LIBRARY WITH TIERED MEMBERSHIPS AND RESERVATION

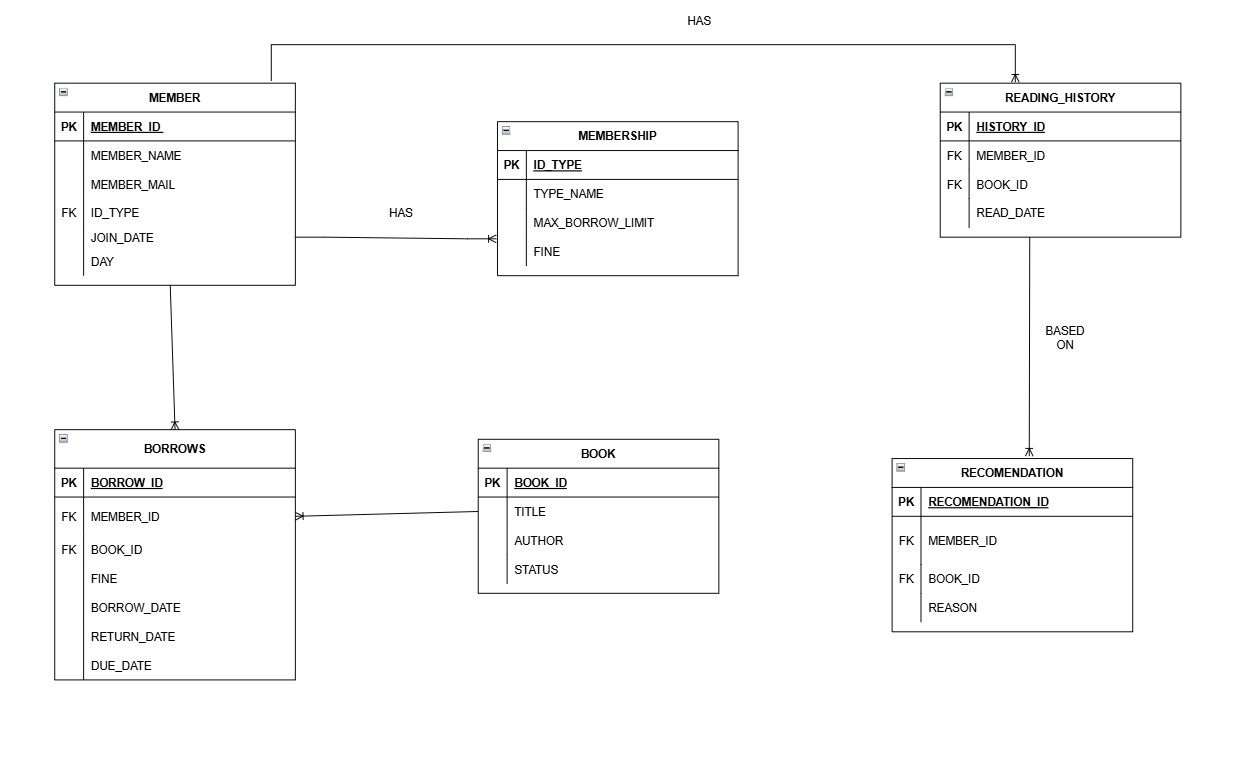
ENTITIES AND ATTRIBUTES

1. **Member** 
   * **Member\_ID** *(PK)*
   * Name
   * Type\_ID *(FK)*
   * Email
   * Join\_Date
2. **Membership** 
   * **TYPE\_ID** *(PK)*
   * Membership\_type
   * Reservation\_Duration
   * Max\_borrow\_limit
   * Fine
3. **Books** 
   * **Book\_ID** *(PK)*
   * Title
   * Author
   * Status
4. **Borrows** 
   * **Borrow\_ID** *(PK)*
   * Member\_ID *(FK → Member.Member\_ID)*
   * Book\_ID *(FK → Books.Book\_ID)*
   * Fine
   * Borrow\_date
   * Due\_date
   * Return\_date
5. **Reservation** 
   * **Reservation\_ID** *(PK)*
   * Member\_ID *(FK → Member.Member\_ID)*
   * Book\_ID *(FK → Books.Book\_ID)*
   * Date
6. **Reading\_History** 
   * **History\_ID** *(PK)*
   * Member\_ID *(FK → Member.Member\_ID)*
   * Book\_ID *(FK → Books.Book\_ID)*
   * Read\_Date
7. **Recommendation** 
   * **Recommendation\_ID**
   * Member\_ID *(FK → Member.Member\_ID)*
   * Book\_ID *(FK → Books.Book\_ID)*
   * Reason

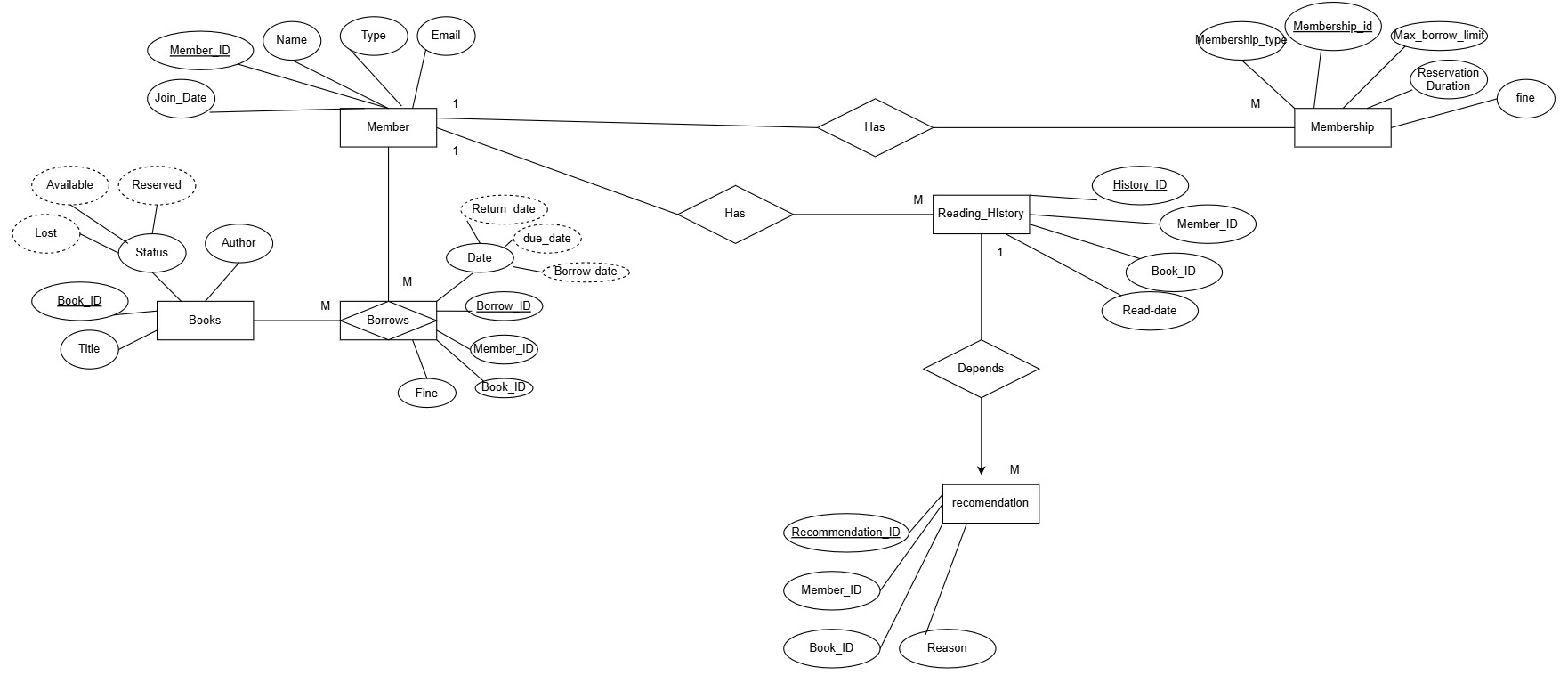
**RELATION:**

* + One member can have one membership.
  + A member can borrow multiple books.
  + A book can be reserved or borrowed.
  + Borrowing includes tracking dates and fines.

# ERD\_DIAGRAM CROW FEET



ERD\_DIAGRAM CHEN MODEL



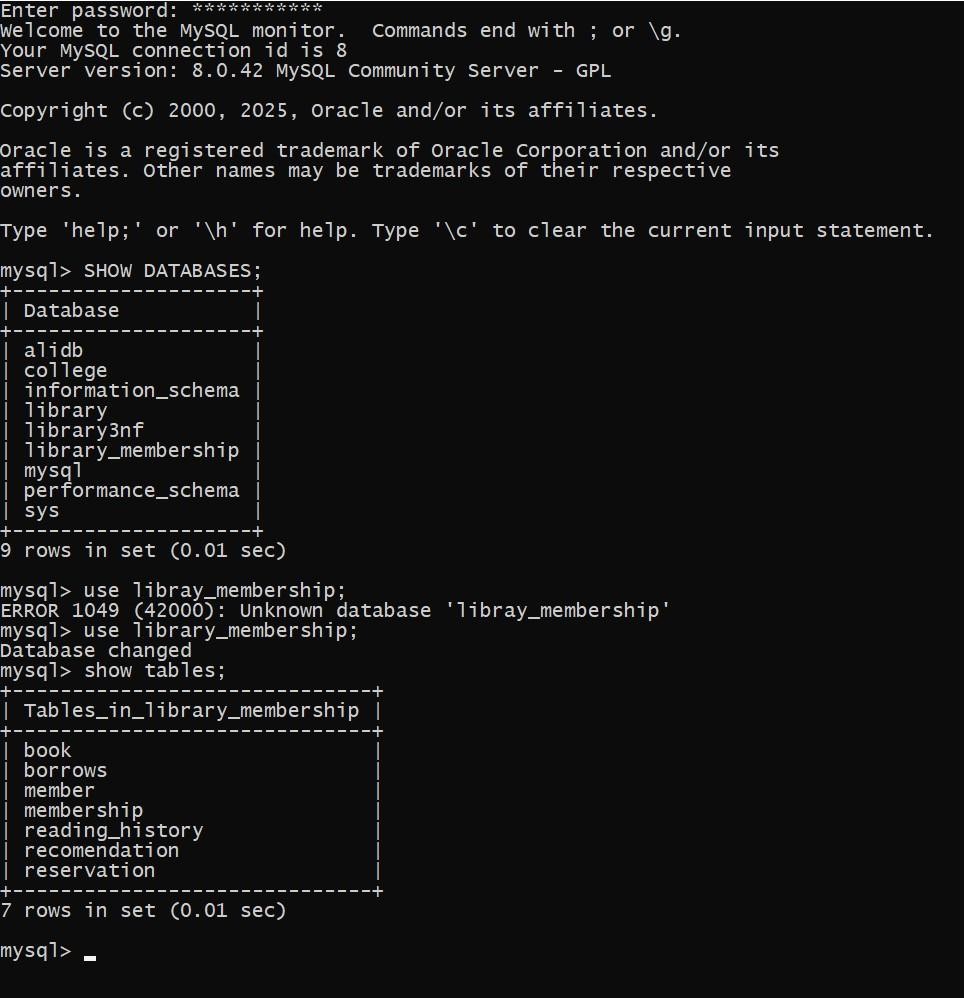
DATABASE TABLES AND QUERIES IN MYSQL:

IMPORTANT COMMANDS:

**SHOW DATABASES;**

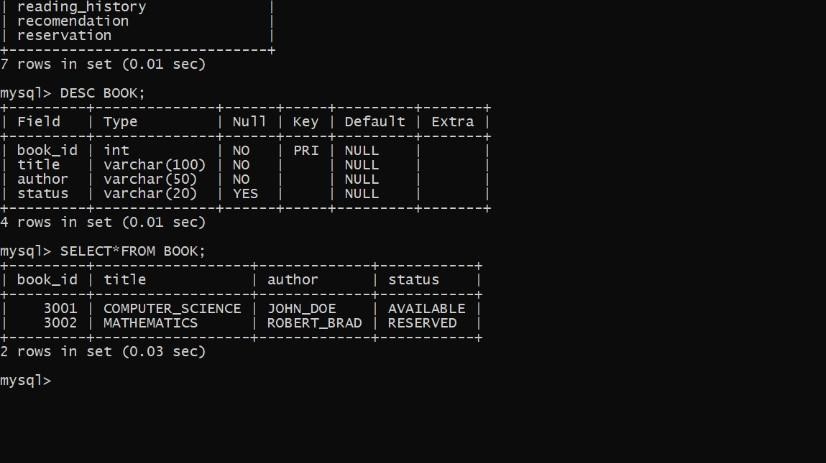
**USE LIBRARY\_MEMBERSHIP;**

**SHOW TABLES;**



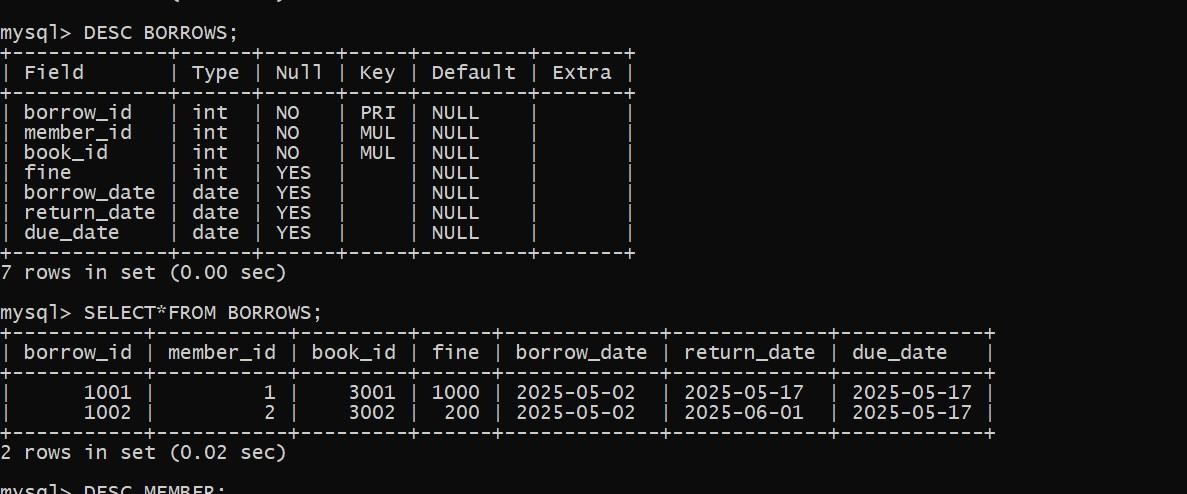
**DESC BOOK;**

**SELECT\*FROM BOOK;**



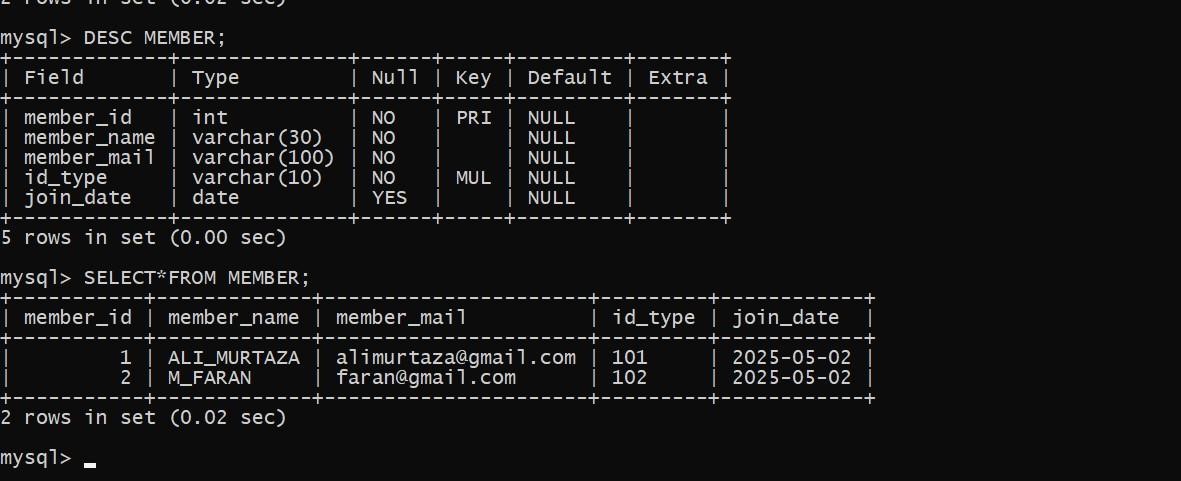
**DESC BORROWS;**

**SELECT\*FROM BORROWS;**



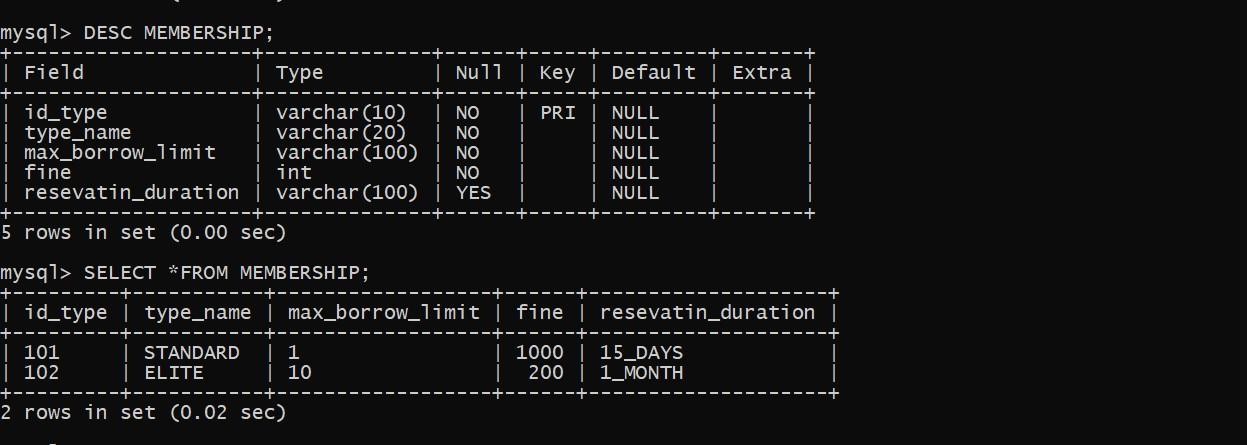
**DESC MEMBER;**

**SELECT\*FROM MEMBER;**



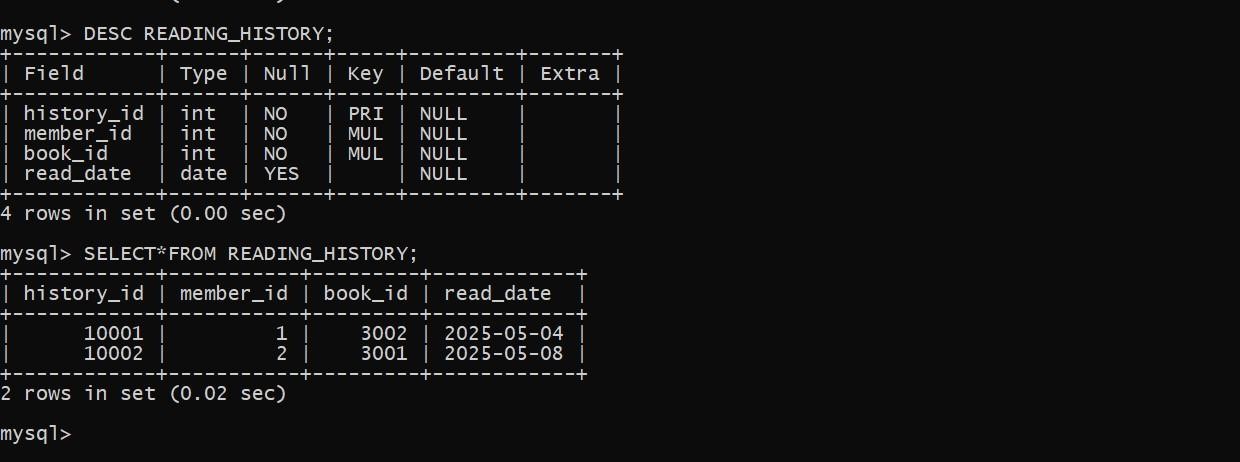
**DESC MEMBERSHIP;**

**SELECT\*FROM MEMBERSHIP;**



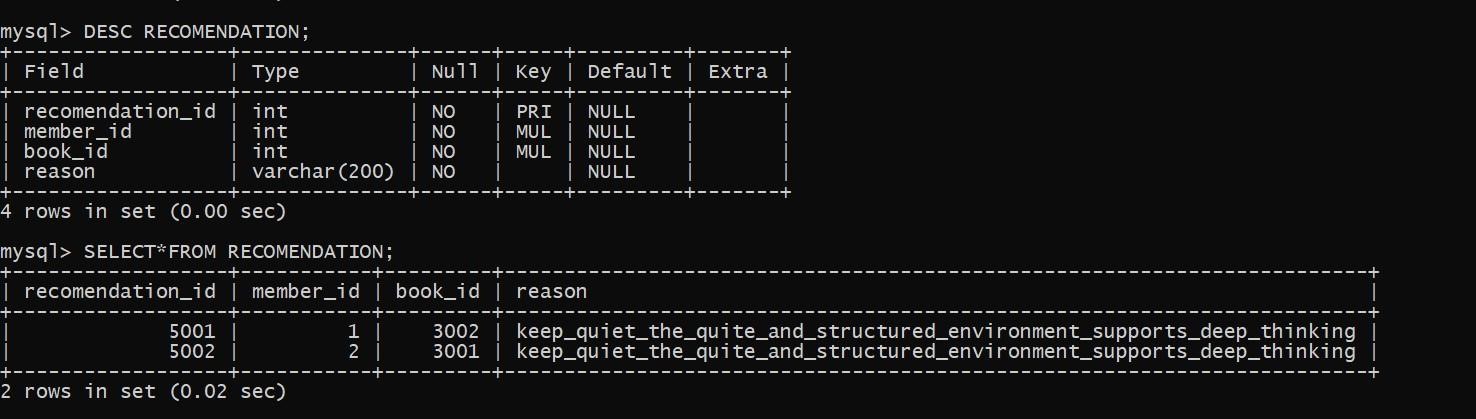
**DESC READING\_HISTORY;**

**SELECT\*FROM READING\_HISTORY;**



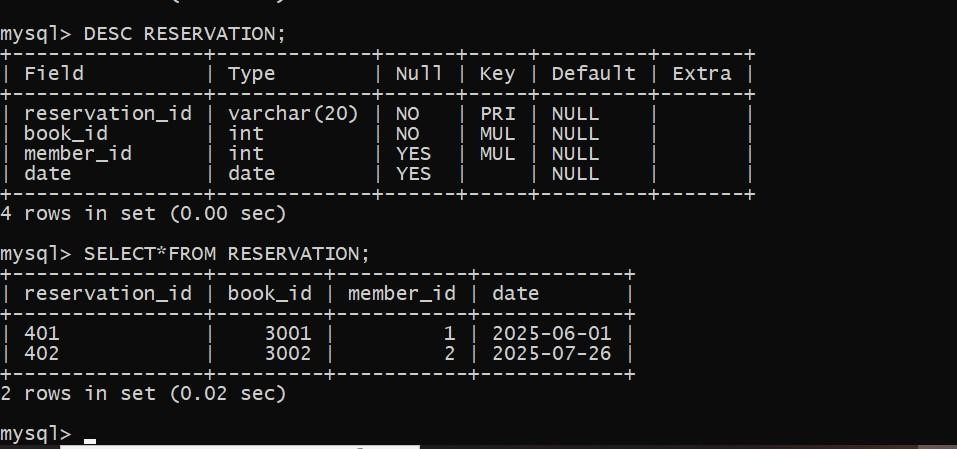
**DESC RECOMENDATION;**

**SELECT \*FROM RECOMENDATION;**



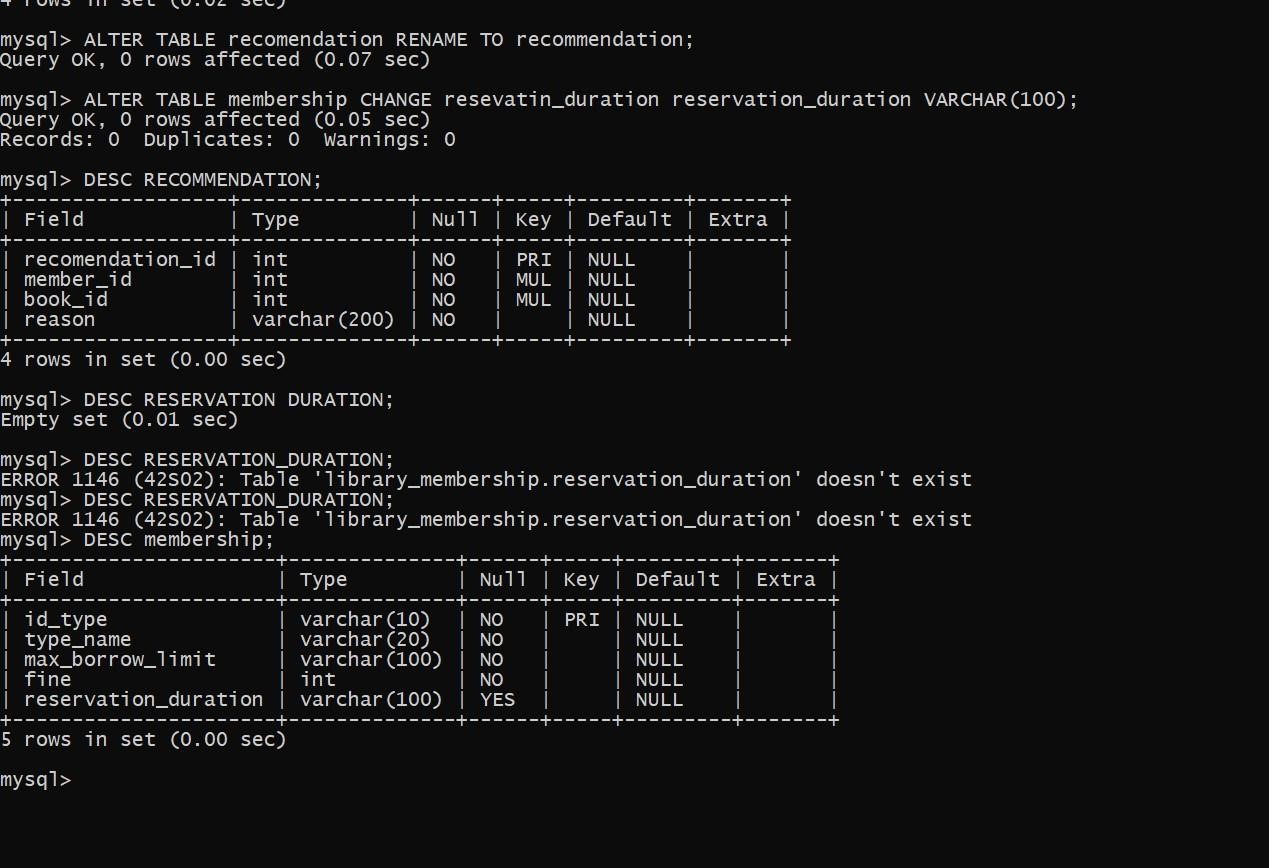
**DESC RESERVATION;**

**SELECT\*FROM RESERVATION;**

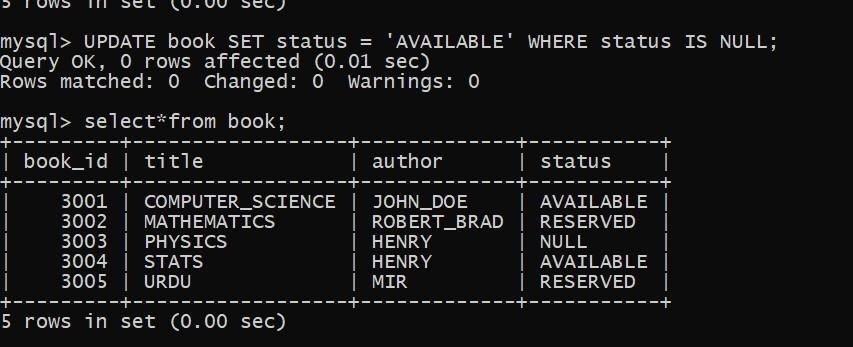


**QUERIES:**

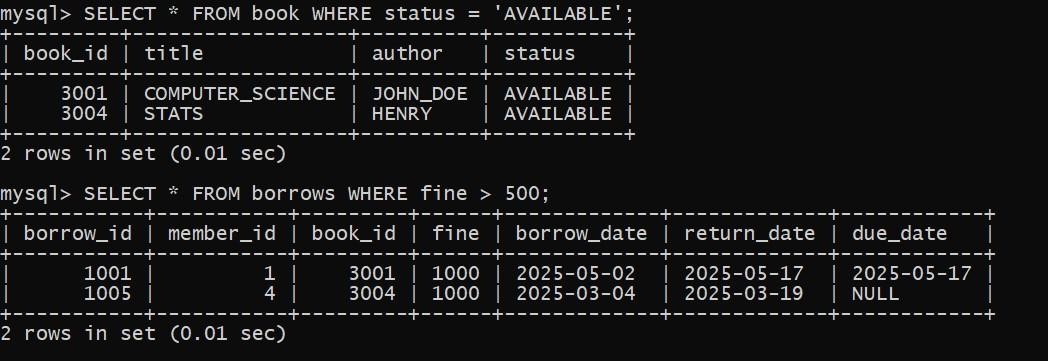
**Fix incorrect column:**



**Data cleaning:**



**Where clause:**



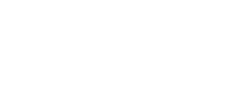
**A**



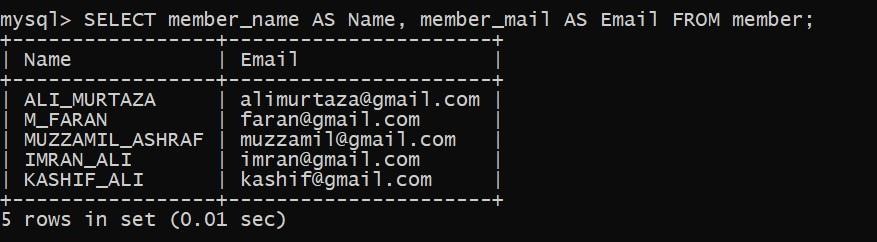
**s**



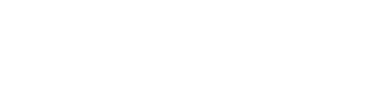
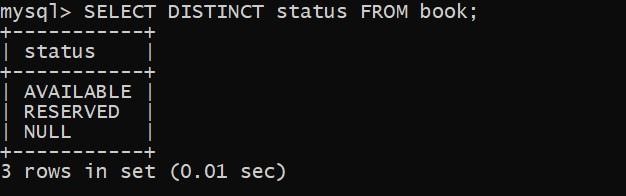
**Q**



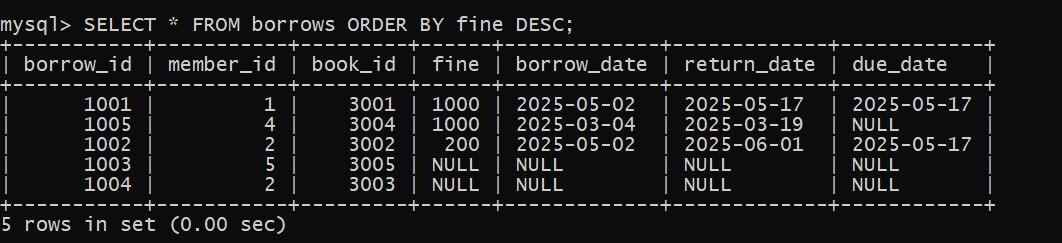
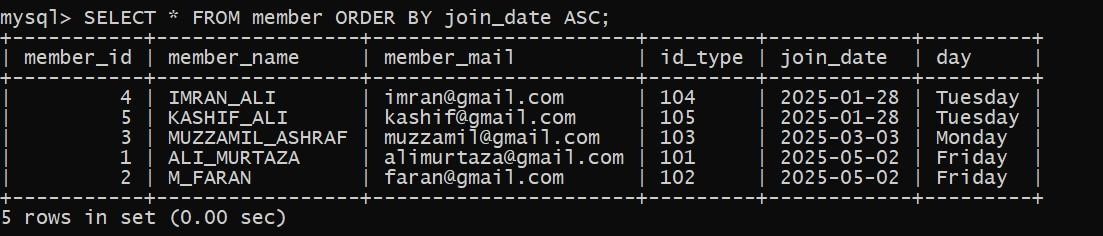
**uerie:**



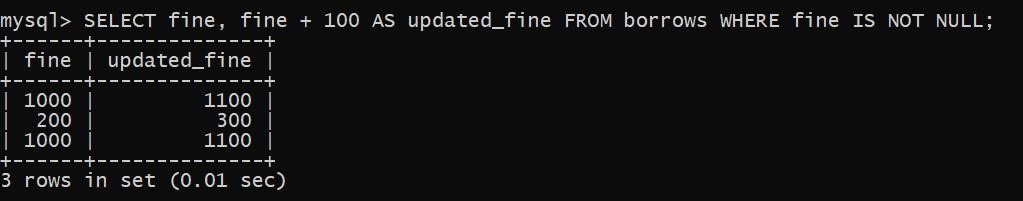
**Distinct:**



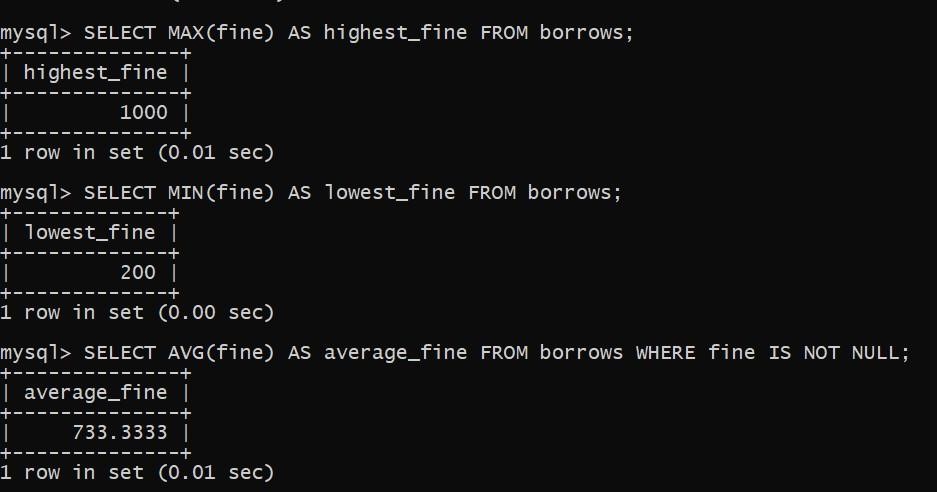
**Order by:**



**Arithmetic operations:**

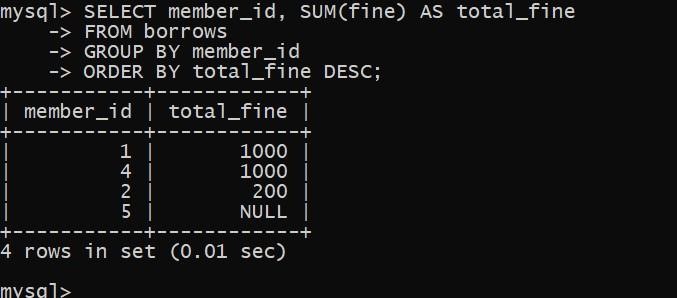


**Aggregate functions:**



**Using Functions with Aliases and ORDER**

**BY:**



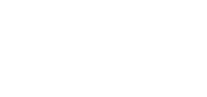
Joins:

Inner join:

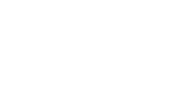
Show Member Name, Book id, Date, Member\_id:



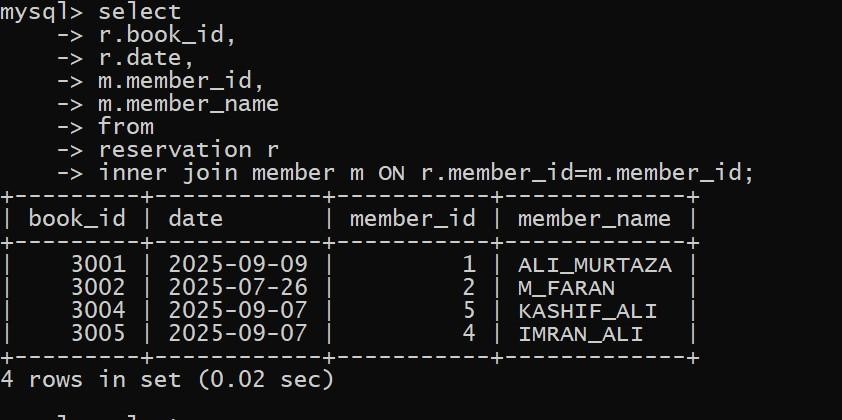
I



nner



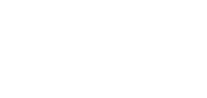
join:



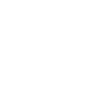
Show Member Name, Book id, Date, Member\_id, with WHERE Clause.



I



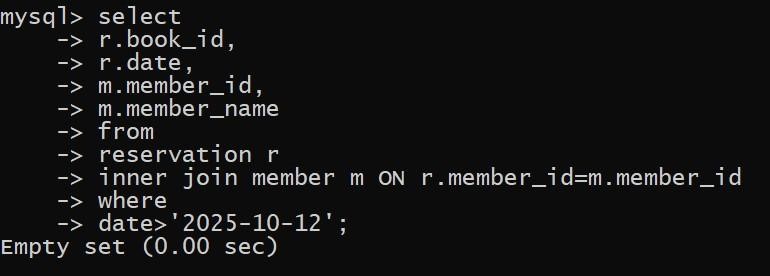
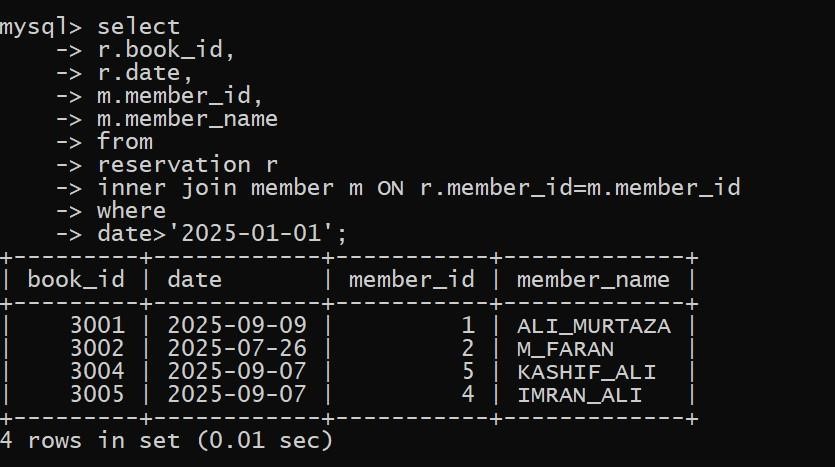
nner



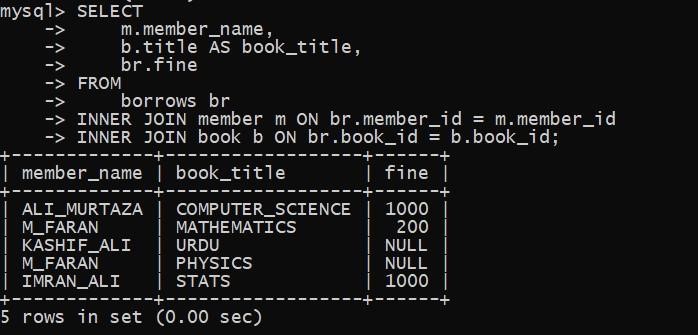
joi



n:



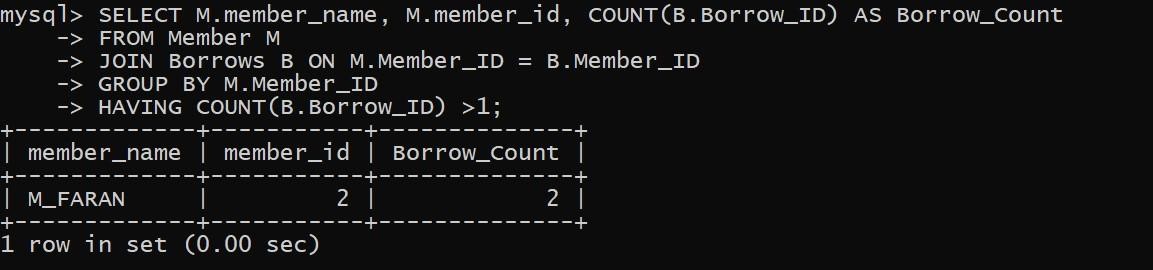
Member details with book title and fine from member, book, borrows.



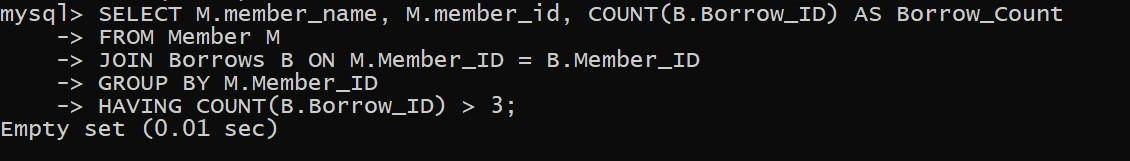
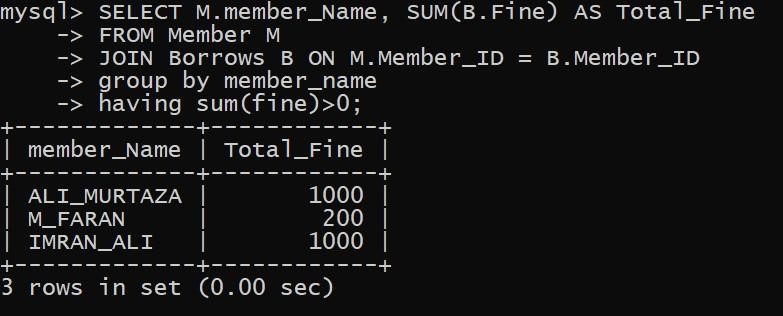
Group by and having with Joins:

Members who borrowed more than 1 book:

Members who’s total fine is >0:



Members who borrowed more than 3 books:



**Denormalization:**

1. Combine Tables (Joining Tables**)**

Avoid runtime JOINs by embedding related table info directly into borrows.

current borrows table already includes:

* book\_title (from book)
* membership\_type (from membership)
* Member\_name (from member)

## 2. Precompute Derived Data (Storing Aggregates)

**Example Additions**:  
You could add a column like:

sql

CopyEdit

ALTER TABLE borrows ADD COLUMN borrow\_duration INT;

Then calculate borrow\_duration as:

sql

CopyEdit

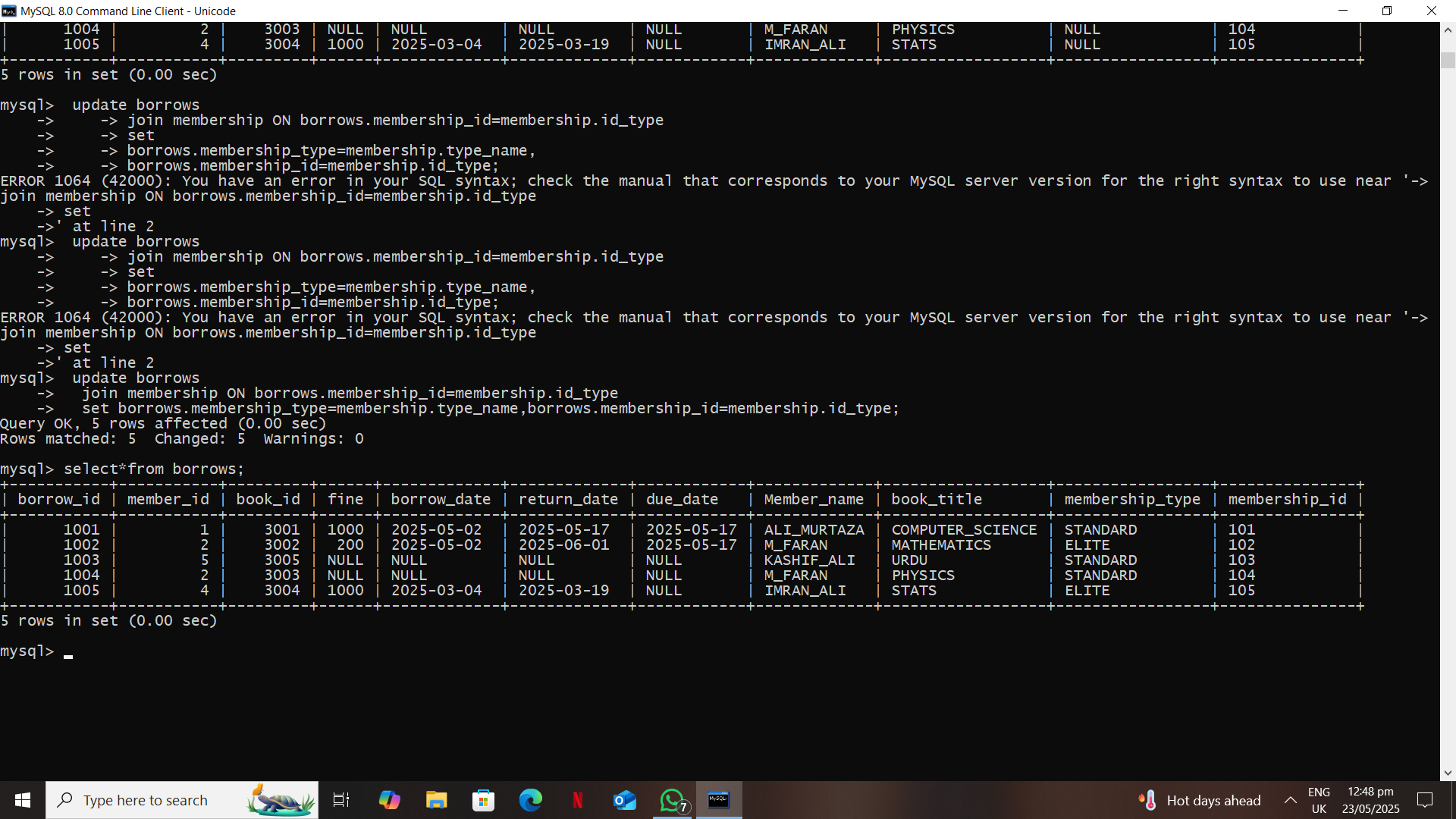
UPDATE borrows

SET borrow\_duration = DATEDIFF(return\_date, borrow\_date)

WHERE return\_date IS NOT NULL AND borrow\_date IS NOT NULL;

**Purpose**: Speeds up reports like average borrow duration per member.

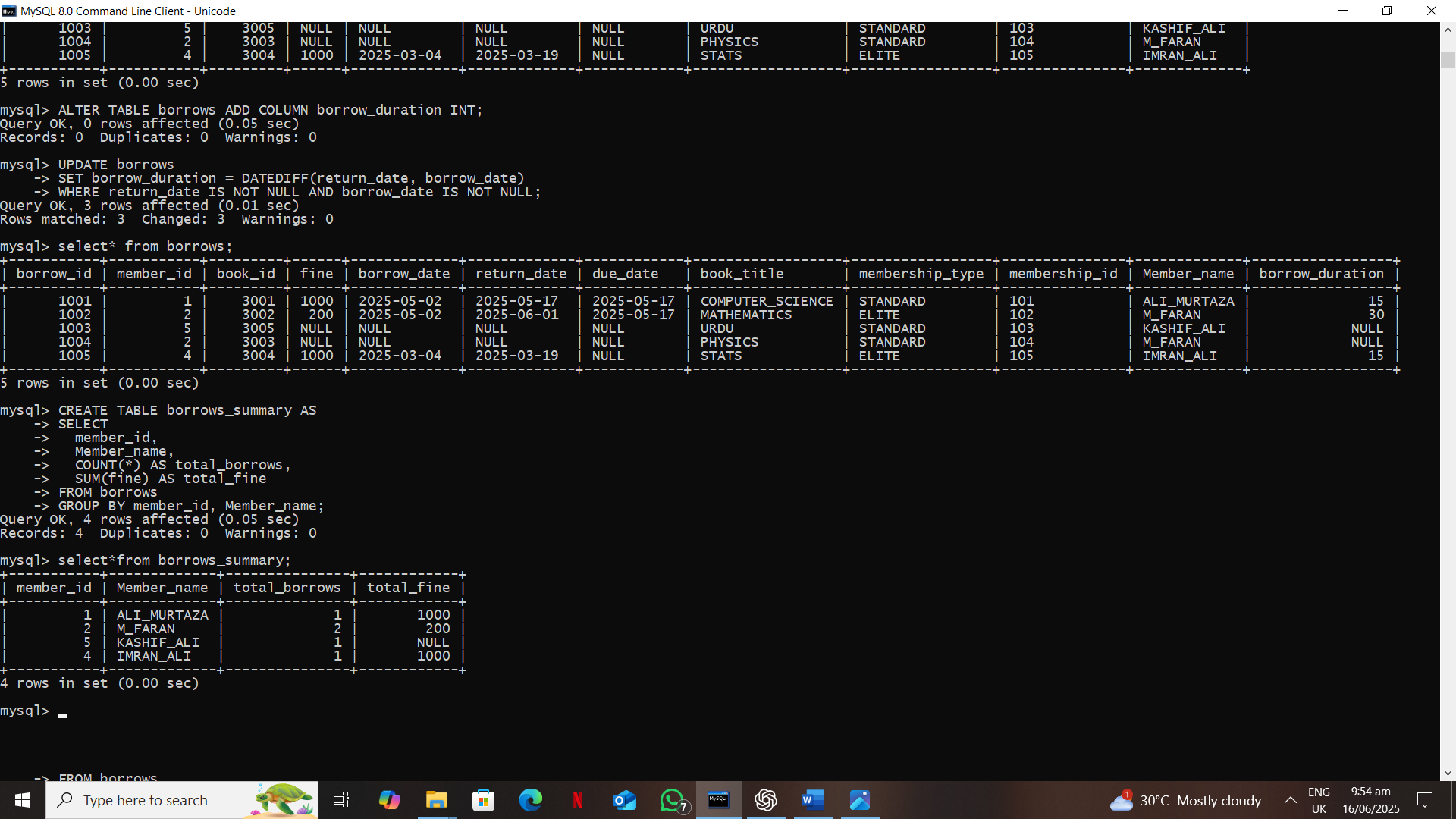
✅ **3. Add Redundant Data (Copy from Parent Tables)**

****

**4. Use Data Replication (Across Systems or Tables):**

This is generally handled **outside MySQL** (e.g., cache or reporting DB), but for **in-table replication**, you can do:

* Create a summary or reporting table (e.g., borrows\_summary) with key redundant info:



**5. Introduce Summary Tables (Materialized Views):**

Create summary tables for faster reporting: