Database Systems Project

Library Management System Database Design

By: Shahd Anwar : 211002472
Shahd Ahmed : 211000410
Salma Khaled : 211003394

Description

The objective of this project is to design a database for a Library Management System that manages library operations, including tracking books, authors, members, loans, staff, categories, fine, fine payments and reservations. This database organizes the processes of book borrowing and returning and overall library administration.

Business Rules

Books and loans
-Only active members can borrow books.
-Each book can be loaned out to one member at a time.
-Each book must have a quantity of copies available for lending.
- Book cannot be loaned if it's checked out.
-Every book in the system must have a unique BookID for tracking purposes.
-A member can borrow up to 3 books at one time.
-A loan record has a set duration of 14 days. After this period, the book must be returned, or a fine will be imposed.

Reservations
-A member can reserve a book that is currently unavailable (already on loan).
- Each reservation is associated with a single book.
-Reserved books must be picked up within 3 days of reservation or reservation in cancelled.

Fines
-Fines are imposed for late returns; if a member does not return a book by the due date.
-Members cannot borrow more books until fines are paid.

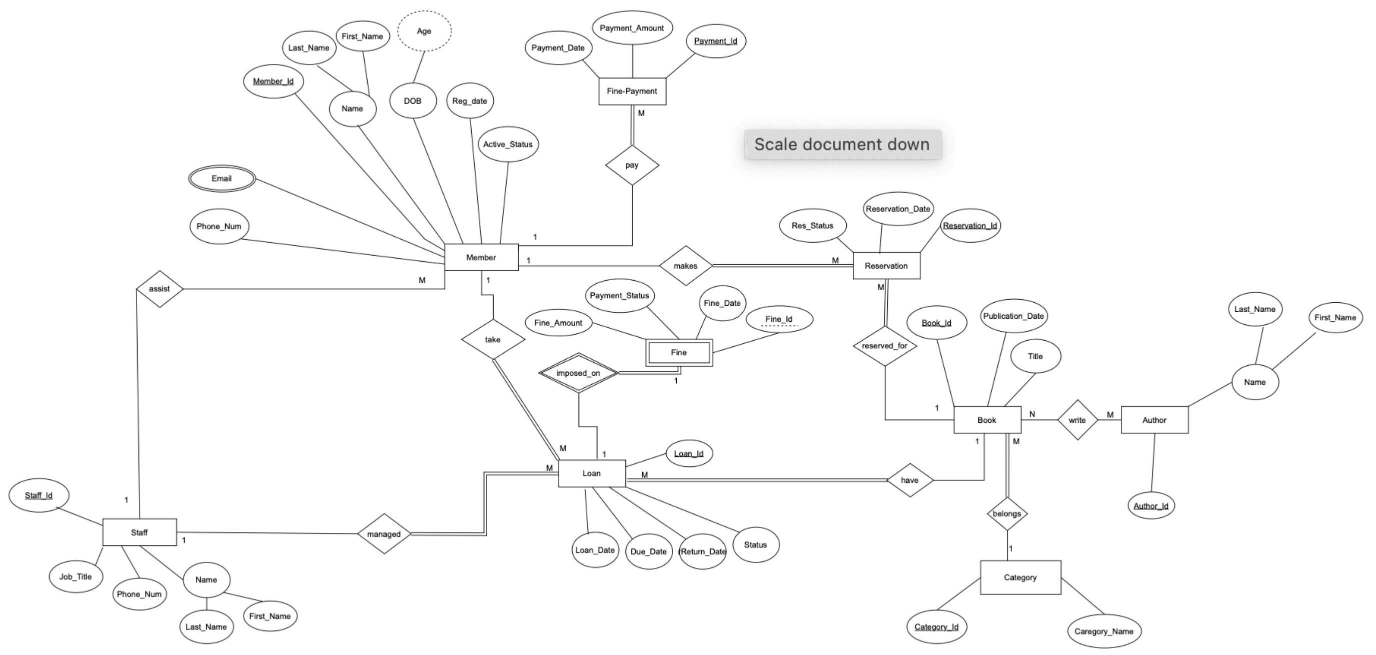
Staff and Members
-Members are identified by a unique id and must provide full contact details to obtain membership.
-Only authorized staff members can assist members and each library member is assigned a single staff member for administrative help.
-A loan record is created by a staff member when issuing a book to a member.

Book Categories and Authors
-Each book must be assigned to a category.
-A book can have multiple authors.

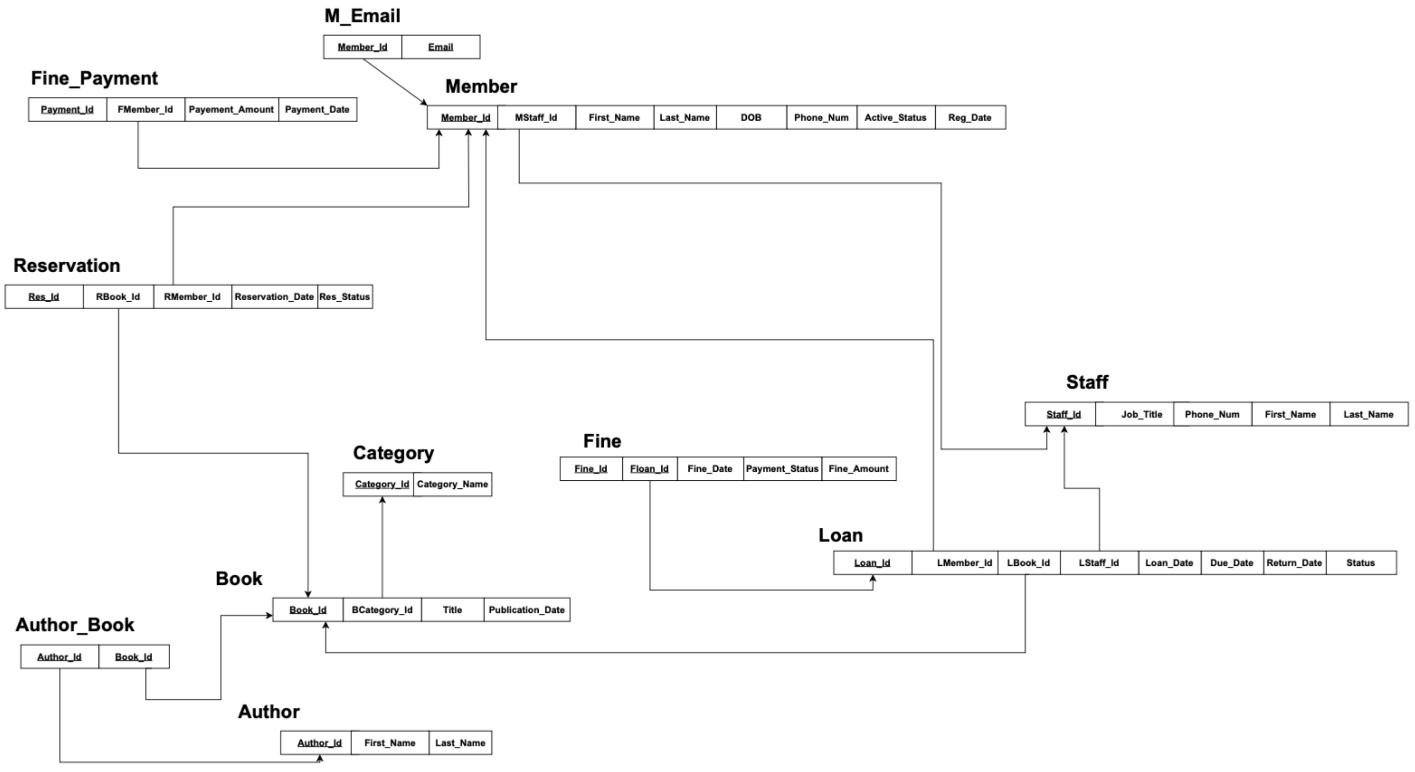
Integrity
-A book can only be deleted from the catalog if it is no longer loaned out or reserved by any member.

-If a book is removed from the library catalog, all related records (such as reservations) should be updated or deleted to maintain referential integrity.

ERD



Relational Model


M_Email
Member_Id | Email
Fine_Payment
Payment_Id | FMember_Id | Payement_Amount | Payment_Date
Member
Member_Id | MStaff_Id | First_Name | Last_Name | DOB | Phone_Num | Active_Status | Reg_Date
Reservation
Res_Id | RBook_Id | RMember_Id | Reservation_Date | Res_Status
Staff
Staff_Id | Job_Title | Phone_Num | First_Name | Last_Name
Category
Category_Id | Category_Name
Fine
Fine_Id | Floan_Id | Fine_Date | Payment_Status | Fine_Amount
Loan
Loan_Id | LMember_Id | LBook_Id | LStaff_Id | Loan_Date | Due_Date | Return_Date | Status
Book
Book_Id | BCategory_Id | Title | Publication_Date
Author_Book
Author_Id | Book_Id
Author
Author_Id | First_Name | Last_Name

Code for tables creation:

```
CREATE TABLE Author (
   Author_Id INT PRIMARY KEY,
   First_Name VARCHAR(50) NOT NULL,
   Last_Name VARCHAR(50) NOT NULL
);



CREATE TABLE Category (
   Category_Id INT PRIMARY KEY,
   Category_Name VARCHAR(100) NOT NULL UNIQUE
);


CREATE TABLE Book (
   Book_Id INT PRIMARY KEY,
   Category_Id INT NOT NULL,
   Title VARCHAR(200) NOT NULL,
   Publication_Date DATE,
   FOREIGN KEY (Category_Id) REFERENCES Category(Category_Id)
      ON UPDATE CASCADE
      ON DELETE RESTRICT
);


CREATE TABLE Author_Book (
   Author_Id INT,
   Book_Id INT,
   PRIMARY KEY (Author_Id, Book_Id),
   FOREIGN KEY (Author_Id) REFERENCES Author(Author_Id)
      ON UPDATE CASCADE
      ON DELETE CASCADE,
   FOREIGN KEY (Book_Id) REFERENCES Book(Book_Id)
      ON UPDATE CASCADE
      ON DELETE CASCADE
);
```

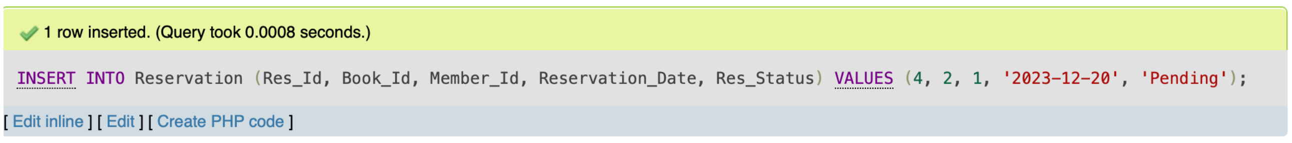
```
CREATE TABLE Staff (
   Staff_Id INT PRIMARY KEY,
   Job_Title VARCHAR(100) NOT NULL,
   Phone_Num VARCHAR(20),
   First_Name VARCHAR(50) NOT NULL,
   Last_Name VARCHAR(50) NOT NULL
);


CREATE TABLE Member (
   Member_Id INT PRIMARY KEY,
   Staff_Id INT,
   First_Name VARCHAR(50) NOT NULL,
   Last_Name VARCHAR(50) NOT NULL,
   DOB DATE,
   Phone_Num VARCHAR(20),
   Active_Status BOOLEAN DEFAULT TRUE,
   Reg_Date DATE NOT NULL,
   FOREIGN KEY (Staff_Id) REFERENCES Staff(Staff_Id)
      ON UPDATE CASCADE
      ON DELETE SET NULL
);


CREATE TABLE M_Email (
   Member_Id INT,
   Email VARCHAR(100),
   PRIMARY KEY (Member_Id, Email),
   FOREIGN KEY (Member_Id) REFERENCES Member(Member_Id)
      ON UPDATE CASCADE
      ON DELETE CASCADE
);
```

```sql
CREATE TABLE Reservation (
   Res_Id INT PRIMARY KEY,
   Book_Id INT NOT NULL,
   Member_Id INT NOT NULL,
   Reservation_Date DATE NOT NULL,
   Res_Status VARCHAR(20) CHECK (Res_Status IN ('Pending', 'Confirmed', 'Cancel
   FOREIGN KEY (Book_Id) REFERENCES Book(Book_Id)
      ON UPDATE CASCADE
      ON DELETE RESTRICT,
   FOREIGN KEY (Member_Id) REFERENCES Member(Member_Id)
      ON UPDATE CASCADE
      ON DELETE CASCADE
);

CREATE TABLE Loan (
   Loan_Id INT PRIMARY KEY,
   Member_Id INT NOT NULL,
   Book_Id INT NOT NULL,
   Staff_Id INT NOT NULL,
   Loan_Date DATE NOT NULL,
   Due_Date DATE NOT NULL CHECK (Due_Date > Loan_Date),
   Return_Date DATE CHECK (Return_Date >= Loan_Date),
   Status VARCHAR(20) CHECK (Status IN ('Active', 'Returned', 'Overdue')),
   FOREIGN KEY (Member_Id) REFERENCES Member(Member_Id)
      ON UPDATE CASCADE
      ON DELETE RESTRICT,
   FOREIGN KEY (Book_Id) REFERENCES Book(Book_Id)
      ON UPDATE CASCADE
      ON DELETE RESTRICT,
   FOREIGN KEY (Staff_Id) REFERENCES Staff(Staff_Id)
      ON UPDATE CASCADE
      ON DELETE RESTRICT
);


 CREATE TABLE Fine (
    Fine_Id INT,
    Loan_Id INT,
    Fine_Date DATE NOT NULL,
    Payment_Status VARCHAR(20) CHECK (Payment_Status IN ('Pending', 'Paid', 'Overdue')),
    Fine_Amount DECIMAL(10,2) CHECK (Fine_Amount > 0),
    PRIMARY KEY (Fine_Id, Loan_Id),
    FOREIGN KEY (Loan_Id) REFERENCES Loan(Loan_Id)
       ON UPDATE CASCADE
       ON DELETE RESTRICT
 );


 CREATE TABLE Fine_Payment (
    Payment_Id INT PRIMARY KEY,
    Member_Id INT NOT NULL,
    Payment_Amount DECIMAL(10,2) NOT NULL CHECK (Payment_Amount > 0),
    Payment_Date DATE NOT NULL,
    FOREIGN KEY (Member_Id) REFERENCES Member(Member_Id)
       ON UPDATE CASCADE
       ON DELETE RESTRICT
 );
```

Code for Inserts:

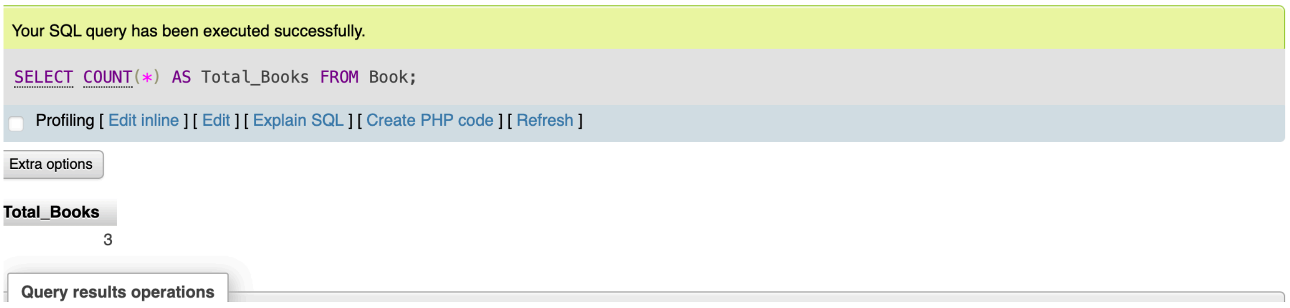
```
INSERT INTO Author (Author_Id, First_Name, Last_Name)
VALUES (1, 'Jane', 'Austen'),(2,'Fyodor','Dostoevsky'),(3,'William','Shakespeare');

INSERT INTO Category (Category_Id, Category_Name)
VALUES (1, 'Fiction'),(2, 'Non Fiction'),(3, 'Romance');

INSERT INTO Book (Book_Id, Category_Id, Title, Publication_Date)
VALUES (1, 1, 'Pride and Prejudice', '1813-01-28'),(2, 1, '1984', '1949-06-08'),(3, 3, 'The Adventures of Tom Sawyer', '1876-06-01');


INSERT INTO Author_Book (Author_Id, Book_Id)
VALUES (1, 3),(2,1),(3,2);

INSERT INTO Staff (Staff_Id, Job_Title, Phone_Num, First_Name, Last_Name)
VALUES (1, 'Librarian', '123-456-7890', 'Emily', 'Smith'),(2, 'Assistant Librarian', '321-654-0987', 'Sarah', 'Brown'),(3, 'Library Clerk', '654-321-8765', 'Michael', 'Johnson');

INSERT INTO Member (Member_Id, Staff_Id, First_Name, Last_Name, DOB, Phone_Num, Active_Status, Reg_Date)
VALUES (1, 1, 'Salma', 'Khaled', '1990-05-15', '987-654-3210', TRUE, '2023-01-01'),(2, 2, 'Shahd', 'Ahmed', '1995-08-20', '123-123-1234', TRUE, '2023-02-01'),(3, 3, 'Shahd',
'Anwar', '1985-03-15', '456-456-4567', FALSE, '2023-03-01');

INSERT INTO M_Email (Member_Id, Email)
VALUES (1, 'salmakhaled@gmail.com.com'), (2, 'shahdahmed@yahoo.com'),(3, 'shahdanwar@gmail.com');


INSERT INTO Reservation (Res_Id, Book_Id, Member_Id, Reservation_Date, Res_Status)
VALUES (1, 1, 1, '2023-12-01', 'Pending'),(2, 2, 2, '2023-12-10', 'Confirmed'),(3, 3, 3, '2023-12-15', 'Cancelled');


INSERT INTO Loan (Loan_Id, Member_Id, Book_Id, Staff_Id, Loan_Date, Due_Date, Return_Date, Status)
VALUES (1, 1, 1, 1, '2023-11-01', '2023-11-15', NULL, 'Active'),(2, 2, 2, 2, '2023-11-05', '2023-11-20', '2023-11-19', 'Returned'),(3, 3, 3, 3, '2023-11-10', '2023-11-25', NULL,
'Active');

INSERT INTO Fine (Fine_Id, Loan_Id, Fine_Date, Payment_Status, Fine_Amount)
VALUES (1, 1, '2023-11-20', 'Pending', 50.00),(2, 2, '2023-11-21', 'Paid', 20.00),(3, 3, '2023-11-26', 'Overdue', 75.00);

INSERT INTO Fine_Payment (Payment_Id, Member_Id, Payment_Amount, Payment_Date)
VALUES (1, 1, 50.00, '2023-11-25'),(2, 2, 20.00, '2023-11-22'),(3, 3, 75.00, '2023-11-27');
```

Sample Queries:



1 row inserted. (Query took 0.0008 seconds.)

```sql
INSERT INTO Reservation (Res_Id, Book_Id, Member_Id, Reservation_Date, Res_Status) VALUES (4, 2, 1, '2023-12-20', 'Pending');
```

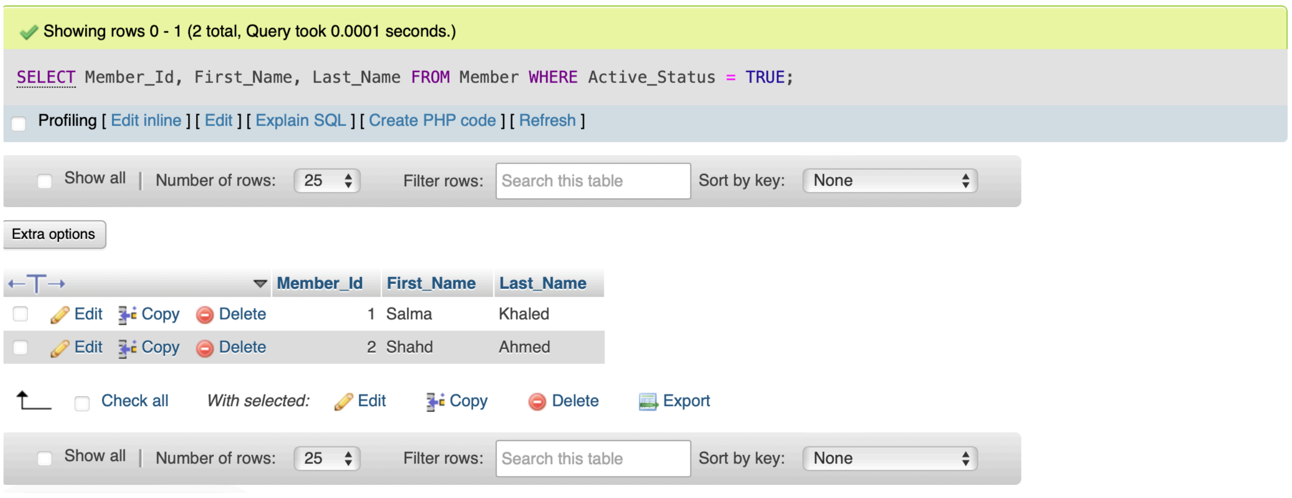


Your SQL query has been executed successfully.

```sql
SELECT COUNT(*) AS Total_Books FROM Book;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Extra options

Total_Books
3



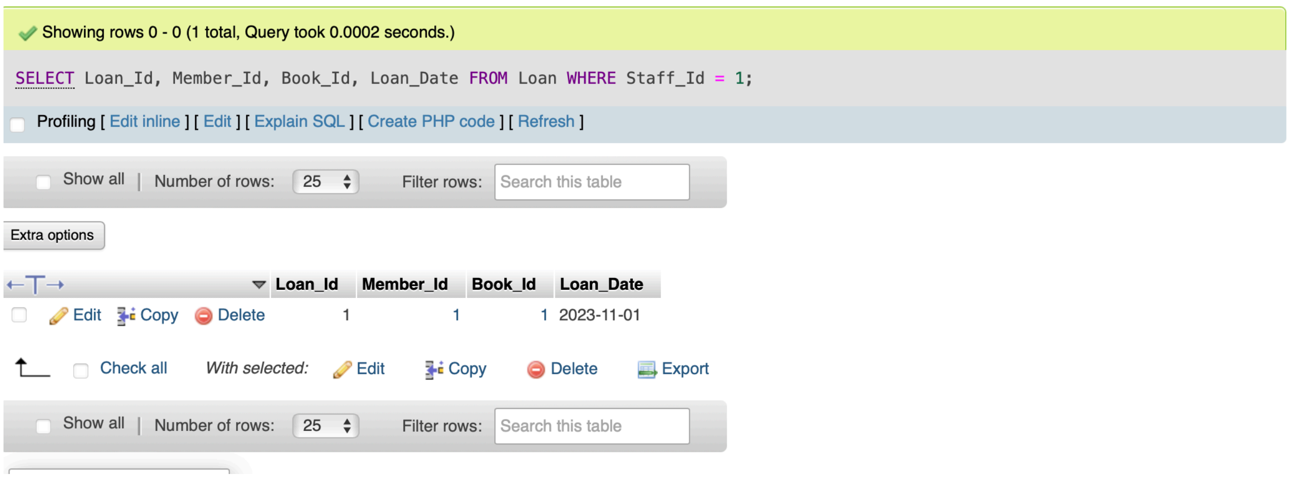
Showing rows 0 - 1 (2 total, Query took 0.0001 seconds.)

```sql
SELECT Member_Id, First_Name, Last_Name FROM Member WHERE Active_Status = TRUE;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options

	Member_Id	First_Name	Last_Name
Edit Copy Delete	1	Salma	Khaled
Edit Copy Delete	2	Shahd	Ahmed

Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None



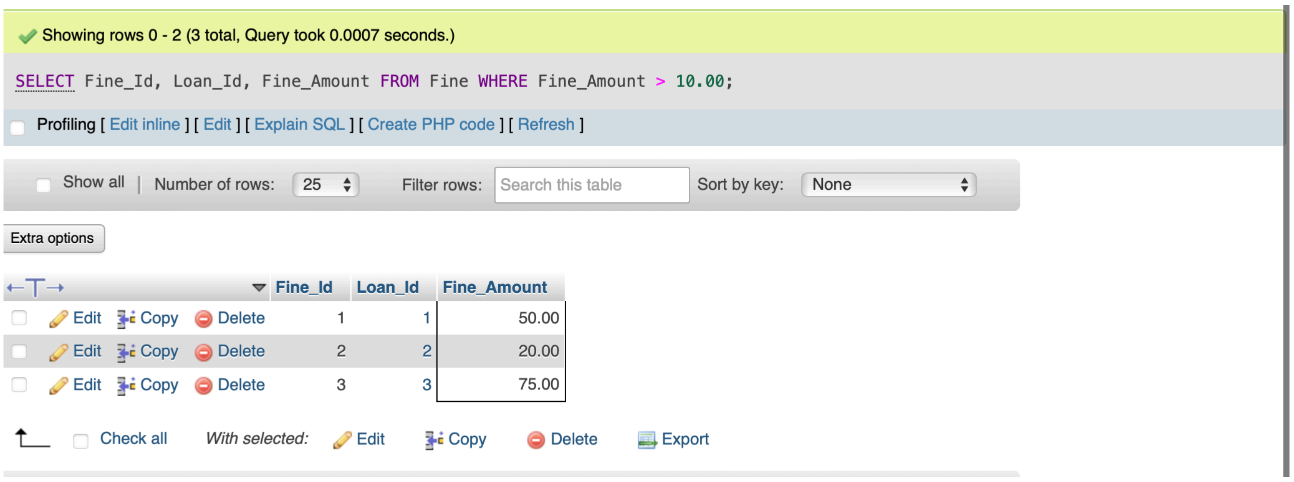
Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)

```sql
SELECT Loan_Id, Member_Id, Book_Id, Loan_Date FROM Loan WHERE Staff_Id = 1;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table
Extra options

	Loan_Id	Member_Id	Book_Id	Loan_Date
Edit Copy Delete	1	1	1	2023-11-01

Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table



✓ Showing rows 0 - 2 (3 total, Query took 0.0007 seconds.)

```sql
SELECT Fine_Id, Loan_Id, Fine_Amount FROM Fine WHERE Fine_Amount > 10.00;
```

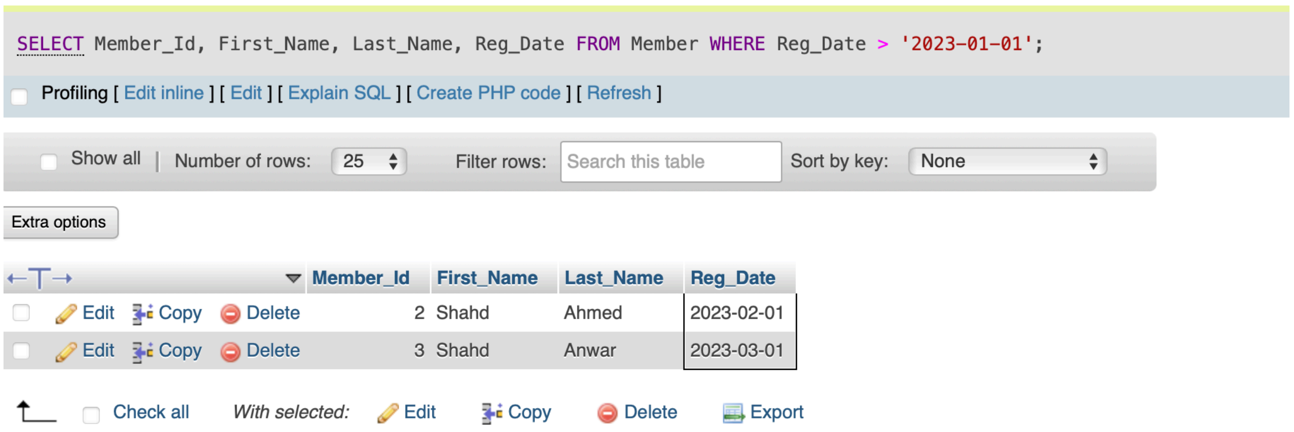
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	Fine_Id	Loan_Id	Fine_Amount
Edit Copy Delete	1	1	50.00
Edit Copy Delete	2	2	20.00
Edit Copy Delete	3	3	75.00

Check all *With selected:* Edit Copy Delete Export



```sql
SELECT Member_Id, First_Name, Last_Name, Reg_Date FROM Member WHERE Reg_Date > '2023-01-01';
```

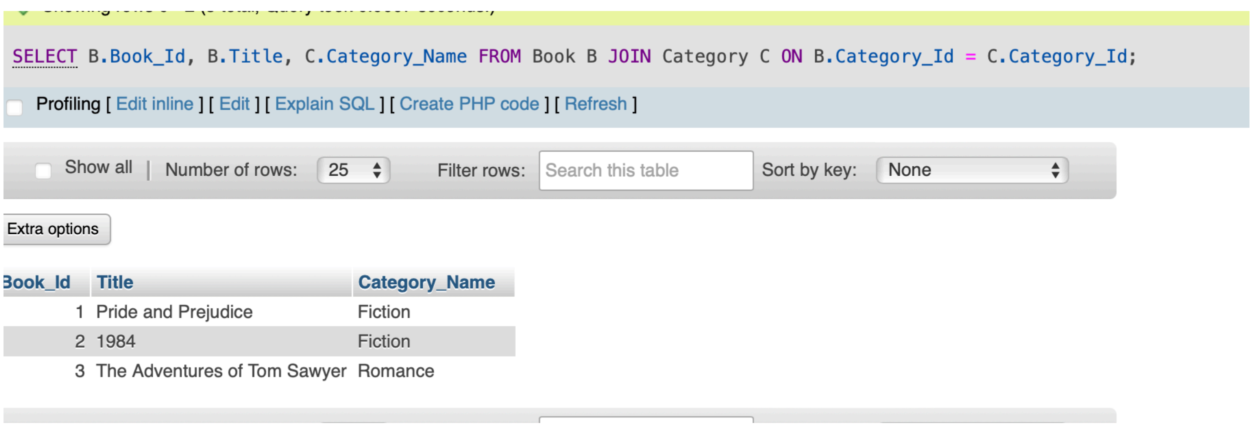
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	Member_Id	First_Name	Last_Name	Reg_Date
Edit Copy Delete	2	Shahd	Ahmed	2023-02-01
Edit Copy Delete	3	Shahd	Anwar	2023-03-01

Check all *With selected:* Edit Copy Delete Export



```sql
SELECT B.Book_Id, B.Title, C.Category_Name FROM Book B JOIN Category C ON B.Category_Id = C.Category_Id;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

Book_Id	Title	Category_Name
1	Pride and Prejudice	Fiction
2	1984	Fiction
3	The Adventures of Tom Sawyer	Romance