**AI ASSISTED CODING**

NAME: GUGGILLA ANUJA

ROLL NO : 2403A51101

BATCH : 06

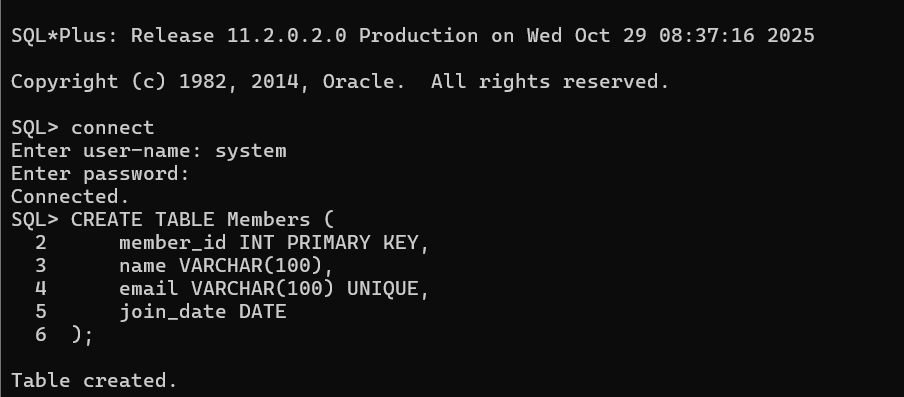
ASSIGNMENT : 16

#TASK-1

PROMPT :

Design a schema for a Library Management System  
(Tables: Books, Members, Loans).

CODE :



A screen shot of a computer

AI-generated content may be incorrect.

A computer screen with white text

AI-generated content may be incorrect.

OBSERVATION :

 loan\_id: Unique ID for each loan transaction.

 member\_id: References the borrowing member.

 book\_id: References the borrowed book.

 loan\_date: The date the book was borrowed.

 return\_date: The date the book is (or should be) returned.

 **Foreign keys** ensure referential integrity — a loan must be linked to an existing member and book.

#TASK – 2

PROMPT :

Generate INSERT INTO queries for the schema above  
(3 sample records per table).

CODE:

A black screen with white text

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

OBSERVATION :

 Proper table relationship (foreign key integrity) was maintained throughout.

 Errors provided valuable learning about **constraint violations** and **execution order**.

#TASK -3

PROMPT:

Generate a query to list all books borrowed by a specific  
member

CODE:

A computer screen with white text

AI-generated content may be incorrect.

OBSERVATION :

1. Used **JOIN operations** between Members, Books, and Loans to fetch related data.
2. Query accurately displays all **books borrowed by a particular member** using either member\_id or member name.
3. Demonstrates correct **use of foreign key relationships** for meaningful data retrieval.
4. Output confirms the logical link between tables works properly.

#TASK – 4

PROMPT :

Generate queries with AI for:  
• Updating a book’s availability to FALSE when borrowed.  
• Deleting a member record safely.

CODE :

A computer screen with white text

AI-generated content may be incorrect.

OBSERVATION :

1. **Update Query** correctly changes a book’s status from 'Y' to 'N' to mark it unavailable.
2. **Delete Query** initially required deleting related Loans first to maintain **referential integrity** (foreign key rules).
3. Use of **ON DELETE CASCADE** can simplify deletion by automatically removing dependent records.
4. Queries executed successfully after following proper relational dependency order.