Advance Data Base
Midterm

Name
Muhammad Baihaqi Aulia Asy'ari

NIM
2241720145

Class
2I

Department
Information Technology

Study Program
D4 Informatics Engineering

1 Case Study Setup

1.1 Create Tables

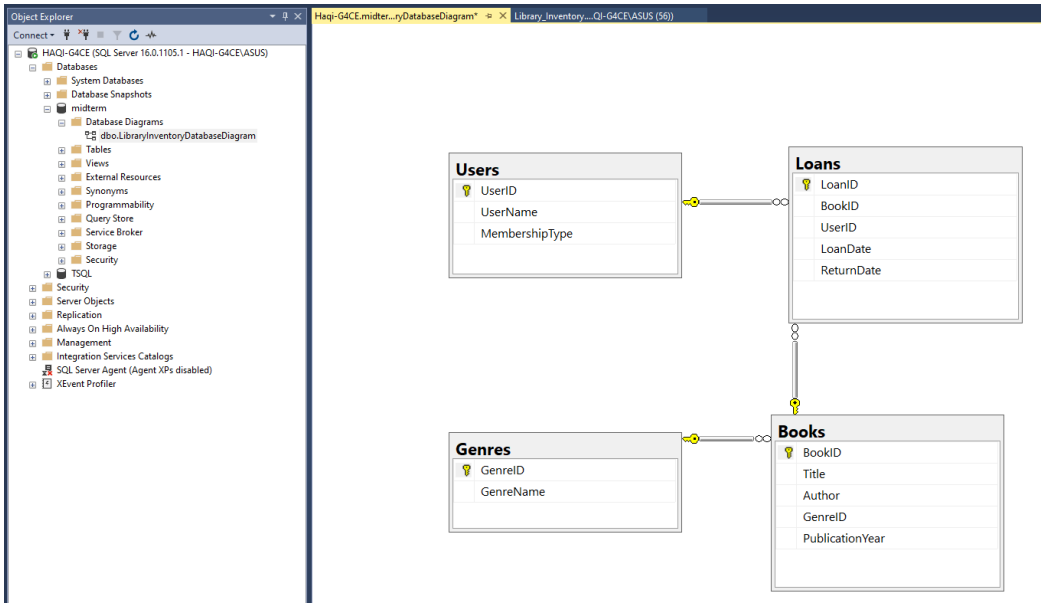
```sql
-- Create tables in case study
CREATE TABLE Genres (
    GenreID     INT             NOT NULL    PRIMARY KEY,
    GenreName   VARCHAR(100)    NOT NULL
);

CREATE TABLE Books (
    BookID          INT             NOT NULL    PRIMARY KEY,
    Title           VARCHAR(255)    NOT NULL,
    Author          VARCHAR(100)    NOT NULL,
    GenreID         INT             NOT NUll,
    PublicationYear INT             NOT NULL,
    CONSTRAINT FK_GenreID FOREIGN KEY (GenreID) REFERENCES Genres(GenreID)
);

CREATE TABLE Users (
    UserID          INT             NOT NULL    PRIMARY KEY,
    UserName        VARCHAR(100)    NOT NULl,
    MembershipType  VARCHAR(25)     NOT NULL
    CHECK (MembershipType IN ('platinum', 'gold', 'silver'))
);

CREATE TABLE Loans (
    LoanID      INT         NOT NULL    PRIMARY KEY,
    BookID      INT         NOT NULL,
    UserID      INT         NOT NULL,
    LoanDate    DATETIME    NOT NULL,
    ReturnDate  DATETIME    NOT NULL,
    CONSTRAINT FK_BookID FOREIGN KEY (BookID) REFERENCES Books(BookID),
    CONSTRAINT FK_UserID FOREIGN KEY (UserID) REFERENCES Users(UserID)
);
```

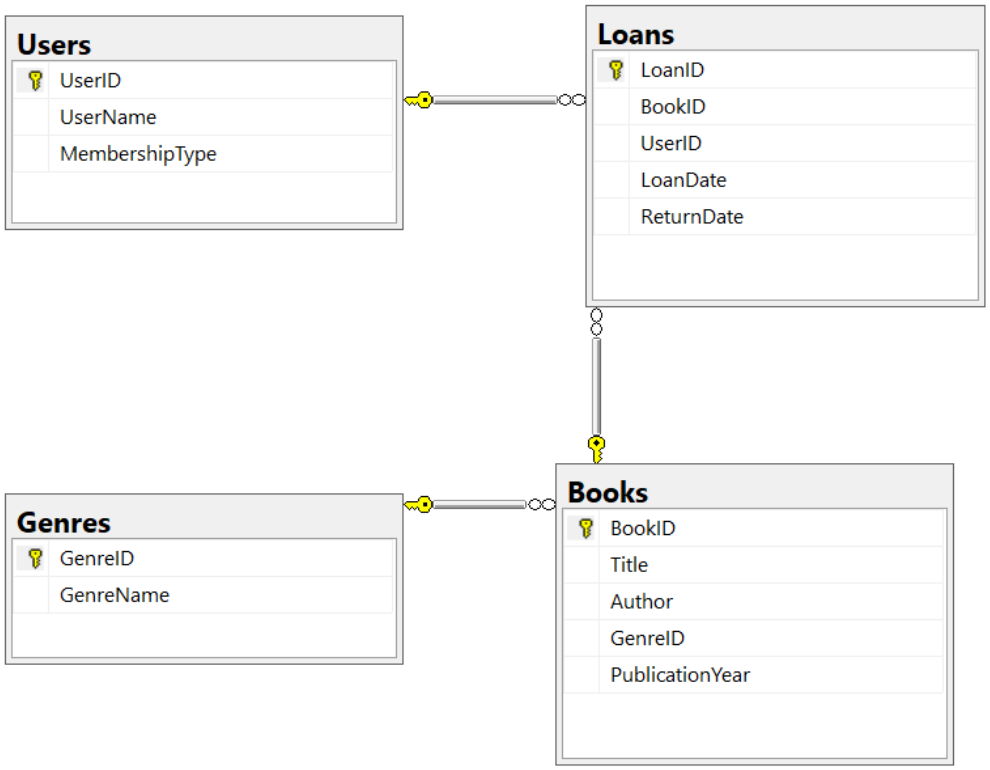
1.2 Insert Data

```sql
-- Insert data into the tables
/*  10 data each for tables Genres and Books
 *  5 data in the Users table
 *  25 data in the Loans table
 */

INSERT INTO Genres VALUES
(1, 'Drama'),
(2, 'Action'),
(3, 'Comedy'),
(4, 'Sci-Fi'),
(5, 'Fantasy'),
(6, 'Romance'),
(7, 'Cookbook'),
(8, 'Childrens Literature'),
(9, 'Philosophy'),
(10, 'Self-Help');

INSERT INTO Books VALUES
(1, 'My Mom Is My Cousin From Alabama', 'Jesse Earl Montgomery', 1, 2016),
(2, 'How To Not Overspent', 'Jared Bankman', 10, 2021),
(3, 'Roy and Marty: The Seeds of Life', 'Greg Harmon', 4, 2019),
(4, 'Need For Fuel', 'Earl Armisen', 2, 2013),
(5, 'Outside', 'Ben Frozeham', 3, 2020),
(6, 'My ABCs', 'Saltfork Utensils', 8, 2022),
(7, 'Small and Big', 'Cise Matter', 9, 2015),
(8, 'Beans and Toast', 'Gideon Ramly', 7, 2017),
(9, 'Letter in My Locker', 'Roomie Saddler', 6, 2014),
(10, 'Big Dragon In Nut Kingdom', 'Bofa Deez', 5, 2018);

INSERT INTO Users VALUES
(1, 'Baihaqi', 'platinum'),
(2, 'Elmira', 'gold'),
(3, 'Dave', 'silver'),
(4, 'Mike', 'gold'),
(5, 'Eli', 'platinum');

INSERT INTO Loans VALUES
-- loanID, bookID, userID, loanDate, returnDate
(1, 2, 1, '2023-01-10 12:00:00.000', '2023-01-15 12:00:00.000'),
(2, 10, 5, '2023-01-12 12:00:00.000', '2023-01-16 12:00:00.000'),
(3, 5, 2, '2023-01-14 12:00:00.000', '2023-01-19 12:00:00.000'),
(4, 3, 3, '2023-01-13 12:00:00.000', '2023-01-16 12:00:00.000'),
(5, 5, 5, '2023-01-11 12:00:00.000', '2023-01-17 12:00:00.000'),
(6, 4, 4, '2023-01-15 12:00:00.000', '2023-01-18 12:00:00.000'),
(7, 8, 3, '2023-01-13 12:00:00.000', '2023-01-17 12:00:00.000'),
(8, 6, 1, '2023-01-12 12:00:00.000', '2023-01-16 12:00:00.000'),
(9, 1, 3, '2023-01-10 12:00:00.000', '2023-01-15 12:00:00.000'),
(10, 9, 2, '2023-01-11 12:00:00.000', '2023-01-16 12:00:00.000'),
(11, 1, 1, '2023-02-10 12:00:00.000', '2023-02-15 12:00:00.000'),
(12, 2, 5, '2023-02-12 12:00:00.000', '2023-02-16 12:00:00.000'),
(13, 4, 2, '2023-02-14 12:00:00.000', '2023-02-19 12:00:00.000'),
(14, 3, 3, '2023-02-13 12:00:00.000', '2023-02-16 12:00:00.000'),
(15, 6, 5, '2023-02-11 12:00:00.000', '2023-02-16 12:00:00.000'),
(16, 5, 4, '2023-02-15 12:00:00.000', '2023-02-18 12:00:00.000'),
(17, 8, 3, '2023-02-13 12:00:00.000', '2023-02-17 12:00:00.000'),
(18, 9, 1, '2023-02-12 12:00:00.000', '2023-02-16 12:00:00.000'),
(19, 5, 3, '2023-02-10 12:00:00.000', '2023-02-15 12:00:00.000'),
(20, 10, 2, '2023-02-11 12:00:00.000', '2023-02-16 12:00:00.000'),
(21, 2, 5, '2023-03-15 12:00:00.000', '2023-03-18 12:00:00.000'),
(22, 4, 2, '2023-03-13 12:00:00.000', '2023-03-17 12:00:00.000'),
(23, 5, 1, '2023-03-12 12:00:00.000', '2023-03-16 12:00:00.000'),
(24, 9, 5, '2023-03-10 12:00:00.000', '2023-03-15 12:00:00.000'),
(25, 8, 1, '2023-03-11 12:00:00.000', '2023-03-16 12:00:00.000');
```

2 Assignment

1. -





2. Query

```
-- Show all data inside all of the tables
SELECT * FROM Genres;
SELECT * FROM Books;
SELECT * FROM Users;
SELECT * FROM Loans;
```

Result

Results Messages

	GenreID	GenreName
1	1	Drama
2	2	Action
3	3	Comedy
4	4	Sci-Fi
5	5	Fantasy
6	6	Romance
7	7	Cookbook
8	8	Childrens Literature
9	9	Philosophy
10	10	Self-Help

Results Messages

	BookID	Title	Author	GenreID	PublicationYear
1	1	My Mom Is My Cousin From Alabama	Jesse Earl Montgomery	1	2016
2	2	How To Not Overspent	Jared Bankman	10	2021
3	3	Roy and Marty: The Seeds of Life	Greg Harmon	4	2019
4	4	Need For Fuel	Earl Arnisen	2	2013
5	5	Outside	Ben Frozeham	3	2020
6	6	My ABCs	Saltfork Utensils	8	2022
7	7	Small and Big	Cise Matter	9	2015
8	8	Beans and Toast	Gideon Ramly	7	2017
9	9	Letter in My Locker	Roomie Saddler	6	2014
10	10	Big Dragon In Nut Kingdom	Bofa Deez	5	2018

Results Messages

	UserID	UserName	MembershipType
1	1	Baihaqi	platinum
2	2	Elmira	gold
3	3	Dave	silver
4	4	Mike	gold
5	5	Eli	platinum

Results Messages

	LoanID	BookID	UserID	LoanDate	ReturnDate
1	1	2	1	2023-01-10 12:00:00.000	2023-01-15 12:00:00.000
2	2	10	5	2023-01-12 12:00:00.000	2023-01-16 12:00:00.000
3	3	5	2	2023-01-14 12:00:00.000	2023-01-19 12:00:00.000
4	4	3	3	2023-01-13 12:00:00.000	2023-01-16 12:00:00.000
5	5	5	5	2023-01-11 12:00:00.000	2023-01-17 12:00:00.000
6	6	4	4	2023-01-15 12:00:00.000	2023-01-18 12:00:00.000
7	7	8	3	2023-01-13 12:00:00.000	2023-01-17 12:00:00.000
8	8	6	1	2023-01-12 12:00:00.000	2023-01-16 12:00:00.000
9	9	1	3	2023-01-10 12:00:00.000	2023-01-15 12:00:00.000
10	10	9	2	2023-01-11 12:00:00.000	2023-01-16 12:00:00.000
11	11	1	1	2023-02-10 12:00:00.000	2023-02-15 12:00:00.000
12	12	2	5	2023-02-12 12:00:00.000	2023-02-16 12:00:00.000
13	13	4	2	2023-02-14 12:00:00.000	2023-02-19 12:00:00.000
14	14	3	3	2023-02-13 12:00:00.000	2023-02-16 12:00:00.000
15	15	6	5	2023-02-11 12:00:00.000	2023-02-16 12:00:00.000
16	16	5	4	2023-02-15 12:00:00.000	2023-02-18 12:00:00.000
17	17	8	3	2023-02-13 12:00:00.000	2023-02-17 12:00:00.000

Query executed successfully.

3. Query

```
-- Show Book name and author of unread books
SELECT Title, Author FROM Books
WHERE BookID NOT IN (SELECT BookID FROM Loans);
```

Result

Results Messages

	Title	Author
1	Small and Big	Cise Matter

4. Query

```sql
-- Show how many time a book of a certain genre have been borrowed
SELECT g.GenreName, COUNT(b.BookID) AS loan_amount
FROM Genres AS g
JOIN Books AS b ON b.GenreID = g.GenreID
JOIN Loans AS l ON b.BookID = l.BookID
GROUP BY g.GenreName;
```

Result

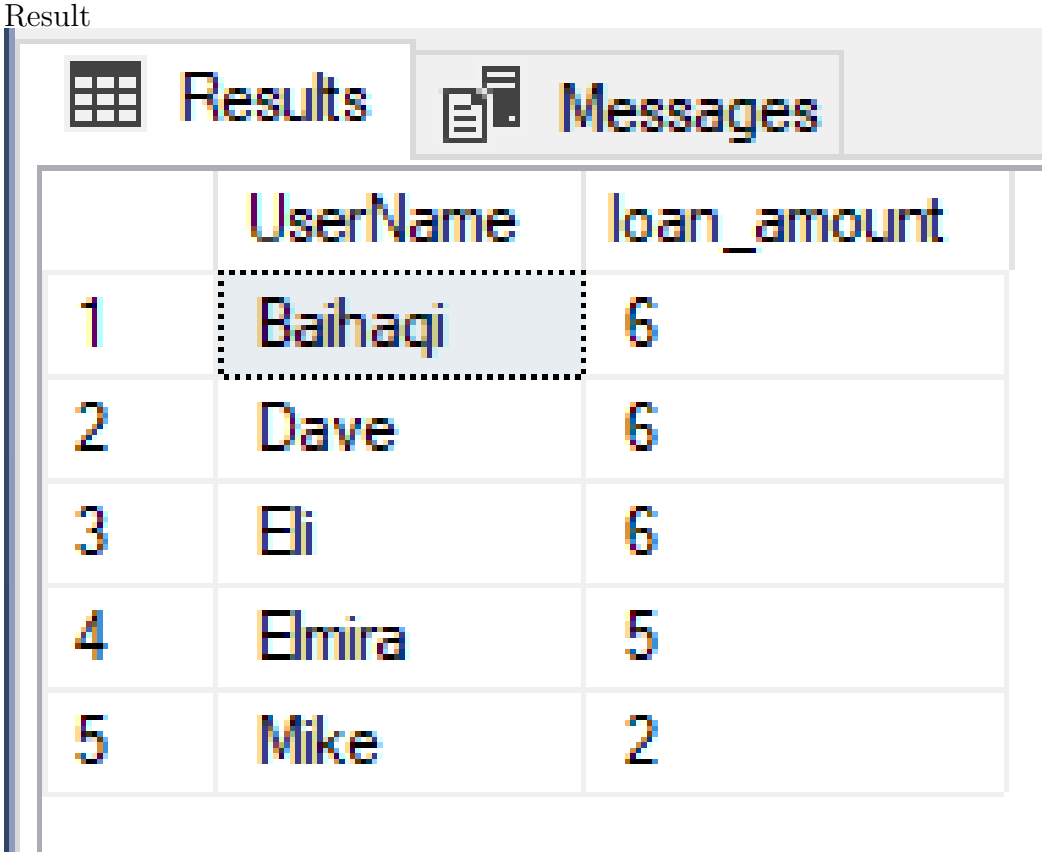
Results | Messages

	GenreName	loan_amount
1	Action	3
2	Childrens Literature	2
3	Comedy	5
4	Cookbook	3
5	Drama	2
6	Fantasy	2
7	Romance	3
8	Sci-Fi	2
9	Self-Help	3

5. Query

```sql
-- Show loan amount per user
SELECT u.UserName, COUNT(l.LoanID) AS loan_amount
FROM Users AS u
JOIN Loans AS l ON l.UserID = u.UserID
GROUP BY u.UserName
```

Result



	UserName	loan_amount
1	Baihaqi	6
2	Dave	6
3	Eli	6
4	Elmira	5
5	Mike	2

6. Query

```
-- Show frequently borrowed books
SELECT b.Title, COUNT(l.LoanID) AS loan_amount
FROM Books AS b
JOIN Loans AS l ON l.BookID = b.BookID
GROUP BY b.Title
ORDER BY loan_amount DESC
```

Result

Results | Messages

	Title	loan_amount
1	Outside	5
2	Beans and Toast	3
3	How To Not Overspent	3
4	Letter in My Locker	3
5	Need For Fuel	3
6	My ABCs	2
7	My Mom Is My Cousin From Alabama	2
8	Big Dragon In Nut Kingdom	2
9	Roy and Marty: The Seeds of Life	2

7. Query

```
-- Show average length of borrowing
SELECT b.Title, AVG(DATEDIFF(DAY, l.LoanDate, l.ReturnDate)) AS AverageBorrowingLength
FROM Books AS b
JOIN Loans AS l ON l.BookID = b.BookID
GROUP BY b.Title;
```

Result

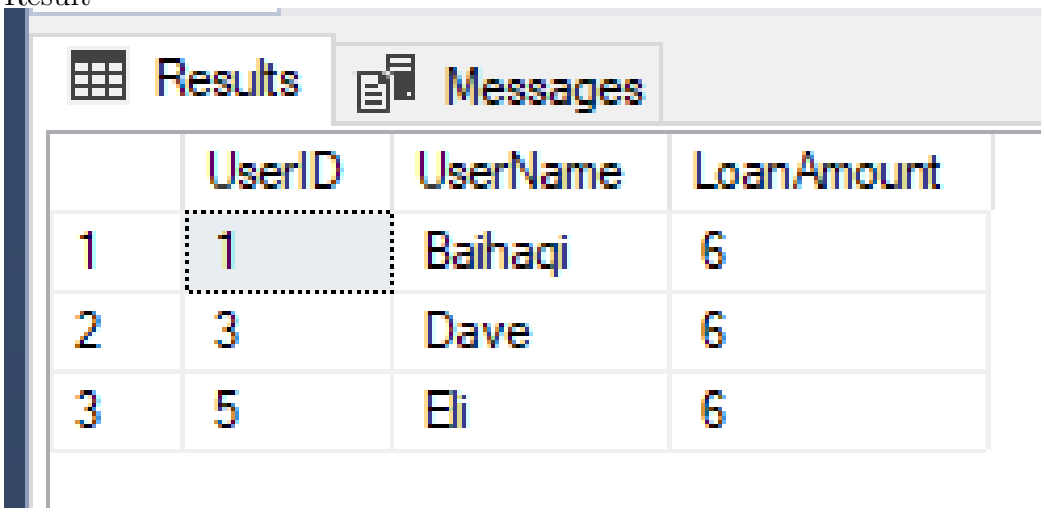
Results | Messages

	Title	AverageBorrowingLength
1	Beans and Toast	4
2	Big Dragon In Nut Kingdom	4
3	How To Not Overspent	4
4	Letter in My Locker	4
5	My ABCs	4
6	My Mom Is My Cousin From Alabama	5
7	Need For Fuel	4
8	Outside	4
9	Roy and Marty: The Seeds of Life	3

8. Query

```sql
-- Show user with more than average borrowing
SELECT u.UserID, u.UserName, COUNT(l.LoanID) AS LoanAmount
FROM Users AS u
LEFT JOIN Loans AS l ON u.UserID = l.UserID
GROUP BY u.UserID, u.UserName
HAVING COUNT(l.LoanID) > (
    SELECT AVG(LoanAmount)
    FROM (
        SELECT u.UserID, COUNT(l.LoanID) AS LoanAmount
        FROM Users AS u
        LEFT JOIN Loans AS l ON u.UserID = l.UserID
        GROUP BY u.UserID
    ) AS UserLoanAmounts
);
```

Result



Results | Messages

	UserID	UserName	LoanAmount
1	1	Baihaqi	6
2	3	Dave	6
3	5	Eli	6

9. Query

```sql
-- show history of a certain book
SELECT l.LoanID, b.Title, u.UserName, l.LoanDate, l.ReturnDate
FROM Loans AS l
JOIN Books AS b ON l.BookID = b.BookID
JOIN Users AS u ON l.UserID = u.UserID
WHERE b.BookID = 5;
```

Result

Results | Messages

	LoanID	Title	UserName	LoanDate	ReturnDate
1	3	Outside	Elmira	2023-01-14 12:00:00.000	2023-01-19 12:00:00.000
2	5	Outside	Eli	2023-01-11 12:00:00.000	2023-01-17 12:00:00.000
3	16	Outside	Mike	2023-02-15 12:00:00.000	2023-02-18 12:00:00.000
4	19	Outside	Dave	2023-02-10 12:00:00.000	2023-02-15 12:00:00.000
5	23	Outside	Baihaqi	2023-03-12 12:00:00.000	2023-03-16 12:00:00.000

10. Query

```
-- show a book borrowed on  a specific date
SELECT l.LoanID, b.Title, u.UserName, l.LoanDate, l.ReturnDate
FROM Loans AS l
JOIN Books AS b ON l.BookID = b.BookID
JOIN Users AS u ON l.UserID = u.UserID
WHERE b.BookID = 5 AND CONVERT(DATE, l.LoanDate) = '2023-01-14';
```

Result

Results | Messages

	LoanID	Title	UserName	LoanDate	ReturnDate
1	3	Outside	Elmira	2023-01-14 12:00:00.000	2023-01-19 12:00:00.000

11. Query

```
-- longest borrowed book
SELECT b.BookID, b.Title, DATEDIFF(DAY, l.LoanDate, l.ReturnDate) AS BorrowingLength
FROM Books AS b
JOIN Loans AS l ON b.BookID = l.BookID
ORDER BY BorrowingLength DESC;
```

Result

Results Messages

	BookID	Title	BorrowingLength
1	5	Outside	6
2	2	How To Not Overspent	5
3	5	Outside	5
4	1	My Mom Is My Cousin From Alabama	5
5	9	Letter in My Locker	5
6	1	My Mom Is My Cousin From Alabama	5
7	4	Need For Fuel	5
8	6	My ABCs	5
9	5	Outside	5
10	10	Big Dragon In Nut Kingdom	5
11	9	Letter in My Locker	5
12	8	Beans and Toast	5
13	4	Need For Fuel	4
14	5	Outside	4
15	8	Beans and Toast	4
16	9	Letter in My Locker	4
17	2	How To Not Overspent	4

Query executed successfully.

12. Query

```
-- Monthly book borrowing in the last 1 year
SELECT * FROM (
    SELECT COUNT(l.LoanID) AS loan_amount, DATENAME(MONTH, l.LoanDate) AS Mon
    FROM Loans AS l
    WHERE YEAR(l.LoanDate) = 2023
    GROUP BY DATENAME(MONTH, l.LoanDate)
) AS MonthlySum
PIVOT (
    SUM(loan_amount)
    FOR Mon IN (January, February, March, April, May, June, July, August, September, October, November, December)
) AS MonthlyPivot
```

Result

Results Messages

	January	February	March	April	May	June	July	August	September	October	November	December
1	10	10	5	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL