Library Management System

Entities Detail:

The database is designed for a **Library Management System**, encompassing essential entities for efficient library operations. **Authors** are recorded with their details like name, birthdate, nationality, and biography. **Borrowers** are identified by unique IDs and have personal information like name, address, contact details, membership type, and expiry date. **Genres** categorize books based on type, with additional attributes for language, rating, and description. **Publishers** are listed with their IDs, names, addresses, and contact information.

Books form the **core** of the system, each having a unique ID, title, publication year, and associated author, publisher, and genre IDs. **Fiction books** and **non-fiction books** are **specialized** types, each with unique identifiers and specific attributes like themes, summaries, subjects, and academic levels. **Transactions** track book borrowing, including details like transaction IDs, book IDs, borrower IDs, dates of issue, due dates, return dates, fine amounts, and branch IDs.

The system also manages **staff members** with IDs, names, positions, joining dates, salaries, contact details, and branch assignments. **Branches** represent the physical locations of the library, each with an ID, name, location, contact number, and manager ID.

Business rules:

Business rules dictate categorizing books into fiction and non-fiction genres, tracking borrower memberships and due dates to avoid fines, and efficiently managing staff responsibilities across different branches. Overall, the database ensures smooth library functions, from cataloging books to managing loans, fines, and staff duties.

Entities:

Here are the entities for LMS:

- Authors
- Borrowers
- Genres
- Publishers
- Books
- FictionBooks
- NonFictionBooks
- Transactions
- Staff
- Branches

Tables and Attributes:

Here are the tables and attributes for LMS:

✓ Authors:

- AuthorId (Primary Key)
- AuthorName
- BirthDate
- Nationality
- Biography

✓ Borrowers:

- BorrowerId (Primary Key)
- BorrowerName
- Address
- PhoneNo
- Email
- MembershipType
- MembershipExpiry

✓ Genres:

- GenreId (Primary Key)
- GenreType
- GenreLanguage
- Rating
- Description

✓ Publishers:

- PublisherId (Primary Key)
- PublisherName
- Address
- PhoneNo

✓ Books:

- BookId (Primary Key)
- Title
- PubYear
- AuthorId (Foreign Key referencing Authors)
- PublisherId (Foreign Key referencing Publishers)
- GenreId (Foreign Key referencing Genres)

✓ **Fiction Book:**

- FictionBookId (Primary Key)
- BookId (Foreign Key referencing Books)
- Theme
- Summary

✓ **Non-Fiction Book:**

- NonFictionBookId (Primary Key)
- BookId (Foreign Key referencing Books)
- Subject
- AcademicLevel

✓ **Transactions:**

- TransactionId (Primary Key)
- BookId (Foreign Key referencing Books)
- BorrowerId (Foreign Key referencing Borrowers)
- IssueDate
- DueDate
- ReturnDate
- FineAmount
- BranchId (Foreign Key referencing Branches)

✓ **Staff:**

- StaffId (Primary Key)
- Name
- Position
- JoiningDate
- Salary
- PhoneNo
- BranchId (Foreign Key referencing Branches)

✓ **Branches:**

- BranchId (Primary Key)
- BranchName
- Location
- BranchContactId
- BranchManagerId (Foreign Key referencing Staff)

Relationships:

Here are the relationships between the tables in the LMS database schema:

- **Authors and Books:**
  One-to-Many relationship: An author can write multiple books, but a book is written by only one author.
- **Publishers and Books:**
  One-to-Many relationship: A publisher can publish multiple books, but a book is published by only one publisher.
- **Genres and Books:**
  One-to-Many relationship: A genre can be assigned to multiple books, but a book is categorized under only one genre.
- **Fiction Books and Non-fiction Books:**
  Inheritance relationship: Both fiction books and non-fiction books are types of books, where fiction books have additional attributes like theme and summary, and non-fiction books have attributes like subject and academic level.
- **Transactions and Borrowers:**
  One-to-Many relationship: A borrower can have multiple transactions (e.g., borrowing multiple books), but each transaction is associated with only one borrower.
- **Branches and Staff:**
  One-to-Many relationship: A branch can have multiple staff members, but each staff member is assigned to only one branch.
- **Branches and Transactions:**
  One-to-Many relationship: A branch can have multiple transactions (e.g., books borrowed from that branch), but each transaction is associated with only one branch.

DATABASE SYSTEMS FINAL PROJECT

LMS

QUERIES

Update Queries:

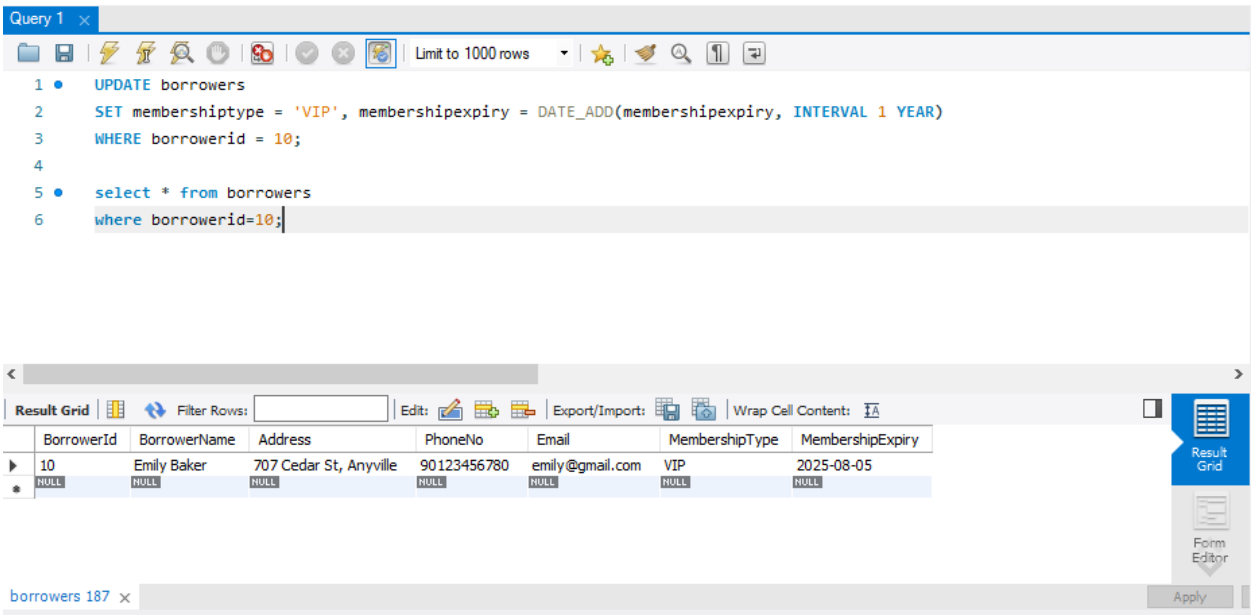
Query Number 1:

Query:

UPDATE borrowers

SET membertype = 'VIP', membershipexpiry = DATE_ADD(membershipexpiry, INTERVAL 1 YEAR)

WHERE borrowerid = 101;



Explanation:

This SQL statement updates the "membertype" column to 'VIP' and extends the "membershipexpiry" date by one year for the borrower with ID 101 in the "borrowers" table.

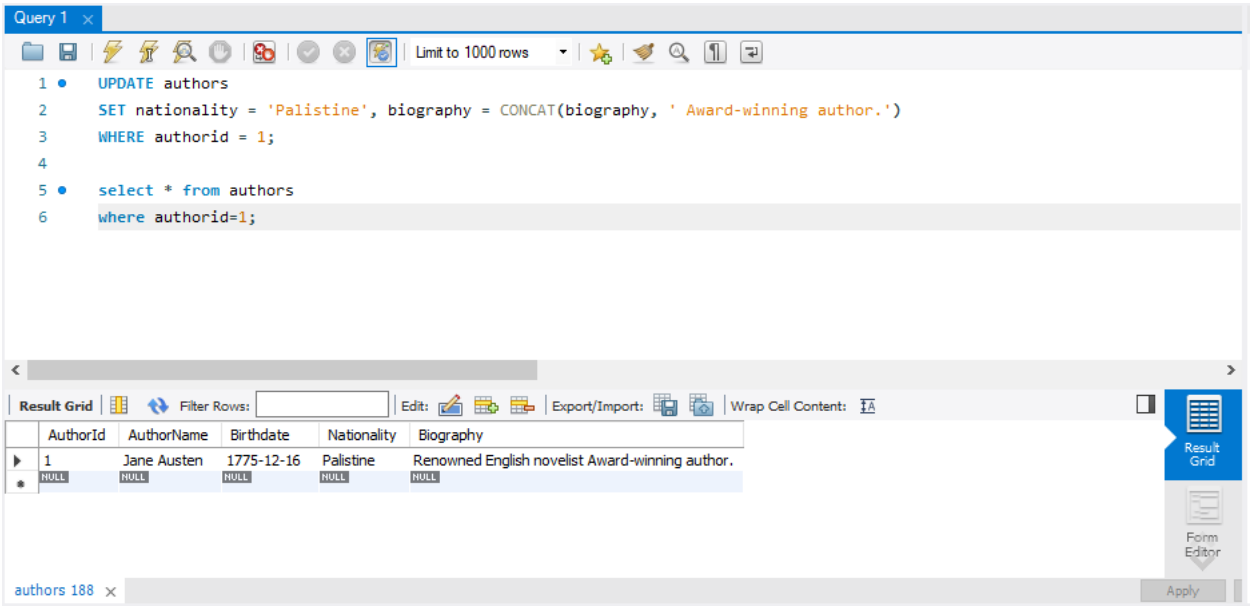
Query Number 2:

Query:

UPDATE authors

SET nationality = 'Palistine', biography = CONCAT(biography, ' Award-winning author.')

WHERE authorid = 1;



Explanation:

This SQL query updates the "nationality" column to 'Palestine' and appends 'Award-winning author.' to the "biography" column for the author with ID 1 in the "authors" table.

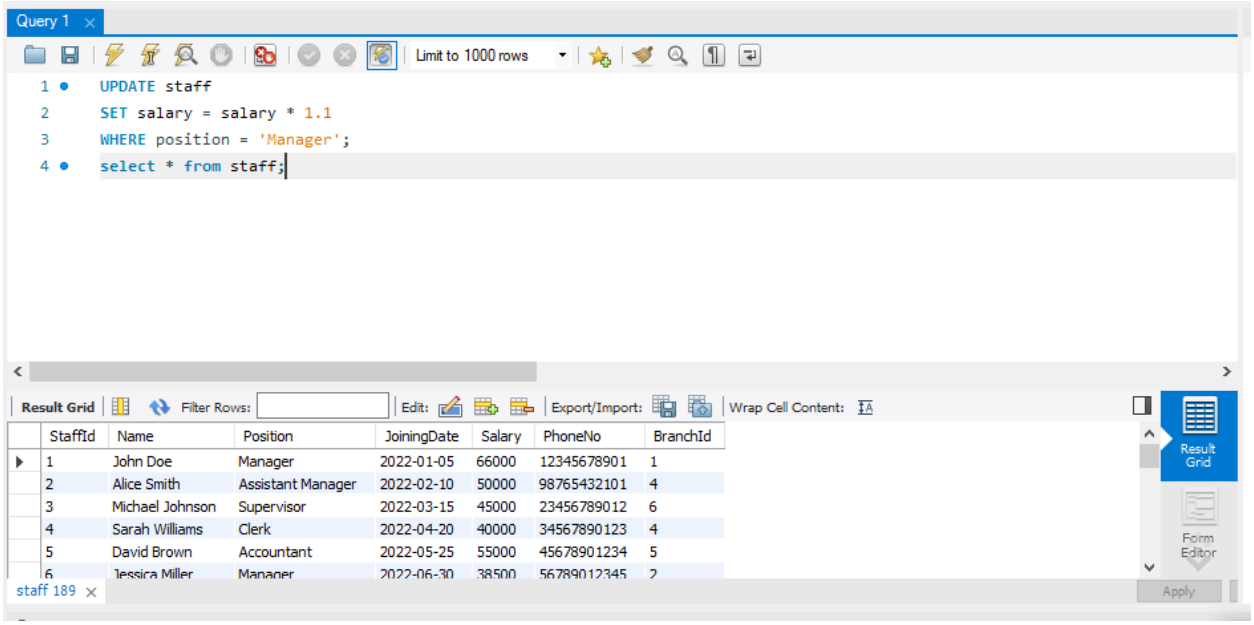
Query Number 3:

Query:

UPDATE staff

SET salary = salary * 1.1

WHERE position = 'Manager';



StaffId	Name	Position	JoiningDate	Salary	PhoneNo	BranchId
1	John Doe	Manager	2022-01-05	66000	12345678901	1
2	Alice Smith	Assistant Manager	2022-02-10	50000	98765432101	4
3	Michael Johnson	Supervisor	2022-03-15	45000	23456789012	6
4	Sarah Williams	Clerk	2022-04-20	40000	34567890123	4
5	David Brown	Accountant	2022-05-25	55000	45678901234	5
6	Jessica Miller	Manager	2022-06-30	38500	56789012345	2

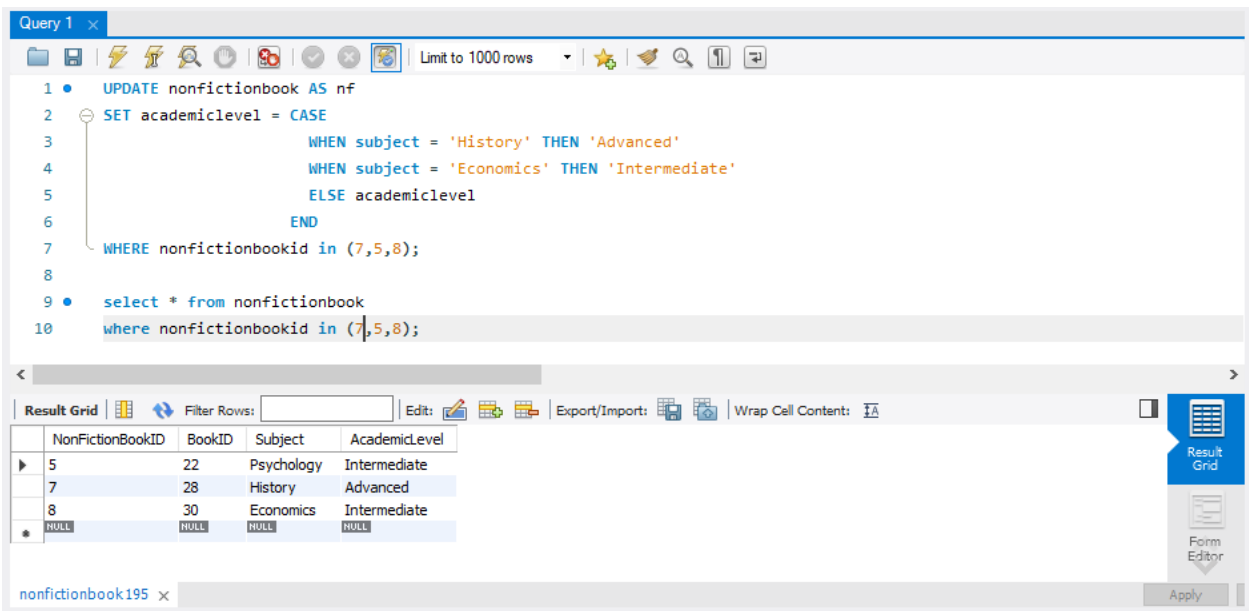
Explanation:

This SQL query increases the "salary" by 10% for all staff members whose "position" is 'Manager' in the "staff" table.

Query Number 4:

Query:

```
UPDATE nonfictionbook AS nf

SET academiclevel = CASE

                WHEN subject = 'History' THEN 'Advanced'

                WHEN subject = 'Economics' THEN 'Intermediate'

                ELSE academiclevel

              END

WHERE nonfictionbookid in (7,5,8);
```



Explanation:

This SQL statement updates the "academiclevel" column for non-fiction books with IDs 7, 5, and 8 in the "nonfictionbook" table. If the book's "subject" is 'History', it sets the "academiclevel" to 'Advanced'. If the subject is 'Economics', it sets the "academiclevel" to 'Intermediate'. Otherwise, it leaves the "academiclevel" unchanged.

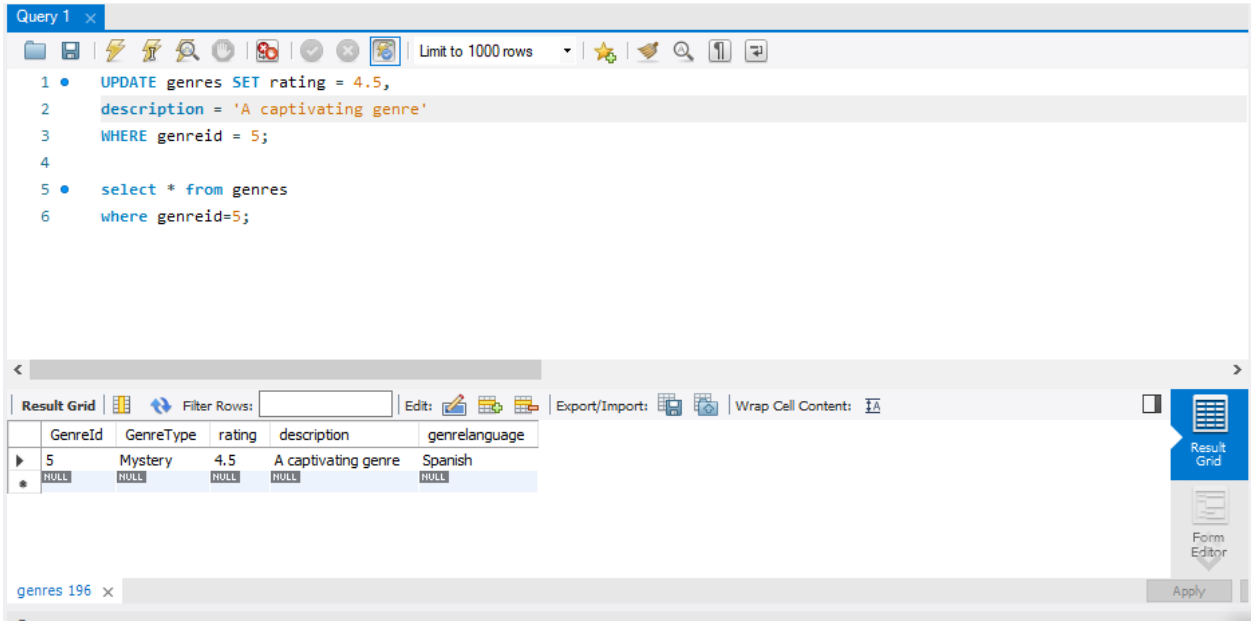
Query Number 5:

Query:

UPDATE genres

SET rating = 4.5, description = 'A captivating genre'

WHERE genreid = 5;



Explanation:

This SQL query updates the "rating" to 4.5 and sets the "description" to 'A captivating genre' for the genre with ID 5 in the "genres" table.

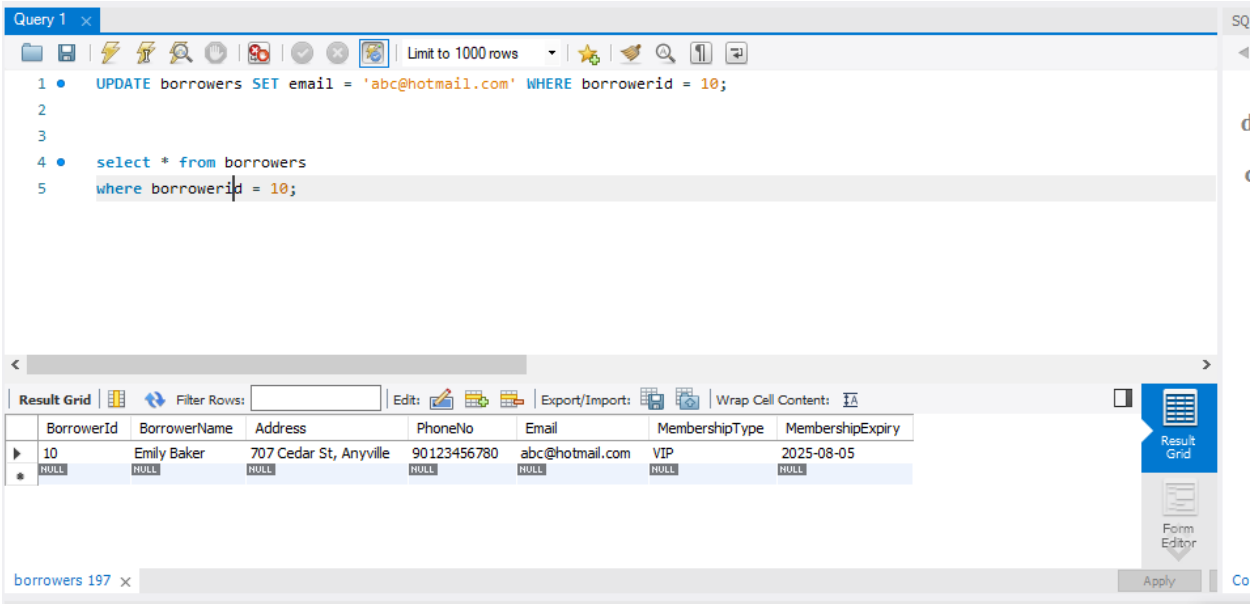
Query Number 6:

Query:

UPDATE borrowers

SET email = 'abc@hotmail.com'

WHERE borrowerid = 10;



Explanation:

This SQL query updates the "email" to 'abc@hotmail.com' for the borrower with ID 10 in the "borrowers" table.

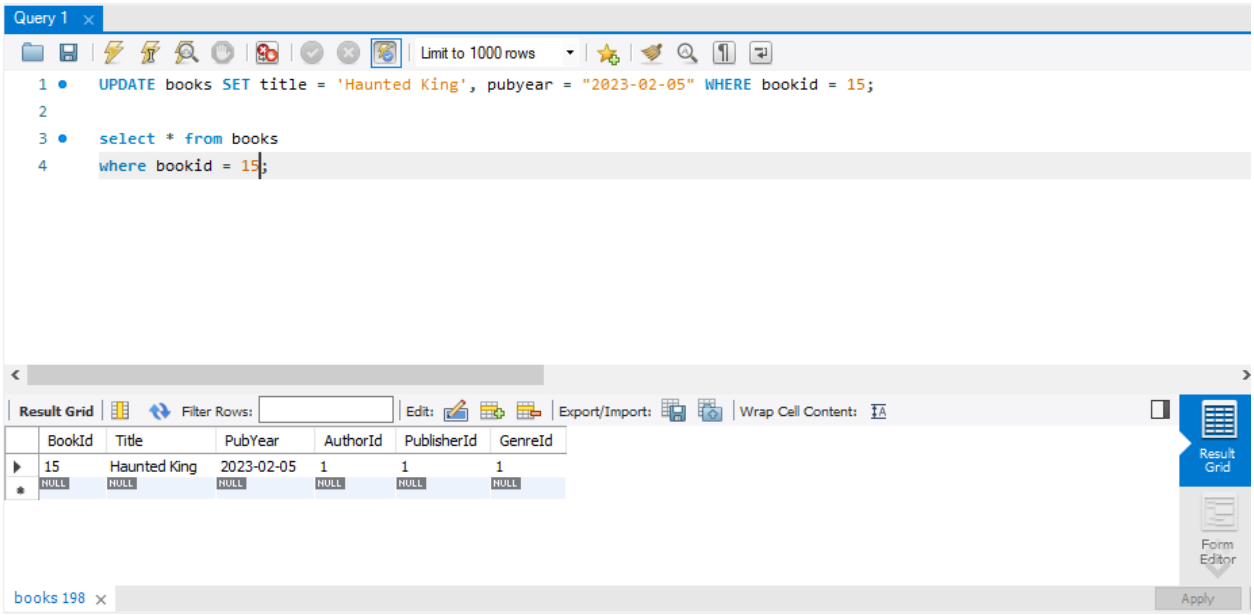
Query Number 7:

Query:

UPDATE books

SET title = 'Haunted King', pubyear = "2023-02-05"

WHERE bookid = 15;



Explanation:

This SQL query updates the "title" to 'Haunted King' and the "pubyear" to '2023-02-05' for the book with ID 15 in the "books" table.

Query Number 8:

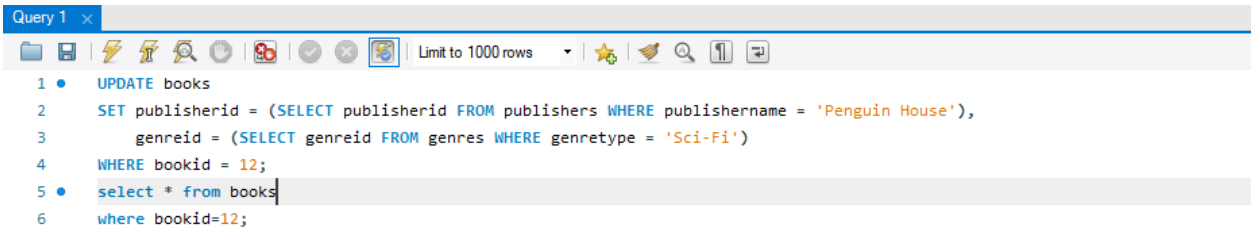
Query:

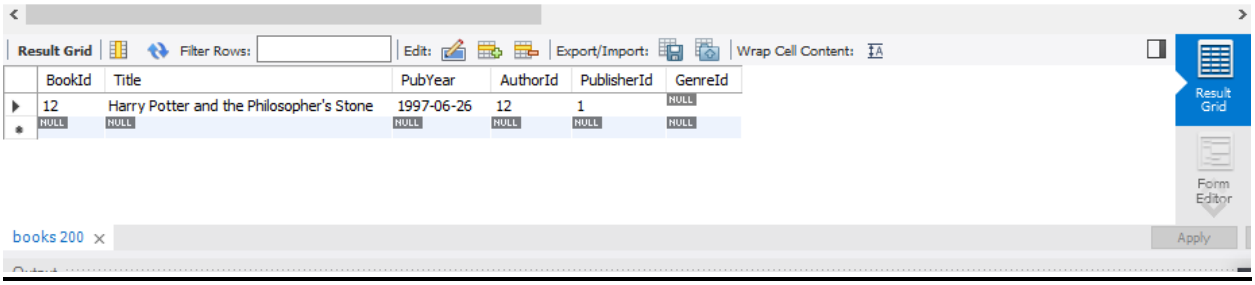
UPDATE books

SET publisherid = (SELECT publisherid FROM publishers WHERE publishername = 'Penguin House'),

genreid = (SELECT genreid FROM genres WHERE genretype = 'Sci-Fi')

WHERE bookid = 12;





Explanation:

This SQL query updates the "publisherid" to the ID of the publisher 'Penguin House' and the "genreid" to the ID of the genre 'Sci-Fi' for the book with ID 12 in the "books" table.

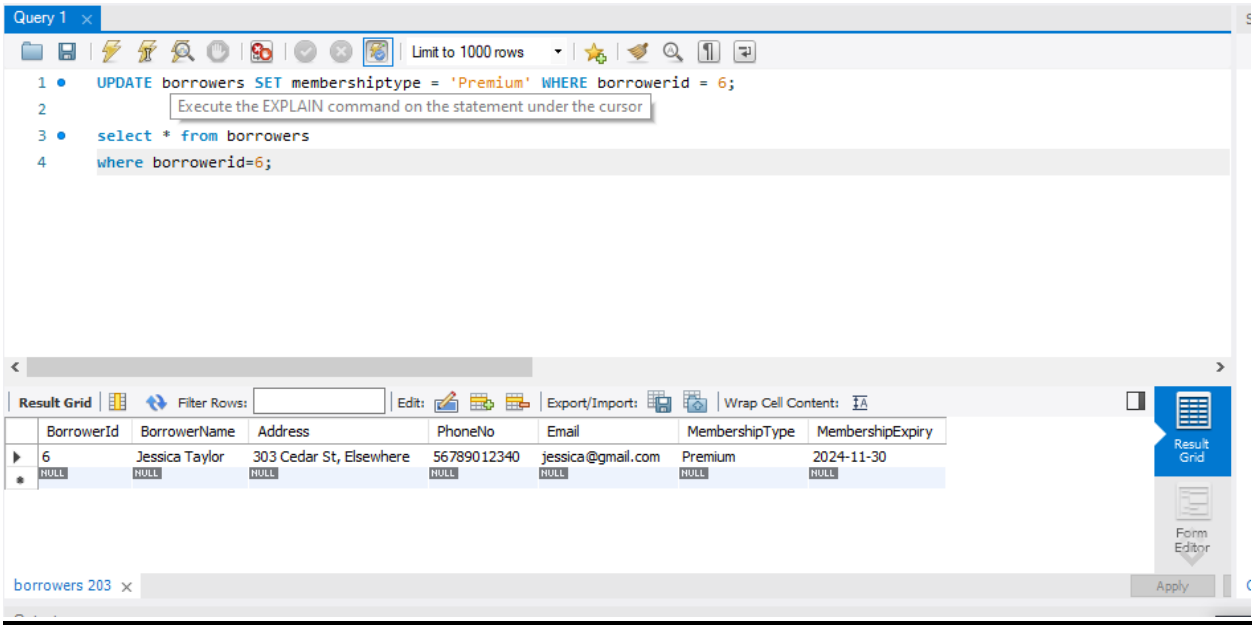
Query Number 9:

Query:

UPDATE borrowers

SET membershiptype = 'Premium'

WHERE borrowerid = 6;



Explanation:

This SQL query updates the "membershiptype" to 'Premium' for the borrower with ID 6 in the "borrowers" table.

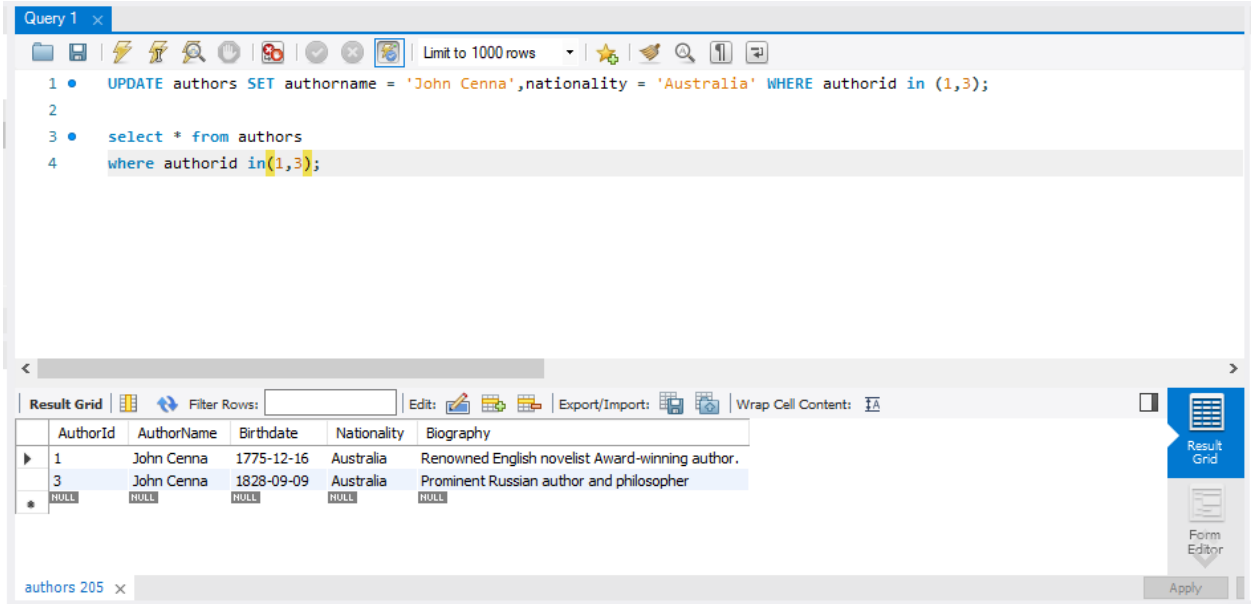
Query Number 10:

Query:

UPDATE authors

SET nationality = 'Australia'

WHERE authorid in (1,3);



Explanation:

This SQL query updates the "nationality" to 'Australia' for the authors with IDs 1 and 3 in the "authors" table.

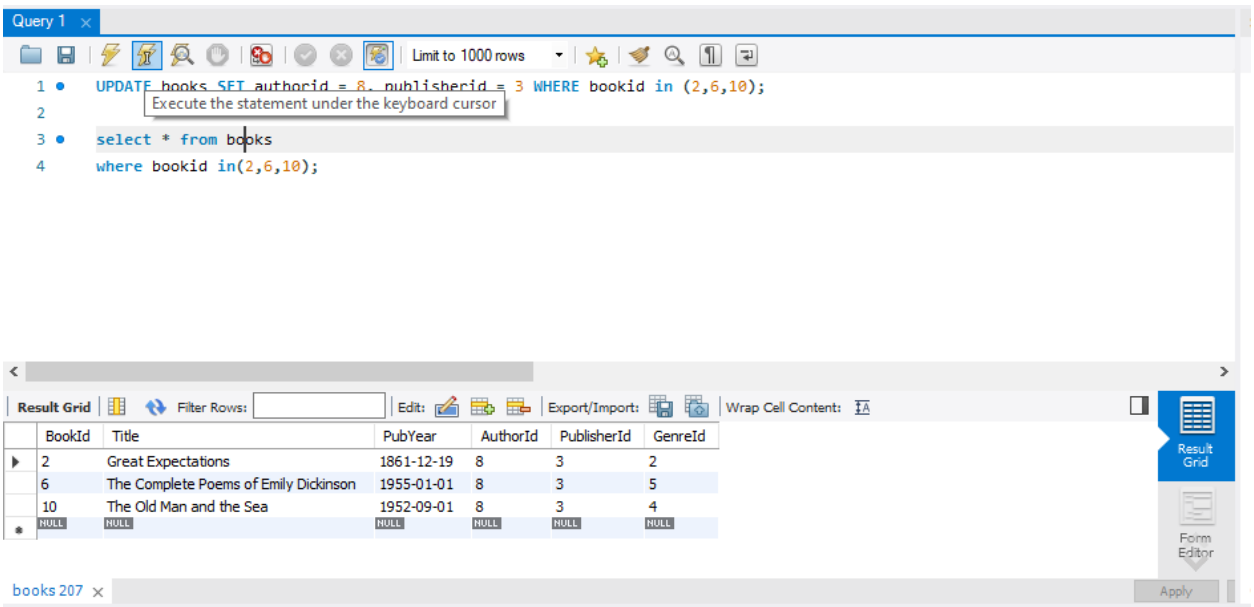
Query Number 11:

Query:

UPDATE books

SET authorid = 8, publisherid = 3

WHERE bookid in (2,6,10);



Explanation:

This SQL query updates the "authorid" to 8 and the "publisherid" to 3 for the books with IDs 2, 6, and 10 in the "books" table.

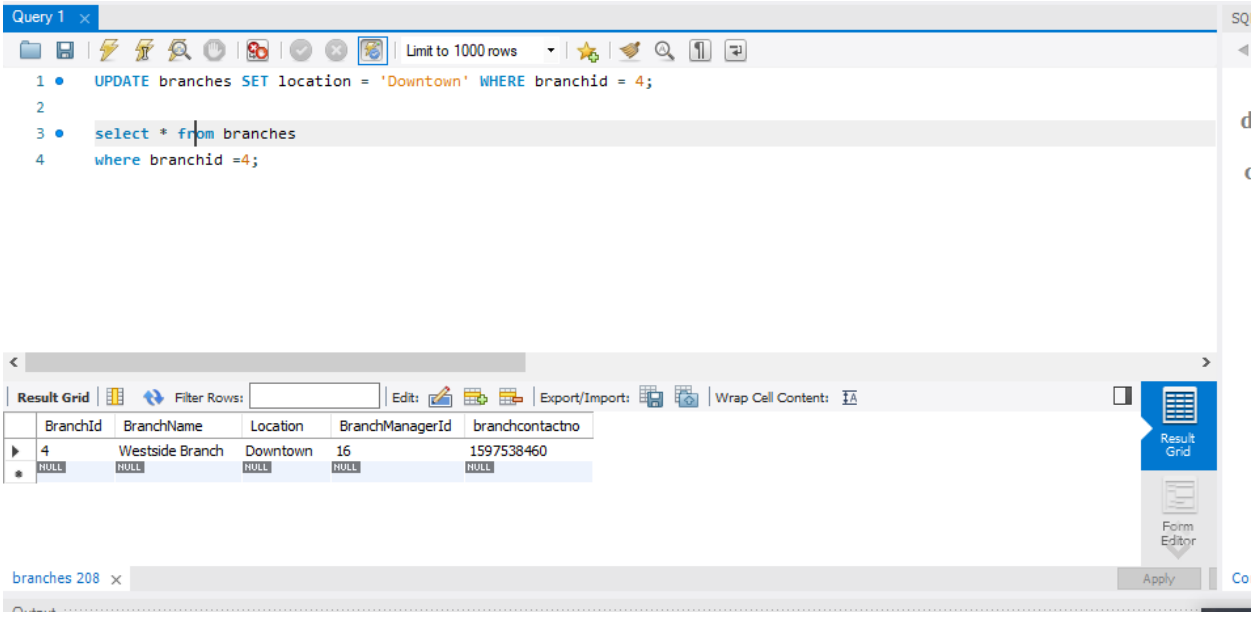
Query Number 12:

Query:

UPDATE branches

SET location = 'Downtown'

WHERE branchid = 4;



Explanation:

This SQL query updates the "location" to 'Downtown' for the branch with ID 4 in the "branches" table.

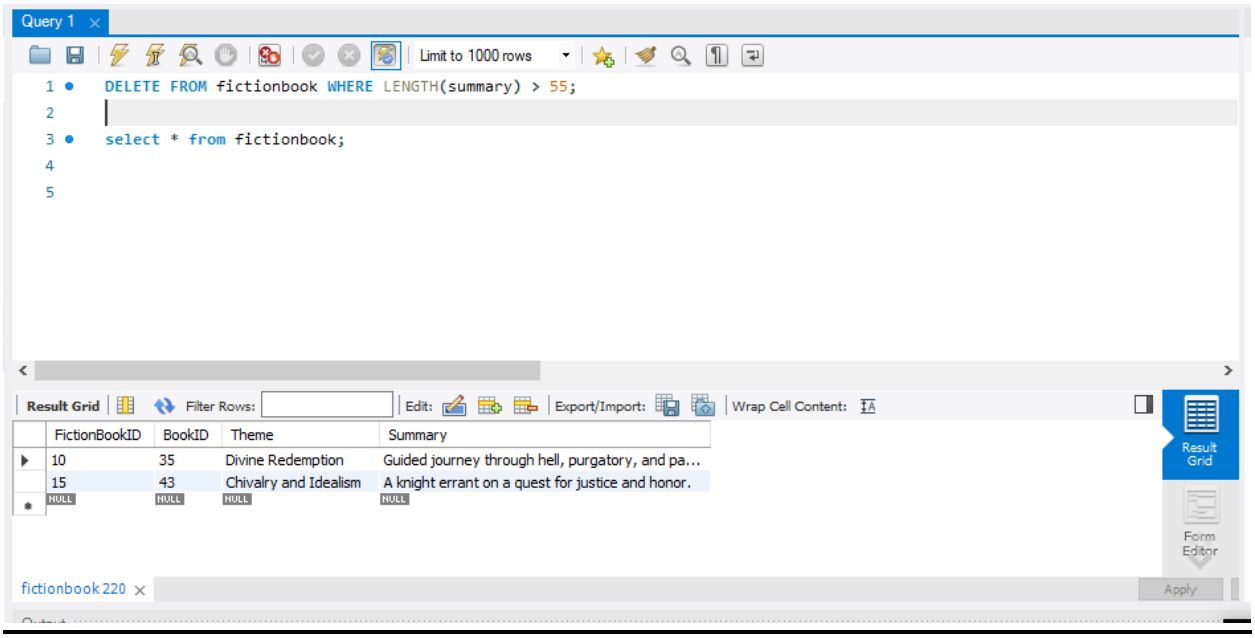
Delete Queries:

Query Number 1:

Query:

DELETE FROM fictionbook

WHERE LENGTH(summary) > 55;



Explanation:

This SQL query deletes rows from the "fictionbook" table where the length of the "summary" column is greater than 55 characters.

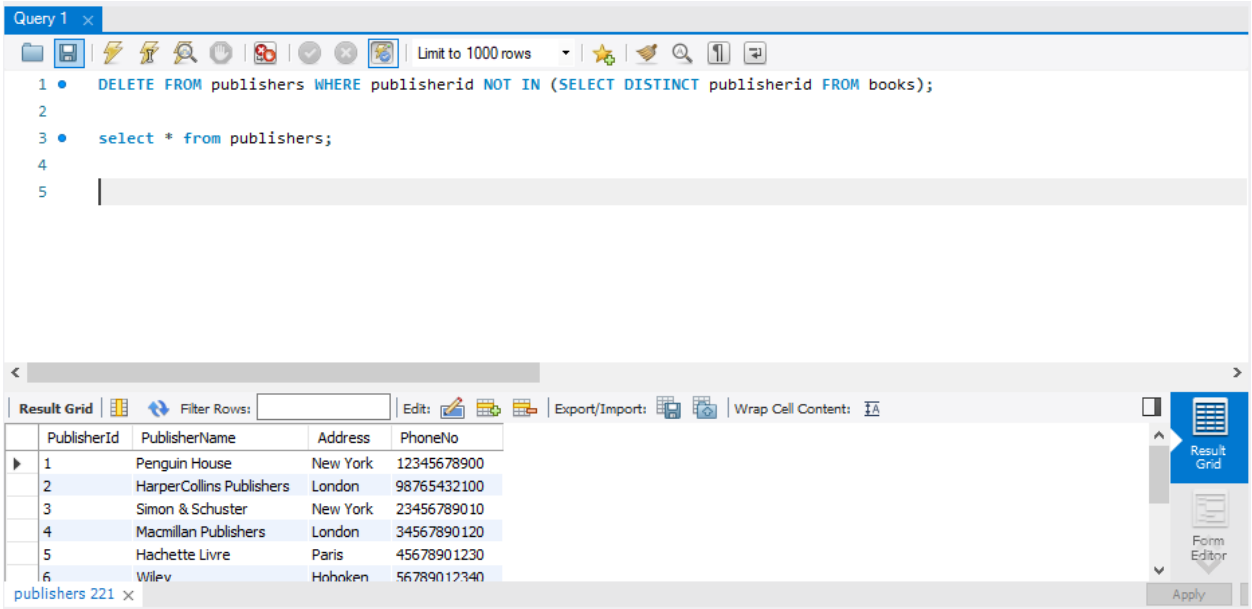
Query Number 2:

Query:

DELETE FROM publishers

WHERE publisherid NOT IN

(SELECT DISTINCT publisherid FROM books);



Explanation:

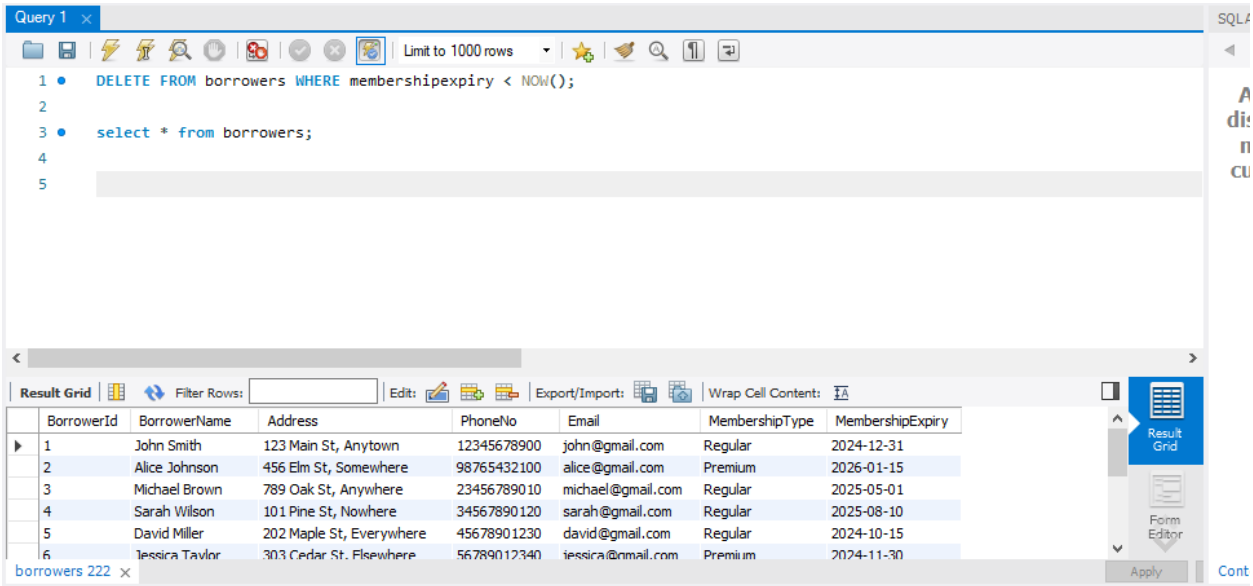
This SQL query deletes rows from the "publishers" table where the "publisherid" is not present in the list of distinct "publisherid" values from the "books" table.

Query Number 3:

Query:

DELETE FROM borrowers

WHERE membershipexpiry < NOW();



BorrowerId	BorrowerName	Address	PhoneNo	Email	MembershipType	MembershipExpiry
1	John Smith	123 Main St, Anytown	12345678900	john@gmail.com	Regular	2024-12-31
2	Alice Johnson	456 Elm St, Somewhere	98765432100	alice@gmail.com	Premium	2026-01-15
3	Michael Brown	789 Oak St, Anywhere	23456789010	michael@gmail.com	Regular	2025-05-01
4	Sarah Wilson	101 Pine St, Nowhere	34567890120	sarah@gmail.com	Regular	2025-08-10
5	David Miller	202 Maple St, Everywhere	45678901230	david@gmail.com	Regular	2024-10-15
6	Jessica Taylor	303 Cedar St, Elsewhere	56789012340	jessica@gmail.com	Premium	2024-11-30

Explanation:

This SQL query deletes rows from the "borrowers" table where the "membershipexpiry" date is earlier than the current date and time.

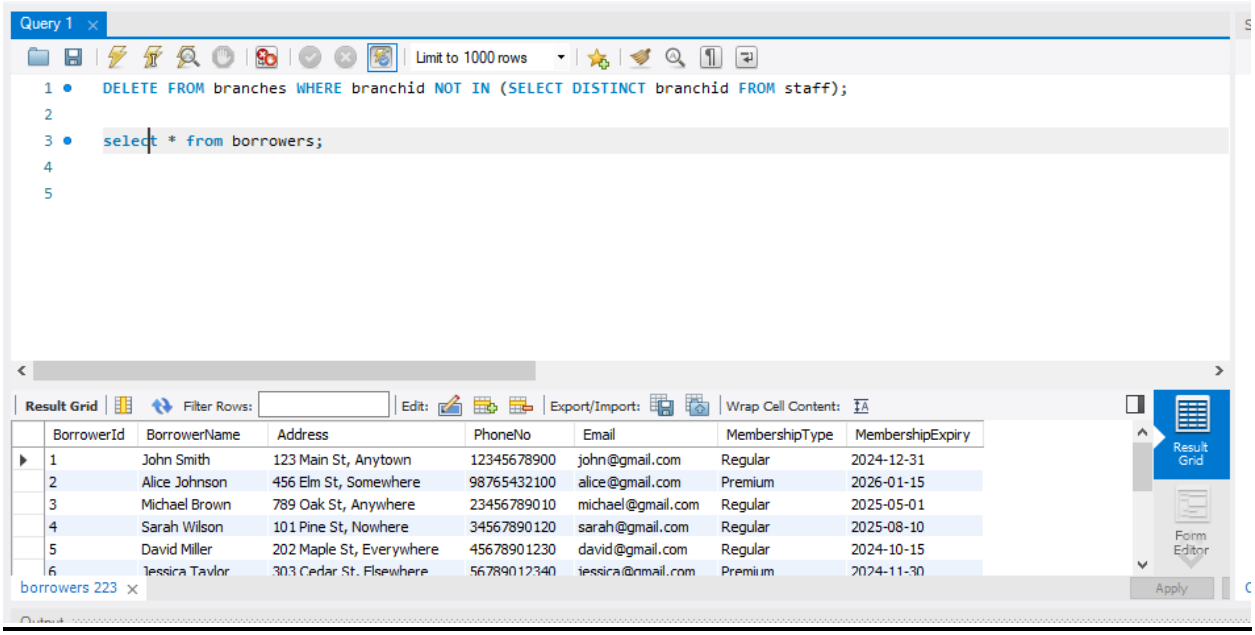
Query Number 4:

Query:

DELETE FROM branches

WHERE branchid NOT IN

(SELECT DISTINCT branchid FROM staff);



Explanation:

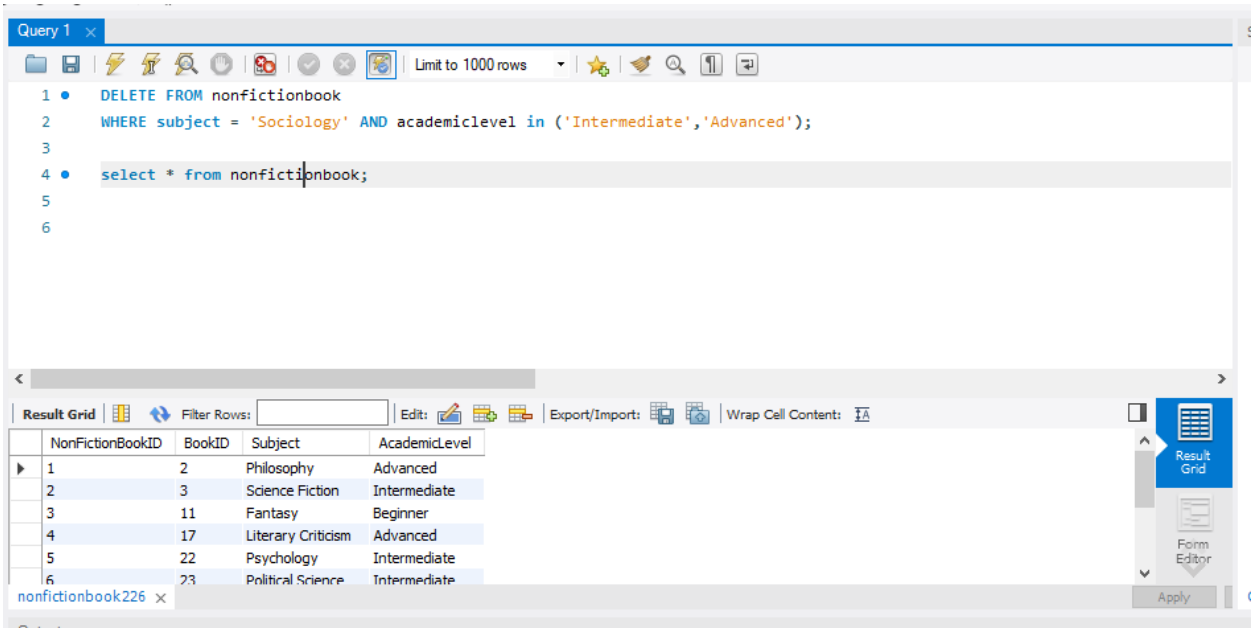
This SQL query deletes rows from the "branches" table where the "branchid" is not present in the list of distinct "branchid" values from the "staff" table.

Query Number 5:

Query:

DELETE FROM nonfictionbook

WHERE subject = 'Sociology' AND academiclevel IN ('Intermediate','Advanced');



Explanation:

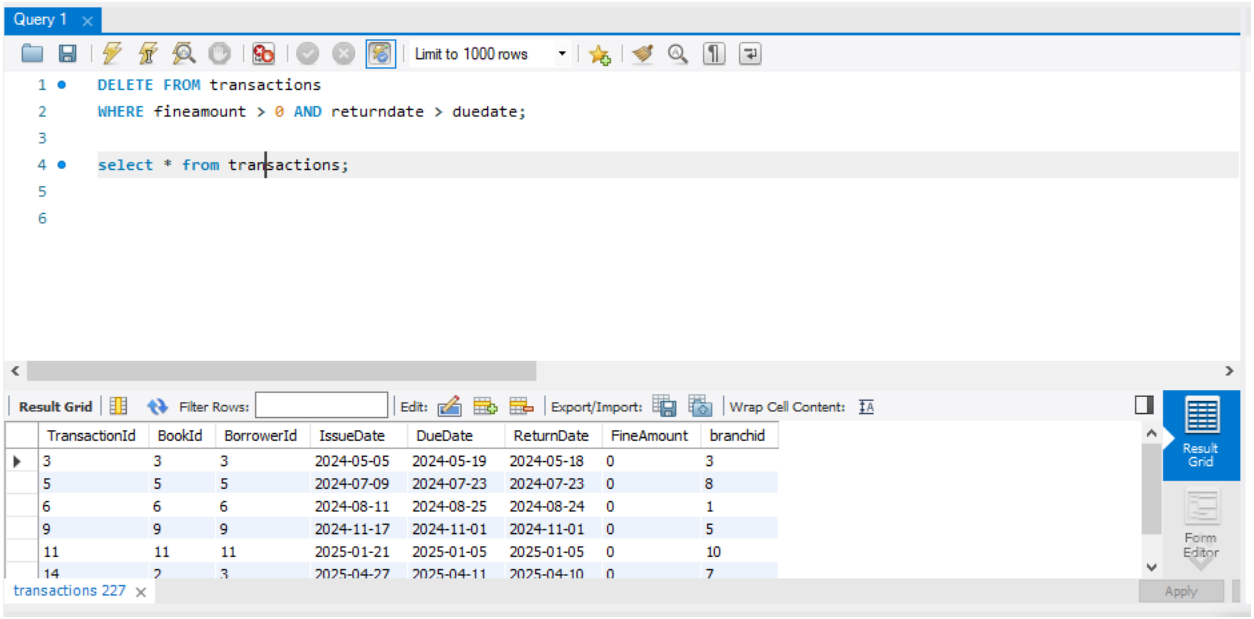
This SQL query deletes rows from the "nonfictionbook" table where the "subject" is 'Sociology' and the "academiclevel" is either 'Intermediate' or 'Advanced'.

Query Number 6:

Query:

DELETE FROM transactions

WHERE fineamount > 0 AND returndate > duedate;



TransactionId	BookId	BorrowerId	IssueDate	DueDate	ReturnDate	FineAmount	branchid
3	3	3	2024-05-05	2024-05-19	2024-05-18	0	3
5	5	5	2024-07-09	2024-07-23	2024-07-23	0	8
6	6	6	2024-08-11	2024-08-25	2024-08-24	0	1
9	9	9	2024-11-17	2024-11-01	2024-11-01	0	5
11	11	11	2025-01-21	2025-01-05	2025-01-05	0	10
14	2	3	2025-04-27	2025-04-11	2025-04-10	0	7

Explanation:

This SQL query deletes rows from the "transactions" table where the "fineamount" is greater than 0 and the "returndate" is later than the "duedate".

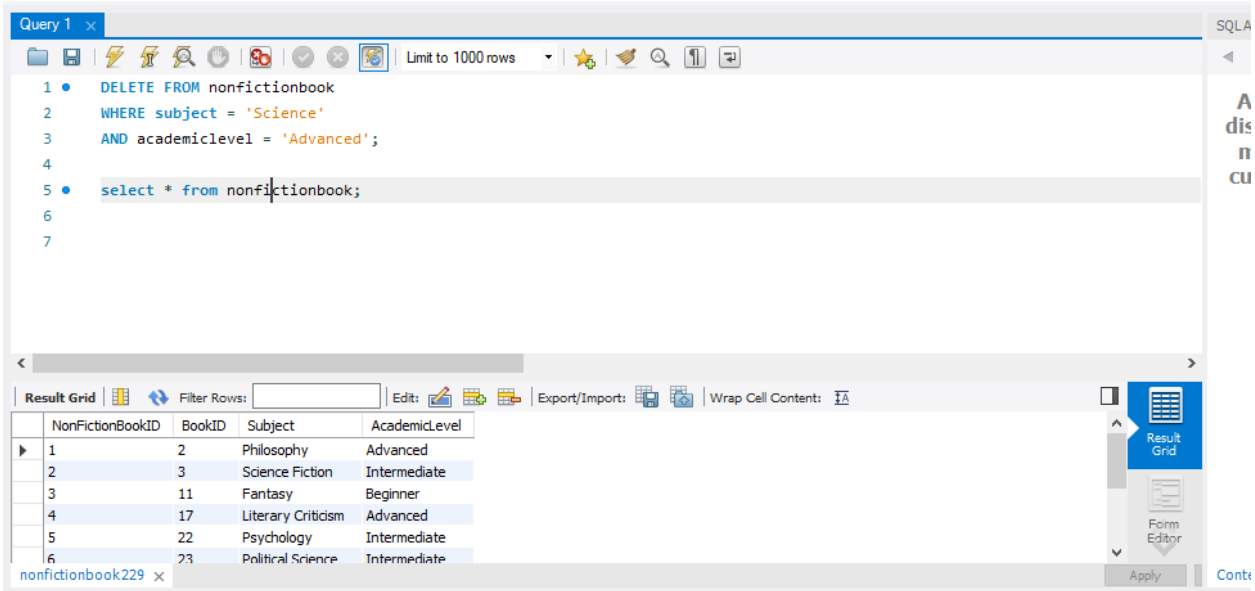
Query Number 7:

Query:

DELETE FROM nonfictionbook

WHERE subject = 'Science'

AND academic_level = 'Advanced';



Explanation:

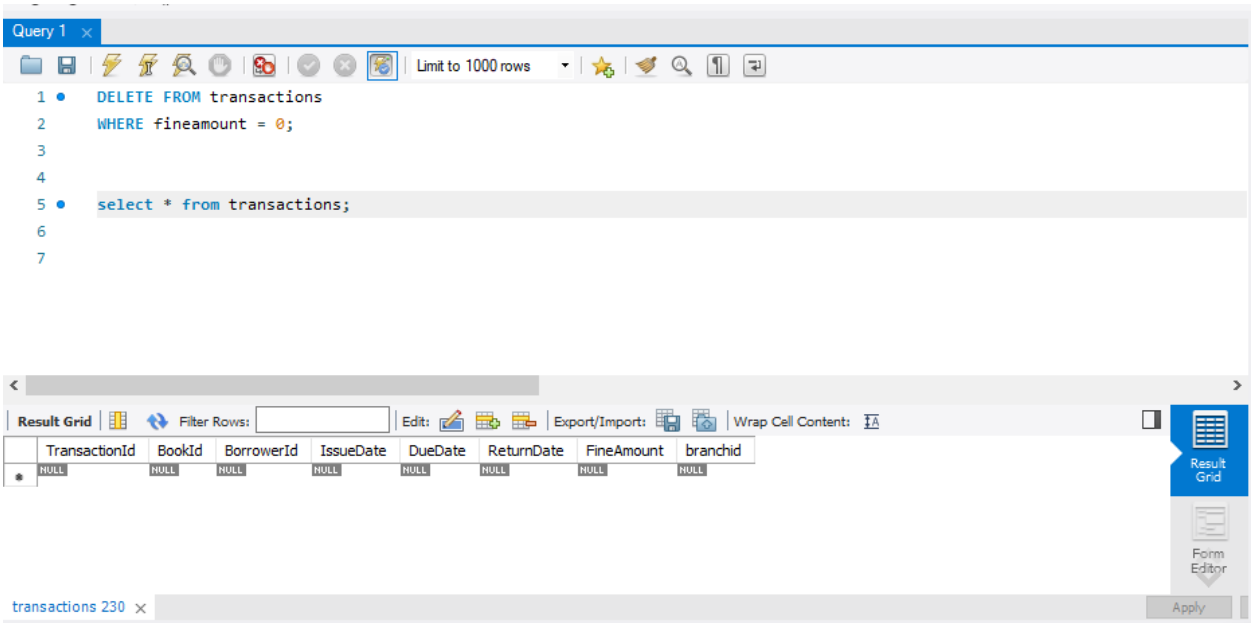
This SQL query deletes rows from the "nonfictionbook" table where the "subject" is 'Science' and the "academiclevel" is 'Advanced'.

Query Number 8:

Query:

DELETE FROM transactions

WHERE fineamount = 0;



Explanation:

This SQL query deletes rows from the "transactions" table where the "fineamount" is equal to 0.

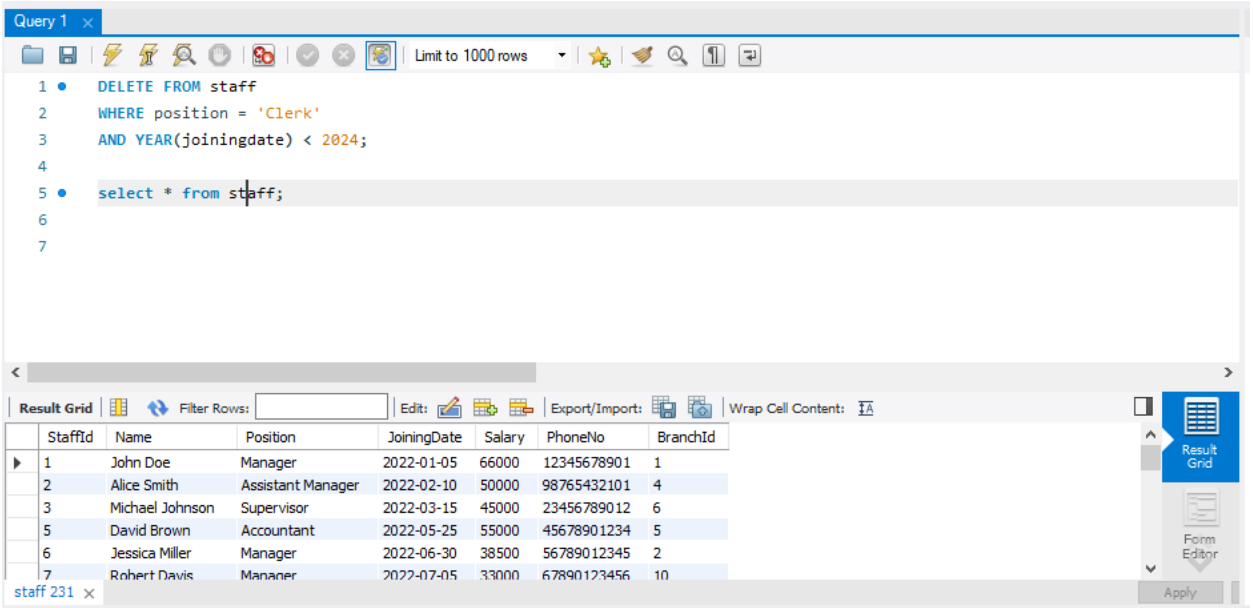
Query Number 9:

Query:

DELETE FROM staff

WHERE position = 'Clerk'

AND YEAR (joiningdate) < 2024;



StaffId	Name	Position	JoiningDate	Salary	PhoneNo	BranchId
1	John Doe	Manager	2022-01-05	66000	12345678901	1
2	Alice Smith	Assistant Manager	2022-02-10	50000	98765432101	4
3	Michael Johnson	Supervisor	2022-03-15	45000	23456789012	6
5	David Brown	Accountant	2022-05-25	55000	45678901234	5
6	Jessica Miller	Manager	2022-06-30	38500	56789012345	2
7	Robert Davis	Manager	2022-07-05	33000	67890123456	10

Explanation:

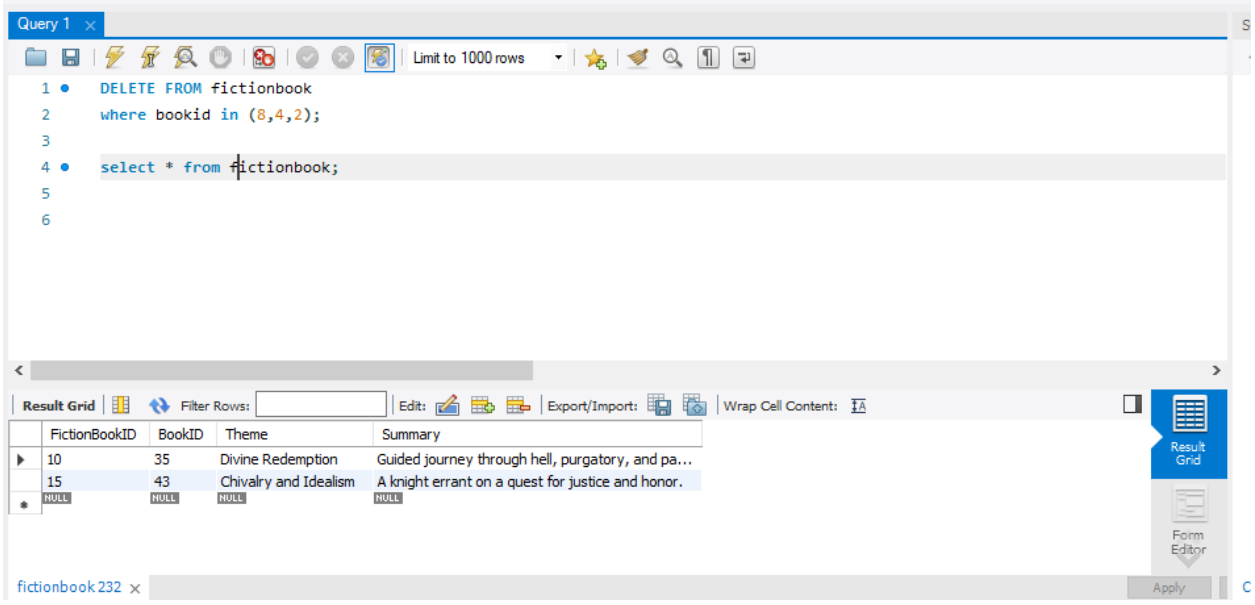
This SQL query deletes rows from the "staff" table where the "position" is 'Clerk' and the year of the "joiningdate" is before 2024.

Query Number 10:

Query:

DELETE FROM fictionbook

WHERE bookid IN (8,4,2);



Explanation:

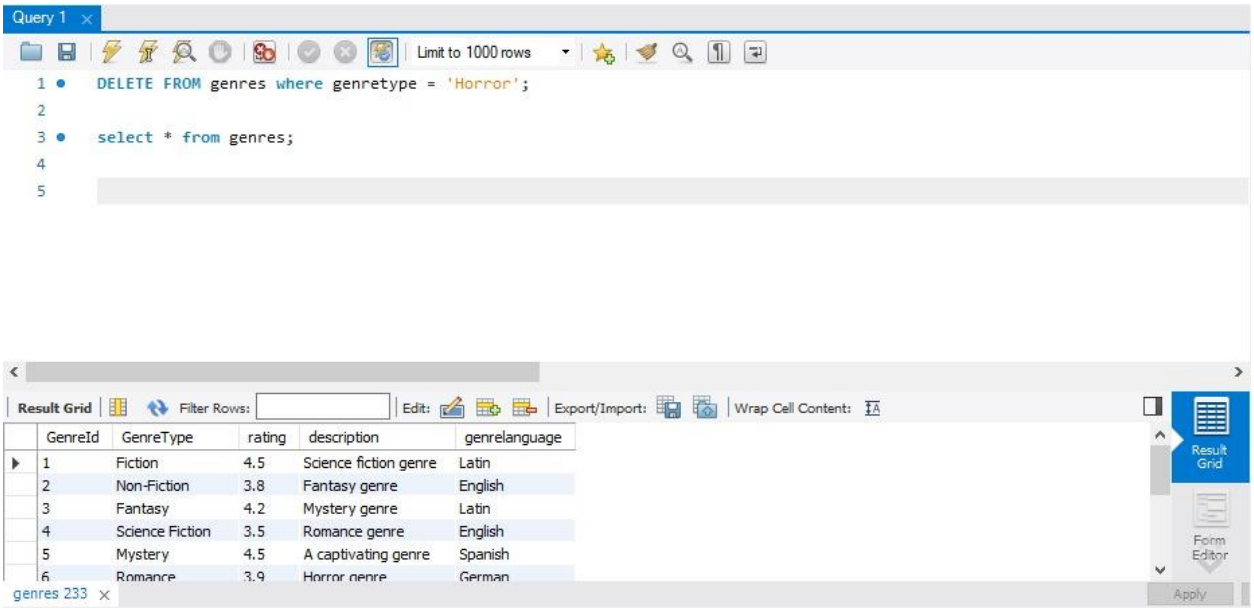
This SQL query deletes rows from the "fictionbook" table where the "bookid" is either 8, 4, or 2.

Query Number 11:

Query:

DELETE FROM genres

WHERE genretype = 'Horror';



Explanation:

This SQL query deletes rows from the "genres" table where the "genretype" is 'Horror'.

Join Queries:

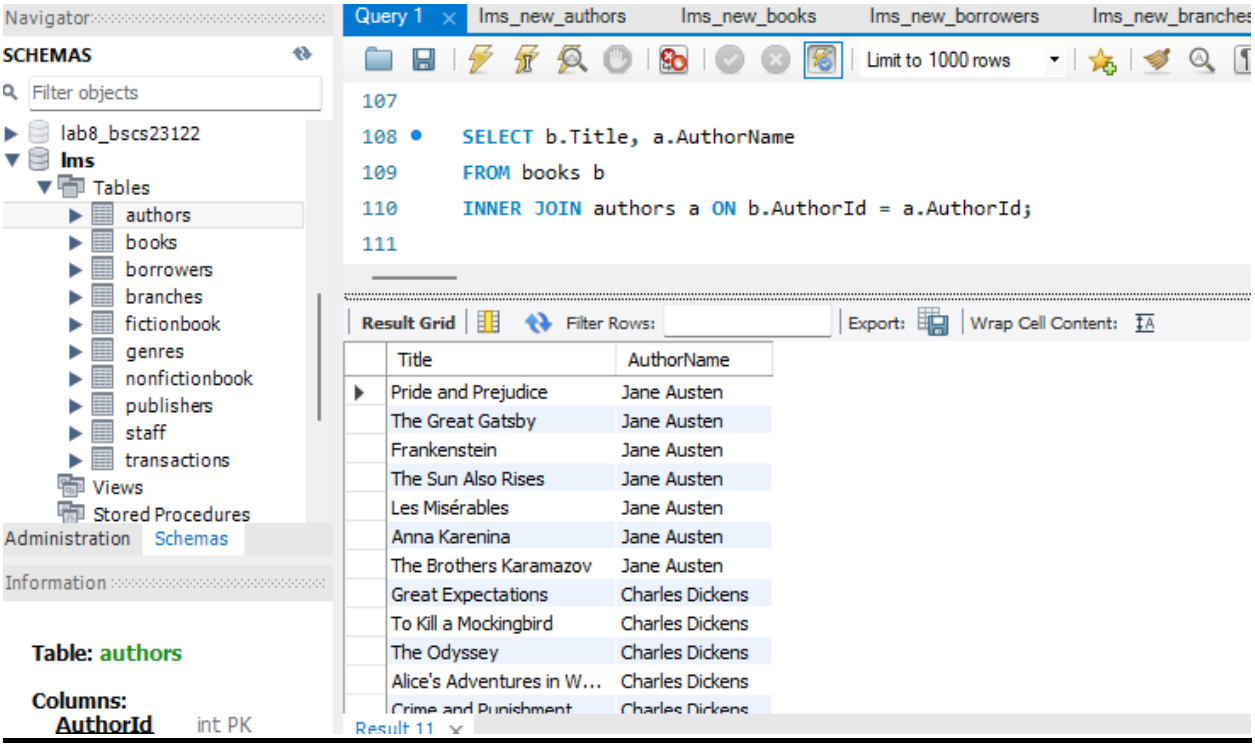
Query Number 1:

Query:

SELECT b.Title, a.AuthorName

FROM books b

INNER JOIN authors a ON b.AuthorId = a.AuthorId;



Explanation:

This query retrieves book titles and their corresponding author names by joining the books and authors tables based on the author's ID.

Query Number 2:

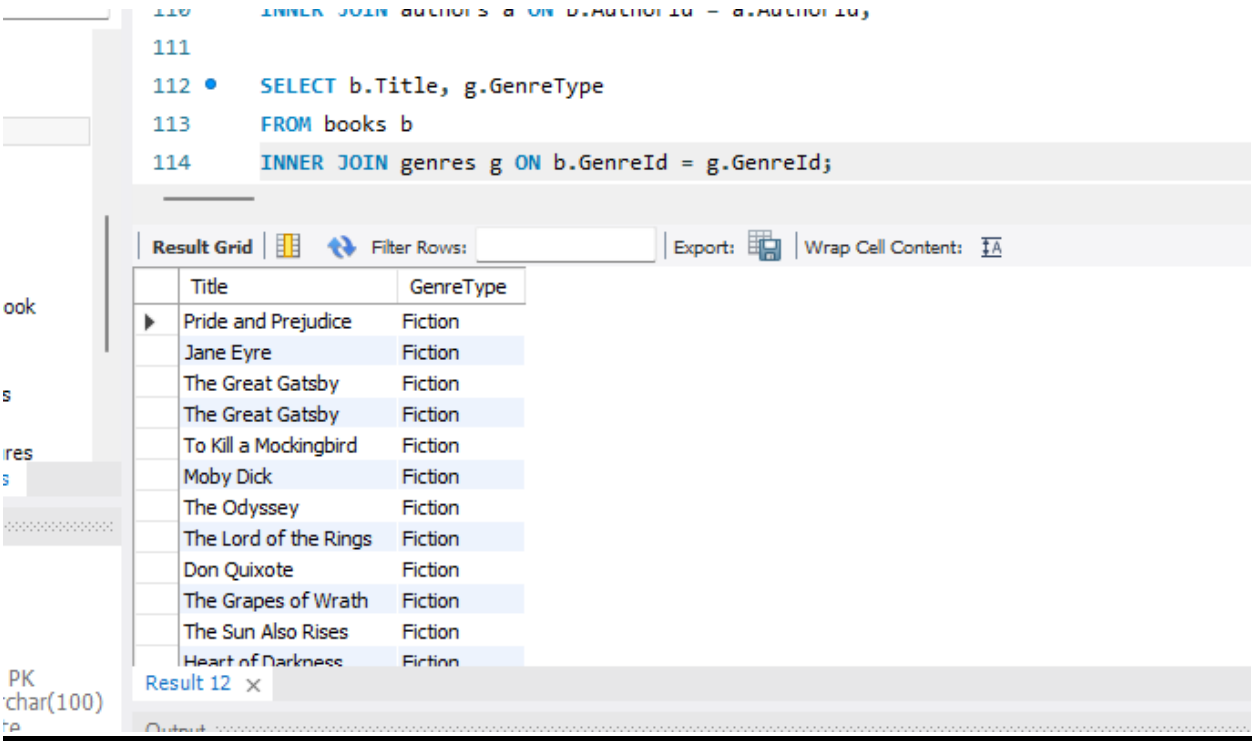
Query:

SELECT b.Title, g.GenreType

FROM books b

INNER JOIN genres g

ON b.GenreId = g.GenreId;



Title	GenreType
Pride and Prejudice	Fiction
Jane Eyre	Fiction
The Great Gatsby	Fiction
The Great Gatsby	Fiction
To Kill a Mockingbird	Fiction
Moby Dick	Fiction
The Odyssey	Fiction
The Lord of the Rings	Fiction
Don Quixote	Fiction
The Grapes of Wrath	Fiction
The Sun Also Rises	Fiction
Heart of Darkness	Fiction

Explanation:

This query fetches book titles along with their corresponding genre types by joining the books and genres tables based on the genre ID.

Query Number 3:

Query:

SELECT t.TransactionId, b.Title

AS BookTitle, br.BorrowerName, br.Email, bs.BranchName

FROM transactions t

INNER JOIN books b ON t.BookId = b.BookId

INNER JOIN borrowers br ON t.BorrowerId = br.BorrowerId

INNER JOIN branches bs ON t.branchid = bs.BranchId;

Limit to 1000 rows

```
SELECT t.TransactionId, b.Title AS BookTitle, br.BorrowerName, br.Email, bs.BranchName
FROM transactions t
INNER JOIN books b ON t.BookId = b.BookId
INNER JOIN borrowers br ON t.BorrowerId = br.BorrowerId
INNER JOIN branches bs ON t.branchid = bs.BranchId;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TransactionId	BookTitle	BorrowerName	Email	BranchName
6	The Complete Poems of Emily Dickinson	Jessica Taylor	jessica@gmail.com	Main Branch
1	Pride and Prejudice	John Smith	john@gmail.com	Downtown Branch
13	Pride and Prejudice	Alice Johnson	alice@gmail.com	Downtown Branch
3	War and Peace	Michael Brown	michael@gmail.com	City Center Branch
2	Great Expectations	Alice Johnson	alice@gmail.com	Westside Branch
10	The Old Man and the Sea	Emily Baker	emily@gmail.com	Westside Branch
9	The Great Gatsby	Daniel Clark	daniel@gmail.com	Eastside Branch
12	Harry Potter and the Philosopher's Stone	Olivia Adams	olivia@gmail.com	Northside Branch
4	To the Lighthouse	Sarah Wilson	sarah@gmail.com	Southside Branch
14	Great Expectations	Michael Brown	michael@gmail.com	Southside Branch
5	The Adventures of Huckleberry Finn	David Miller	david@gmail.com	Central Branch
8	Jane Eyre	Laura Martinez	laura@gmail.com	Midtown Branch

Result 13

Explanation:

This query retrieves transaction IDs, book titles, borrower names, borrower emails, and branch names by joining the transactions, books, borrowers, and branches tables based on their respective IDs.

Query Number 4:

Query:

SELECT b.Title, a.Nationality

AS AuthorNationality

FROM books b

INNER JOIN authors a ON b.AuthorId = a.AuthorId;

```
SELECT b.Title, a.Nationality AS AuthorNationality
FROM books b
INNER JOIN authors a ON b.AuthorId = a.AuthorId;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Title	AuthorNationality
Pride and Prejudice	British
The Great Gatsby	British
Frankenstein	British
The Sun Also Rises	British
Les Misérables	British
Anna Karenina	British
The Brothers Karamazov	British
Great Expectations	British
To Kill a Mockingbird	British
The Odyssey	British
Alice's Adventures in W...	British
Crime and Punishment	British

Result 14

Explanation:

This query fetches book titles along with the nationality of their respective authors by joining the books and authors tables based on the author's ID.

Query Number 5:

Query:

SELECT t.TransactionId, b.Title AS BookTitle, br.BorrowerName, br.Email, bs.BranchName, t.FineAmount

FROM transactions t

INNER JOIN books b ON t.BookId = b.BookId

INNER JOIN borrowers br ON t.BorrowerId = br.BorrowerId

INNER JOIN branches bs ON t.branchid = bs.BranchId

WHERE t.FineAmount > 0;

```
SELECT t.TransactionId, b.Title AS BookTitle, br.BorrowerName, br.Email, bs.BranchName, t.FineAmount
FROM transactions t
INNER JOIN books b ON t.BookId = b.BookId
INNER JOIN borrowers br ON t.BorrowerId = br.BorrowerId
INNER JOIN branches bs ON t.branchid = bs.BranchId
WHERE t.FineAmount > 0;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TransactionId	BookTitle	BorrowerName	Email	BranchName	FineAmount
1	Pride and Prejudice	John Smith	john@gmail.com	Downtown Branch	200
2	Great Expectations	Alice Johnson	alice@gmail.com	Westside Branch	400
4	To the Lighthouse	Sarah Wilson	sarah@gmail.com	Southside Branch	500
7	1984	Robert Anderson	robert@gmail.com	Uptown Branch	300
8	Jane Eyre	Laura Martinez	laura@gmail.com	Midtown Branch	100
10	The Old Man and the Sea	Emily Baker	emily@gmail.com	Westside Branch	200
12	Harry Potter and the Philosopher's Stone	Olivia Adams	olivia@gmail.com	Northside Branch	800
13	Pride and Prejudice	Alice Johnson	alice@gmail.com	Downtown Branch	1200

Result 15

Explanation:

This query retrieves transaction IDs, book titles, borrower names, borrower emails, branch names, and fine amounts for transactions with fines greater than 0, joining the transactions, books, borrowers, and branches tables based on their respective IDs.

Query Number 6:

Query:

SELECT b.Title

AS BookTitle, nf.Subject, nf.AcademicLevel

FROM nonfictionbook nf

INNER JOIN books b ON nf.BookID = b.BookId;

```
SELECT b.Title AS BookTitle, nf.Subject, nf.AcademicLevel
FROM nonfictionbook nf
INNER JOIN books b ON nf.BookID = b.BookId;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

BookTitle	Subject	AcademicLevel
Great Expectations	Philosophy	Advanced
War and Peace	Science Fiction	Intermediate
One Hundred Years of Solitude	Fantasy	Beginner
1984	Literary Criticism	Advanced
War and Peace	Psychology	Intermediate
The Adventures of Huckleberry Finn	Political Science	Intermediate
Brave New World	History	Intermediate
The Picture of Dorian Gray	Economics	Advanced
Alice's Adventures in Wonderland	Sociology	Intermediate
The Stranger	Anthropology	Intermediate

Result 16

Explanation:

This query selects the titles of non-fiction books along with their subjects and academic levels, joining the nonfictionbook and books tables based on the book IDs.

Query Number 7:

Query:

SELECT t.TransactionId, b.Title

AS BookTitle, br.BorrowerName, br.Email, bs.BranchName, a.Nationality

AS AuthorNationality

FROM transactions t

INNER JOIN books b ON t.BookId = b.BookId

INNER JOIN borrowers br ON t.BorrowerId = br.BorrowerId

INNER JOIN branches bs ON t.branchid = bs.BranchId

INNER JOIN authors a ON b.AuthorId = a.AuthorId;

```
SELECT t.TransactionId, b.Title AS BookTitle, br.BorrowerName, br.Email, bs.BranchName, a.Nationality
AS AuthorNationality
FROM transactions t
INNER JOIN books b ON t.BookId = b.BookId
INNER JOIN borrowers br ON t.BorrowerId = br.BorrowerId
INNER JOIN branches bs ON t.branchid = bs.BranchId
INNER JOIN authors a ON b.AuthorId = a.AuthorId;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TransactionId	BookTitle	BorrowerName	Email	BranchName	AuthorNationality
6	The Complete Poems of Emily Dickinson	Jessica Taylor	jessica@gmail.com	Main Branch	American
1	Pride and Prejudice	John Smith	john@gmail.com	Downtown Branch	British
13	Pride and Prejudice	Alice Johnson	alice@gmail.com	Downtown Branch	British
3	War and Peace	Michael Brown	michael@gmail.com	City Center Branch	Russian
2	Great Expectations	Alice Johnson	alice@gmail.com	Westside Branch	British
10	The Old Man and the Sea	Emily Baker	emily@gmail.com	Westside Branch	American
9	The Great Gatsby	Daniel Clark	daniel@gmail.com	Eastside Branch	American
12	Harry Potter and the Philosopher's Stone	Olivia Adams	olivia@gmail.com	Northside Branch	British
4	To the Lighthouse	Sarah Wilson	sarah@gmail.com	Southside Branch	British

Result 17

Explanation:

This query retrieves transaction IDs, book titles, borrower names, borrower emails, branch names, and author nationalities by joining the transactions, books, borrowers, branches, and authors tables based on their respective IDs.

Query Number 8:

Query:

SELECT b.Title AS BookTitle, p.PublisherName, p.Address

AS PublisherAddress, p.PhoneNo AS PublisherPhone

FROM books b

INNER JOIN publishers p ON b.PublisherId = p.PublisherId;

```
SELECT b.Title AS BookTitle, p.PublisherName, p.Address AS PublisherAddress, p.PhoneNo AS PublisherPhone
FROM books b
INNER JOIN publishers p ON b.PublisherId = p.PublisherId;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

BookTitle	PublisherName	PublisherAddress	PublisherPhone
Pride and Prejudice	Penguin House	New York	12345678900
Harry Potter and the Philosopher's Stone	Penguin House	New York	12345678900
The Great Gatsby	Penguin House	New York	12345678900
The Lord of the Rings	Penguin House	New York	12345678900
The Stranger	Penguin House	New York	12345678900
The Sun Also Rises	Penguin House	New York	12345678900
The Bell Jar	Penguin House	New York	12345678900
Les Misérables	Penguin House	New York	12345678900
The Brothers Karamazov	Penguin House	New York	12345678900

Result 18

Explanation:

This query selects the titles of books along with their publisher names, addresses, and phone numbers by joining the books and publishers tables based on the publisher IDs.

Query Number 9:

Query:

SELECT b.BranchName, s.Name AS ManagerName

FROM branches b

INNER JOIN staff s ON b.BranchManagerId = s.StaffId;

```
SELECT b.BranchName, s.Name AS ManagerName
FROM branches b
INNER JOIN staff s ON b.BranchManagerId = s.StaffId;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

BranchName	ManagerName
Main Branch	John Doe
Downtown Branch	Jessica Miller
City Center Branch	William Anderson
Westside Branch	Sophia Martinez
Eastside Branch	James Taylor
Northside Branch	Sophia Clark
Southside Branch	Emma White
Central Branch	William Wilson
Midtown Branch	Christopher Moore

Explanation:

This query retrieves the branch names along with the names of their respective managers by joining the branches and staff tables based on the manager's staff ID.

Query Number 10:

Query:

SELECT p.PublisherName, COUNT(b.BookId)

AS NumberOfBooksPublished

FROM publishers p

LEFT JOIN books b ON p.PublisherId = b.PublisherId

GROUP BY p.PublisherName;

```
SELECT p.PublisherName, COUNT(b.BookId) AS NumberOfBooksPublished
FROM publishers p
LEFT JOIN books b ON p.PublisherId = b.PublisherId
GROUP BY p.PublisherName;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PublisherName	NumberOfBooksPublished
Penguin House	9
HarperCollins Publishers	9
Simon & Schuster	2
Macmillan Publishers	5
Hachette Livre	2
Wiley	3
Oxford University Press	4
Pearson	3
Scholastic Corporation	2

Explanation:

This query counts the number of books published by each publisher, listing their publisher names alongside the corresponding count. It performs a left join between the publishers and books tables on the publisher ID, grouping the results by publisher name.

Query Number 11:

Query:

SELECT a.AuthorName, COUNT(b.BookId)

AS NumberOfBooks

FROM authors a

LEFT JOIN books b ON a.AuthorId = b.AuthorId

GROUP BY a.AuthorName;

```
SELECT a.AuthorName, COUNT(b.BookId) AS NumberOfBooks
FROM authors a
LEFT JOIN books b ON a.AuthorId = b.AuthorId
GROUP BY a.AuthorName;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

AuthorName	NumberOfBooks
Jane Austen	7
Charles Dickens	7
Leo Tolstoy	6
Virginia Woolf	2
Mark Twain	4
Emily Dickinson	4
George Orwell	4
Charlotte Bronte	2
F. Scott Fitzgerald	2

Explanation:

This query counts the number of books authored by each author, displaying their names alongside the corresponding count. It utilizes a left join between the authors and books tables on the author ID, grouping the results by author name.

Query Number 12:

Query:

SELECT br.BorrowerName, b.Title

AS BookTitle

FROM borrowers br

INNER JOIN transactions t ON br.BorrowerId = t.BorrowerId

INNER JOIN books b ON t.BookId = b.BookId;

```
SELECT br.BorrowerName, b.Title AS BookTitle
FROM borrowers br
INNER JOIN transactions t ON br.BorrowerId = t.BorrowerId
INNER JOIN books b ON t.BookId = b.BookId;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

BorrowerName	BookTitle
John Smith	Pride and Prejudice
Alice Johnson	Great Expectations
Michael Brown	War and Peace
Sarah Wilson	To the Lighthouse
David Miller	The Adventures of Huckleberry Finn
Jessica Taylor	The Complete Poems of Emily Dickinson
Robert Anderson	1984
Laura Martinez	Jane Eyre
Daniel Clark	The Great Gatsby

Result 22

Explanation:

This query retrieves borrower names along with the titles of books they have borrowed by joining the borrowers, transactions, and books tables based on their respective IDs.

Query Number 13:

Query:

SELECT b.Title AS BookTitle, br.BorrowerName, t.IssueDate, t.DueDate, t.ReturnDate, t.FineAmount

FROM transactions t

INNER JOIN books b ON t.BookId = b.BookId

INNER JOIN borrowers br ON t.BorrowerId = br.BorrowerId

WHERE t.ReturnDate IS NOT NULL;

```
SELECT b.Title AS BookTitle, br.BorrowerName, t.IssueDate, t.DueDate, t.ReturnDate, t.FineAmount
FROM transactions t
INNER JOIN books b ON t.BookId = b.BookId
INNER JOIN borrowers br ON t.BorrowerId = br.BorrowerId
WHERE t.ReturnDate IS NOT NULL;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

BookTitle	BorrowerName	IssueDate	DueDate	ReturnDate	FineAmount
Pride and Prejudice	John Smith	2024-04-01	2024-04-15	2024-04-17	200
Great Expectations	Alice Johnson	2024-04-03	2024-04-17	2024-04-21	400
War and Peace	Michael Brown	2024-05-05	2024-05-19	2024-05-18	0
To the Lighthouse	Sarah Wilson	2024-06-07	2024-06-21	2024-06-26	500
The Adventures of Huckleberry Finn	David Miller	2024-07-09	2024-07-23	2024-07-23	0
The Complete Poems of Emily Dickinson	Jessica Taylor	2024-08-11	2024-08-25	2024-08-24	0
1984	Robert Anderson	2024-09-13	2024-09-27	2024-09-30	300
Jane Eyre	Laura Martinez	2024-10-15	2024-10-29	2024-10-30	100
The Great Gatsby	Daniel Clark	2024-11-17	2024-11-01	2024-11-01	0

Result 25

Explanation:

This query selects the title of books, borrower names, issue date, due date, return date, and fine amount for transactions where the return date is not null, by joining the transactions, books, and borrowers tables based on their respective IDs.

Query Number 14:

Query:

SELECT br.BorrowerName, COUNT(t.TransactionId)

AS TotalTransactions

FROM borrowers br

INNER JOIN transactions t ON br.BorrowerId = t.BorrowerId

GROUP BY br.BorrowerName;

```
SELECT br.BorrowerName, COUNT(t.TransactionId) AS TotalTransactions
FROM borrowers br
INNER JOIN transactions t ON br.BorrowerId = t.BorrowerId
GROUP BY br.BorrowerName;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

BorrowerName	TotalTransactions
John Smith	1
Alice Johnson	2
Michael Brown	2
Sarah Wilson	1
David Miller	1
Jessica Taylor	1
Robert Anderson	1
Laura Martinez	1
Daniel Clark	1

Explanation:

This query counts the total number of transactions for each borrower, displaying their names alongside the corresponding count. It performs an inner join between the borrowers and transactions tables on the borrower ID, grouping the results by borrower name.

Query Number 15:

Query:

SELECT b.MembershipType, SUM(t.FineAmount)

AS TotalFine

FROM borrowers b

INNER JOIN transactions t ON b.BorrowerId = t.BorrowerId

GROUP BY b.MembershipType

HAVINGTotalFine > 500;

```
SELECT b.MembershipType, SUM(t.FineAmount) AS TotalFine
FROM borrowers b
INNER JOIN transactions t ON b.BorrowerId = t.BorrowerId
GROUP BY b.MembershipType
HAVING TotalFine > 500;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

MembershipType	TotalFine
Regular	2100
Premium	1600

Explanation:

This query calculates the total fine amount for each membership type of borrowers, filtering the results to only include those with a total fine amount greater than 500. It involves an inner join between the borrowers and transactions tables on the borrower ID, grouping the results by membership type, and applying the filter using the HAVING clause.

Members:

M.Talha Qureshi (BSCS-23122)

Abdullah Hussain Yasim (BSCS-23008)

M.Ibrahim Butt (BSCS-23086)