Consider the computer games industry scenario described below and answer the questions in part a) and b) below. The following rules describe the scenario:

- A game has a title and an age rating.
- A game has exactly one genre (for example, 'role play').
- A genre is described by its name.
- A game is made by a producer.
- Producers can collaborate on making one game.
- A producer can make more than one game (or might not yet have made a game).
- A producer has a name.
- A game runs on one or more platforms.
- Each platform supports one or more games.
- A platform has exactly one company making it.
- A company might create more than one platform.
- A platform has a year of introduction to the market and a name.
- A company has a name and a country for its head quarter.

a) Create an entity relationship diagram for the scenario provided using a suitable notation. Your answer must show entities with their attributes and their relations (including cardinality and optionality).

(13 marks)

b) Using your design from part a) design a set of tables. Clearly identify all primary and foreign keys and show a few (up to 3) rows of data.

(12 marks)

END OF EXAMINATION
b) The following symbols are used in ERD diagrams (crow’s foot notation or Chen’s notation) and relate to the cardinality and modality of associations. Explain what they mean:

i) \( \circ \) or the equivalent \((0,M)\) (2 marks)

ii) \( \leq \) or the equivalent \((1,1)\) (2 marks)

c) Relational algebra provides a mathematical foundation to relation databases. Consider the following relational algebra statements and provide their SQL equivalent.

i) \( \forall \text{name}, \text{capital} \; \text{name} = "Spain" \; \text{countries} \) (2 marks)

ii) \( \exists \text{movies}.\text{title} \; \text{actor}.\text{name} = "Brad Pitt" \; (\text{movies} \times \text{actor}) \) (2 marks)

iii) \( \forall \text{customers}.\text{customername}, \forall \text{orders}.\text{orderid} \; (\text{customers} \Join \text{orders}) \) (3 marks)

d) Assume tables A and B, with their respective keys AID and BID. Write the SQL query that returns all values in B that are not in A using a suitable JOIN notation (as indicated in the shaded section in the following diagram):

(4 marks)
Section B
Answer Section B questions in Answer Book B

B4.

a)

i) Provide the definitions for Boyce Codd Normal Form. (3 marks)

ii) Explain the concept of ‘update anomaly’ that might occur in a database that is not normalised. Provide an example. (3 marks)

b) Consider the following unnormalised table:

<table>
<thead>
<tr>
<th>Full Name</th>
<th>Title</th>
<th>Address</th>
<th>Movies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adam Smith</td>
<td>Mr</td>
<td>15 Park Lane</td>
<td>Pirates of the Caribbean, Robin Hood</td>
</tr>
<tr>
<td>Adam Smith</td>
<td>Mr</td>
<td>28 The Mall</td>
<td>Harry Potter, Love Actually</td>
</tr>
<tr>
<td>Berta Miller</td>
<td>Ms</td>
<td>12 Princess Street</td>
<td>Harry Potter, Robin Hood</td>
</tr>
</tbody>
</table>

i) Convert the table into 1st Normal Form. (3 marks)

ii) Convert the table into 2nd Normal Form. (6 marks)

iii) Convert the table into 3rd Normal Form. (3 marks)

c) Provide the SQL create table statements required to create the set of 3rd NF tables identified in part b). Suitable data types for the attributes should be chosen. Ensure that suitable referential integrity constraints are included. You do not need to show the insertion of any data in the tables. (7 marks)

B5.

a) Define the following concepts and provide an example for each:

a) Attribute. (2 marks)

b) Tuple. (2 marks)

c) One-to-many relationship. (2 marks)

d) Functional dependency. (2 marks)

e) Modality (of a relation). (2 marks)
Refer to the **Products**, **Shops** and **ShopProducts** tables supplied below for this question.

### Products Table

<table>
<thead>
<tr>
<th>PRODUCTID</th>
<th>PRODUCTNAME</th>
<th>PRODUCTTYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Apple</td>
<td>Fruit</td>
</tr>
<tr>
<td>2</td>
<td>Mango</td>
<td>Fruit</td>
</tr>
<tr>
<td>3</td>
<td>Leeks</td>
<td>Vegetable</td>
</tr>
<tr>
<td>4</td>
<td>Cauliflower</td>
<td>Vegetable</td>
</tr>
<tr>
<td>5</td>
<td>Roses</td>
<td>Flowers</td>
</tr>
<tr>
<td>6</td>
<td>Milk</td>
<td>Dairy</td>
</tr>
</tbody>
</table>

### Shops Table

<table>
<thead>
<tr>
<th>SHOPID</th>
<th>SHOPNAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mandana</td>
</tr>
<tr>
<td>2</td>
<td>Tobeys</td>
</tr>
<tr>
<td>3</td>
<td>Hansels</td>
</tr>
<tr>
<td>4</td>
<td>FritzStores</td>
</tr>
</tbody>
</table>

### ShopProducts table

<table>
<thead>
<tr>
<th>SHOPID</th>
<th>PRODUCTID</th>
<th>QUANTITYINSTOCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>9</td>
</tr>
</tbody>
</table>

---

a) Consider the following SQL query:

**SQL query 1:**

```sql
SELECT SUM(QuantityInStock), sp.ShopID, s.shopname
FROM shopProducts sp
INNER JOIN shops s
ON s.shopID = sp.shopID
GROUP BY sp.shopID, s.shopname;
```

i) State the overall function of SQL statement query1. Show the output produced from SQL query 1. (2 marks)

ii) Explain the SQL keywords **SUM**, **INNER JOIN** and **GROUP BY**. (5 marks)

---

b) Consider the SQL statement in SQL query 2 below:

**SQL query 2:**

```sql
SELECT products.* , shops.shopname
FROM products
INNER JOIN shopProducts
ON products.productID = shopproducts.productID
INNER JOIN shops
ON shopProducts.shopID = shops.shopID
WHERE shopProducts.QuantityInStock > 7
AND products.productType  = 'Fruit';
```

i) Show the result set that is returned when SQL query 2 is executed. (5 marks)

ii) Summarise what the results show about the shops and products data. (2 marks)