Northumbria University
NEWCASTLE

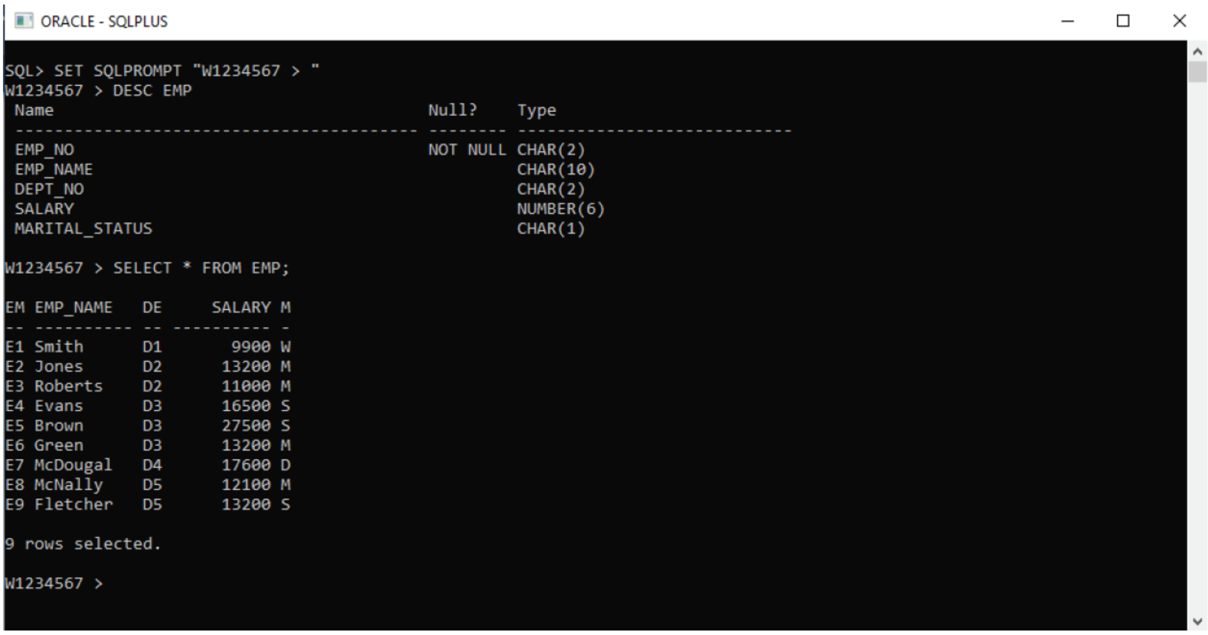
**Department of Computer & Information Sciences**

# ASSESSMENT SUBMISSION

| | |
|---|---|
| **Module Title:** | Advanced Databases |
| **Module Code:** | KL7011 |
| **Academic Year / Semester:** | 2023-24 / Semester 1 |
| **Module Tutor / Email (all queries):** | Akhtar Ali akhtar.ali@northumbria.ac.uk |
| **% Weighting (to overall module):** | 60% |
| **Assessment Title:** | Assignment 1: individual work |
| **Date of Handout to Students:** | 20th October 2023 |
| **Mechanism for Handout:** | Module Blackboard Site |
| **Deadline for Submission Attempt by Students:** | Thursday 30/11/2023 @ 23:59 GMT |
| **Mechanism for Submission:** | Document upload to Module Blackboard Site |
| **Submission Format / Word Count** | Please upload your written report as a single PDF document |
| **Date by which Work, Feedback and Marks will be returned:** | 31st December 2023 |
| **Mechanism for return of Feedback and Marks:** | Mark and individual written feedback will be uploaded to the Module Site on Blackboard. For further queries please email module tutor. |
| **Student ID** | W22058786 |
| **Oracle Username** | advDBusr285 |
| **Student Name** | Christy Joseph |

## Personalising your SQL output/prompt

Before executing any **SQL code** for this assignment, you should personalise your SQL output / prompt by running SET SQLPROMPT “UniversityUserName > ”, i.e., *double-quote* followed by your UniversityUserName followed by > and then a *space* and *double-quote* as shown in the screenshot below:



```
SQL> SET SQLPROMPT "W1234567 > "
W1234567 > DESC EMP
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMP_NO                                    NOT NULL CHAR(2)
 EMP_NAME                                           CHAR(10)
 DEPT_NO                                            CHAR(2)
 SALARY                                             NUMBER(6)
 MARITAL_STATUS                                     CHAR(1)

W1234567 > SELECT * FROM EMP;

EM EMP_NAME   DE     SALARY M
-- ---------- -- ---------- -
E1 Smith      D1       9900 W
E2 Jones      D2      13200 M
E3 Roberts    D2      11000 M
E4 Evans      D3      16500 S
E5 Brown      D3      27500 S
E6 Green      D3      13200 M
E7 McDougal   D4      17600 D
E8 McNally    D5      12100 M
E9 Fletcher   D5      13200 S

9 rows selected.

W1234567 >
```

## Assignment Questions

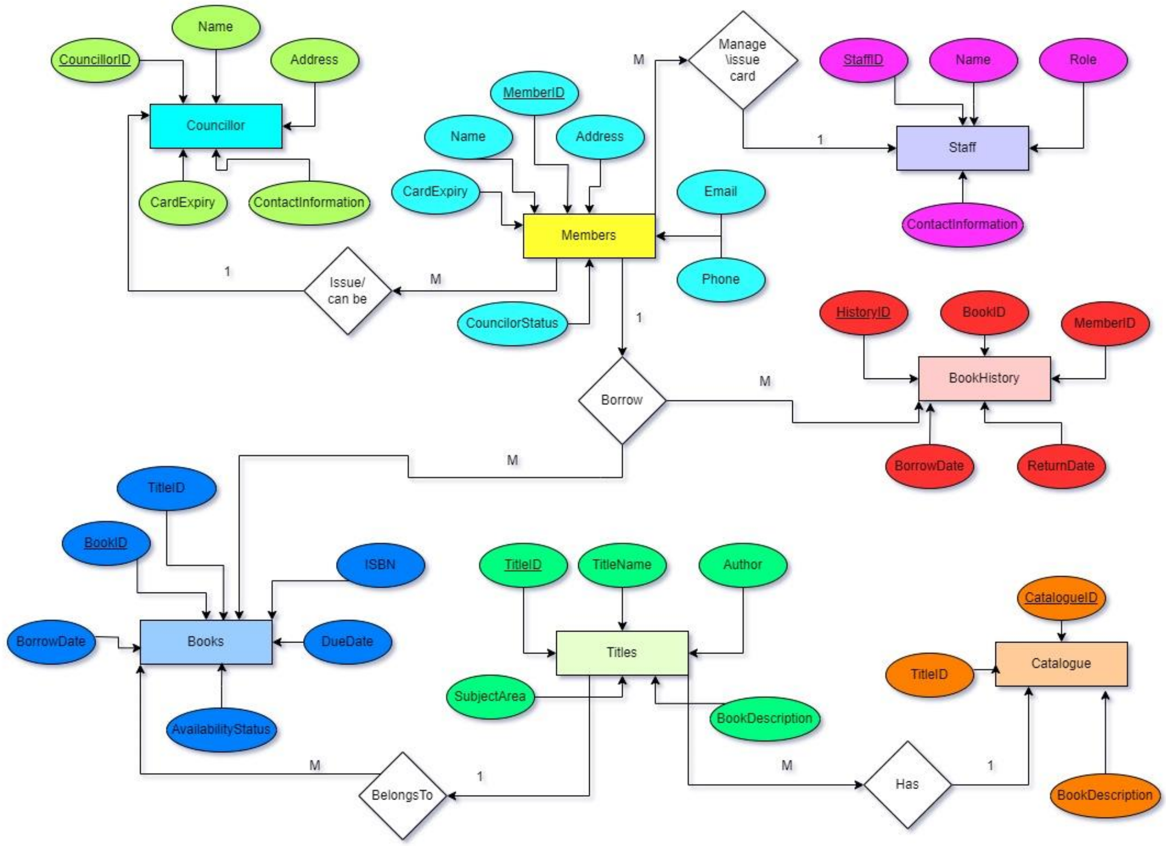
### Part 1 (35 marks)

This part is based on the BLS scenario as described in the Appendix.

(A) *Using entity-relationship (ER) OR enhanced entity-relationship (EER) modelling, produce a conceptual design for the database to support the BLS required by BCC.*

(10 marks)

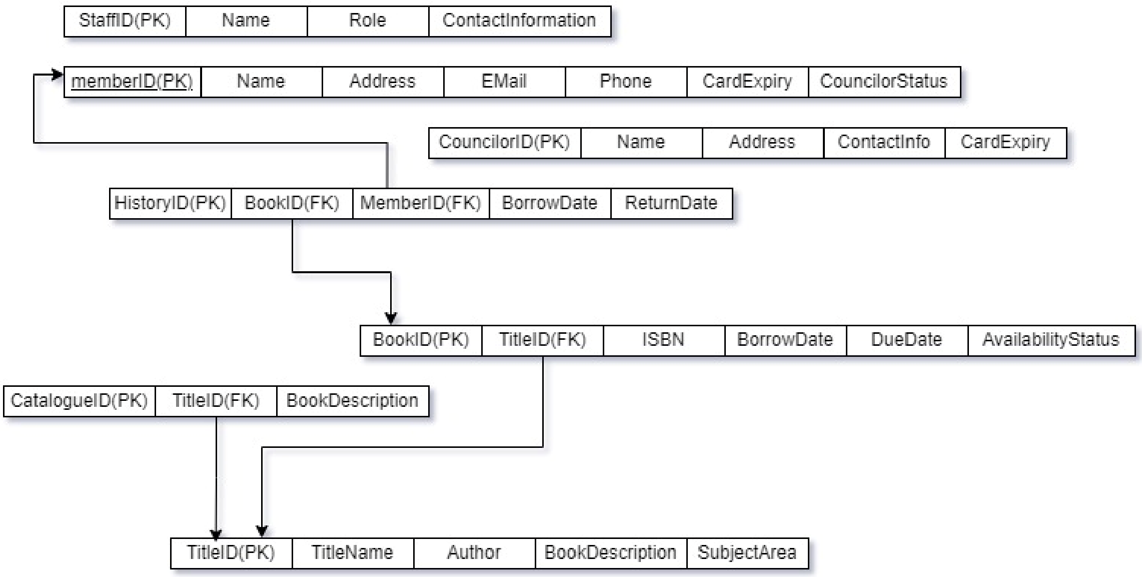
## Answer Part 1 A: Insert your ER or EER Diagram Below



*(B) Convert the ER / EER diagram from Part 1(A) to produce a logical relational schema using ER / EER to relational mapping.*

(15 marks)

## Answer Part 1 B: Provide your Logical Relational Design/Schema Below



**Answer Part 1 B: Provide your Data Dictionary Below (*in a tabular form and must be presented as text rather than an image or picture*)**

(5 marks)

| Table | Attribute | Data Type | Constraints | Description |
|---|---|---|---|---|
| Member | MemberID (PK) | INT | PRIMARY KEY | Unique identifier for a library member |
| Member | Name | VARCHAR | - | Name of the library member |
| Member | Address | VARCHAR | - | Address of the library member |
| Member | Email | VARCHAR | - | Email address of the library member |
| Member | Phone | VARCHAR | - | Phone number of the library member |
| Member | CardExpiry | DATE | - | Expiry date of the library member's card |
| Member | CouncillorStatus | BOOLEAN | - | Status indicating if a member is a councillor |
| Book | BookID (PK) | INT | PRIMARY KEY | Unique identifier for a library copy |
| Book | TitleID (FK) | INT | FOREIGN KEY REFERENCES | Foreign key referencing TitleID in Title table |
| Book | AvailabilityStatus | VARCHAR | - | Status indicating whether the copy is available |
| Book | ISBN | VARCHAR | - | ISBN of the book copy |
| Book | BorrowDate | DATE | - | Date when the copy was borrowed |

| Table | Attribute | Data Type | Constraints | Description |
|---|---|---|---|---|
| Book | DueDate | DATE | - | Due date for the borrowed copy |
| Title | TitleID (PK) | INT | PRIMARY KEY | Unique identifier for a book title |
| Title | TitleName | VARCHAR | - | Name of the book title |
| Title | Author | VARCHAR | - | Author of the book title |
| Title | SubjectArea | VARCHAR | - | Subject area of the book title |
| Title | BookDescription | TEXT | - | Description of the book title |
| Catalogue | CatalogueID (PK) | INT | PRIMARY KEY | Unique identifier for a catalogue entry |
| Catalogue | TitleID (FK) | INT | FOREIGN KEY REFERENCES | Foreign key referencing TitleID in Title table |
| Catalogue | BookDescription | TEXT | - | Description of the book in the catalogue |
| Staff | StaffID (PK) | INT | PRIMARY KEY | Unique identifier for a staff member |
| Staff | Name | VARCHAR | - | Name of the staff member |
| Staff | Role | VARCHAR | - | Role of the staff member |
| Staff | ContactInformation | VARCHAR | - | Contact information of the staff member |
| Councillor | CouncillorID (PK) | INT | PRIMARY KEY | Unique identifier for a councillor |
| Councillor | Name | VARCHAR | - | Name of the councillor |
| Councillor | Address | VARCHAR | - | Address of the councillor |
| Councillor | CardExpiry | DATE | - | Expiry date of the councillor's card |
| Councillor | ContactInformation | VARCHAR | - | Contact information of the councillor |
| BookHistory | HistoryID (PK) | INT | PRIMARY KEY | Unique identifier for a copy history |
| BookHistory | MemberID (FK) | INT | FOREIGN KEY REFERENCES | Foreign key referencing MemberID in Member table |
| BookHistory | BookID (FK) | INT | FOREIGN KEY REFERENCES | Foreign key referencing CopyID in Copy table |
| BookHistory | BorrowDate | DATE | - | Date when the copy was borrowed |
| BookHistory | ReturnDate | DATE | - | Date when the copy was returned |

*(C) Based on your logical design from Part 1 (B) and the information available in the scenario, produce an SQL script file using Oracle 11g/12c/higher.*

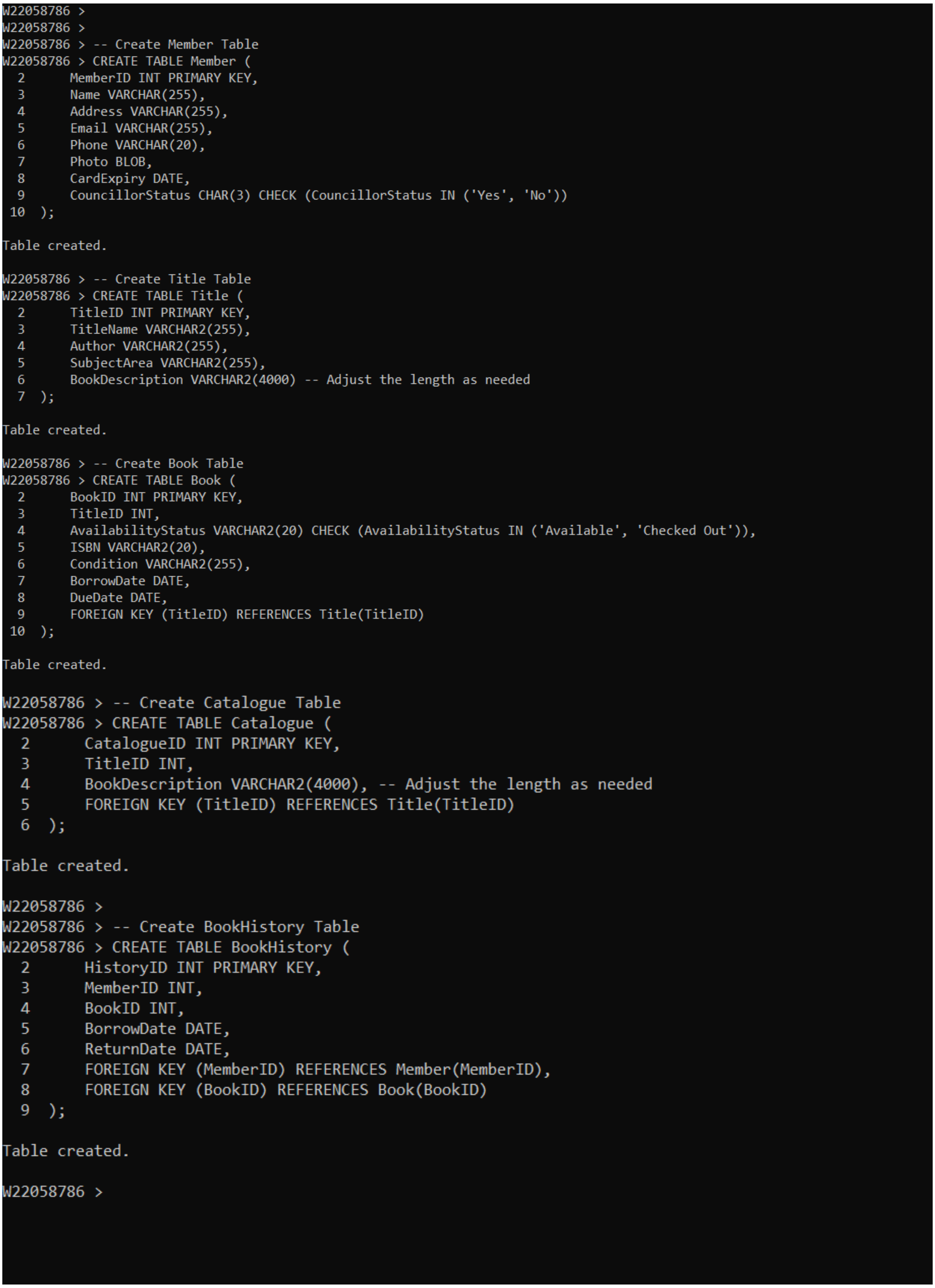
(10 marks)

## Answer Part 1 C: Provide SQL DDL Script file contents (i.e., the SQL code for creating / altering your Tables / Constraints etc)

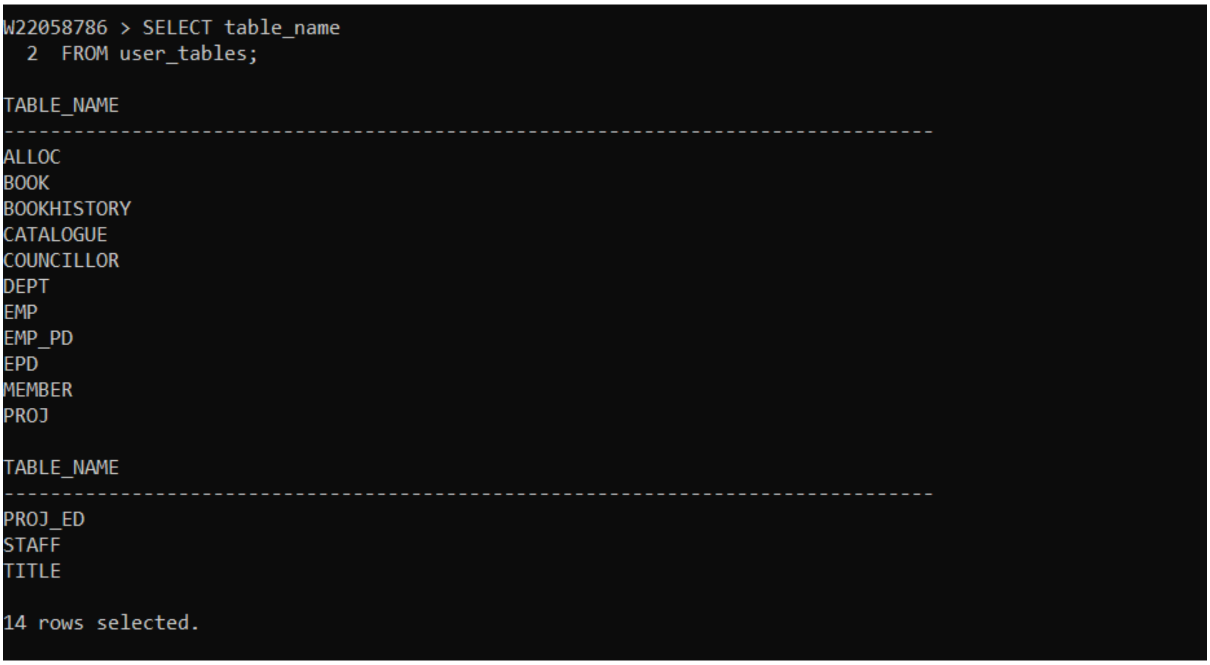
```
-- Create Member Table
CREATE TABLE Member (
    MemberID INT PRIMARY KEY,
    Name VARCHAR(255),
    Address VARCHAR(255),
    Email VARCHAR(255),
    Phone VARCHAR(20),
    Photo BLOB,
    CardExpiry DATE,
    CouncillorStatus CHAR(3) CHECK (CouncillorStatus IN ('Yes',
'No'))
);

-- Create Title Table
CREATE TABLE Title (
    TitleID INT PRIMARY KEY,
    TitleName VARCHAR2(255),
    Author VARCHAR2(255),
    SubjectArea VARCHAR2(255),
    BookDescription VARCHAR2(4000) -- Adjust the length as needed
);

-- Create Book Table
CREATE TABLE Book (
    BookID INT PRIMARY KEY,
    TitleID INT,
    AvailabilityStatus VARCHAR2(20) CHECK (AvailabilityStatus IN
('Available', 'Checked Out')),
    ISBN VARCHAR2(20),
    Condition VARCHAR2(255),
    BorrowDate DATE,
    DueDate DATE,
    FOREIGN KEY (TitleID) REFERENCES Title(TitleID)
);

-- Create Catalogue Table
CREATE TABLE Catalogue (
    CatalogueID INT PRIMARY KEY,
    TitleID INT,
    BookDescription VARCHAR2(4000), -- Adjust the length as needed
    FOREIGN KEY (TitleID) REFERENCES Title(TitleID)
);

-- Create BookHistory Table
CREATE TABLE BookHistory (
    HistoryID INT PRIMARY KEY,
    MemberID INT,
    BookID INT,
    BorrowDate DATE,
    ReturnDate DATE,
    FOREIGN KEY (MemberID) REFERENCES Member(MemberID),
    FOREIGN KEY (BookID) REFERENCES Book(BookID)
);

-- Create Staff Table
CREATE TABLE Staff (
    StaffID INT PRIMARY KEY,
    Name VARCHAR(255),
    Role VARCHAR(255),
    ContactInformation VARCHAR(255)
);

-- Create Councillor Table
CREATE TABLE Councillor (
    CouncillorID INT PRIMARY KEY,
    Name VARCHAR(255),
    Address VARCHAR(255),
    CardExpiry DATE,
    ContactInformation VARCHAR(255)
);
```

You can use this code to execute and create the Member table in your database.
doing so for every table

**Answer Part 1 C: SQL DDL Output (e.g., SPOOL file contents or output you got when you executed your above SQL Table Creation code, this should show the SQL code as well as its output). Make sure the output is a screenshot / image or picture and NOT a simple TEXT as it may increase your Turn-it-in similarity score.**



```
W22058786 >
W22058786 >
W22058786 > -- Create Member Table
W22058786 > CREATE TABLE Member (
         MemberID INT PRIMARY KEY,
         Name VARCHAR(255),
         Address VARCHAR(255),
         Email VARCHAR(255),
         Phone VARCHAR(20),
         Photo BLOB,
         CardExpiry DATE,
         CouncillorStatus CHAR(3) CHECK (CouncillorStatus IN ('Yes', 'No'))
     );

Table created.

W22058786 > -- Create Title Table
W22058786 > CREATE TABLE Title (
         TitleID INT PRIMARY KEY,
         TitleName VARCHAR2(255),
         Author VARCHAR2(255),
         SubjectArea VARCHAR2(255),
         BookDescription VARCHAR2(4000) -- Adjust the length as needed
     );

Table created.

W22058786 > -- Create Book Table
W22058786 > CREATE TABLE Book (
         BookID INT PRIMARY KEY,
         TitleID INT,
         AvailabilityStatus VARCHAR2(20) CHECK (AvailabilityStatus IN ('Available', 'Checked Out')),
         ISBN VARCHAR2(20),
         Condition VARCHAR2(255),
         BorrowDate DATE,
         DueDate DATE,
         FOREIGN KEY (TitleID) REFERENCES Title(TitleID)
     );

Table created.

W22058786 > -- Create Catalogue Table
W22058786 > CREATE TABLE Catalogue (
         CatalogueID INT PRIMARY KEY,
         TitleID INT,
         BookDescription VARCHAR2(4000), -- Adjust the length as needed
         FOREIGN KEY (TitleID) REFERENCES Title(TitleID)
     );

Table created.

W22058786 >
W22058786 > -- Create BookHistory Table
W22058786 > CREATE TABLE BookHistory (
         HistoryID INT PRIMARY KEY,
         MemberID INT,
         BookID INT,
         BorrowDate DATE,
         ReturnDate DATE,
         FOREIGN KEY (MemberID) REFERENCES Member(MemberID),
         FOREIGN KEY (BookID) REFERENCES Book(BookID)
     );

Table created.

W22058786 >
```



```
W22058786 > SELECT table_name
  2  FROM user_tables;

TABLE_NAME
--------------------------------------------------------------------------------
ALLOC
BOOK
BOOKHISTORY
CATALOGUE
COUNCILLOR
DEPT
EMP
EMP_PD
EPD
MEMBER
PROJ

TABLE_NAME
--------------------------------------------------------------------------------
PROJ_ED
STAFF
TITLE

14 rows selected.
```

## Part 2 (20 marks)

This part is based on your answer / solution to Part 1, i.e., design and implementation of the database for the BLS scenario.

*(A) Populate the database with some sample data (e.g., you should generate your own dummy data and load it into the BLS database, consider 5 to 10 rows for each table and enough data to see meaningful output for the queries below)*.

(10 marks)

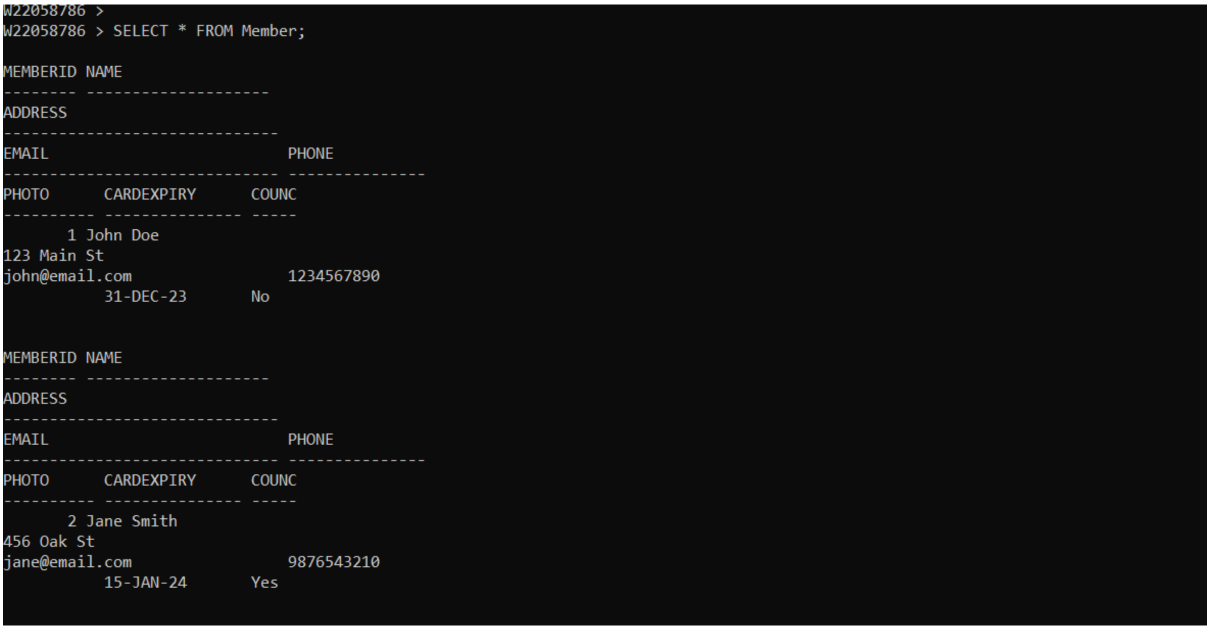
**Answer Part 2 A: Provide SQL code below for populating the above relational database.**

```
-- Insert sample data into Title Table
INSERT INTO Title VALUES (14, 'The Art of War', 'Sun Tzu',
'Philosophy', 'An ancient Chinese military treatise.');
```

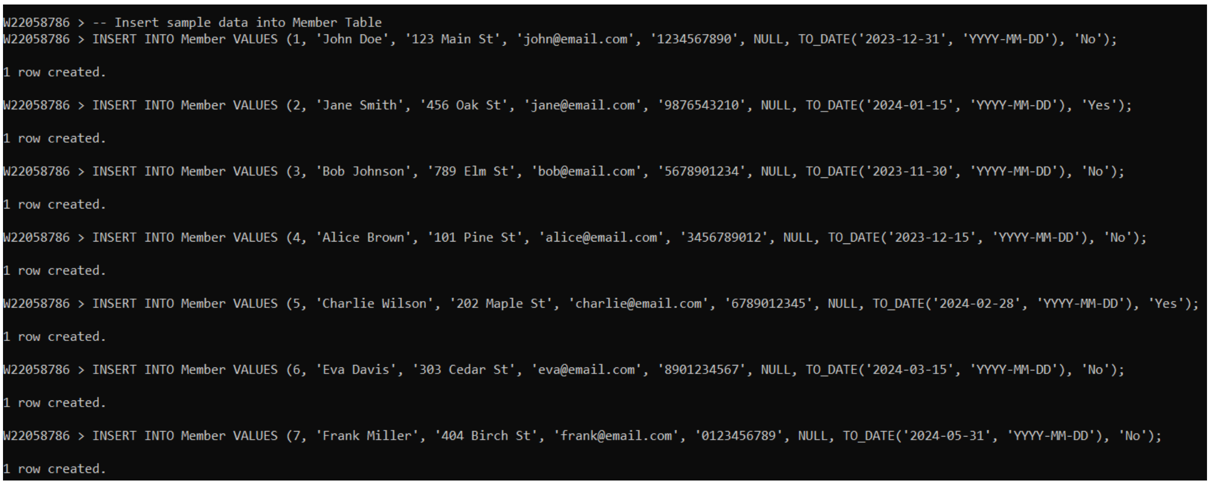
This code inserts a new record into the Title table with the specified values for TitleID, TitleName, Author, SubjectArea, and BookDescription. Make sure to execute this SQL statement after the table creation and any other existing data insertion statements in your database.

**Answer Part 2 A: Provide below output from running the above SQL code for populating your relational database (e.g., contents from Spool file or copy &**

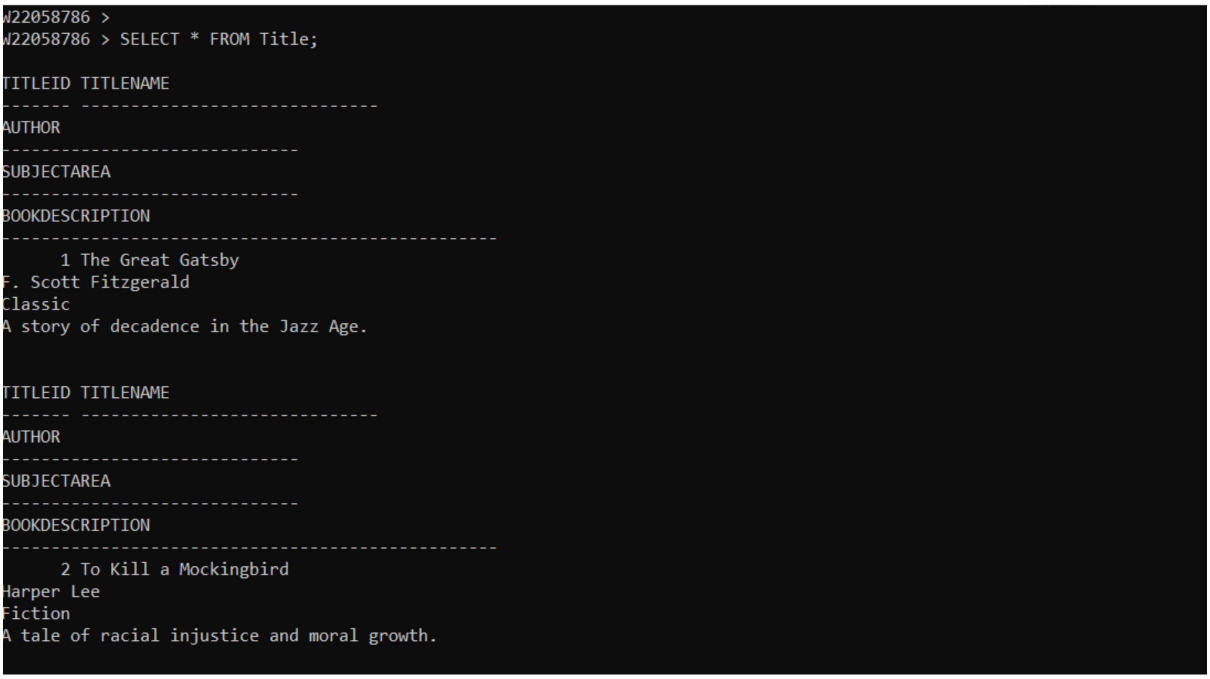
**paste of outputs from the SQL plus window). Make sure the output is a screenshot / image or picture and NOT a simple TEXT as it may increase your Turn-it-in similarity score.**



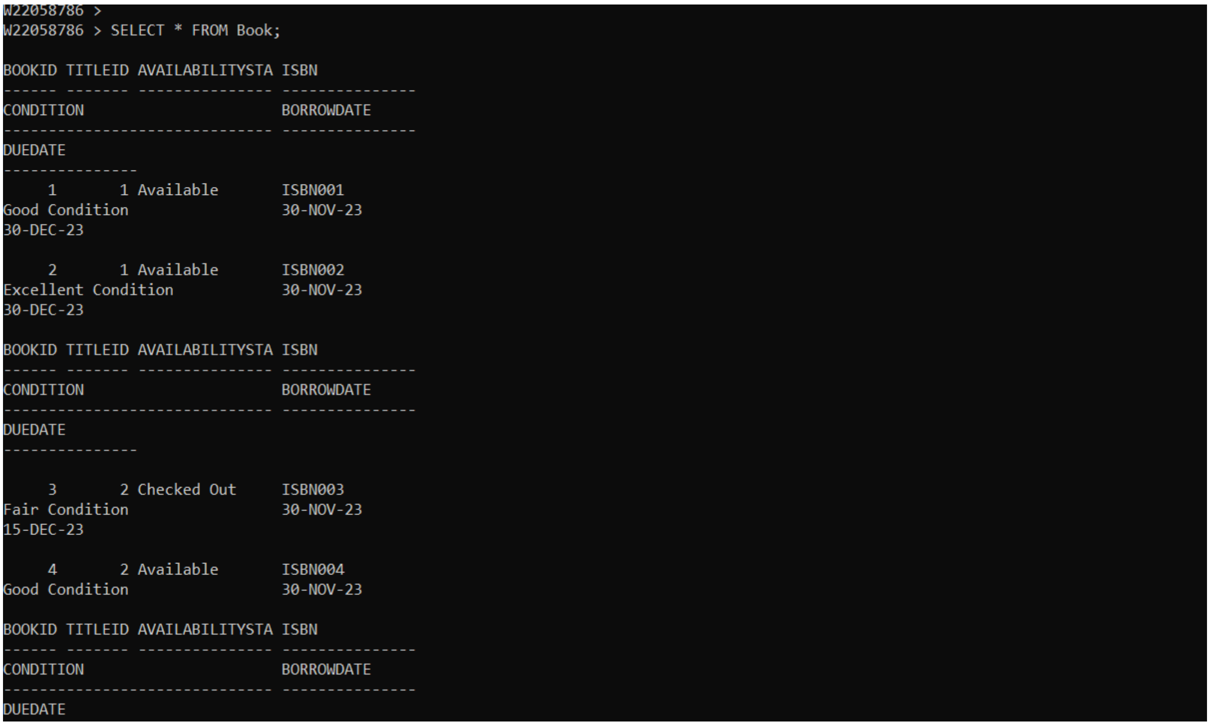
```
W22058786 >
W22058786 > SELECT * FROM Member;

MEMBERID NAME
-------- --------------------
ADDRESS
------------------------------
EMAIL                          PHONE
------------------------------ ---------------
PHOTO      CARDEXPIRY      COUNC
---------- --------------- -----
       1 John Doe
123 Main St
john@email.com                 1234567890
           31-DEC-23       No


MEMBERID NAME
-------- --------------------
ADDRESS
------------------------------
EMAIL                          PHONE
------------------------------ ---------------
PHOTO      CARDEXPIRY      COUNC
---------- --------------- -----
       2 Jane Smith
456 Oak St
jane@email.com                 9876543210
           15-JAN-24       Yes
```



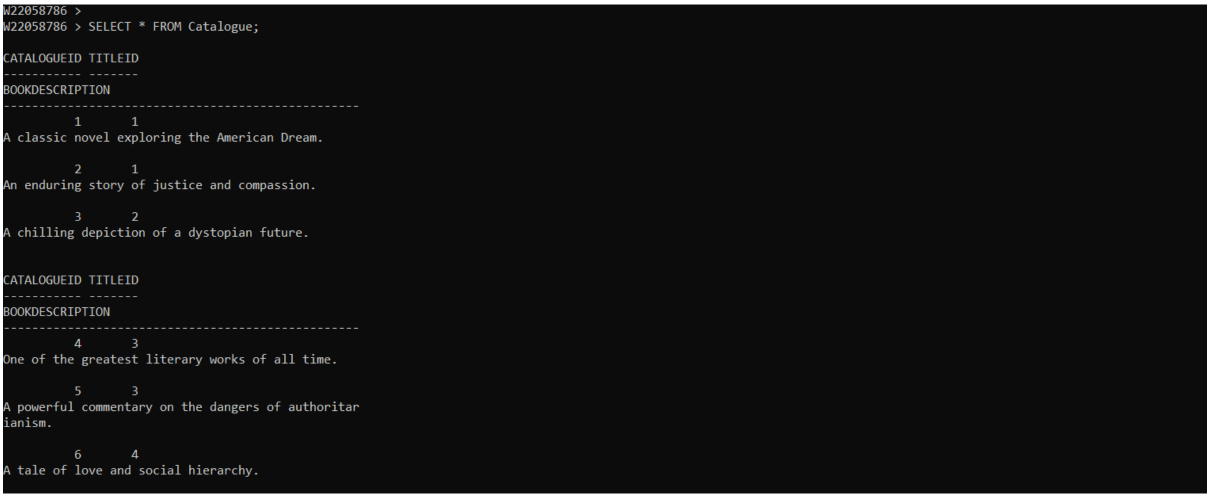
```
W22058786 > -- Insert sample data into Member Table
W22058786 > INSERT INTO Member VALUES (1, 'John Doe', '123 Main St', 'john@email.com', '1234567890', NULL, TO_DATE('2023-12-31', 'YYYY-MM-DD'), 'No');

1 row created.

W22058786 > INSERT INTO Member VALUES (2, 'Jane Smith', '456 Oak St', 'jane@email.com', '9876543210', NULL, TO_DATE('2024-01-15', 'YYYY-MM-DD'), 'Yes');

1 row created.

W22058786 > INSERT INTO Member VALUES (3, 'Bob Johnson', '789 Elm St', 'bob@email.com', '5678901234', NULL, TO_DATE('2023-11-30', 'YYYY-MM-DD'), 'No');

1 row created.

W22058786 > INSERT INTO Member VALUES (4, 'Alice Brown', '101 Pine St', 'alice@email.com', '3456789012', NULL, TO_DATE('2023-12-15', 'YYYY-MM-DD'), 'No');

1 row created.

W22058786 > INSERT INTO Member VALUES (5, 'Charlie Wilson', '202 Maple St', 'charlie@email.com', '6789012345', NULL, TO_DATE('2024-02-28', 'YYYY-MM-DD'), 'Yes');

1 row created.

W22058786 > INSERT INTO Member VALUES (6, 'Eva Davis', '303 Cedar St', 'eva@email.com', '8901234567', NULL, TO_DATE('2024-03-15', 'YYYY-MM-DD'), 'No');

1 row created.

W22058786 > INSERT INTO Member VALUES (7, 'Frank Miller', '404 Birch St', 'frank@email.com', '0123456789', NULL, TO_DATE('2024-05-31', 'YYYY-MM-DD'), 'No');

1 row created.
```



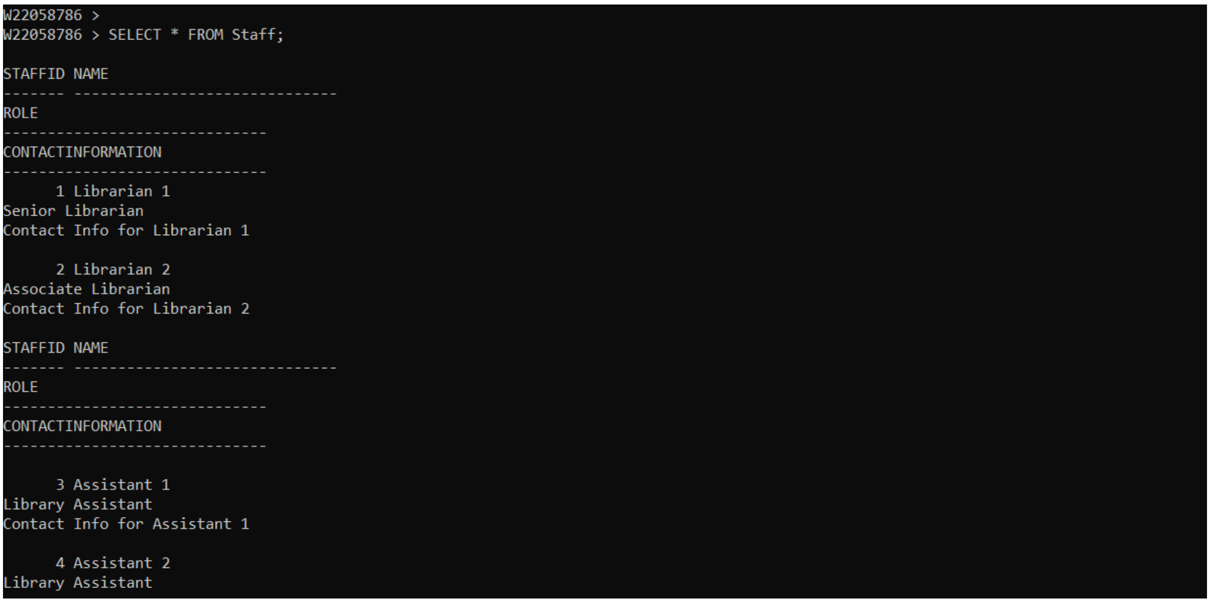
```
W22058786 >
W22058786 > SELECT * FROM Title;

TITLEID TITLENAME
------- ------------------------------
AUTHOR
------------------------------
SUBJECTAREA
------------------------------
BOOKDESCRIPTION
--------------------------------------------------
      1 The Great Gatsby
F. Scott Fitzgerald
Classic
A story of decadence in the Jazz Age.


TITLEID TITLENAME
------- ------------------------------
AUTHOR
------------------------------
SUBJECTAREA
------------------------------
BOOKDESCRIPTION
--------------------------------------------------
      2 To Kill a Mockingbird
Harper Lee
Fiction
A tale of racial injustice and moral growth.
```



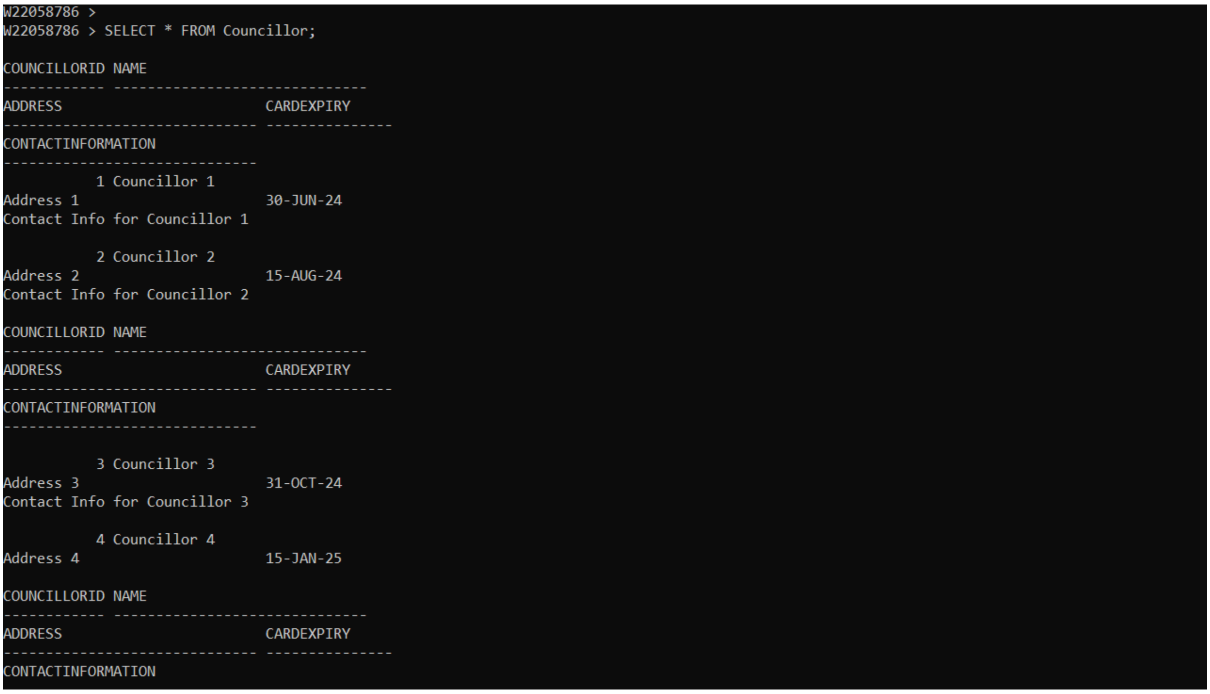
```
W22058786 >
W22058786 > SELECT * FROM Book;

BOOKID TITLEID AVAILABILITYSTA ISBN
------ ------- --------------- ---------------
CONDITION                      BORROWDATE
------------------------------ ---------------
DUEDATE
---------------
     1       1 Available       ISBN001
Good Condition                 30-NOV-23
30-DEC-23

     2       1 Available       ISBN002
Excellent Condition            30-NOV-23
30-DEC-23

BOOKID TITLEID AVAILABILITYSTA ISBN
------ ------- --------------- ---------------
CONDITION                      BORROWDATE
------------------------------ ---------------
DUEDATE
---------------

     3       2 Checked Out     ISBN003
Fair Condition                 30-NOV-23
15-DEC-23

     4       2 Available       ISBN004
Good Condition                 30-NOV-23

BOOKID TITLEID AVAILABILITYSTA ISBN
------ ------- --------------- ---------------
CONDITION                      BORROWDATE
------------------------------ ---------------
DUEDATE
```



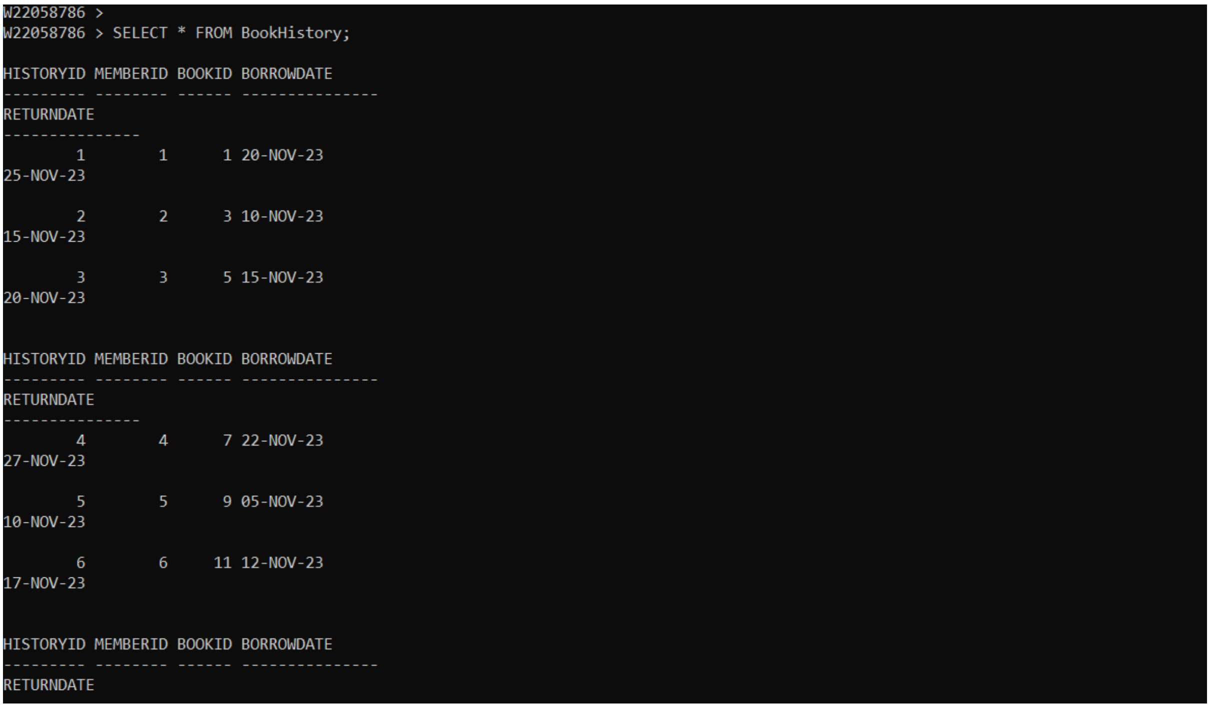
```
W22058786 >
W22058786 > SELECT * FROM Catalogue;

CATALOGUEID TITLEID
----------- -------
BOOKDESCRIPTION
--------------------------------------------------
          1       1
A classic novel exploring the American Dream.

          2       1
An enduring story of justice and compassion.

          3       2
A chilling depiction of a dystopian future.


CATALOGUEID TITLEID
----------- -------
BOOKDESCRIPTION
--------------------------------------------------
          4       3
One of the greatest literary works of all time.

          5       3
A powerful commentary on the dangers of authoritar
ianism.

          6       4
A tale of love and social hierarchy.
```



```
W22058786 >
W22058786 > SELECT * FROM Staff;

STAFFID NAME
------- ------------------------------
ROLE
------------------------------
CONTACTINFORMATION
------------------------------
      1 Librarian 1
Senior Librarian
Contact Info for Librarian 1

      2 Librarian 2
Associate Librarian
Contact Info for Librarian 2

STAFFID NAME
------- ------------------------------
ROLE
------------------------------
CONTACTINFORMATION
------------------------------

      3 Assistant 1
Library Assistant
Contact Info for Assistant 1

      4 Assistant 2
Library Assistant
```



```
W22058786 >
W22058786 > SELECT * FROM Councillor;

COUNCILLORID NAME
----------- ----------------------------
ADDRESS                       CARDEXPIRY
----------------------------- --------------
CONTACTINFORMATION
-----------------------------
          1 Councillor 1
Address 1                     30-JUN-24
Contact Info for Councillor 1

          2 Councillor 2
Address 2                     15-AUG-24
Contact Info for Councillor 2

COUNCILLORID NAME
----------- ----------------------------
ADDRESS                       CARDEXPIRY
----------------------------- --------------
CONTACTINFORMATION
-----------------------------

          3 Councillor 3
Address 3                     31-OCT-24
Contact Info for Councillor 3

          4 Councillor 4
Address 4                     15-JAN-25

COUNCILLORID NAME
----------- ----------------------------
ADDRESS                       CARDEXPIRY
----------------------------- --------------
CONTACTINFORMATION
```



```
W22058786 >
W22058786 > SELECT * FROM BookHistory;

HISTORYID MEMBERID BOOKID BORROWDATE
--------- -------- ------ ---------------
RETURNDATE
---------------
        1        1      1 20-NOV-23
25-NOV-23

        2        2      3 10-NOV-23
15-NOV-23

        3        3      5 15-NOV-23
20-NOV-23


HISTORYID MEMBERID BOOKID BORROWDATE
--------- -------- ------ ---------------
RETURNDATE
---------------
        4        4      7 22-NOV-23
27-NOV-23

        5        5      9 05-NOV-23
10-NOV-23

        6        6     11 12-NOV-23
17-NOV-23


HISTORYID MEMBERID BOOKID BORROWDATE
--------- -------- ------ ---------------
RETURNDATE
```

*(B) Answer the following queries (retrievals) using Relational Algebra and SQL.*

(10 marks)

q1) Display details of the top 12 most frequent users (members) of the library with the number of books/titles/items they have borrowed in the past 12 months.

**Provide Relational Algebra expression below:**

```
Query 1
```

```
π_{MemberID, Name, NumBorrowed}
    (σ_{BorrowDate >= (CURRENT_DATE - 365) ^ BorrowDate <=
CURRENT_DATE}
        (BookHistory ⋈_{MemberID}
            (σ_{BorrowDate >= (CURRENT_DATE - 365) ^ BorrowDate <=
CURRENT_DATE}
                (Book))))
```

In these expressions:

π denotes the projection operation.
σ denotes the selection operation.
⋈ denotes the natural join operation.
∧ denotes the logical AND operation.
≥ denotes "greater than or equal to."
≤ denotes "less than or equal to."

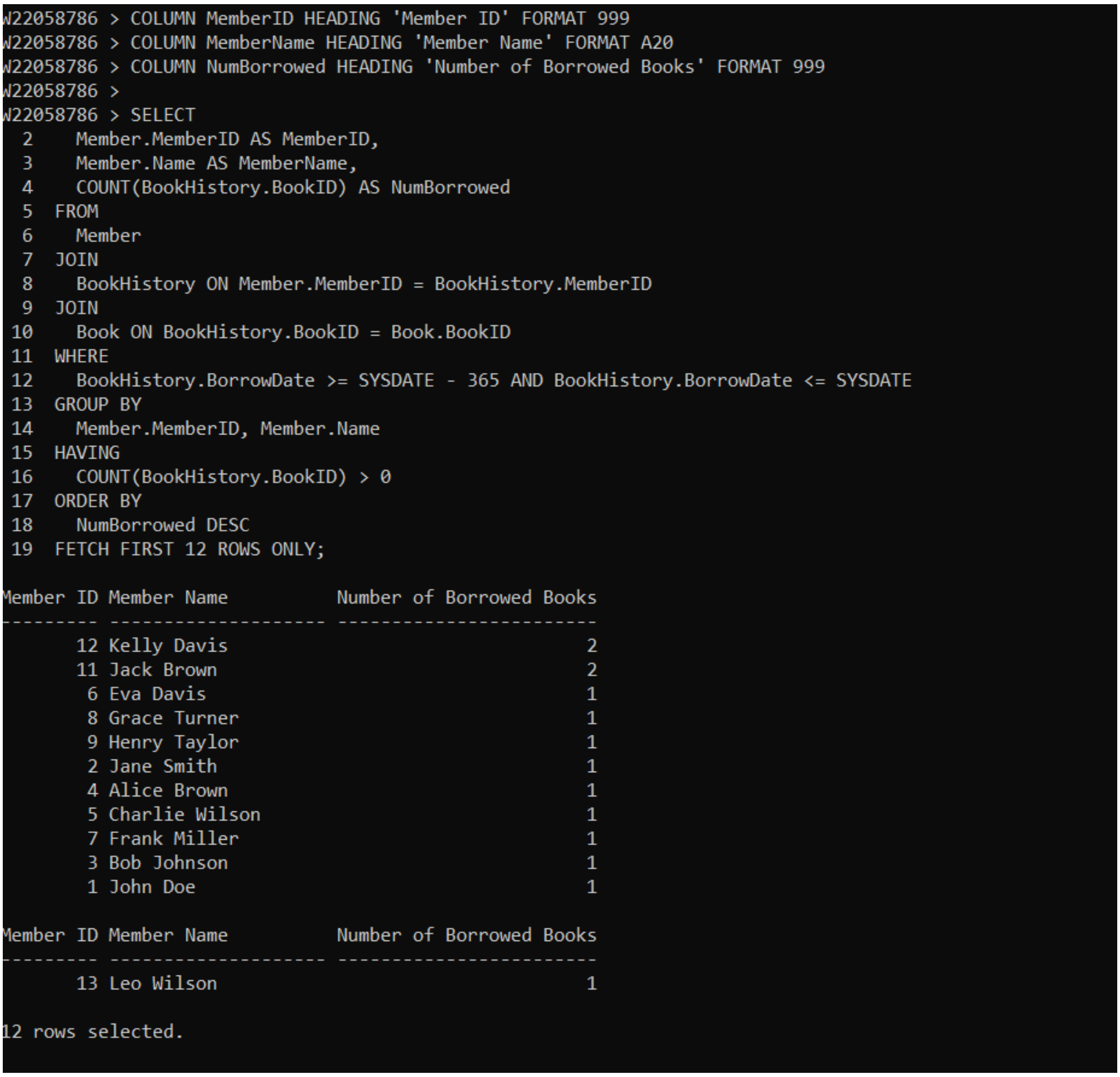
The expressions aim to capture the operations needed to retrieve the desired information from the relational database.

Here's a breakdown of the process:
Relational Algebra Operations:
Mapping the query requirements to relational algebra operations.
Using standard relational algebra symbols (π for projection, σ for selection, ⋈ for natural join, etc.) to represent these operations.
Logical Structure:
Considering the logical structure of the tables and their relationships to formulate the expressions.
Conditions:
Incorporating conditions specified in the queries (e.g., filtering by date ranges) using selection (σ) operations.
Projection:
Using projection (π) to specify the attributes that need to be included in the final result.

**Provide SQL query code as TEXT and output below (as image / screenshot / picture):**

```
  SELECT
  Member.MemberID AS MemberID,
  Member.Name AS MemberName,
  COUNT(BookHistory.BookID) AS NumBorrowed
FROM
  Member
JOIN
  BookHistory ON Member.MemberID = BookHistory.MemberID
JOIN
  Book ON BookHistory.BookID = Book.BookID
WHERE
  BookHistory.BorrowDate >= SYSDATE - 365 AND BookHistory.BorrowDate
<= SYSDATE
```

```
GROUP BY
  Member.MemberID, Member.Name
HAVING
  COUNT(BookHistory.BookID) > 0
ORDER BY
  NumBorrowed DESC
FETCH FIRST 12 ROWS ONLY;
```



```
W22058786 > COLUMN MemberID HEADING 'Member ID' FORMAT 999
W22058786 > COLUMN MemberName HEADING 'Member Name' FORMAT A20
W22058786 > COLUMN NumBorrowed HEADING 'Number of Borrowed Books' FORMAT 999
W22058786 >
W22058786 > SELECT
    Member.MemberID AS MemberID,
    Member.Name AS MemberName,
    COUNT(BookHistory.BookID) AS NumBorrowed
  FROM
    Member
  JOIN
    BookHistory ON Member.MemberID = BookHistory.MemberID
  JOIN
    Book ON BookHistory.BookID = Book.BookID
  WHERE
    BookHistory.BorrowDate >= SYSDATE - 365 AND BookHistory.BorrowDate <= SYSDATE
  GROUP BY
    Member.MemberID, Member.Name
  HAVING
    COUNT(BookHistory.BookID) > 0
  ORDER BY
    NumBorrowed DESC
  FETCH FIRST 12 ROWS ONLY;

Member ID Member Name          Number of Borrowed Books
--------- -------------------- ------------------------
       12 Kelly Davis                                 2
       11 Jack Brown                                  2
        6 Eva Davis                                   1
        8 Grace Turner                                1
        9 Henry Taylor                                1
        2 Jane Smith                                  1
        4 Alice Brown                                 1
        5 Charlie Wilson                              1
        7 Frank Miller                                1
        3 Bob Johnson                                 1
        1 John Doe                                    1

Member ID Member Name          Number of Borrowed Books
--------- -------------------- ------------------------
       13 Leo Wilson                                  1

12 rows selected.
```

q2) Display details of the top 15 most frequently borrowed titles/books/items of the BLS with the total number of borrowings in the past two financial years.

## Provide Relational Algebra expression below:

Query 2

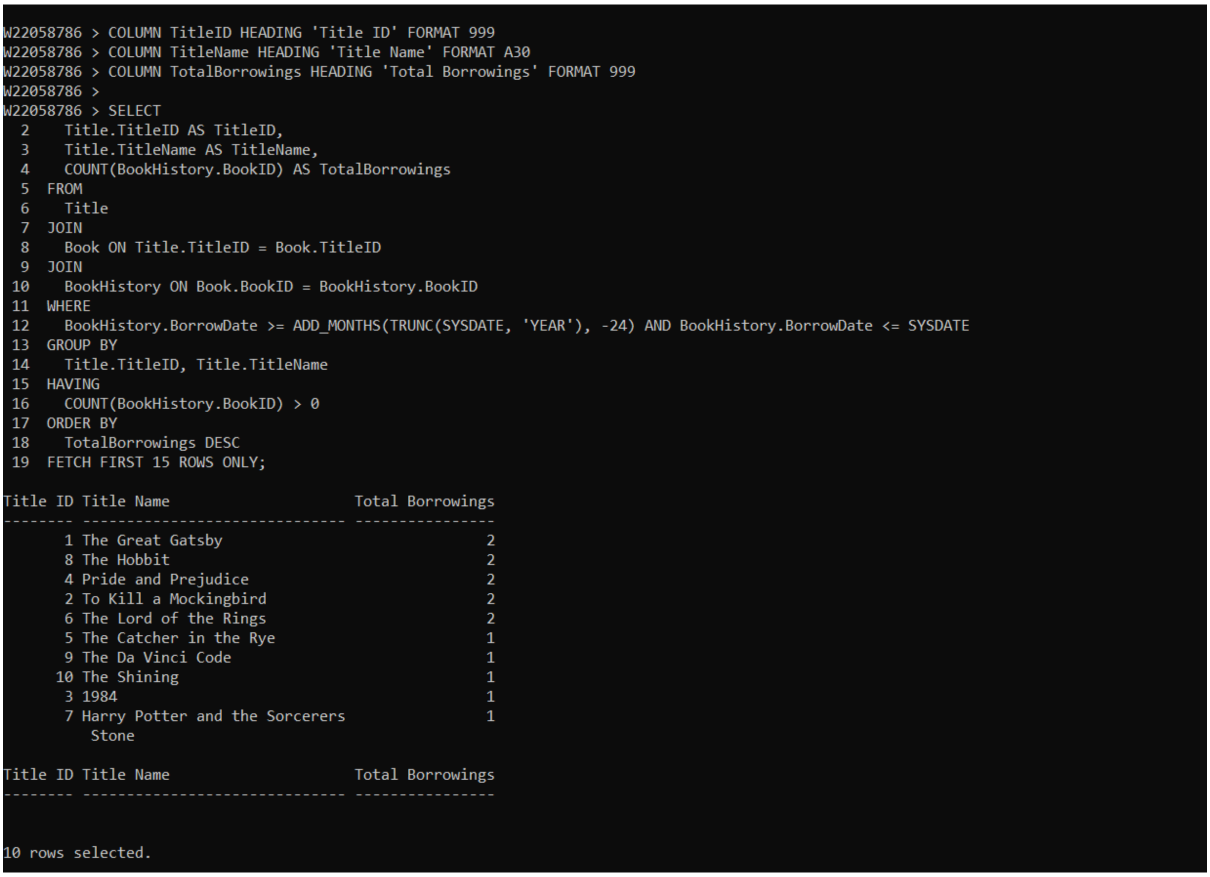
```
π_{TitleID, TitleName, TotalBorrowings}
  (σ_{BorrowDate >= (CURRENT_DATE - 730) ^ BorrowDate <= CURRENT_DATE}
    (BookHistory ⋈_{TitleID}
      (σ_{BorrowDate >= (CURRENT_DATE - 730) ^ BorrowDate <= CURRENT_DATE}
        (Book ⋈_{TitleID} Title))))
```

In these expressions:

π denotes the projection operation.
σ denotes the selection operation.
⋈ denotes the natural join operation.
∧ denotes the logical AND operation.
≥ denotes "greater than or equal to."
≤ denotes "less than or equal to."

## Provide SQL query code as TEXT and output below (as image / screenshot / picture):

```
SELECT
  Title.TitleID AS TitleID,
  Title.TitleName AS TitleName,
  COUNT(BookHistory.BookID) AS TotalBorrowings
FROM
  Title
JOIN
  Book ON Title.TitleID = Book.TitleID
JOIN
  BookHistory ON Book.BookID = BookHistory.BookID
WHERE
  BookHistory.BorrowDate >= ADD_MONTHS(TRUNC(SYSDATE, 'YEAR'), -24)
AND BookHistory.BorrowDate <= SYSDATE
GROUP BY
  Title.TitleID, Title.TitleName
HAVING
  COUNT(BookHistory.BookID) > 0
ORDER BY
  TotalBorrowings DESC
FETCH FIRST 15 ROWS ONLY;
```



```
W22058786 > COLUMN TitleID HEADING 'Title ID' FORMAT 999
W22058786 > COLUMN TitleName HEADING 'Title Name' FORMAT A30
W22058786 > COLUMN TotalBorrowings HEADING 'Total Borrowings' FORMAT 999
W22058786 >
W22058786 > SELECT
    Title.TitleID AS TitleID,
    Title.TitleName AS TitleName,
    COUNT(BookHistory.BookID) AS TotalBorrowings
  FROM
    Title
  JOIN
    Book ON Title.TitleID = Book.TitleID
  JOIN
    BookHistory ON Book.BookID = BookHistory.BookID
  WHERE
    BookHistory.BorrowDate >= ADD_MONTHS(TRUNC(SYSDATE, 'YEAR'), -24) AND BookHistory.BorrowDate <= SYSDATE
  GROUP BY
    Title.TitleID, Title.TitleName
  HAVING
    COUNT(BookHistory.BookID) > 0
  ORDER BY
    TotalBorrowings DESC
  FETCH FIRST 15 ROWS ONLY;

Title ID Title Name                     Total Borrowings
-------- ------------------------------ ----------------
       1 The Great Gatsby                              2
       8 The Hobbit                                    2
       4 Pride and Prejudice                           2
       2 To Kill a Mockingbird                         2
       6 The Lord of the Rings                         2
       5 The Catcher in the Rye                        1
       9 The Da Vinci Code                             1
      10 The Shining                                   1
       3 1984                                          1
       7 Harry Potter and the Sorcerers                1
          Stone

Title ID Title Name                     Total Borrowings
-------- ------------------------------ ----------------


10 rows selected.
```

## Part 3 (35 marks)

This part is based on your answer / solution to Part 1 (A), i.e., conceptual design of the database for the BLS scenario.

*(A) Choose and justify what aspects of BLS conceptual design would be better off if implemented using object-relational database; then provide logical design and implementation of the subset of the BLS using ER/EER to object-relational mapping and object-relational features of Oracle Database System (Kannan); populate the object-tables with sample data and demonstrate your choice of design and implementation by running two complex queries on your object-tables.*

(20 marks)

## Answer Part 3 A

## 1) Provide below your choice and justification of what aspects (subset) of the BLS conceptual design from Part 1.A you would like to implement using object relational databases (2 marks)

The Borrowing System is a crucial component of the BLS, involving complex relationships and diverse data types. Implementing it with an object-relational database offers several advantages. Firstly, object-relational databases excel in handling intricate relationships, ensuring efficient retrieval of information. Secondly, they accommodate rich data types, such as dates and statuses, allowing for more accurate representation and efficient querying. Additionally, the inheritance capabilities of the object-relational model can be leveraged for representing Councillors as a specific type of Member, avoiding redundancy in the database schema. Moreover, object-relational databases provide a flexible schema that can adapt to changes in the Borrowing System. Lastly, they support encapsulation, which can be beneficial for maintaining data integrity and security within the system

## 2) Provide below the logical design for your chosen subset using ER/EER to object-relational mapping (2 marks)

Logical Design for Chosen Subset

Entities:

Member:
Attributes: MemberID (Primary Key), Name, Address, Email, Phone, Photo, CardExpiry, CouncillorStatus
Title:
Attributes: TitleID (Primary Key), TitleName, Author, SubjectArea, BookDescription
Book:
Attributes: BookID (Primary Key), TitleID (Foreign Key), AvailabilityStatus, ISBN, Condition, BorrowDate, DueDate
Catalogue:
Attributes: CatalogueID (Primary Key), TitleID (Foreign Key), BookDescription
Staff:
Attributes: StaffID (Primary Key), Name, Role, ContactInformation
Councillor:
Attributes: CouncillorID (Primary Key), Name, Address, CardExpiry, ContactInformation
BookHistory:
Attributes: HistoryID (Primary Key), MemberID (Foreign Key), BookID (Foreign Key), BorrowDate, ReturnDate
Relationships:

Member-BookHistory (1:M):
A Member can have multiple BookHistories (Borrowing history).
Title-Book (1:M):
A Title can have multiple Books (copies).
Title-Catalogue (1:1):
A Title is uniquely represented in the Catalogue.
Staff-BookHistory (1:M):
A Staff member can be associated with multiple BookHistories.
Councillor-Member (1:1):
A Councillor is a specific type of Member.
Book-BookHistory (1:M):
A Book can be borrowed multiple times, creating multiple BookHistories.
Attributes and Constraints:

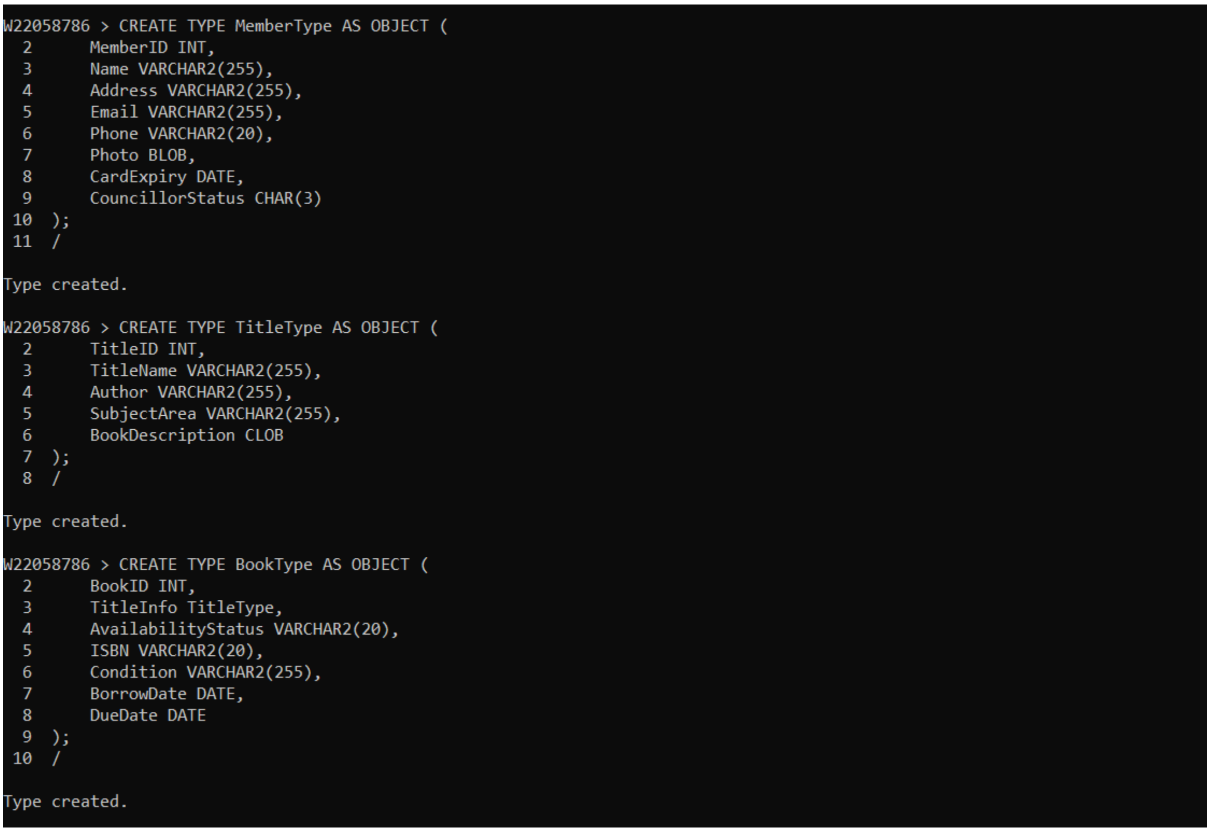
Appropriate data types assigned to each attribute.
Foreign key constraints linking related entities.
Check constraints for attributes like AvailabilityStatus.
Object-Relational Mapping:

Each entity maps to a corresponding table in the relational database.
Attributes become columns, and relationships are represented through foreign key constraints.
Subtypes (e.g., Councillor as a subtype of Member) can be implemented using an inheritance approach, where common attributes are in the base table (Member) and specific attributes are in the subtype table (Councillor).

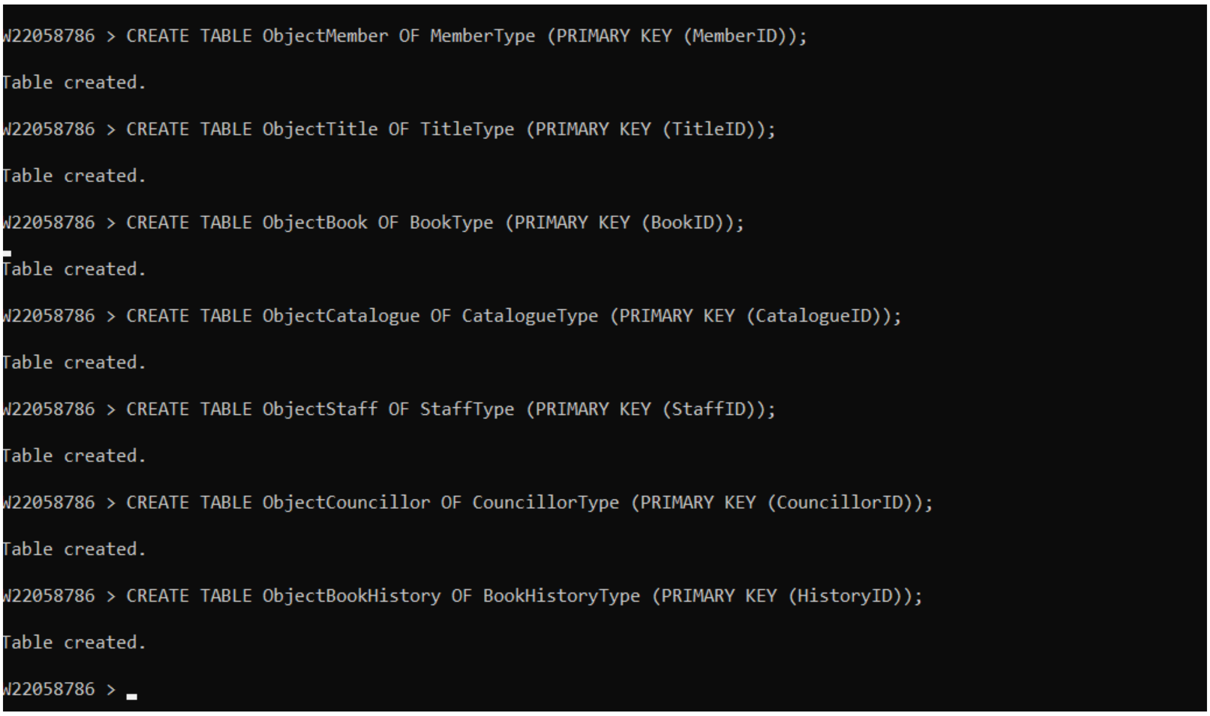
### 3) Provide below the SQL query code as TEXT and output below (as image / screenshot / picture) for implementing your above logical object-relational design (8 marks)

```
CREATE TYPE MemberType AS OBJECT (
    MemberID INT,
    Name VARCHAR2(255),
    Address VARCHAR2(255),
    Email VARCHAR2(255),
    Phone VARCHAR2(20),
    Photo BLOB,
    CardExpiry DATE,
    CouncillorStatus CHAR(3)
);
/

CREATE TYPE TitleType AS OBJECT (
    TitleID INT,
    TitleName VARCHAR2(255),
    Author VARCHAR2(255),
    SubjectArea VARCHAR2(255),
    BookDescription CLOB
);
/

CREATE TYPE BookType AS OBJECT (
    BookID INT,
    TitleInfo TitleType,
    AvailabilityStatus VARCHAR2(20),
    ISBN VARCHAR2(20),
    Condition VARCHAR2(255),
    BorrowDate DATE,
    DueDate DATE
);
/

CREATE TYPE CatalogueType AS OBJECT (
    CatalogueID INT,
    TitleInfo TitleType,
    BookDescription CLOB
);
/

CREATE TYPE StaffType AS OBJECT (
    StaffID INT,
    Name VARCHAR2(255),
    Role VARCHAR2(255),
    ContactInformation VARCHAR2(255)
);
/

CREATE TYPE CouncillorType AS OBJECT (
    CouncillorID INT,
    Name VARCHAR2(255),
    Address VARCHAR2(255),
    CardExpiry DATE,
    ContactInformation VARCHAR2(255)
);
/

CREATE TYPE BookHistoryType AS OBJECT (
    HistoryID INT,
    MemberID INT,
    BookInfo BookType,
    BorrowDate DATE,
    ReturnDate DATE
);
```

```
/

CREATE TABLE ObjectMember OF MemberType (PRIMARY KEY (MemberID));
CREATE TABLE ObjectTitle OF TitleType (PRIMARY KEY (TitleID));
CREATE TABLE ObjectBook OF BookType (PRIMARY KEY (BookID));
CREATE TABLE ObjectCatalogue OF CatalogueType (PRIMARY KEY (CatalogueID));
CREATE TABLE ObjectStaff OF StaffType (PRIMARY KEY (StaffID));
CREATE TABLE ObjectCouncillor OF CouncillorType (PRIMARY KEY (CouncillorID));
CREATE TABLE ObjectBookHistory OF BookHistoryType (PRIMARY KEY (HistoryID));
```



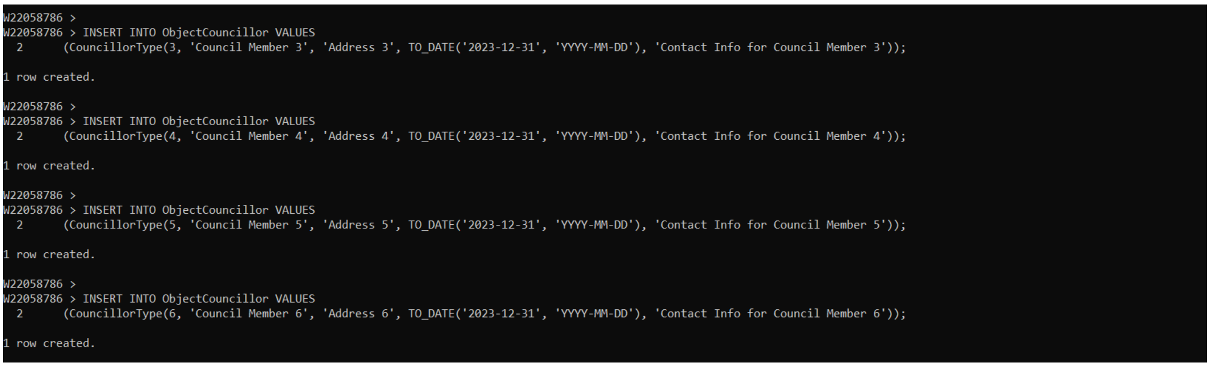
```
W22058786 > CREATE TYPE MemberType AS OBJECT (
      MemberID INT,
      Name VARCHAR2(255),
      Address VARCHAR2(255),
      Email VARCHAR2(255),
      Phone VARCHAR2(20),
      Photo BLOB,
      CardExpiry DATE,
      CouncillorStatus CHAR(3)
  );
  /

Type created.

W22058786 > CREATE TYPE TitleType AS OBJECT (
      TitleID INT,
      TitleName VARCHAR2(255),
      Author VARCHAR2(255),
      SubjectArea VARCHAR2(255),
      BookDescription CLOB
  );
  /

Type created.

W22058786 > CREATE TYPE BookType AS OBJECT (
      BookID INT,
      TitleInfo TitleType,
      AvailabilityStatus VARCHAR2(20),
      ISBN VARCHAR2(20),
      Condition VARCHAR2(255),
      BorrowDate DATE,
      DueDate DATE
  );
  /

Type created.
```



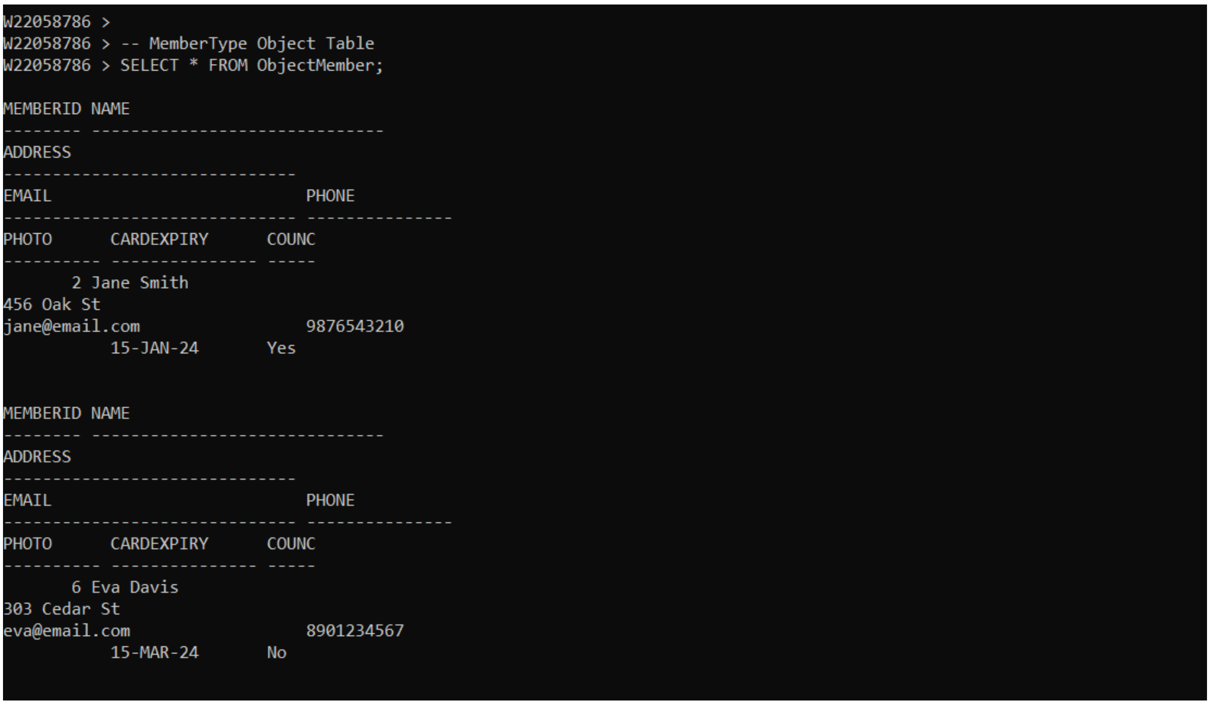
```
W22058786 > CREATE TABLE ObjectMember OF MemberType (PRIMARY KEY (MemberID));

Table created.

W22058786 > CREATE TABLE ObjectTitle OF TitleType (PRIMARY KEY (TitleID));

Table created.

W22058786 > CREATE TABLE ObjectBook OF BookType (PRIMARY KEY (BookID));

Table created.

W22058786 > CREATE TABLE ObjectCatalogue OF CatalogueType (PRIMARY KEY (CatalogueID));

Table created.

W22058786 > CREATE TABLE ObjectStaff OF StaffType (PRIMARY KEY (StaffID));

Table created.

W22058786 > CREATE TABLE ObjectCouncillor OF CouncillorType (PRIMARY KEY (CouncillorID));

Table created.

W22058786 > CREATE TABLE ObjectBookHistory OF BookHistoryType (PRIMARY KEY (HistoryID));

Table created.

W22058786 >
```

## 4) Provide below the SQL query code as TEXT and the output below (as image / screenshot / picture) for populating your above object-relational subset of the BLS database (4 marks)

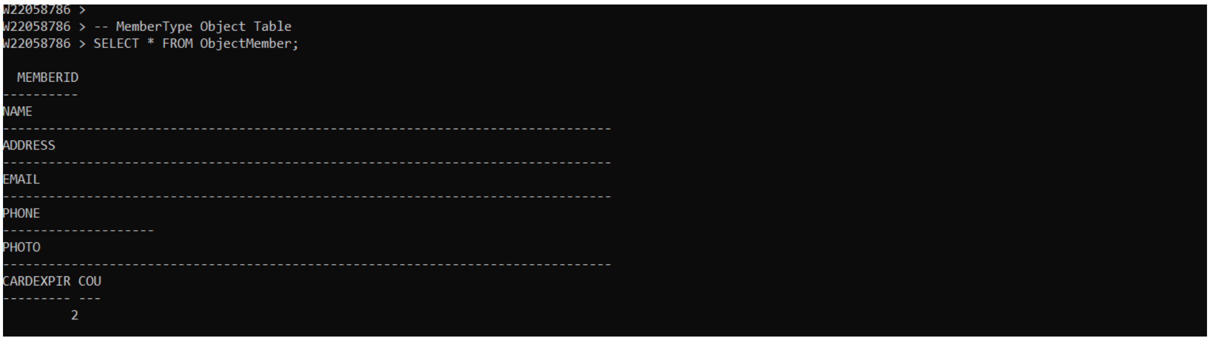
```
-- Inserting sample data into ObjectMember Table
INSERT INTO ObjectMember VALUES (MemberType(1, 'John Doe', '123 Main St', 'john@email.com', '1234567890', NULL, TO_DATE('2023-12-31', 'YYYY-MM-DD'), 'No'));
```



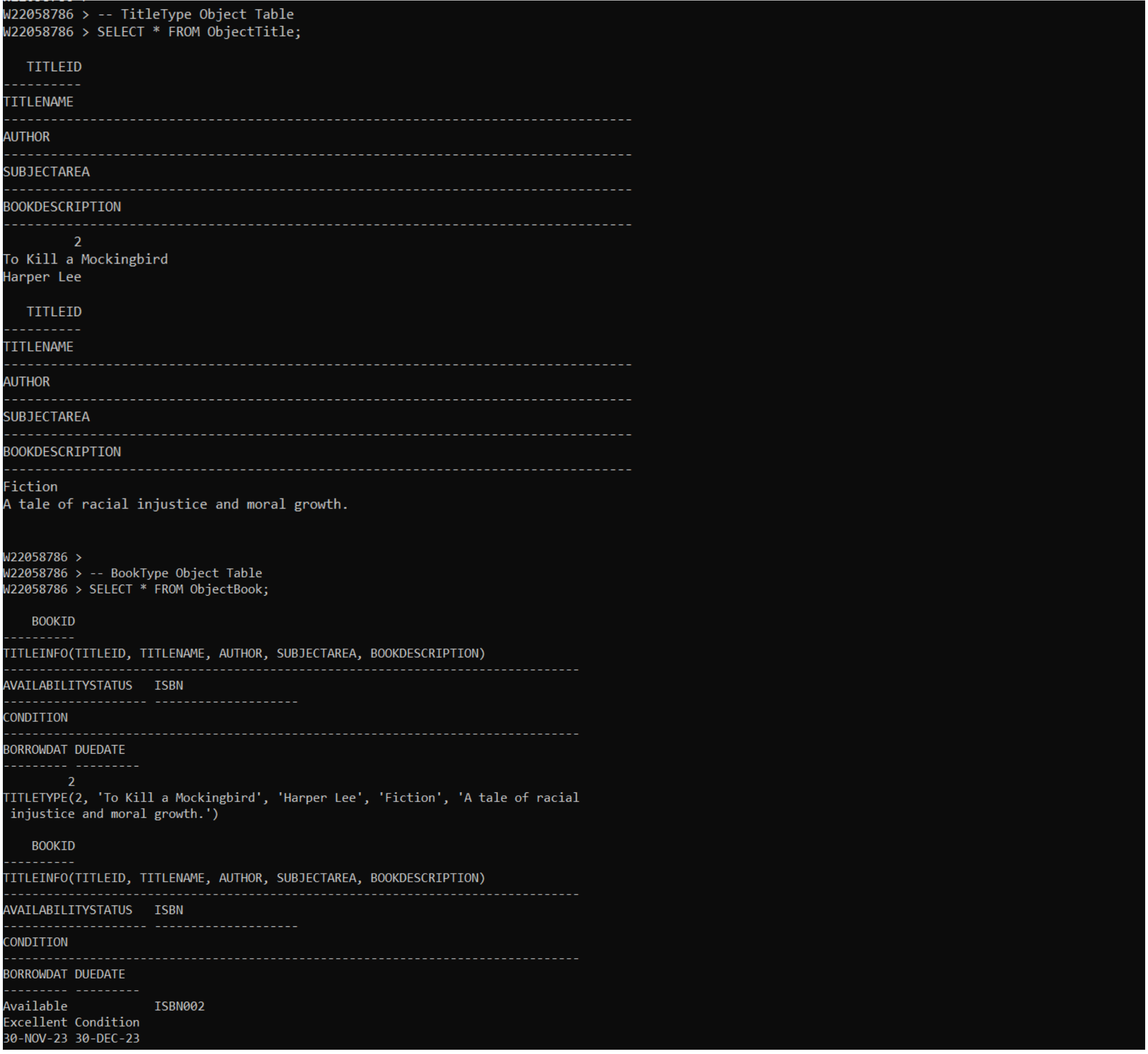
```
W22058786 >
W22058786 > INSERT INTO ObjectCouncillor VALUES
  2      (CouncillorType(3, 'Council Member 3', 'Address 3', TO_DATE('2023-12-31', 'YYYY-MM-DD'), 'Contact Info for Council Member 3'));

1 row created.

W22058786 >
W22058786 > INSERT INTO ObjectCouncillor VALUES
  2      (CouncillorType(4, 'Council Member 4', 'Address 4', TO_DATE('2023-12-31', 'YYYY-MM-DD'), 'Contact Info for Council Member 4'));

1 row created.

W22058786 >
W22058786 > INSERT INTO ObjectCouncillor VALUES
  2      (CouncillorType(5, 'Council Member 5', 'Address 5', TO_DATE('2023-12-31', 'YYYY-MM-DD'), 'Contact Info for Council Member 5'));

1 row created.

W22058786 >
W22058786 > INSERT INTO ObjectCouncillor VALUES
  2      (CouncillorType(6, 'Council Member 6', 'Address 6', TO_DATE('2023-12-31', 'YYYY-MM-DD'), 'Contact Info for Council Member 6'));

1 row created.
```



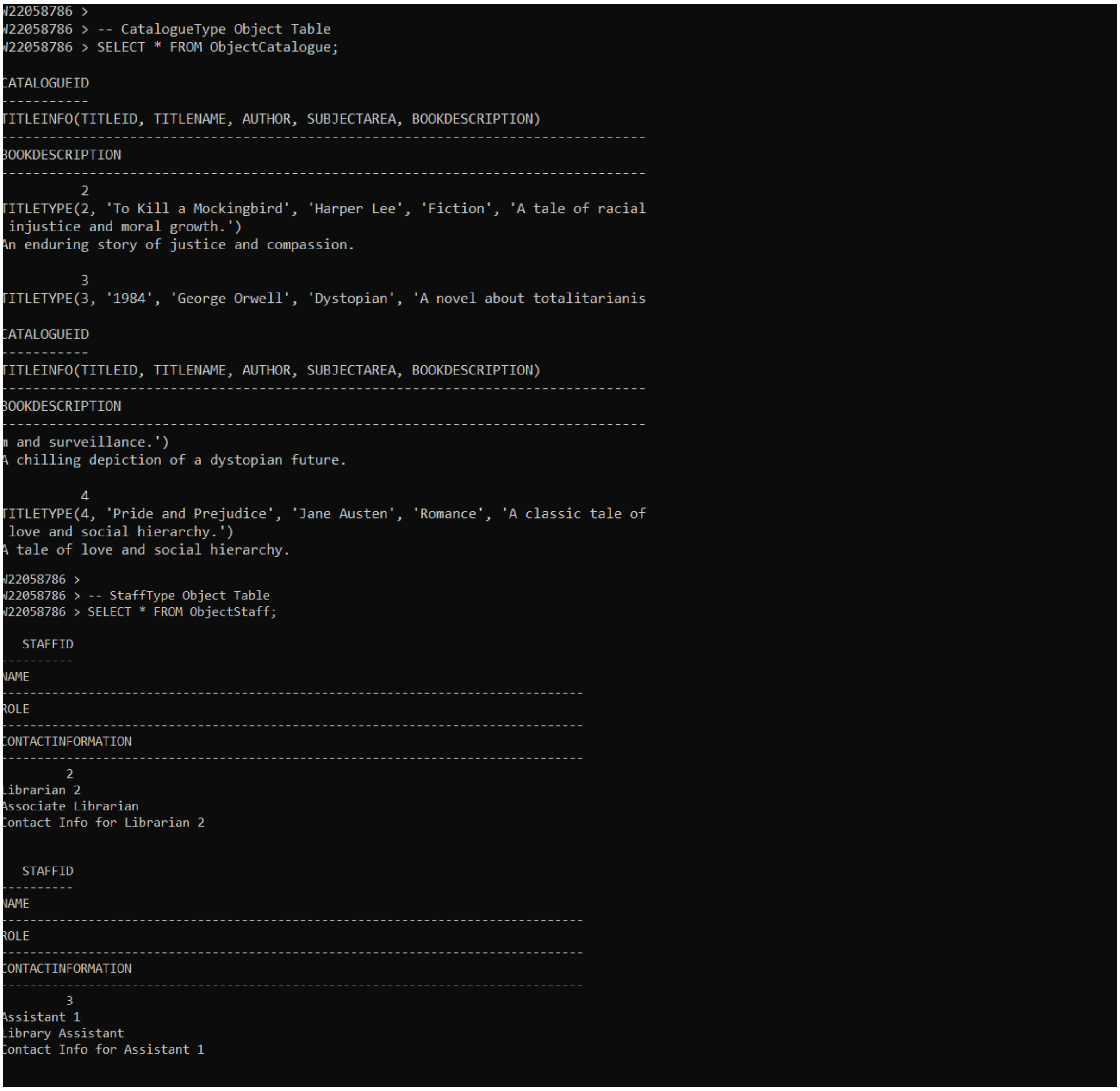
```
W22058786 >
W22058786 > -- MemberType Object Table
W22058786 > SELECT * FROM ObjectMember;

MEMBERID NAME
-------- ------------------------------
ADDRESS
------------------------------
EMAIL                          PHONE
------------------------------ ---------------
PHOTO      CARDEXPIRY      COUNC
---------- --------------- -----
       2 Jane Smith
456 Oak St
jane@email.com                 9876543210
           15-JAN-24       Yes


MEMBERID NAME
-------- ------------------------------
ADDRESS
------------------------------
EMAIL                          PHONE
------------------------------ ---------------
PHOTO      CARDEXPIRY      COUNC
---------- --------------- -----
       6 Eva Davis
303 Cedar St
eva@email.com                  8901234567
           15-MAR-24       No
```



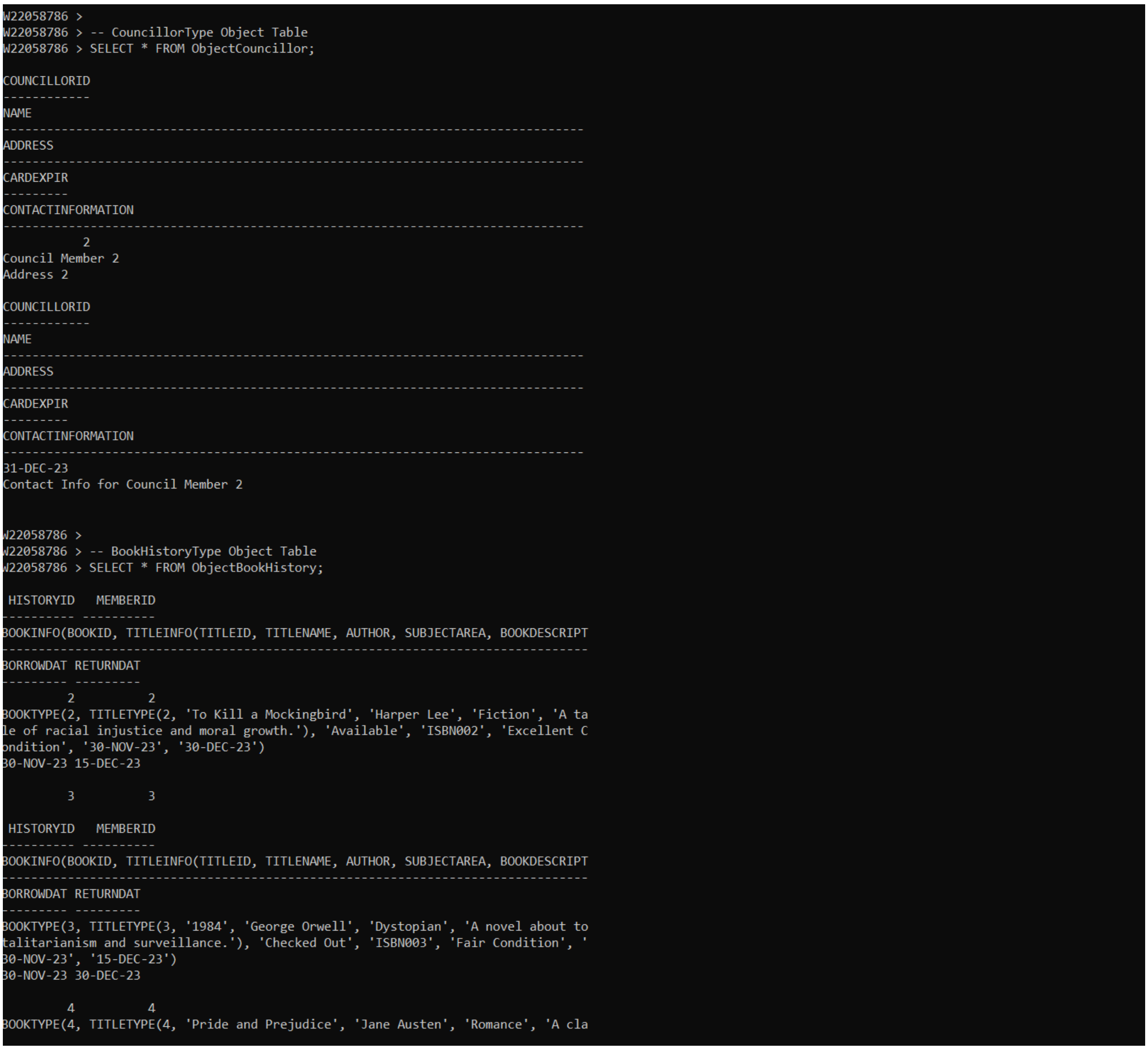
```
W22058786 >
W22058786 > -- MemberType Object Table
W22058786 > SELECT * FROM ObjectMember;

  MEMBERID
----------
NAME
--------------------------------------------------------------------------------
ADDRESS
--------------------------------------------------------------------------------
EMAIL
--------------------------------------------------------------------------------
PHONE
--------------------
PHOTO
--------------------------------------------------------------------------------
CARDEXPIR COU
--------- ---
         2
```



```
W22058786 > -- TitleType Object Table
W22058786 > SELECT * FROM ObjectTitle;

   TITLEID
----------
TITLENAME
--------------------------------------------------------------------------------
AUTHOR
--------------------------------------------------------------------------------
SUBJECTAREA
--------------------------------------------------------------------------------
BOOKDESCRIPTION
--------------------------------------------------------------------------------
         2
To Kill a Mockingbird
Harper Lee

   TITLEID
----------
TITLENAME
--------------------------------------------------------------------------------
AUTHOR
--------------------------------------------------------------------------------
SUBJECTAREA
--------------------------------------------------------------------------------
BOOKDESCRIPTION
--------------------------------------------------------------------------------
Fiction
A tale of racial injustice and moral growth.


W22058786 >
W22058786 > -- BookType Object Table
W22058786 > SELECT * FROM ObjectBook;

    BOOKID
----------
TITLEINFO(TITLEID, TITLENAME, AUTHOR, SUBJECTAREA, BOOKDESCRIPTION)
--------------------------------------------------------------------------------
AVAILABILITYSTATUS   ISBN
-------------------- --------------------
CONDITION
--------------------------------------------------------------------------------
BORROWDAT DUEDATE
--------- ---------
         2
TITLETYPE(2, 'To Kill a Mockingbird', 'Harper Lee', 'Fiction', 'A tale of racial
 injustice and moral growth.')

    BOOKID
----------
TITLEINFO(TITLEID, TITLENAME, AUTHOR, SUBJECTAREA, BOOKDESCRIPTION)
--------------------------------------------------------------------------------
AVAILABILITYSTATUS   ISBN
-------------------- --------------------
CONDITION
--------------------------------------------------------------------------------
BORROWDAT DUEDATE
--------- ---------
Available            ISBN002
Excellent Condition
30-NOV-23 30-DEC-23
```



```
W22058786 >
W22058786 > -- CatalogueType Object Table
W22058786 > SELECT * FROM ObjectCatalogue;

CATALOGUEID
-----------
TITLEINFO(TITLEID, TITLENAME, AUTHOR, SUBJECTAREA, BOOKDESCRIPTION)
--------------------------------------------------------------------------------
BOOKDESCRIPTION
--------------------------------------------------------------------------------
          2
TITLETYPE(2, 'To Kill a Mockingbird', 'Harper Lee', 'Fiction', 'A tale of racial
 injustice and moral growth.')
An enduring story of justice and compassion.

          3
TITLETYPE(3, '1984', 'George Orwell', 'Dystopian', 'A novel about totalitarianis

CATALOGUEID
-----------
TITLEINFO(TITLEID, TITLENAME, AUTHOR, SUBJECTAREA, BOOKDESCRIPTION)
--------------------------------------------------------------------------------
BOOKDESCRIPTION
--------------------------------------------------------------------------------
m and surveillance.')
A chilling depiction of a dystopian future.

          4
TITLETYPE(4, 'Pride and Prejudice', 'Jane Austen', 'Romance', 'A classic tale of
 love and social hierarchy.')
A tale of love and social hierarchy.

W22058786 >
W22058786 > -- StaffType Object Table
W22058786 > SELECT * FROM ObjectStaff;

   STAFFID
----------
NAME
--------------------------------------------------------------------------------
ROLE
--------------------------------------------------------------------------------
CONTACTINFORMATION
--------------------------------------------------------------------------------
         2
Librarian 2
Associate Librarian
Contact Info for Librarian 2


   STAFFID
----------
NAME
--------------------------------------------------------------------------------
ROLE
--------------------------------------------------------------------------------
CONTACTINFORMATION
--------------------------------------------------------------------------------
         3
Assistant 1
Library Assistant
Contact Info for Assistant 1
```

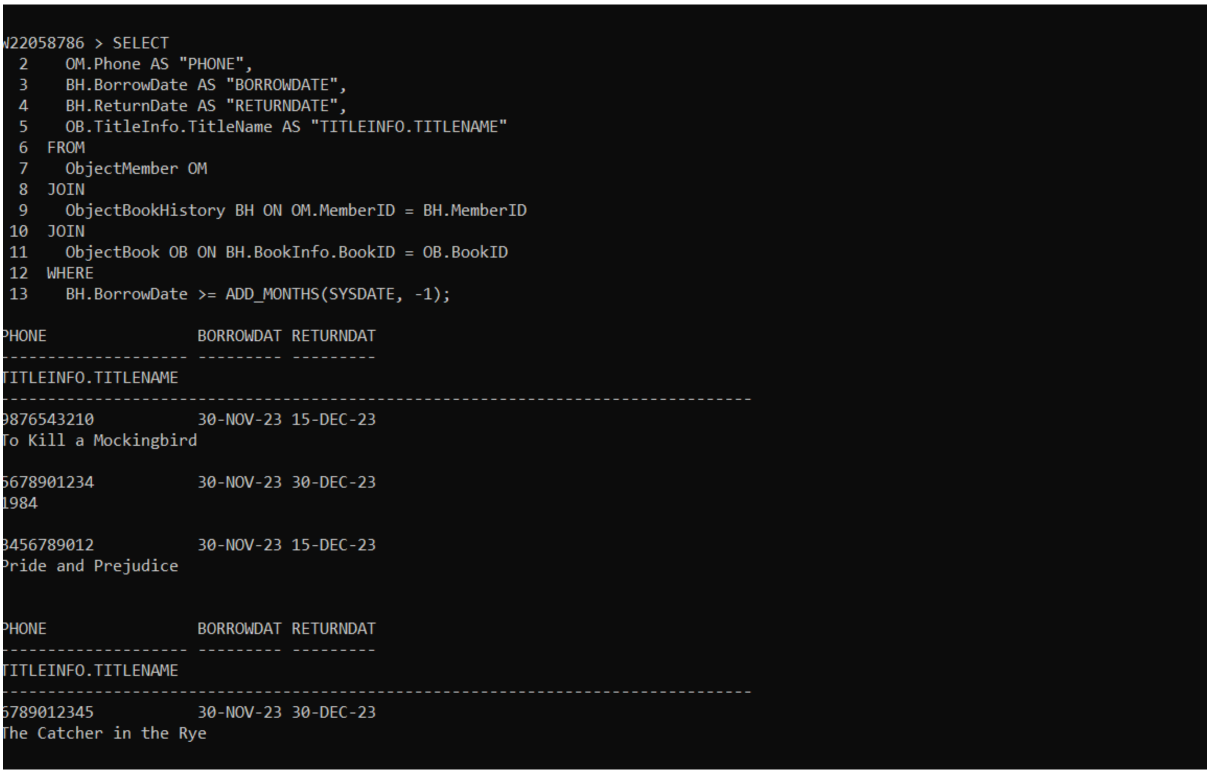


```
W22058786 >
W22058786 > -- CouncillorType Object Table
W22058786 > SELECT * FROM ObjectCouncillor;

COUNCILLORID
-----------
NAME
--------------------------------------------------------------------------------
ADDRESS
--------------------------------------------------------------------------------
CARDEXPIR
---------
CONTACTINFORMATION
--------------------------------------------------------------------------------
          2
Council Member 2
Address 2

COUNCILLORID
-----------
NAME
--------------------------------------------------------------------------------
ADDRESS
--------------------------------------------------------------------------------
CARDEXPIR
---------
CONTACTINFORMATION
--------------------------------------------------------------------------------
31-DEC-23
Contact Info for Council Member 2


W22058786 >
W22058786 > -- BookHistoryType Object Table
W22058786 > SELECT * FROM ObjectBookHistory;

 HISTORYID   MEMBERID
---------- ----------
BOOKINFO(BOOKID, TITLEINFO(TITLEID, TITLENAME, AUTHOR, SUBJECTAREA, BOOKDESCRIPT
--------------------------------------------------------------------------------
BORROWDAT RETURNDAT
--------- ---------
         2          2
BOOKTYPE(2, TITLETYPE(2, 'To Kill a Mockingbird', 'Harper Lee', 'Fiction', 'A ta
le of racial injustice and moral growth.'), 'Available', 'ISBN002', 'Excellent C
ondition', '30-NOV-23', '30-DEC-23')
30-NOV-23 15-DEC-23

         3          3

 HISTORYID   MEMBERID
---------- ----------
BOOKINFO(BOOKID, TITLEINFO(TITLEID, TITLENAME, AUTHOR, SUBJECTAREA, BOOKDESCRIPT
--------------------------------------------------------------------------------
BORROWDAT RETURNDAT
--------- ---------
BOOKTYPE(3, TITLETYPE(3, '1984', 'George Orwell', 'Dystopian', 'A novel about to
talitarianism and surveillance.'), 'Checked Out', 'ISBN003', 'Fair Condition', '
30-NOV-23', '15-DEC-23')
30-NOV-23 30-DEC-23

         4          4
BOOKTYPE(4, TITLETYPE(4, 'Pride and Prejudice', 'Jane Austen', 'Romance', 'A cla
```

## 5) Provide below the SQL query code as TEXT and output below (as image / screenshot / picture) for running two complex queries on the object-relational subset of the above BLS database (4 marks)

Query 1 Recent Book Borrowings with Member Information

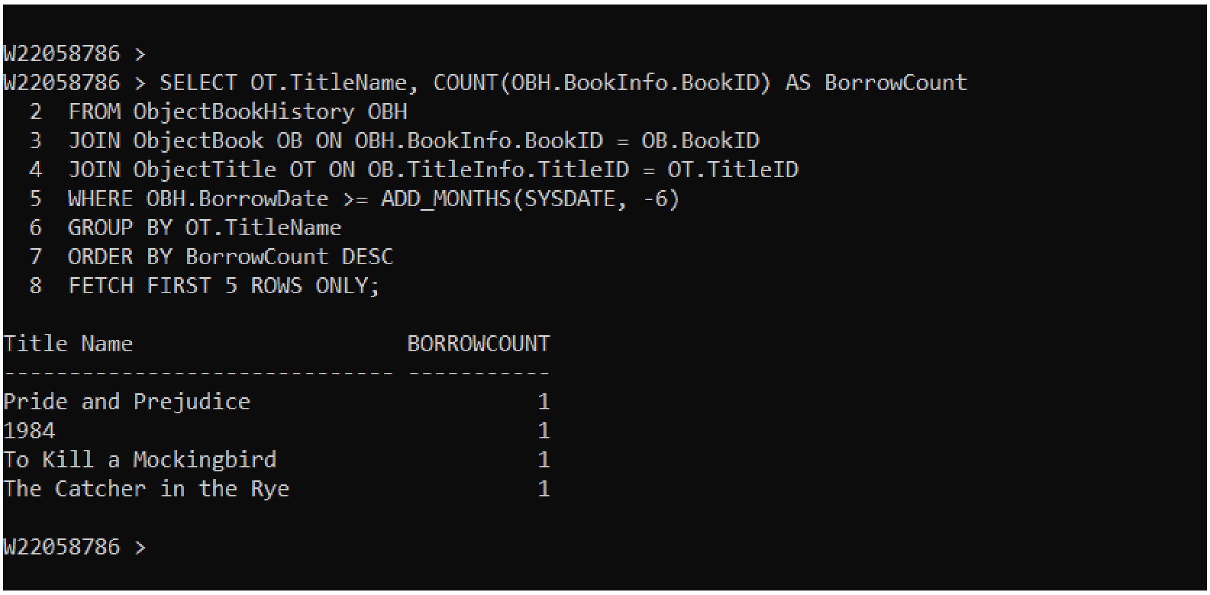
```
SELECT
  OM.Phone AS "PHONE",
  BH.BorrowDate AS "BORROWDATE",
  BH.ReturnDate AS "RETURNDATE",
  OB.TitleInfo.TitleName AS "TITLEINFO.TITLENAME"
FROM
  ObjectMember OM
JOIN
  ObjectBookHistory BH ON OM.MemberID = BH.MemberID
JOIN
  ObjectBook OB ON BH.BookInfo.BookID = OB.BookID
WHERE
  BH.BorrowDate >= ADD_MONTHS(SYSDATE, -1);
```



```
W22058786 > SELECT
      OM.Phone AS "PHONE",
      BH.BorrowDate AS "BORROWDATE",
      BH.ReturnDate AS "RETURNDATE",
      OB.TitleInfo.TitleName AS "TITLEINFO.TITLENAME"
    FROM
      ObjectMember OM
    JOIN
      ObjectBookHistory BH ON OM.MemberID = BH.MemberID
    JOIN
      ObjectBook OB ON BH.BookInfo.BookID = OB.BookID
    WHERE
      BH.BorrowDate >= ADD_MONTHS(SYSDATE, -1);

PHONE                BORROWDAT RETURNDAT
-------------------- --------- ---------
TITLEINFO.TITLENAME
--------------------------------------------------------------------------------
9876543210           30-NOV-23 15-DEC-23
To Kill a Mockingbird

5678901234           30-NOV-23 30-DEC-23
1984

3456789012           30-NOV-23 15-DEC-23
Pride and Prejudice


PHONE                BORROWDAT RETURNDAT
-------------------- --------- ---------
TITLEINFO.TITLENAME
--------------------------------------------------------------------------------
6789012345           30-NOV-23 30-DEC-23
The Catcher in the Rye
```

Query 2 Top 5 Titles by Borrow Count in the Last 6 Months

```
SELECT
  OT.TitleName,
  COUNT(OBH.BookInfo.BookID) AS BorrowCount
FROM
  ObjectBookHistory OBH
JOIN
  ObjectBook OB ON OBH.BookInfo.BookID = OB.BookID
JOIN
  ObjectTitle OT ON OB.TitleInfo.TitleID = OT.TitleID
WHERE
  OBH.BorrowDate >= ADD_MONTHS(SYSDATE, -6)
GROUP BY
  OT.TitleName
ORDER BY
  BorrowCount DESC
FETCH FIRST 5 ROWS ONLY;
```



```
W22058786 >
W22058786 > SELECT OT.TitleName, COUNT(OBH.BookInfo.BookID) AS BorrowCount
  2  FROM ObjectBookHistory OBH
  3  JOIN ObjectBook OB ON OBH.BookInfo.BookID = OB.BookID
  4  JOIN ObjectTitle OT ON OB.TitleInfo.TitleID = OT.TitleID
  5  WHERE OBH.BorrowDate >= ADD_MONTHS(SYSDATE, -6)
  6  GROUP BY OT.TitleName
  7  ORDER BY BorrowCount DESC
  8  FETCH FIRST 5 ROWS ONLY;

Title Name                     BORROWCOUNT
------------------------------ -----------
Pride and Prejudice                      1
1984                                     1
To Kill a Mockingbird                    1
The Catcher in the Rye                   1

W22058786 >
```

*(B) Analyse the conceptual database design from Part 1 (A) and the BLS scenario in the Appendix and propose what aspects of the BLS database would benefit from incorporating NoSQL Database concepts. Illustrate your answer with code from a representative code from NoSQL Database implementation.*

(15 marks)

## Answer Part 3 B

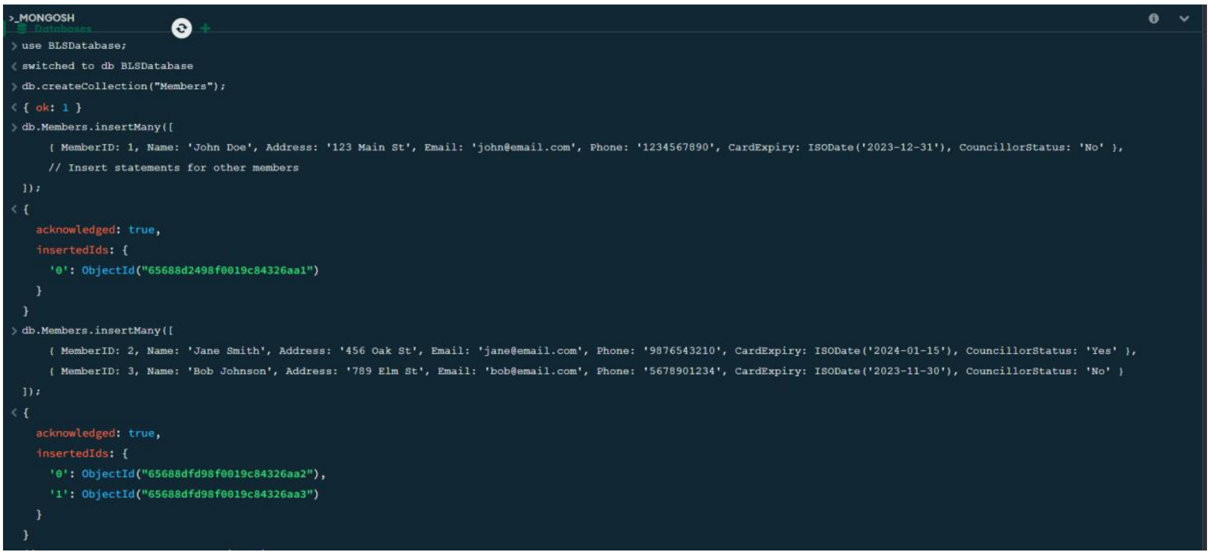
1) **Provide below your choice and justification of what aspects (subset) of the BLS databases would benefit from incorporating NoSQL Database concepts (3 marks)**

In the context of the Beacon Library System (BLS) scenario, integrating NoSQL database concepts, especially in catalog management, can offer several advantages. One key benefit is the flexible schema provided by NoSQL databases, such as document-oriented databases like MongoDB, which can adapt to varying data formats without a predefined structure. Additionally, NoSQL databases excel in scalability, allowing efficient handling of large volumes of unstructured data as the number of titles and associated descriptions grows. They also offer optimized query performance and support for large documents, making them suitable for managing diverse and evolving data structures. This integration can lead to improved flexibility, scalability, and performance in handling catalog-related data.

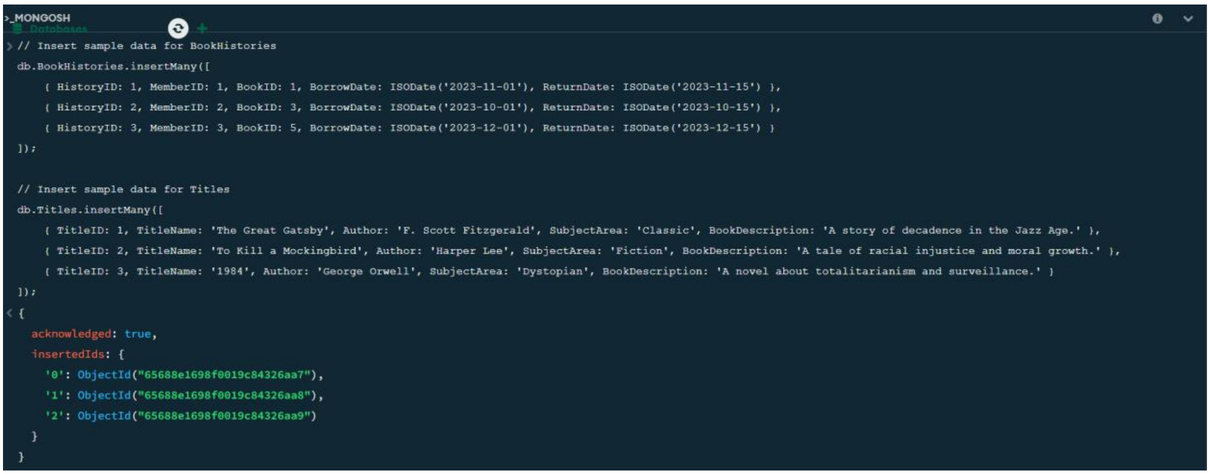
2) **Provide below, code as TEXT and output below (as image / screenshot / picture) for implementing your proposed NoSQL Database subset of the BLS database, populate it with some data, and example queries & outputs (12 Marks)**

```
use BLSDatabase;
switched to db BLSDatabase
db.createCollection("Members");
db.Members.insertMany([
    { MemberID: 1, Name: 'John Doe', Address: '123 Main St', Email: 'john@email.com', Phone: '1234567890', CardExpiry: ISODate('2023-12-31'), CouncillorStatus: 'No' },
    // Insert statements for other members
]);
db.Members.insertMany([
    { MemberID: 2, Name: 'Jane Smith', Address: '456 Oak St', Email: 'jane@email.com', Phone: '9876543210', CardExpiry: ISODate('2024-01-15'), CouncillorStatus: 'Yes' },
    { MemberID: 3, Name: 'Bob Johnson', Address: '789 Elm St', Email: 'bob@email.com', Phone: '5678901234', CardExpiry: ISODate('2023-11-30'), CouncillorStatus: 'No' }
]);
}
// Insert sample data for BookHistories
```

```
db.BookHistories.insertMany([
    { HistoryID: 1, MemberID: 1, BookID: 1, BorrowDate:
ISODate('2023-11-01'), ReturnDate: ISODate('2023-11-15') },
    { HistoryID: 2, MemberID: 2, BookID: 3, BorrowDate:
ISODate('2023-10-01'), ReturnDate: ISODate('2023-10-15') },
    { HistoryID: 3, MemberID: 3, BookID: 5, BorrowDate:
ISODate('2023-12-01'), ReturnDate: ISODate('2023-12-15') }
]);

// Insert sample data for Titles
db.Titles.insertMany([
    { TitleID: 1, TitleName: 'The Great Gatsby', Author: 'F. Scott
Fitzgerald', SubjectArea: 'Classic', BookDescription: 'A story of
decadence in the Jazz Age.' },
    { TitleID: 2, TitleName: 'To Kill a Mockingbird', Author:
'Harper Lee', SubjectArea: 'Fiction', BookDescription: 'A tale of
racial injustice and moral growth.' },
    { TitleID: 3, TitleName: '1984', Author: 'George Orwell',
SubjectArea: 'Dystopian', BookDescription: 'A novel about
totalitarianism and surveillance.' }
]);
```

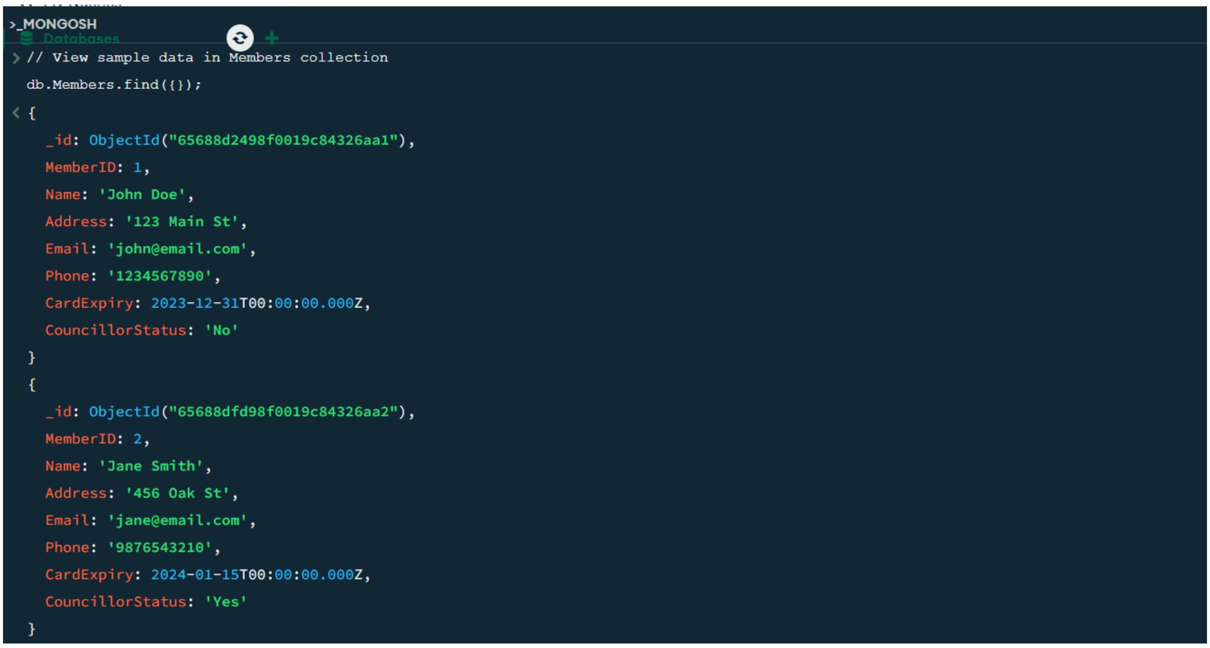


```
> use BLSDatabase;
< switched to db BLSDatabase
> db.createCollection("Members");
< { ok: 1 }
> db.Members.insertMany([
      { MemberID: 1, Name: 'John Doe', Address: '123 Main St', Email: 'john@email.com', Phone: '1234567890', CardExpiry: ISODate('2023-12-31'), CouncillorStatus: 'No' },
      // Insert statements for other members
  ]);
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId("65688d2498f0019c84326aa1")
    }
  }
> db.Members.insertMany([
      { MemberID: 2, Name: 'Jane Smith', Address: '456 Oak St', Email: 'jane@email.com', Phone: '9876543210', CardExpiry: ISODate('2024-01-15'), CouncillorStatus: 'Yes' },
      { MemberID: 3, Name: 'Bob Johnson', Address: '789 Elm St', Email: 'bob@email.com', Phone: '5678901234', CardExpiry: ISODate('2023-11-30'), CouncillorStatus: 'No' }
  ]);
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId("65688dfd98f0019c84326aa2"),
      '1': ObjectId("65688dfd98f0019c84326aa3")
    }
  }
```

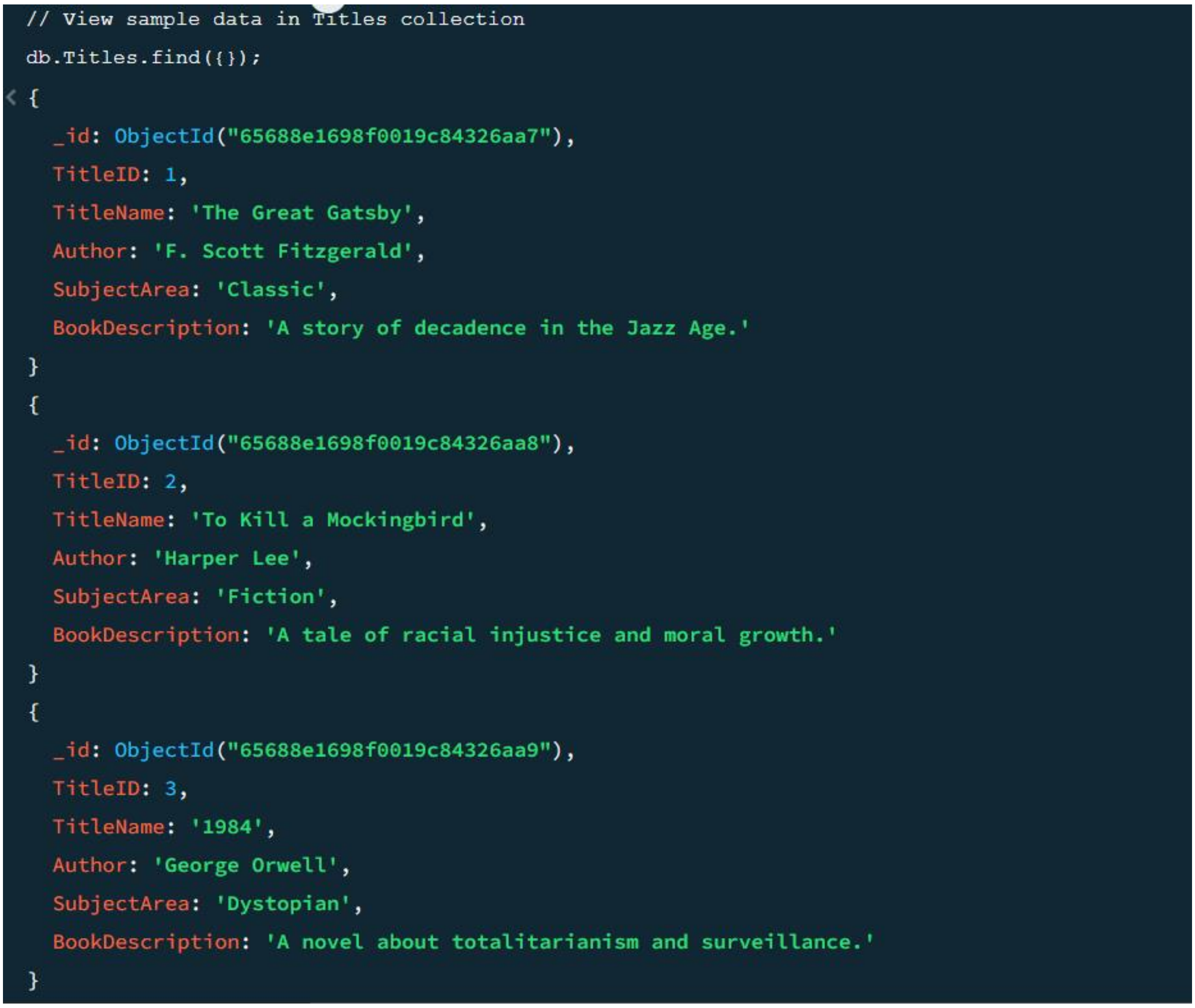


```
> // Insert sample data for BookHistories
  db.BookHistories.insertMany([
      { HistoryID: 1, MemberID: 1, BookID: 1, BorrowDate: ISODate('2023-11-01'), ReturnDate: ISODate('2023-11-15') },
      { HistoryID: 2, MemberID: 2, BookID: 3, BorrowDate: ISODate('2023-10-01'), ReturnDate: ISODate('2023-10-15') },
      { HistoryID: 3, MemberID: 3, BookID: 5, BorrowDate: ISODate('2023-12-01'), ReturnDate: ISODate('2023-12-15') }
  ]);

  // Insert sample data for Titles
  db.Titles.insertMany([
      { TitleID: 1, TitleName: 'The Great Gatsby', Author: 'F. Scott Fitzgerald', SubjectArea: 'Classic', BookDescription: 'A story of decadence in the Jazz Age.' },
      { TitleID: 2, TitleName: 'To Kill a Mockingbird', Author: 'Harper Lee', SubjectArea: 'Fiction', BookDescription: 'A tale of racial injustice and moral growth.' },
      { TitleID: 3, TitleName: '1984', Author: 'George Orwell', SubjectArea: 'Dystopian', BookDescription: 'A novel about totalitarianism and surveillance.' }
  ]);
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId("65688e1698f0019c84326aa7"),
      '1': ObjectId("65688e1698f0019c84326aa8"),
      '2': ObjectId("65688e1698f0019c84326aa9")
    }
  }
```

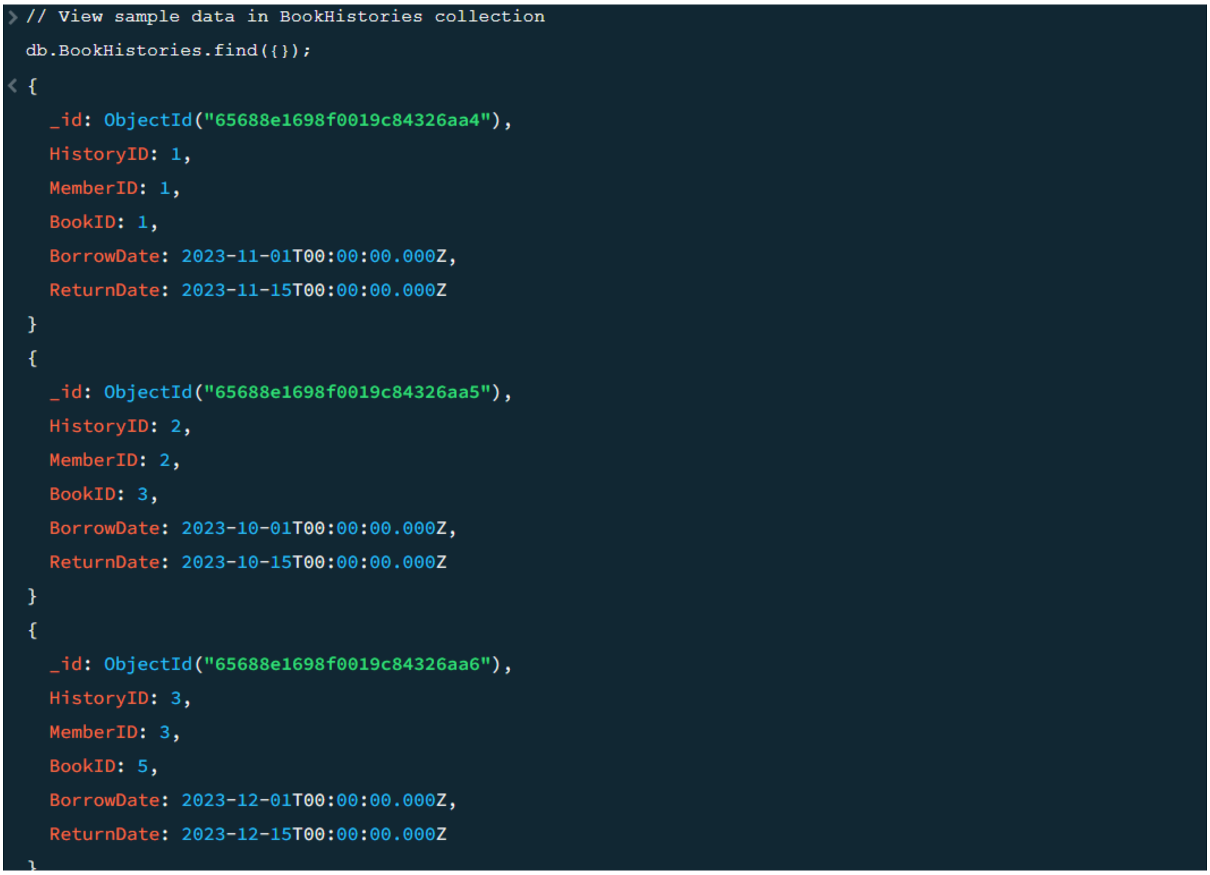
```
// View sample data in Members collection
db.Members.find({});

// View sample data in BookHistories collection
db.BookHistories.find({});

// View sample data in Titles collection
db.Titles.find({});

// View sample data in Members collection
db.Members.find({});

// View sample data in BookHistories collection
db.BookHistories.find({});
```



```
> // View sample data in Members collection
  db.Members.find({});
< {
    _id: ObjectId("65688d2498f0019c84326aa1"),
    MemberID: 1,
    Name: 'John Doe',
    Address: '123 Main St',
    Email: 'john@email.com',
    Phone: '1234567890',
    CardExpiry: 2023-12-31T00:00:00.000Z,
    CouncillorStatus: 'No'
  }
  {
    _id: ObjectId("65688dfd98f0019c84326aa2"),
    MemberID: 2,
    Name: 'Jane Smith',
    Address: '456 Oak St',
    Email: 'jane@email.com',
    Phone: '9876543210',
    CardExpiry: 2024-01-15T00:00:00.000Z,
    CouncillorStatus: 'Yes'
  }
```

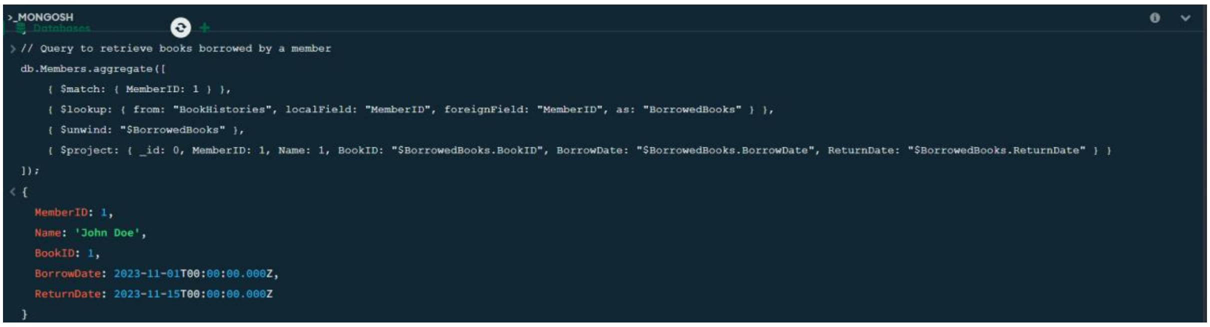


```
  // View sample data in Titles collection
  db.Titles.find({});
< {
    _id: ObjectId("65688e1698f0019c84326aa7"),
    TitleID: 1,
    TitleName: 'The Great Gatsby',
    Author: 'F. Scott Fitzgerald',
    SubjectArea: 'Classic',
    BookDescription: 'A story of decadence in the Jazz Age.'
  }
  {
    _id: ObjectId("65688e1698f0019c84326aa8"),
    TitleID: 2,
    TitleName: 'To Kill a Mockingbird',
    Author: 'Harper Lee',
    SubjectArea: 'Fiction',
    BookDescription: 'A tale of racial injustice and moral growth.'
  }
  {
    _id: ObjectId("65688e1698f0019c84326aa9"),
    TitleID: 3,
    TitleName: '1984',
    Author: 'George Orwell',
    SubjectArea: 'Dystopian',
    BookDescription: 'A novel about totalitarianism and surveillance.'
  }
```

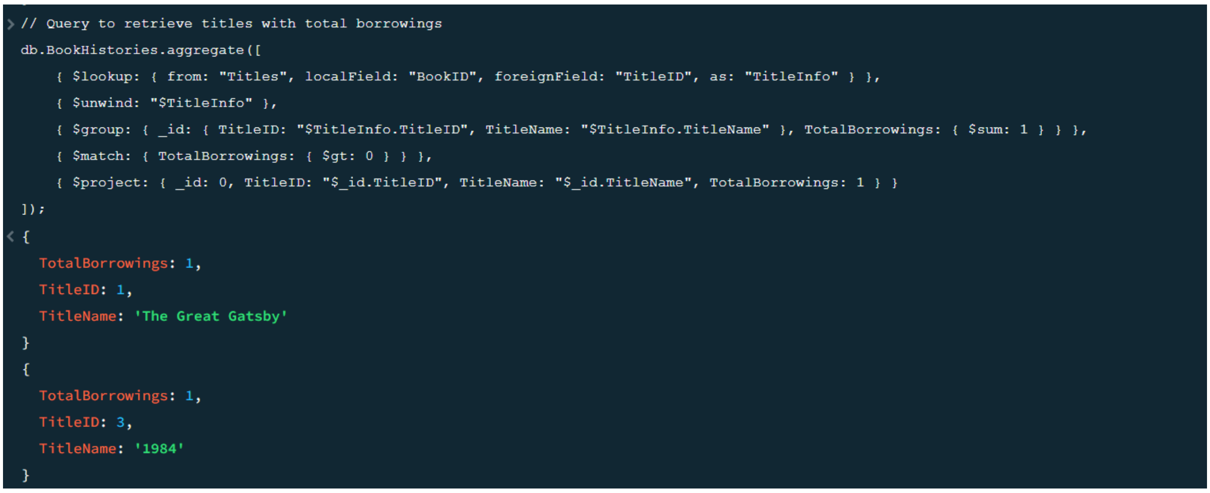


```
> // View sample data in BookHistories collection
  db.BookHistories.find({});
< {
    _id: ObjectId("65688e1698f0019c84326aa4"),
    HistoryID: 1,
    MemberID: 1,
    BookID: 1,
    BorrowDate: 2023-11-01T00:00:00.000Z,
    ReturnDate: 2023-11-15T00:00:00.000Z
  }
  {
    _id: ObjectId("65688e1698f0019c84326aa5"),
    HistoryID: 2,
    MemberID: 2,
    BookID: 3,
    BorrowDate: 2023-10-01T00:00:00.000Z,
    ReturnDate: 2023-10-15T00:00:00.000Z
  }
  {
    _id: ObjectId("65688e1698f0019c84326aa6"),
    HistoryID: 3,
    MemberID: 3,
    BookID: 5,
    BorrowDate: 2023-12-01T00:00:00.000Z,
    ReturnDate: 2023-12-15T00:00:00.000Z
```

```
// Query to retrieve books borrowed by a member
db.Members.aggregate([
    { $match: { MemberID: 1 } },
    { $lookup: { from: "BookHistories", localField: "MemberID",
foreignField: "MemberID", as: "BorrowedBooks" } },
    { $unwind: "$BorrowedBooks" },
    { $project: { _id: 0, MemberID: 1, Name: 1, BookID:
"$BorrowedBooks.BookID", BorrowDate: "$BorrowedBooks.BorrowDate",
ReturnDate: "$BorrowedBooks.ReturnDate" } }
]);

// Query to retrieve titles with total borrowings
db.BookHistories.aggregate([
    { $lookup: { from: "Titles", localField: "BookID", foreignField:
"TitleID", as: "TitleInfo" } },
    { $unwind: "$TitleInfo" },
    { $group: { _id: { TitleID: "$TitleInfo.TitleID", TitleName:
"$TitleInfo.TitleName" }, TotalBorrowings: { $sum: 1 } } },
    { $match: { TotalBorrowings: { $gt: 0 } } },
    { $project: { _id: 0, TitleID: "$_id.TitleID", TitleName:
"$_id.TitleName", TotalBorrowings: 1 } }
]);
```



```
> // Query to retrieve books borrowed by a member
  db.Members.aggregate([
      { $match: { MemberID: 1 } },
      { $lookup: { from: "BookHistories", localField: "MemberID", foreignField: "MemberID", as: "BorrowedBooks" } },
      { $unwind: "$BorrowedBooks" },
      { $project: { _id: 0, MemberID: 1, Name: 1, BookID: "$BorrowedBooks.BookID", BorrowDate: "$BorrowedBooks.BorrowDate", ReturnDate: "$BorrowedBooks.ReturnDate" } }
  ]);
< {
    MemberID: 1,
    Name: 'John Doe',
    BookID: 1,
    BorrowDate: 2023-11-01T00:00:00.000Z,
    ReturnDate: 2023-11-15T00:00:00.000Z
  }
```



```
> // Query to retrieve titles with total borrowings
  db.BookHistories.aggregate([
      { $lookup: { from: "Titles", localField: "BookID", foreignField: "TitleID", as: "TitleInfo" } },
      { $unwind: "$TitleInfo" },
      { $group: { _id: { TitleID: "$TitleInfo.TitleID", TitleName: "$TitleInfo.TitleName" }, TotalBorrowings: { $sum: 1 } } },
      { $match: { TotalBorrowings: { $gt: 0 } } },
      { $project: { _id: 0, TitleID: "$_id.TitleID", TitleName: "$_id.TitleName", TotalBorrowings: 1 } }
  ]);
< {
    TotalBorrowings: 1,
    TitleID: 1,
    TitleName: 'The Great Gatsby'
  }
  {
    TotalBorrowings: 1,
    TitleID: 3,
    TitleName: '1984'
  }
```

## Part 4 (10 marks)

*Consider the BLS scenario in the Appendix. Produce a report for the Lord Mayor of BCC – elaborating on sustainability, professional, legal, ethical and security issues, and matters related to diversity, inclusion, cultural, societal and environment as well as risk management and evaluation of commercial risks that need to be considered and make appropriate recommendations for BLS. Whilst answering this question, employ a critical review of current and relevant literature, systems, developments, and standards.*

(10 marks)

The report should be concise and comprehensive and in the region of 900-1000 words.

You should use Harvard style of citation and referencing by following the guidelines in Pears and Shields (2008).

**Answer Part 4: 10 Marks** [8 for the quality of report covering all the above issues, 1 for the quality of referencing and citation and adhering to the Harvard style, 1 for presentation]

Report for the Lord Mayor of BCC on Beacon Library System (BLS)

Introduction

This comprehensive report offers an in-depth analysis of the Beacon Library System (BLS), focusing on sustainability, professional excellence, legal and ethical considerations, security matters, diversity and inclusion, cultural and societal impacts, and environmental responsibilities. Additionally, the report addresses risk management and evaluates commercial risks affecting the BLS. The subsequent recommendations are meticulously tailored to enhance the overall efficiency and effectiveness of the library system.

Sustainability

(Graham) Sustainability is not merely a contemporary trend but a fundamental principle in modern library management. BLS, as a responsible institution, is encouraged to adopt cutting-edge strategies outlined in current literature. This includes embracing energy-efficient lighting, initiating robust recycling programs, and expediting the digitization of resources to not only augment efficiency but also significantly minimize the environmental footprint. Beyond immediate cost reductions, these measures align seamlessly with eco-friendly practices, contributing substantially to environmental conservation and fostering community goodwill.

The implementation of sustainability measures in BLS can serve as a beacon of inspiration, setting a precedent for other institutions within the community. By leading the way in eco-friendly practices, BLS can become a catalyst for positive change, promoting environmental consciousness among both staff and patrons.

Professional Standards

The cornerstone of BLS's success lies in its commitment to upholding high service standards. Achieving this necessitates a comprehensive training program for all staff members, ranging from senior librarians to assistants. A critical literature review underscores the paramount importance of continuous professional development, ensuring that the staff remains not only proficient but also adaptive to the latest trends and technologies in library management.

Expanding further on the theme of professional development, BLS can institute mentorship programs, encouraging experienced staff to guide newer members. This not only enhances the overall skill set and efficiency of the library team but also fosters a culture of teamwork and knowledge sharing within the institution.

Legal and Ethical Considerations

In navigating legal issues related to data protection and privacy laws, BLS must not only adhere to the strict compliance of these regulations but also consider the evolving landscape of data protection. Regular updates to policies and procedures based on the synthesis of relevant literature and standards become imperative to ensure the secure storage and lawful use of member data.

Ethical considerations extend beyond legal compliance to ensuring equitable access to resources for all members and maintaining an unbiased collection of books. BLS, as a hub of knowledge and information, should actively seek out voices and perspectives that may be underrepresented. This not only aligns with ethical considerations but also contributes to the intellectual richness of the library's offerings.

Security Issues

Comprehensive security measures are paramount for safeguarding both physical and digital assets within BLS. A critical literature review recommends adopting a holistic approach to security, combining robust physical measures, such as CCTV cameras, alarm systems, and secure storage, with cutting-edge digital security measures. Proactively mitigating potential risks ensures the integrity of BLS operations, safeguarding both tangible and intangible assets from unforeseen challenges.

In the realm of digital security, ongoing cybersecurity training for staff is essential. This not only ensures that staff remains vigilant against emerging threats but also cultivates a culture of cyber-resilience within the organization.

Diversity, Inclusion, Cultural, Societal, and Environmental Considerations

Promoting diversity and inclusion is not only a strategic imperative but also a moral obligation for BLS to effectively cater to the diverse cultural and societal interests of its members. The literature underscores the transformative power of libraries as inclusive spaces, providing resources that authentically represent a broad range of perspectives and experiences. BLS should actively engage with community groups, seeking input and collaboration to ensure that the library truly reflects the diversity of its patrons.

In assessing and mitigating its environmental impact, BLS can take a leadership role in implementing eco-friendly strategies. Beyond recycling and energy conservation, BLS can explore partnerships with local environmental organizations to host educational programs, creating a platform for community engagement and environmental awareness.

Risk Management and Commercial Risks
Effective risk management is not merely about identifying and mitigating potential risks; it is about fostering a culture of resilience within BLS. The critical literature review strongly recommends the development and implementation of a comprehensive risk management strategy. This strategy should systematically identify, assess, and mitigate potential risks associated with various facets of library operations.

Expanding on risk management, BLS can integrate scenario planning exercises into its regular operations. This proactive approach allows the organization to simulate potential challenges and refine its response strategies, ensuring a more agile and resilient institution in the face of uncertainties.

Recommendations
Sustainability Strategies: While BLS should embrace energy-efficient lighting, recycling programs, and digitization of resources, it can further enhance its sustainability initiatives by actively engaging with the community through educational programs and partnerships with local environmental organizations.

Staff Training: Institute not only regular training sessions but also mentorship programs to foster a culture of continuous learning and collaboration within the organization. Encourage staff to participate in industry conferences and workshops to gain exposure to the latest trends and technologies.

Data Protection Compliance: Beyond compliance, regularly review and update data protection policies based on emerging trends and legal developments. Consider conducting periodic audits to ensure the ongoing security and ethical use of member data.

Diversity Promotion: (Philpotts, 2020) Actively seek input from community groups to ensure that the library's collection authentically represents the diverse interests and demographics of its patrons. Host cultural events and collaborate with local artists to create a vibrant and inclusive library environment.

Comprehensive Security Measures: Regularly update physical security measures to align with emerging technologies. Conduct regular cybersecurity training for staff and explore partnerships with cybersecurity experts to stay ahead of potential threats.

Risk Management Strategy: Integrate scenario planning exercises into regular operations to enhance the organization's ability to respond effectively to unforeseen challenges. Consider establishing a dedicated risk management committee to oversee and refine the implementation of risk mitigation strategies.

## Conclusion

In conclusion, aligning BLS practices with the provided recommendations is not just about meeting operational standards; it is about positioning the library as a dynamic and responsible institution within the community. By drawing insights from relevant literature, systems, developments, and standards, the Beacon Library System has the potential to become a trailblazer in innovative library management. The adoption of these recommendations is not only a strategic imperative but a moral obligation for the Beacon Library System's continued success in serving the community effectively and responsibly. Through proactive measures and a commitment to excellence, BLS can not only meet but exceed the expectations of its patrons, becoming a beacon of knowledge, inclusion, and sustainability within the community.

## References:

Graham, L. (no date) *Supporting sustainability: What can librarians do?*, *Supporting sustainability: what can librarians do? | For Librarians | Springer Nature*. Available at: https://www.springernature.com/gp/librarians/the-link/blog/blogposts-news-initiatives/what-librarians-can-do-to-support-sustainability/23146572 (Accessed: 30 November 2023).

Philpotts, L.L. (2020) *Diversity and inclusion in libraries: A call to action and strategies for Success*, *Journal of the Medical Library Association : JMLA*. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7524626/ (Accessed: 30 November 2023).