	
	`CREATE TABLE PLAYER(`
	`PlayerID: varchar NOT NULL,`
	`PlayerName: varchar,`
	`SkillLevel: int,`
	`PRIMARY KEY(PlayerID),`
	`);`
		5

2(d)(ii)	**1 mark** per bullet - `ALTER TABLE PLAYER` - `ADD DateOfBirth Date;`	**2**

Answer 31

7(a)	**1 mark** for each correct join 	**2**
7(b)	**1 mark** per bullet point ▯ MovieID is the Primary Key in MOVIE ▯ ... links to MovieID which is the Foreign Key in MOVIESCHEDULE ▯ ScreenNumber is the Primary Key in SCREEN ▯ ... links to ScreenNumber which is the Foreign Key in MOVIESCHEDULE	**4**
7(c)	**1 mark** per bullet point ▯ ALTER TABLE MOVIE ▯ ADD (COLUMN) ProductionCompany VARCHAR(25);	**2**

7(d)	**1 mark** per bullet point Answer 1: ▯ SELECT Title, Rating ▯ FROM MOVIE, MOVIESCHEDULE ▯ WHERE MOVIE.MovieID = MOVIESCHEDULE.MovieID ▯ AND MOVIESCHEDULE.ScreenNumber = 3; **Or** Answer 2: ▯ SELECT Title, Rating ▯ FROM MOVIE INNER JOIN MOVIESCHEDULE ▯ ON MOVIE.MovieID = MOVIESCHEDULE.MovieID ▯ WHERE MOVIESCHEDULE.ScreenNumber = 3;	**4**

Answer 32

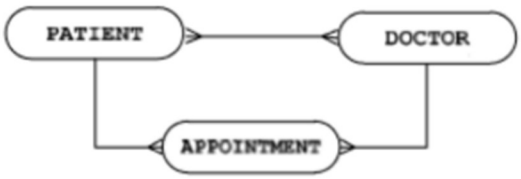
1(a)	Many-to-one	1
1(b)(i)	A-NURSE(NurseID, FirstName, FamilyName, **WardName**)	1
1(b)(ii)	☐ The primary key WardName in the A-WARD table ... 1 ☐ ... links to the foreign key WardName in the A-NURSE table. 1	2
1(c)(i)	Many-to-many relationship	1
1(c)(ii)	B-WARD-NURSE(WardName, NurseID) Both attributes (with no additions) 1 Joint primary key correctly underlined 1	2
1(c)(iii)	 Correct relationship between B-NURSE and B-WARD-NURSE 1 Correct relationship between B-WARD and B-WARD-NURSE 1	2
1(d)(i)	SELECT NurseID, FamilyName 1 FROM B-NURSE 1 WHERE Specialism = 'THEATRE'; 1	3
1(d)(ii)	UPDATE **B-NURSE** 1 SET **FamilyName = 'Chi'** 1 WHERE **NurseID = '076';** 1	3

Answer 33

1(a)	Many-to-many relationship	1
1(b)(i)	 Both entities correctly labelled 1 Correct relationship between SHOP and SHOP-SUPPLIER 1 Correct relationship between SUPPLIER and SHOP-SUPPLIER 1	3
1(b)(ii)	<table><tr><th>Table</th><th>Primary key</th><th>Foreign keys(s) (if any)</th><th>Explanation</th></tr><tr><td>SHOP</td><td>ShopID</td><td>None</td><td></td></tr><tr><td>SUPPLIER</td><td>SupplierID</td><td>None</td><td></td></tr><tr><td>SHOP-SUPPLIER</td><td>ShopID AND SupplierID</td><td>ShopID OR SupplierID (or both)</td><td>To create a link with the SHOP or SUPPLIER table.</td></tr></table> □ SHOP has primary key ShopID and SUPPLIER has primary key SupplierID 1 □ SHOP-SUPPLIER has primary key ShopID + SupplierID 1 □ Both SHOP and SUPPLIER show foreign key as 'None' 1 □ SHOP-SUPPLIER shows foreign key ShopID or SupplierID 1 □ Explanation for SHOP-SUPPLIER foreign key describes ShopID or SupplierID creating a link 1	5

1(b)(iii)	**Two** from: □ The database user will <u>frequently</u> want to search on contact name 1 □ The contact name attribute has been indexed 1 □ It allows for a <u>fast/faster</u> search using contact name 1	**Max 2**
1(c)(i)	`SELECT ShopID, Location` 1 `FROM SHOP` 1 `WHERE RetailSpecialism = 'GROCERY';` 1	3
1(c)(ii)	`INSERT INTO SHOP-SUPPLIER` 1 `(ShopID, SupplierID)` 1 `VALUES (8765, 'SUP89');` 1	3

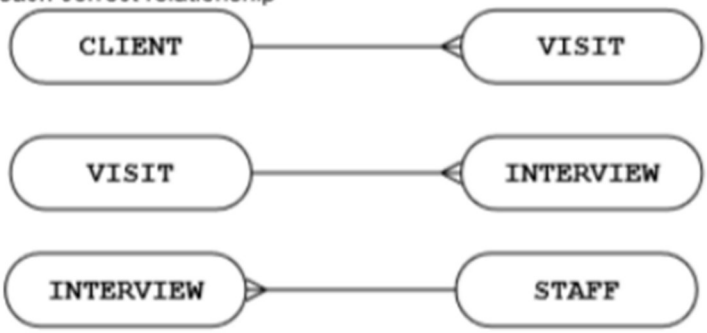
Answer 34

7(a)(i)	PatientID DoctorID } (1) AppointmentDate, AppointmentTime (1)	2
7(a)(ii)	 One PATIENT attends many APPOINTMENTs One DOCTOR takes many APPOINTMENTs Special case for 1 mark only (only if no one to many relationships shown) Many PATIENTs are seen by many DOCTORs	2

Question	Answer	Marks
7(b)	**Two marks** from: Either: ▯ Add an attribute (for example Attended) ▯ To the appointment table // APPOINTMENT Or: ▯ Add an attribute (for example AppointmentsMissed) ▯ To the patient table // PATIENT	2
7(c)(i)	Available to work at both SITE-A and SITE-B	1
7(c)(ii)	APPOINTMENT(Site, AppointmentDate, AppointmentTime, DoctorID, PatientID)	1
7(d)(i)	**One mark** per line UPDATE DOCTOR SET DoctorID = '017' WHERE DoctorID = '117';	3
7(d)(ii)	**1 Mark** per bullet, max 2 ▯ Referential integrity should be maintained // Referential integrity could be violated. ▯ Data becomes inconsistent ▯ There may be records in the APPOINTMENT table showing doctor ID 117 ▯ The APPOINTMENT table might not be automatically updated ▯ Records in the APPOINTMENT table will become orphaned	Max 2
7(e)	**One mark** per line SELECT AppointmentDate, AppointmentTime FROM APPOINTMENT WHERE PatientID = '556';	3

Answer 35

7(a)(i)	**1 Mark** for correct primary key identified in both `STAFF` and `CLIENT`
	`STAFF(StaffID, StaffName, Department)`
	`CLIENT(ClientName, Address, Town)`
	
	1 Mark for correct primary key identified in `VISIT`
	`VISIT(ClientName, VisitDate)`
	
	1 Mark for correct primary key identified in `INTERVIEW`
	`INTERVIEW(ClientName, VisitDate, StaffID, SpecialistFocus, InterviewText)`
		3

7(a)(ii)	**1 Mark** for each correct relationship 	3
7(b)	**1 Mark** for correct answer Add attribute `VisitReportText` to table `VISIT`	1

7(c)(i)	**1 Mark** for each correct line
	
	`UPDATE CLIENT`
	`SET ClientName = 'Albright Holdings'`
	`WHERE ClientName = 'ABC Holdings';`
		3

7(c)(ii)	**1 Mark** per bullet, max 2 - Referential integrity should be maintained // Referential integrity could be violated - Data becomes inconsistent - There may be records in the `VISIT` and `INTERVIEW` tables / other tables with client name ABC Holdings - The `ClientName` in the `VISIT` and `INTERVIEW` tables / other tables might not be automatically updated - Records in the `VISIT` and `INTERVIEW` tables / other tables will become orphaned	2

Question	Answer	Marks

7(d)	**1 Mark** for each correct line
	
	`SELECT StaffID FROM INTERVIEW`
	`WHERE ClientName = 'New Age Toys'`
	`AND VisitDate = '13/10/2016';` (Accept clauses other way round)
		3

7(e)	**1 Mark** for a correct answer Add a suitable attribute, for example, `EuropeTraveller` to the `STAFF` table // Add a suitable attribute, for example, `Country` to the `CLIENT` table	1

Answer 36

(a) (i) Database Management System **[1]**

(ii) **One mark** for identifying the way in which the data security is ensured, and **one mark** for a further description.

Maximum of two marks per method. **Maximum of two** methods. **[4]**

- Issue usernames and passwords...
  - stops unauthorised access to the data
  - any further expansion e.g. strong passwords / passwords should be changed regularly etc…
- Access rights / privileges...
  - so that only relevant staff / certain usernames can read/edit certain parts of the data
  - can be read only, or full access / read, write and delete
  - any relevant example e.g. only class tutors can edit details of pupils in their tutor group
- Create (regular / scheduled) backups...
  - in case of loss/damage to the live data a copy is available
  - any relevant example e.g. backing up the attendance registers at the end of each day and storing the data off-site/to a separate device
- Encryption of data...
  - if there is unauthorised access to the data it cannot be understood // needs a decryption key
  - any relevant example e.g. personal details of pupils are encrypted before being sent over the Internet to examination boards
- Definition of different views...
  - composed of one or more tables
  - controls the scope of the data accessible to authorised users
  - any relevant example e.g. teachers can only see their classes
- Usage monitoring / logging of activity...
  - creation of an audit /activity log
  - records the use of the data in the database / records operations performed by all users / all access to the data
  - any relevant example, e.g. Track who changed a student's grade

(iii) Two points from: **[2]**

- Set up search criteria
- To find / retrieve / return the data that matches the criteria
- Any relevant example e.g. find pupils who were absent on a particular day

(iv) Three points from: **[3]**

- By storing data in (separate) linked tables data redundancy is reduced / data duplication is controlled...
- Compatibility / data integrity issues are reduced as data only needs to be updated once / is only stored once.
- Unwanted or accidental deletion of linked data is prevented as the DBMS will flag an error.
- Program - data dependence is overcome.
- Changes made to the structure of the data have little effect on existing programs.
- Ad-hoc / complex queries can be more easily made as the DBMS will have a query language/ QBE form.
- Unproductive maintenance is eliminated as changes only need to be made once (rather than changing multiple programs).
- Fields can be added or removed without any effect on existing programs (that do not use these fields).
- Security / privacy of the data is improved as each application only has access to the fields it needs.
- There is better control of data integrity as the DBMS (uses its Data Dictionary) to perform validation checks on data entered.

(b) (i) Two points from: **[2]**

- The Primary Key in **CLASS** is ClassID
- The Foreign Key of **CLASS-GROUP** is ClassID.
- The Primary Key of CLASS is also included in CLASS-GROUP as a Foreign Key, (which links to CLASS table)

(ii) Many-to-one **[1]**

(iii) One mark per statement. Several statements may be on the same line. **[4]**

```
SELECT StudentID, FirstName
FROM STUDENT
WHERE TutorGroup = "10B" // WHERE (TutorGroup = "10B")
ORDER BY LastName ASC;
```

(iv) One mark per statement. Several statements may be on the same line. **[4]**

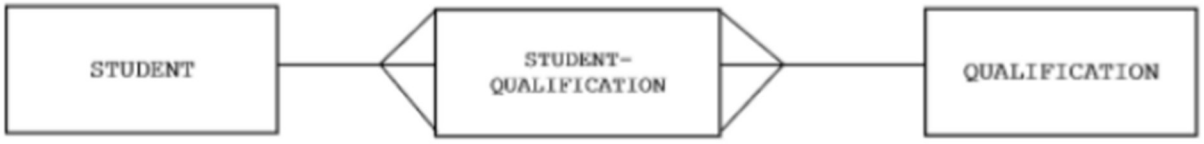
```
SELECT STUDENT.LastName
FROM STUDENT, CLASS-GROUP
WHERE ClassID = "CS1" // WHERE (ClassID = "CS1")
AND CLASS-GROUP.StudentID = STUDENT.StudentID;
```

One mark per statement. Several statements may be on the same line.

```
SELECT STUDENT.LastName
FROM STUDENT INNER JOIN CLASS-GROUP
ON CLASS-GROUP.StudentID = STUDENT.StudentID
WHERE ClassID = "CS1" // WHERE (ClassID = "CS1");
```

Answer 37

(c) (i) One mark for each correct relationship. **[2]**



(ii) One-to-many **[1]**

(iii) Two points from: **[2]**

- The primary key in the QUALIFICATION table is QualCode.
- The foreign key in the STUDENT-QUALIFICATION table is QualCode.
- The primary key of QUALIFICATION is also included in QualCode.

(d) (i) One mark per statement. Several statements may be on one line. **[2]**

```
ALTER TABLE STUDENT
ADD DateOfBirth DATE;
```

(ii) One mark per statement. Several statements may be on one line. **[3]**

```
SELECT StudentID, Grade, DateOfAward
FROM STUDENT-QUALIFICATION
WHERE QualCode = 'SC12';
```

(iii) One mark per statement. Several statements may be on one line. **[4]**

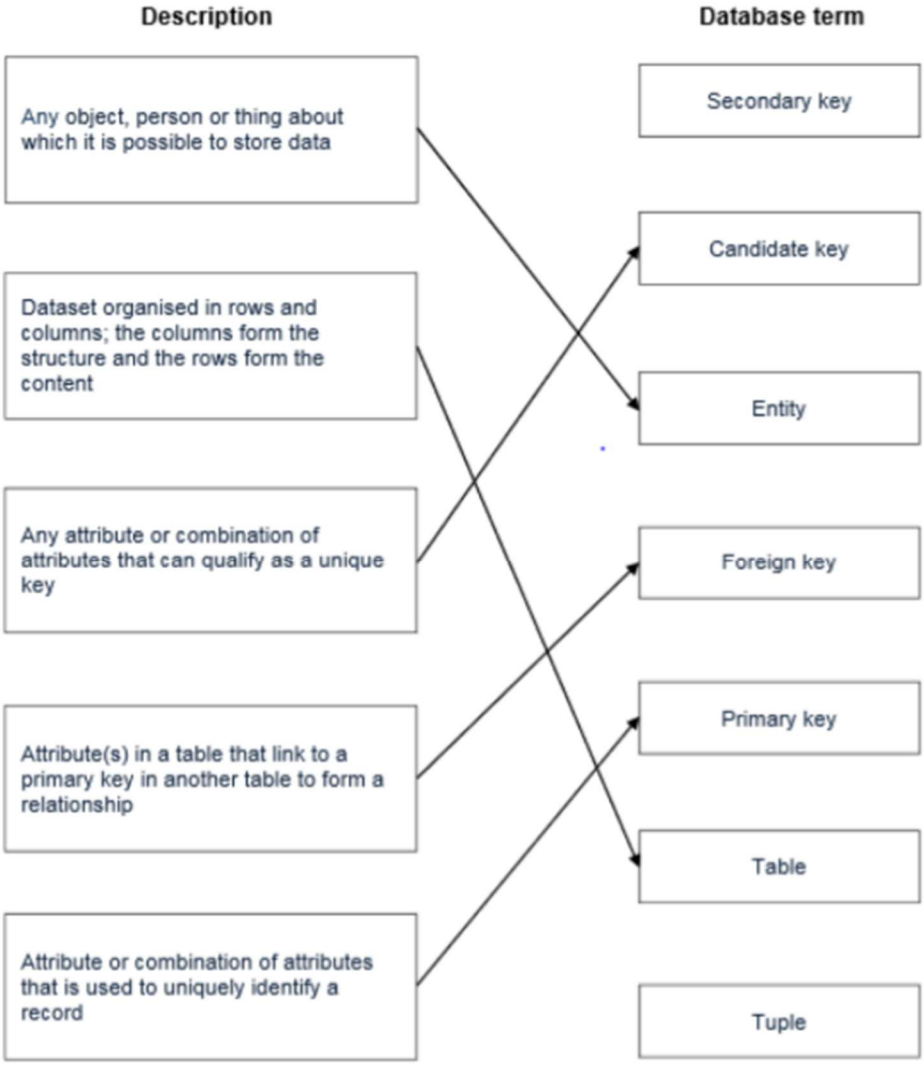
```
SELECT STUDENT.FirstName, STUDENT.LastName, STUDENT-
    QUALIFICATION.QualCode
FROM STUDENT, STUDENT-QUALIFICATION
WHERE STUDENT-QUALIFICATION.Grade = 'A'
AND STUDENT.StudentID = STUDENT-QUALIFICATION.StudentID;
```

Alternative answer:

```
SELECT FirstName, LastName, STUDENT-QUALIFICATION.QualCode
FROM STUDENT, INNER JOIN STUDENT-QUALIFICATION
ON STUDENT.StudentID = STUDENT-QUALIFICATION.StudentID
WHERE Grade = 'A';
```

Answer 38

(a) **One mark** for each correct line.
Two lines from any box on left means no mark for that description.



(b) Any **three** from:

- Ensures related data in tables are consistent

- If one table has a foreign key (the 'foreign' table)...
- ... then it is not possible to add a record to that table / the 'foreign' table
- ... unless there is a corresponding record in the linked table with a corresponding primary key (the 'primary' table)
- Cascading delete
- If a record is deleted in the 'primary' table...
- all corresponding linked records in 'foreign' tables must also be deleted

- Cascading update
- If a record in the 'primary' table is modified...
- ... all linked records in foreign tables will also be modified [3]

Answer 39

(a) **ONE** mark for each reason and **ONE** mark for a further explanation. **MAX THREE** reasons.

- Reduced data redundancy / data duplication
- Data is stored in (separate) linked tables

- The database (generally) stores data only once / data need only be updated once
- Improved data consistency / integrity / associated data will be automatically updated / easier to maintain the data / elimination of unproductive maintenance

- Complex queries can be more easily written
- To search / find specific data // specific example related to the Health Club

- Fields can be more easily added to or removed from tables
- Without affecting existing applications (that do not use these fields)

- Program-data dependence is overcome
- Changes to the data (design) do not require changes to programs // changes to programs do not require changes to data // the data can be accessed by any appropriate program

- Security is improved
- Each application only has access to the fields it needs // different users can be given different access rights

- Different users can be given different views of the data / data privacy is maintained
- So they do not see confidential information

- Allows concurrent access
- Record locking prevents two users updating the same record at the same time // record locking assures data consistency

[6]

(b) **ONE** mark for each correct relationship as shown.

[2]

(c) An example of a script is shown, but different syntax may be used.

```
CREATE TABLE CLASS (
     ClassID VARCHAR(5),
     Description VARCHAR(30),
     StartDate DATE,
     ClassTime TIME,
     NoOfSessions INT,
     AdultsOnly BIT,
     PRIMARY KEY(ClassID)
);
```

Mark as follows:

1 mark for `CREATE TABLE CLASS` and `();`
1 mark for `PRIMARY KEY(ClassID)`
1 mark for both `ClassID VARCHAR(5),` and `Description VARCHAR(30),`
1 mark for both `StartDate DATE,` and `ClassTime TIME,`
1 mark for `NoOfSessions INT,`
1 mark for `AdultsOnly BIT,`

[6]

Answer 40

9 **(a)** Any **one** from:

- (ShopSales) table has repeated group (of attributes)
- each sales person has a number of products
- FirstName, Shop would need to be repeated for each record

(b) **One mark** for `SalesPerson` table

table: `SalesPerson`

FirstName	Shop
Nick	TX
Sean	BH
John	TX

table: `SalesProducts`

FirstName	ProductName	NoOfProducts	Manufacturer
Nick	television set	3	SKC
Nick	refrigerator	2	WP
Nick	digital camera	6	HKC
Sean	hair dryer	1	WG
Sean	electric shaver	8	BG
John	television set	2	SKC
John	mobile phone	8	ARC
John	digital camera	4	HKC
John	toaster	3	GK

(c) (i) Any **two** from:

- primary key of SalesPerson table is FirstName
- links to FirstName in SalesProducts table
- FirstName in SalesProductsS table is foreign key [2]

(ii)
- There is a non-key dependency
- Manufacturer is dependent on ProductName, (which is not the primary key of the SalesProducts table) [2]

(iii) SalesPerson (FirstName, Shop)
-SalesProducts (FirstName, ProductName, NoOfProducts) **OR**
SalesProducts(SalesID, FirstName, ProductName, NoOfProducts)

-Product (ProductName, Manufacturer)

1 mark for correct attributes in SalesProducts and Product tables and **1 mark** for correct identification of both primary keys [2]