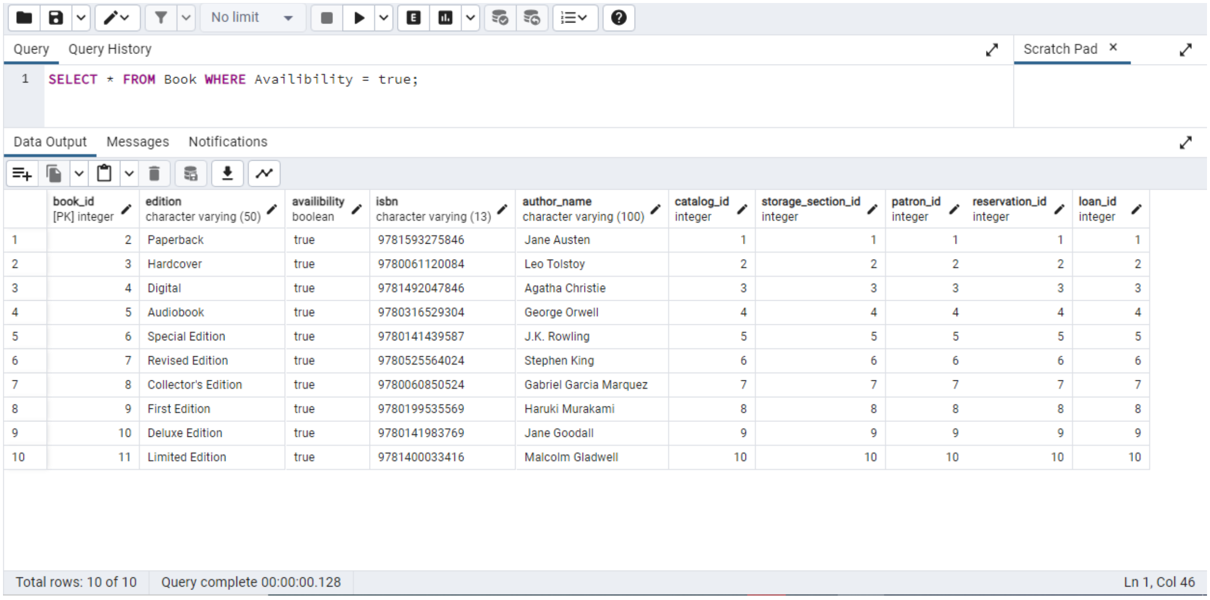
Principal Queries

1. Retrieve all books available in the library.

SELECT * FROM Book WHERE Availibility = true;

Output



```
SELECT * FROM Book WHERE Availibility = true;
```

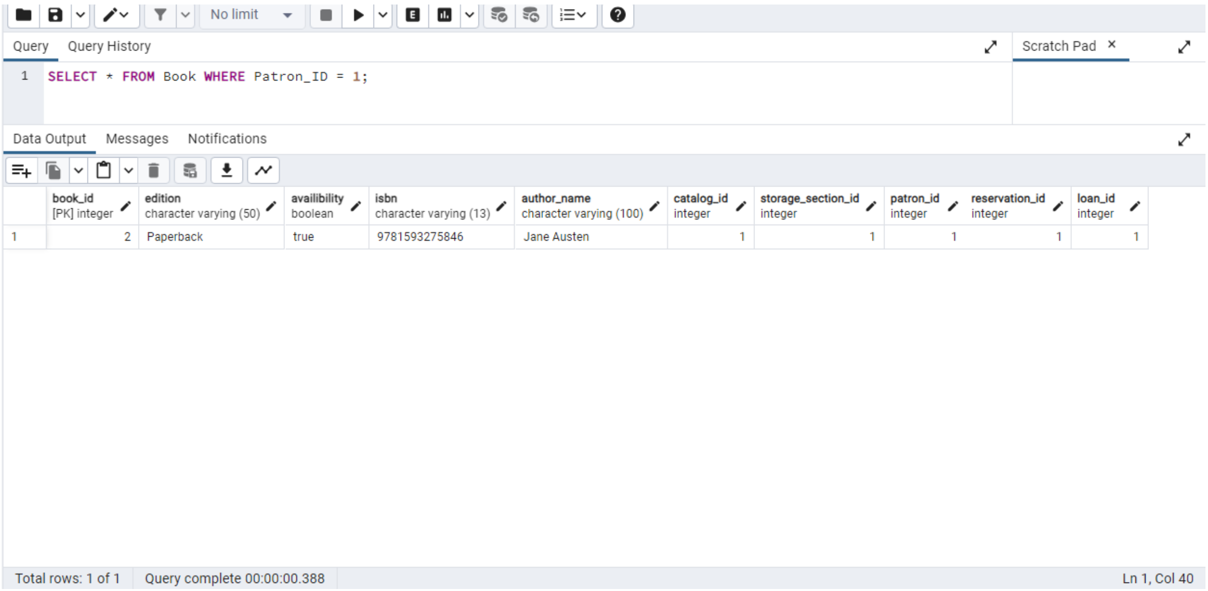
	book_id [PK] integer	edition character varying (50)	availibility boolean	isbn character varying (13)	author_name character varying (100)	catalog_id integer	storage_section_id integer	patron_id integer	reservation_id integer	loan_id integer
1	2	Paperback	true	9781593275846	Jane Austen	1	1	1	1	1
2	3	Hardcover	true	9780061120084	Leo Tolstoy	2	2	2	2	2
3	4	Digital	true	9781492047846	Agatha Christie	3	3	3	3	3
4	5	Audiobook	true	9780316529304	George Orwell	4	4	4	4	4
5	6	Special Edition	true	9780141439587	J.K. Rowling	5	5	5	5	5
6	7	Revised Edition	true	9780525564024	Stephen King	6	6	6	6	6
7	8	Collector's Edition	true	9780060850524	Gabriel Garcia Marquez	7	7	7	7	7
8	9	First Edition	true	9780199535569	Haruki Murakami	8	8	8	8	8
9	10	Deluxe Edition	true	9780141983769	Jane Goodall	9	9	9	9	9
10	11	Limited Edition	true	9781400033416	Malcolm Gladwell	10	10	10	10	10

Total rows: 10 of 10 Query complete 00:00:00.128 Ln 1, Col 46

2. Find books borrowed by a specific patron:

SELECT * FROM Book WHERE Patron_ID = 1;

Output



```
SELECT * FROM Book WHERE Patron_ID = 1;
```

	book_id [PK] integer	edition character varying (50)	availibility boolean	isbn character varying (13)	author_name character varying (100)	catalog_id integer	storage_section_id integer	patron_id integer	reservation_id integer	loan_id integer
1	2	Paperback	true	9781593275846	Jane Austen	1	1	1	1	1

Total rows: 1 of 1 Query complete 00:00:00.388 Ln 1, Col 40

3. Count the number of books in each storage section:

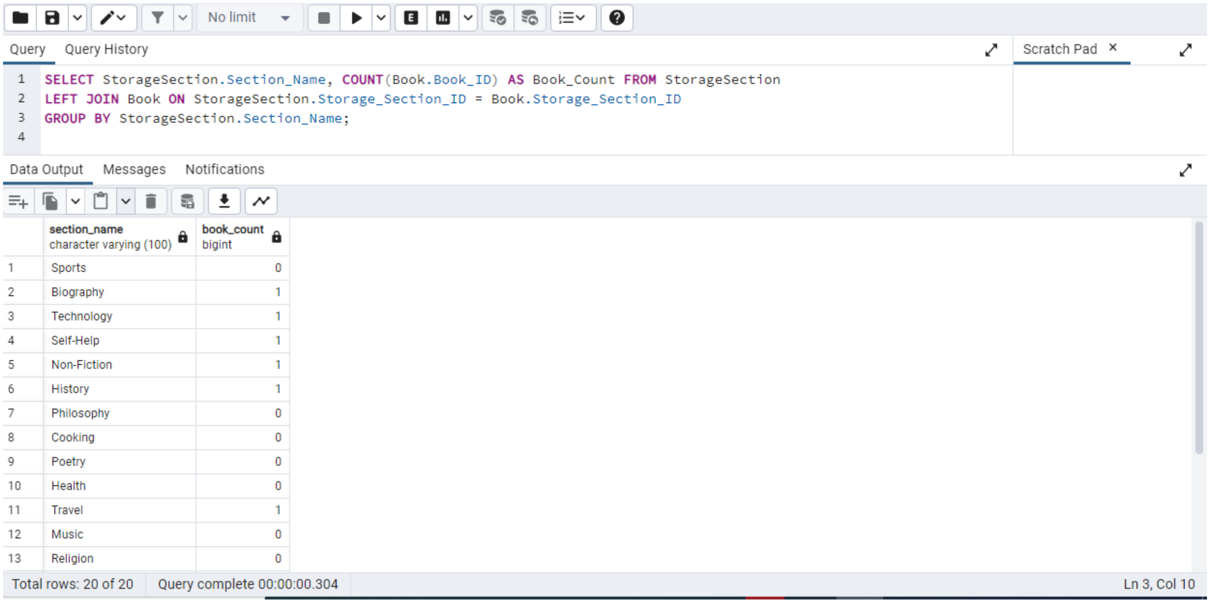
SELECT

StorageSection.Section_Name,

COUNT(Book.Book_ID) AS Book_Count

```
FROM
   StorageSection
LEFT JOIN Book ON StorageSection.Storage_Section_ID = Book.Storage_Section_ID
GROUP BY
   StorageSection.Section_Name;
```

Output



```
SELECT StorageSection.Section_Name, COUNT(Book.Book_ID) AS Book_Count FROM StorageSection
LEFT JOIN Book ON StorageSection.Storage_Section_ID = Book.Storage_Section_ID
GROUP BY StorageSection.Section_Name;
```

	section_name	book_count
	character varying (100)	bigint
1	Sports	0
2	Biography	1
3	Technology	1
4	Self-Help	1
5	Non-Fiction	1
6	History	1
7	Philosophy	0
8	Cooking	0
9	Poetry	0
10	Health	0
11	Travel	1
12	Music	0
13	Religion	0

Total rows: 20 of 20 Query complete 00:00:00.304

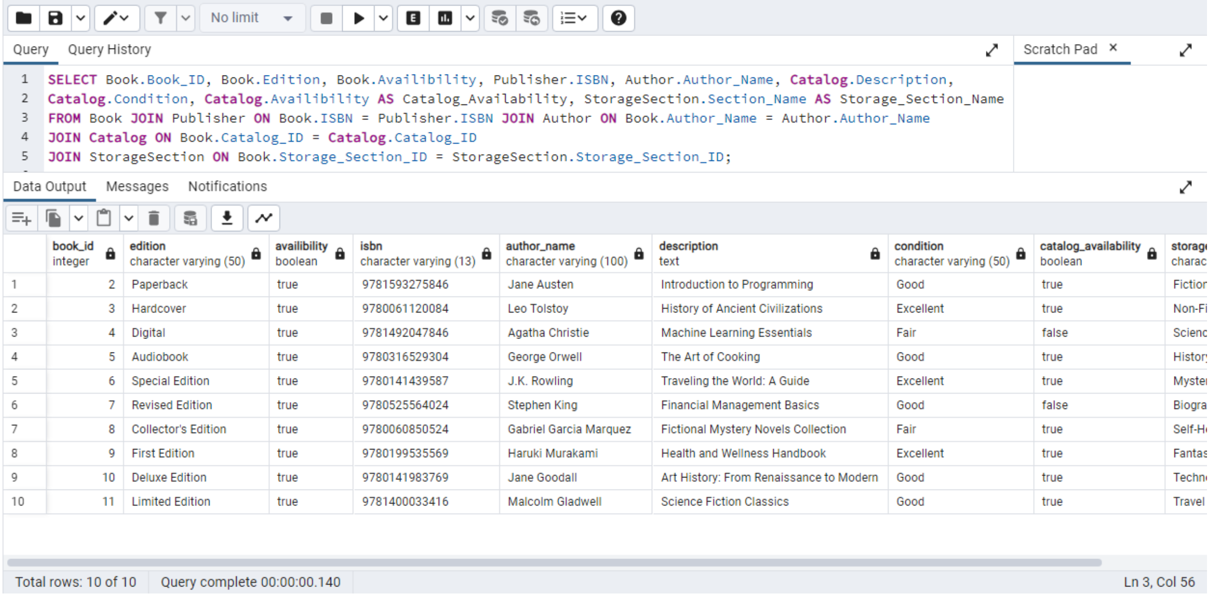
4. Retrieve all books along with their availability status and storage section names:

```
SELECT
   Book.Book_ID,
   Book.Edition,
   Book.Availibility,
   Publisher.ISBN,
   Author.Author_Name,
   Catalog.Description,
   Catalog.Condition,
   Catalog.Availibility AS Catalog_Availability,
   StorageSection.Section_Name AS Storage_Section_Name
FROM
   Book
JOIN Publisher ON Book.ISBN = Publisher.ISBN
JOIN Author ON Book.Author_Name = Author.Author_Name
```

JOIN Catalog ON Book.Catalog_ID = Catalog.Catalog_ID

JOIN StorageSection ON Book.Storage_Section_ID = StorageSection.Storage_Section_ID;

Output



```sql
SELECT Book.Book_ID, Book.Edition, Book.Availibility, Publisher.ISBN, Author.Author_Name, Catalog.Description,
Catalog.Condition, Catalog.Availibility AS Catalog_Availability, StorageSection.Section_Name AS Storage_Section_Name
FROM Book JOIN Publisher ON Book.ISBN = Publisher.ISBN JOIN Author ON Book.Author_Name = Author.Author_Name
JOIN Catalog ON Book.Catalog_ID = Catalog.Catalog_ID
JOIN StorageSection ON Book.Storage_Section_ID = StorageSection.Storage_Section_ID;
```

	book_id	edition	availibility	isbn	author_name	description	condition	catalog_availability	storag
	integer	character varying (50)	boolean	character varying (13)	character varying (100)	text	character varying (50)	boolean	charac
1	2	Paperback	true	9781593275846	Jane Austen	Introduction to Programming	Good	true	Fictior
2	3	Hardcover	true	9780061120084	Leo Tolstoy	History of Ancient Civilizations	Excellent	true	Non-Fi
3	4	Digital	true	9781492047846	Agatha Christie	Machine Learning Essentials	Fair	false	Scienc
4	5	Audiobook	true	9780316529304	George Orwell	The Art of Cooking	Good	true	History
5	6	Special Edition	true	9780141439587	J.K. Rowling	Traveling the World: A Guide	Excellent	true	Myster
6	7	Revised Edition	true	9780525564024	Stephen King	Financial Management Basics	Good	false	Biogra
7	8	Collector's Edition	true	9780060850524	Gabriel Garcia Marquez	Fictional Mystery Novels Collection	Fair	true	Self-H
8	9	First Edition	true	9780199535569	Haruki Murakami	Health and Wellness Handbook	Excellent	true	Fantas
9	10	Deluxe Edition	true	9780141983769	Jane Goodall	Art History: From Renaissance to Modern	Good	true	Techn
10	11	Limited Edition	true	9781400033416	Malcolm Gladwell	Science Fiction Classics	Good	true	Travel

Total rows: 10 of 10 Query complete 00:00:00.140 Ln 3, Col 56

5. Retrieve the total late fees collected:

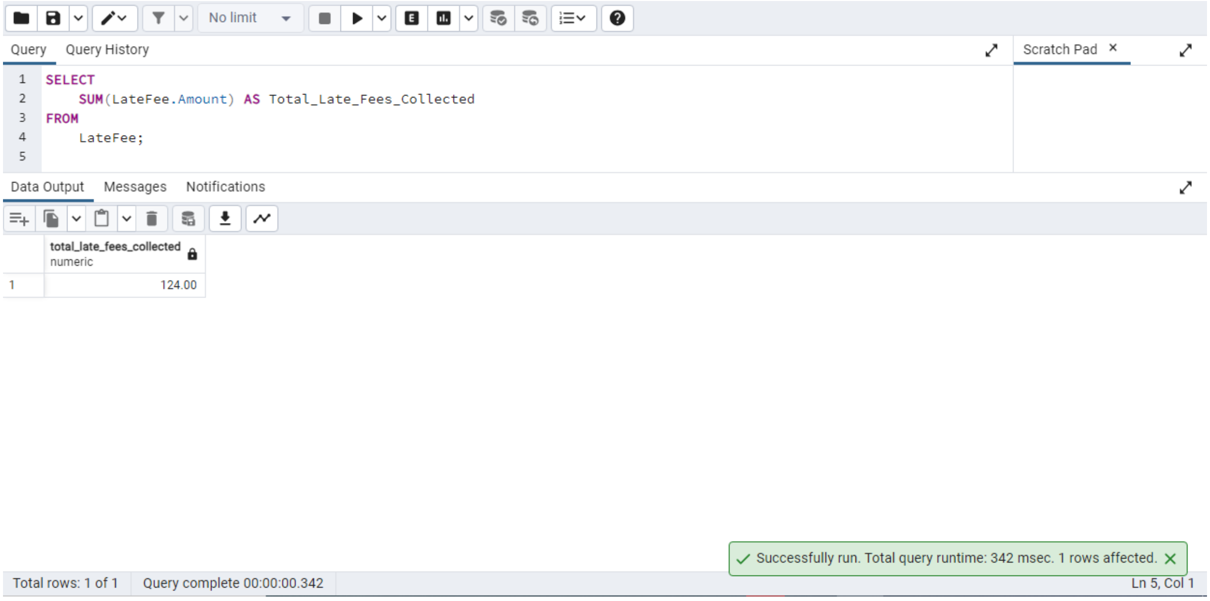
SELECT

SUM(LateFee.Amount) AS Total_Late_Fees_Collected

FROM

LateFee;

Output



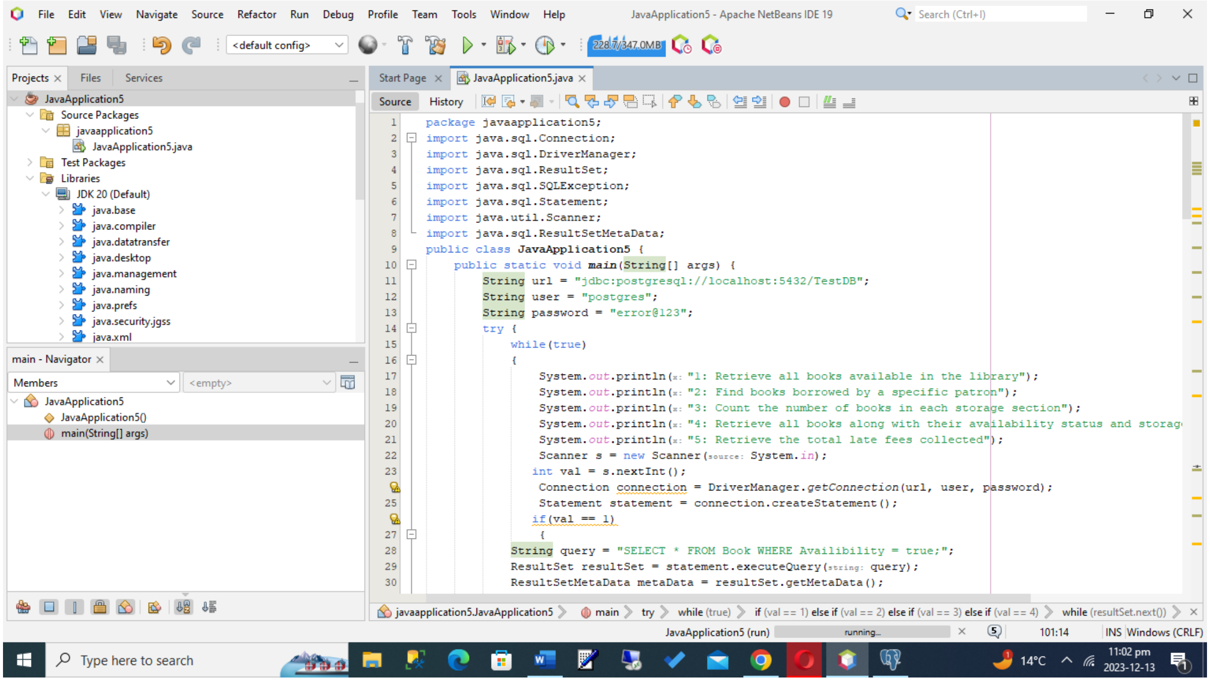
```sql
SELECT
    SUM(LateFee.Amount) AS Total_Late_Fees_Collected
FROM
    LateFee;
```

	total_late_fees_collected
	numeric
1	124.00

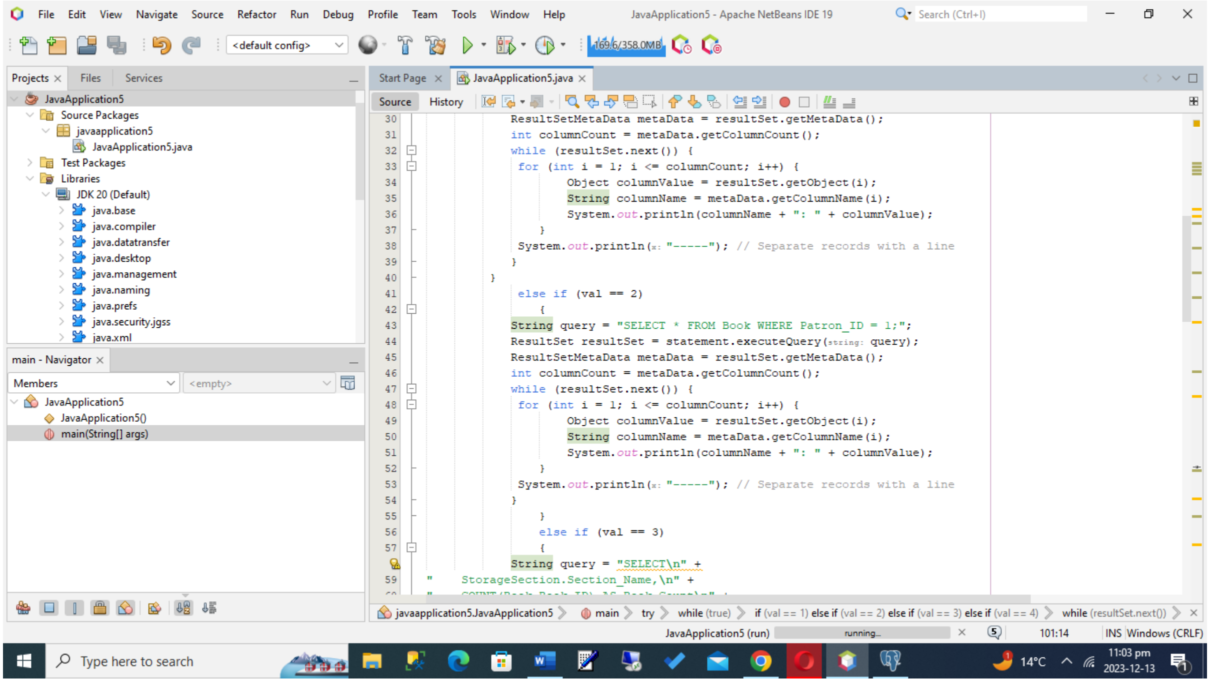
✓ Successfully run. Total query runtime: 342 msec. 1 rows affected.

Total rows: 1 of 1 Query complete 00:00:00.342 Ln 5, Col 1

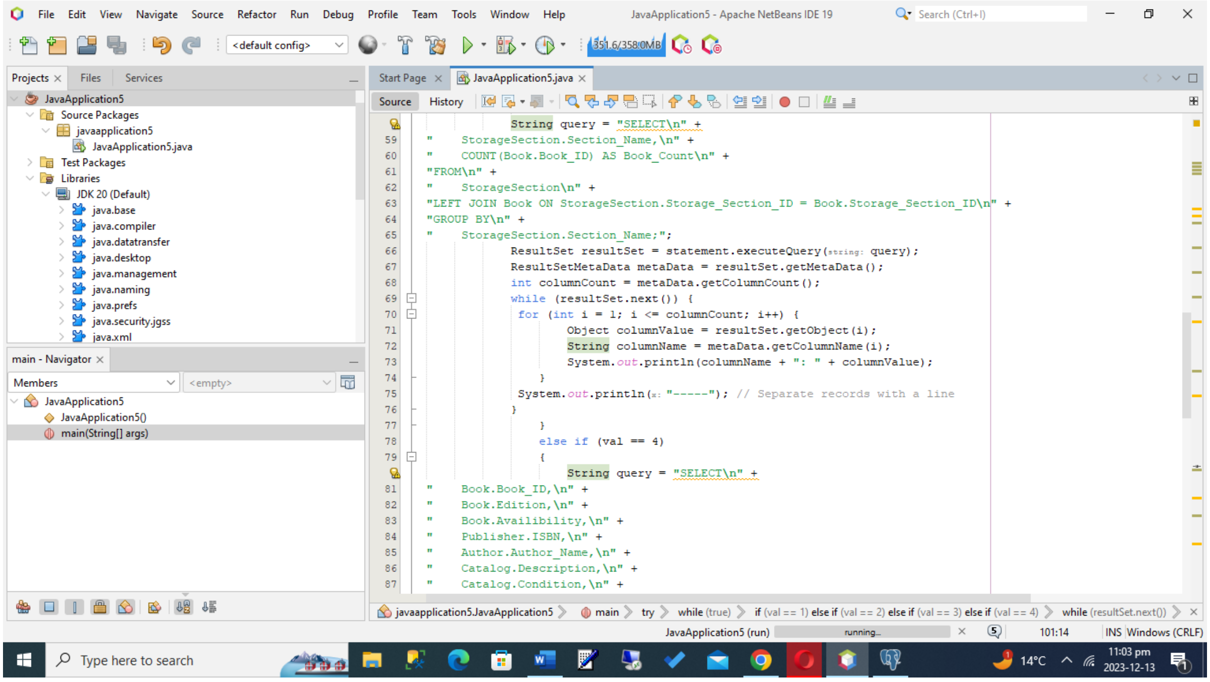
JDBC Implementations of the Principal Queries and Visualization



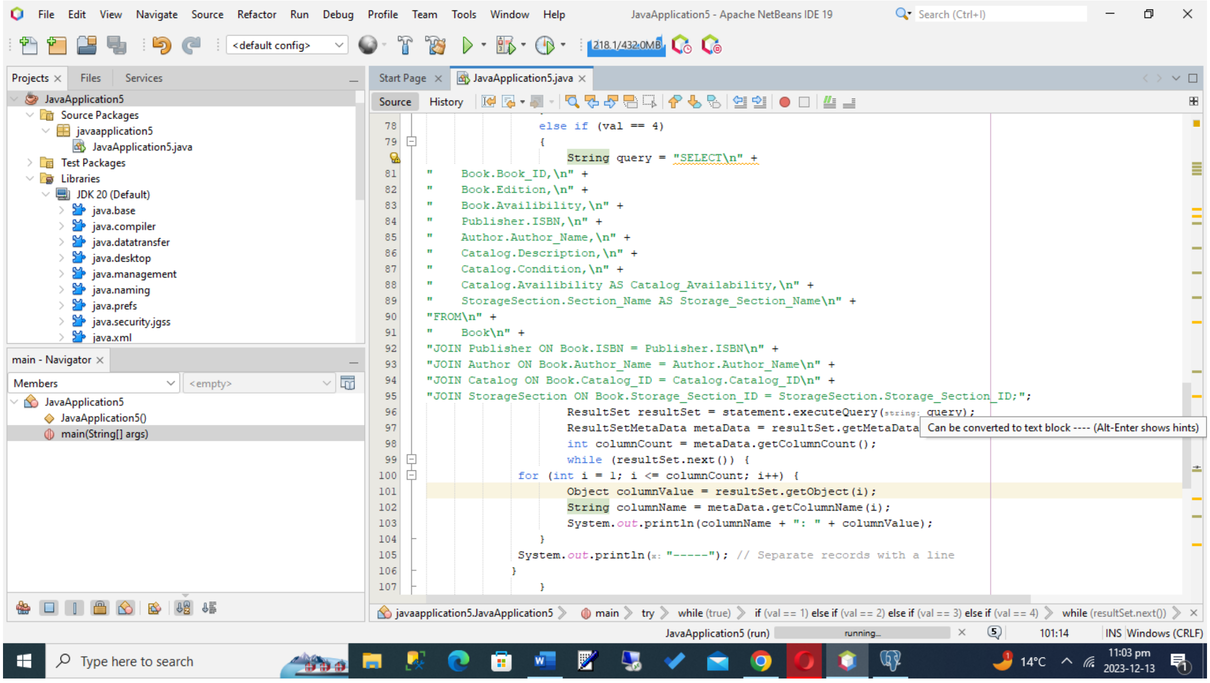
```java
package javaapplication5;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;
import java.sql.ResultSetMetaData;
public class JavaApplication5 {
    public static void main(String[] args) {
        String url = "jdbc:postgresql://localhost:5432/TestDB";
        String user = "postgres";
        String password = "error@123";
        try {
            while(true)
            {
                System.out.println("1: Retrieve all books available in the library");
                System.out.println("2: Find books borrowed by a specific patron");
                System.out.println("3: Count the number of books in each storage section");
                System.out.println("4: Retrieve all books along with their availability status and storag
                System.out.println("5: Retrieve the total late fees collected");
                Scanner s = new Scanner(System.in);
              int val = s.nextInt();
                Connection connection = DriverManager.getConnection(url, user, password);
                Statement statement = connection.createStatement();
              if(val == 1)
                {
            String query = "SELECT * FROM Book WHERE Availibility = true;";
            ResultSet resultSet = statement.executeQuery(query);
            ResultSetMetaData metaData = resultSet.getMetaData();
```



```java
            ResultSetMetaData metaData = resultSet.getMetaData();
            int columnCount = metaData.getColumnCount();
            while (resultSet.next()) {
             for (int i = 1; i <= columnCount; i++) {
                    Object columnValue = resultSet.getObject(i);
                    String columnName = metaData.getColumnName(i);
                    System.out.println(columnName + ": " + columnValue);
                }
             System.out.println("-----"); // Separate records with a line
            }
        }
             else if (val == 2)
                {
            String query = "SELECT * FROM Book WHERE Patron_ID = 1;";
            ResultSet resultSet = statement.executeQuery(query);
            ResultSetMetaData metaData = resultSet.getMetaData();
            int columnCount = metaData.getColumnCount();
            while (resultSet.next()) {
             for (int i = 1; i <= columnCount; i++) {
                    Object columnValue = resultSet.getObject(i);
                    String columnName = metaData.getColumnName(i);
                    System.out.println(columnName + ": " + columnValue);
                }
             System.out.println("-----"); // Separate records with a line
            }
                }
                else if (val == 3)
                {
            String query = "SELECT\n" +
"    StorageSection.Section_Name,\n" +
```

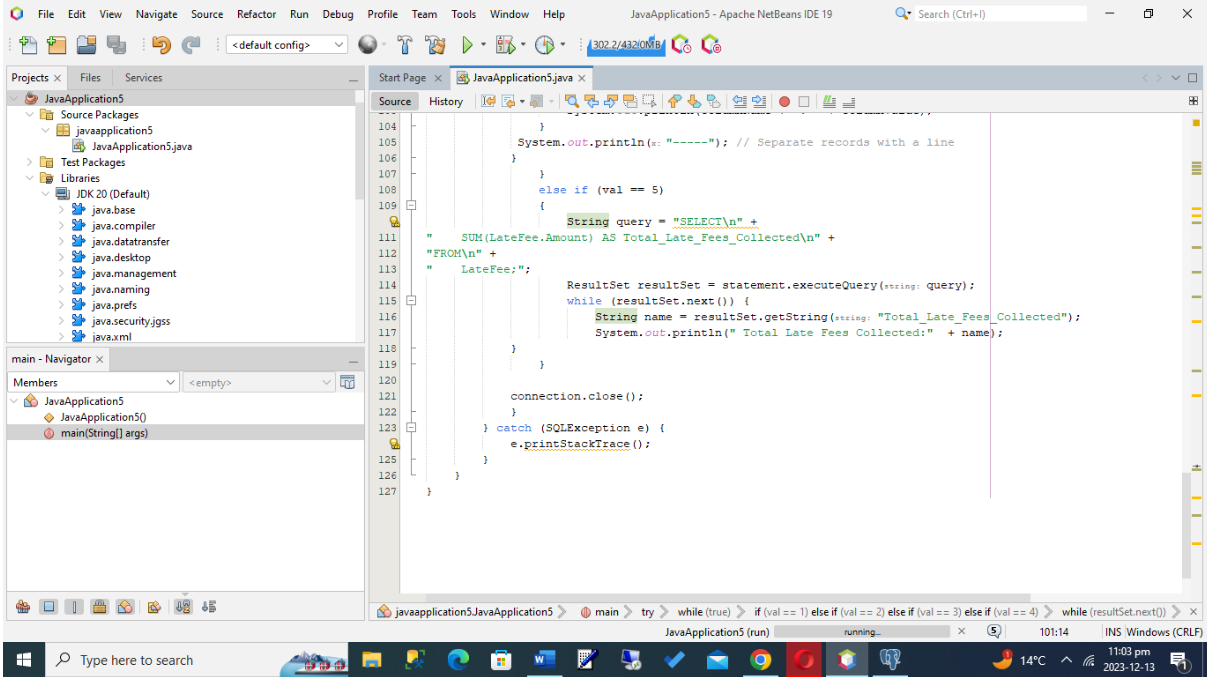


```
                    String query = "SELECT\n" +
"     StorageSection.Section_Name,\n" +
"     COUNT(Book.Book_ID) AS Book_Count\n" +
"FROM\n" +
"     StorageSection\n" +
"LEFT JOIN Book ON StorageSection.Storage_Section_ID = Book.Storage_Section_ID\n" +
"GROUP BY\n" +
"     StorageSection.Section_Name;";
                    ResultSet resultSet = statement.executeQuery(string: query);
                    ResultSetMetaData metaData = resultSet.getMetaData();
                    int columnCount = metaData.getColumnCount();
                    while (resultSet.next()) {
                     for (int i = 1; i <= columnCount; i++) {
                            Object columnValue = resultSet.getObject(i);
                            String columnName = metaData.getColumnName(i);
                            System.out.println(columnName + ": " + columnValue);
                        }
                     System.out.println(x: "-----"); // Separate records with a line
                    }
                        }
                        else if (val == 4)
                        {
                            String query = "SELECT\n" +
"     Book.Book_ID,\n" +
"     Book.Edition,\n" +
"     Book.Availibility,\n" +
"     Publisher.ISBN,\n" +
"     Author.Author_Name,\n" +
"     Catalog.Description,\n" +
"     Catalog.Condition,\n" +
```



```
                        else if (val == 4)
                        {
                            String query = "SELECT\n" +
"     Book.Book_ID,\n" +
"     Book.Edition,\n" +
"     Book.Availibility,\n" +
"     Publisher.ISBN,\n" +
"     Author.Author_Name,\n" +
"     Catalog.Description,\n" +
"     Catalog.Condition,\n" +
"     Catalog.Availibility AS Catalog_Availability,\n" +
"     StorageSection.Section_Name AS Storage_Section_Name\n" +
"FROM\n" +
"     Book\n" +
"JOIN Publisher ON Book.ISBN = Publisher.ISBN\n" +
"JOIN Author ON Book.Author_Name = Author.Author_Name\n" +
"JOIN Catalog ON Book.Catalog_ID = Catalog.Catalog_ID\n" +
"JOIN StorageSection ON Book.Storage_Section_ID = StorageSection.Storage_Section_ID;";
                            ResultSet resultSet = statement.executeQuery(string: query);
                            ResultSetMetaData metaData = resultSet.getMetaData
                            int columnCount = metaData.getColumnCount();
                            while (resultSet.next()) {
                     for (int i = 1; i <= columnCount; i++) {
                            Object columnValue = resultSet.getObject(i);
                            String columnName = metaData.getColumnName(i);
                            System.out.println(columnName + ": " + columnValue);
                        }
                     System.out.println(x: "-----"); // Separate records with a line
                    }
                        }
```

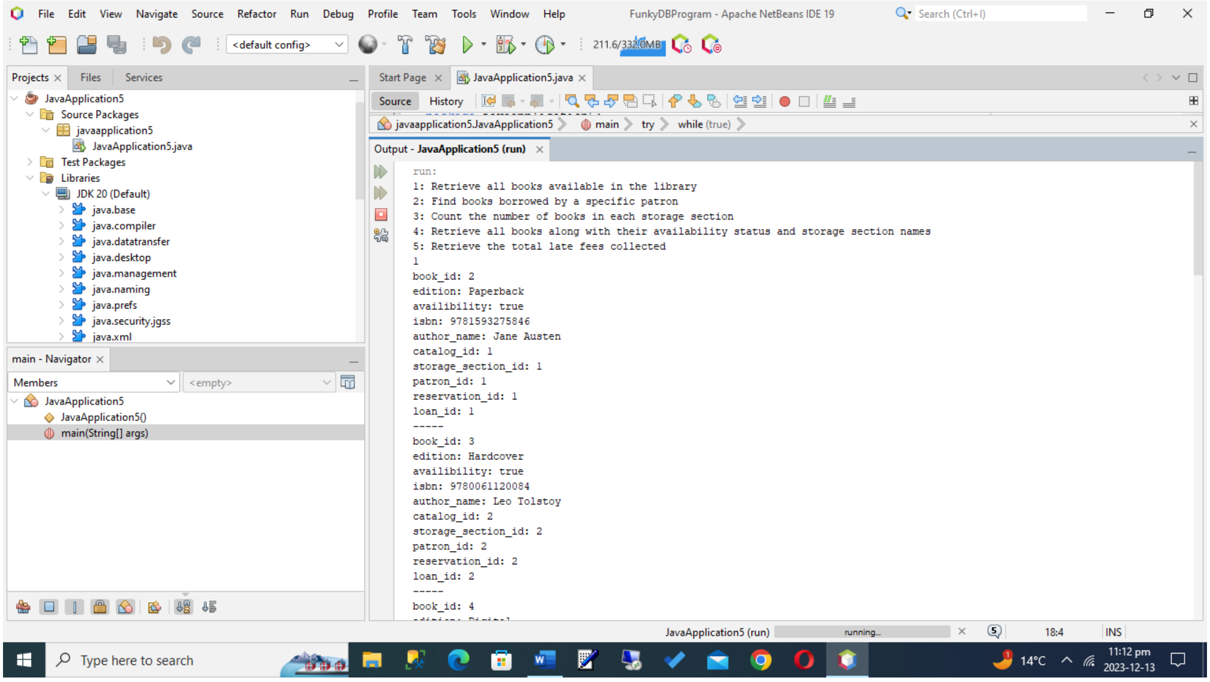
Can be converted to text block ---- (Alt-Enter shows hints)



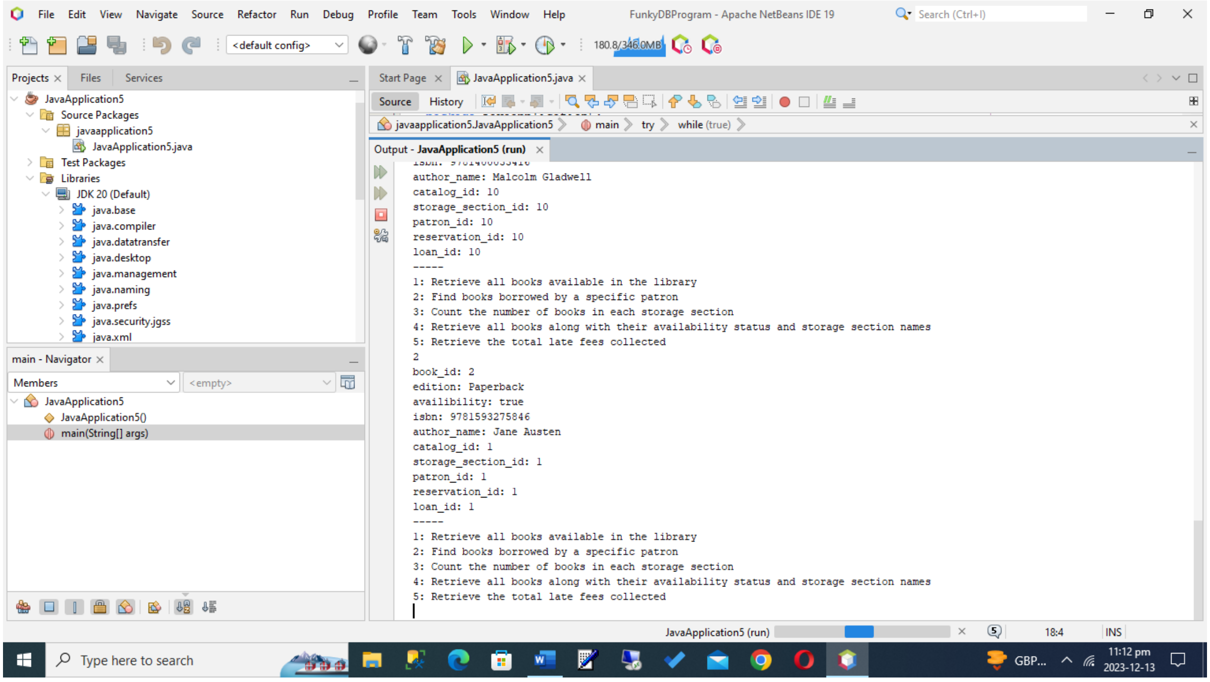
```
                    }
                System.out.println("-----"); // Separate records with a line
            }
                    }
                    else if (val == 5)
                    {
                        String query = "SELECT\n" +
"    SUM(LateFee.Amount) AS Total_Late_Fees_Collected\n" +
"FROM\n" +
"    LateFee;";
                        ResultSet resultSet = statement.executeQuery(query);
                        while (resultSet.next()) {
                            String name = resultSet.getString("Total_Late_Fees_Collected");
                            System.out.println(" Total Late Fees Collected:"  + name);
            }
                    }

            connection.close();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

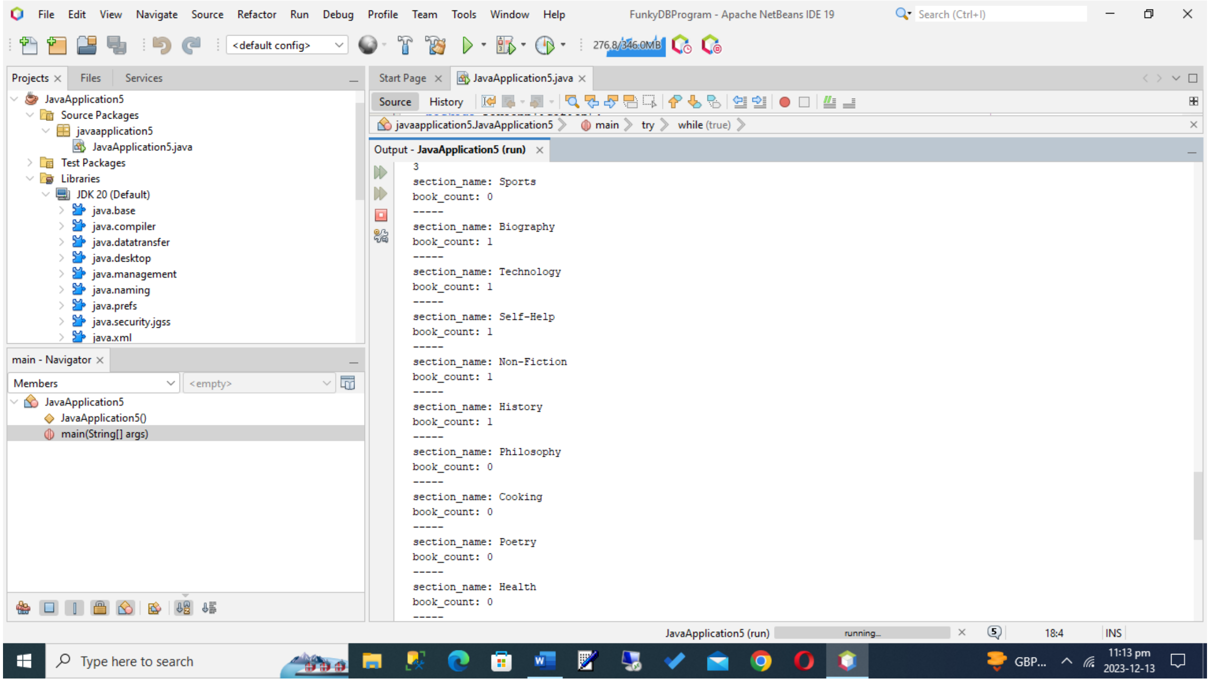
Result:



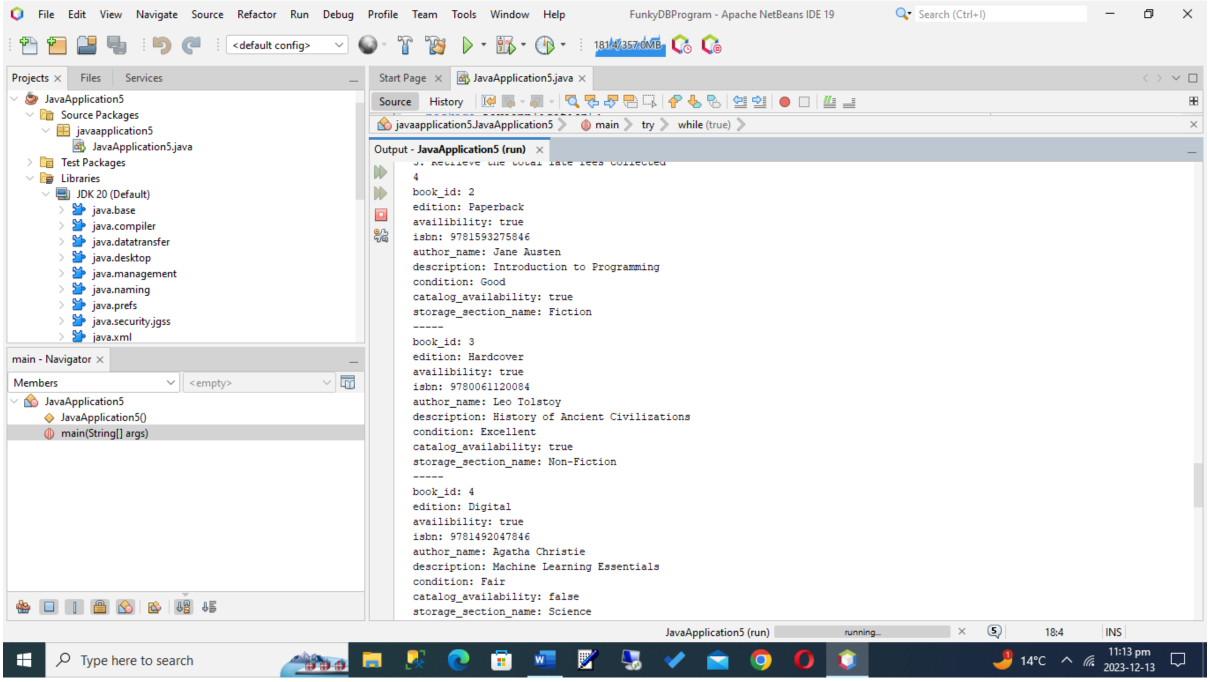
```
run:
1: Retrieve all books available in the library
2: Find books borrowed by a specific patron
3: Count the number of books in each storage section
4: Retrieve all books along with their availability status and storage section names
5: Retrieve the total late fees collected
1
book_id: 2
edition: Paperback
availibility: true
isbn: 9781593275846
author_name: Jane Austen
catalog_id: 1
storage_section_id: 1
patron_id: 1
reservation_id: 1
loan_id: 1
-----
book_id: 3
edition: Hardcover
availibility: true
isbn: 9780061120084
author_name: Leo Tolstoy
catalog_id: 2
storage_section_id: 2
patron_id: 2
reservation_id: 2
loan_id: 2
-----
book_id: 4
```



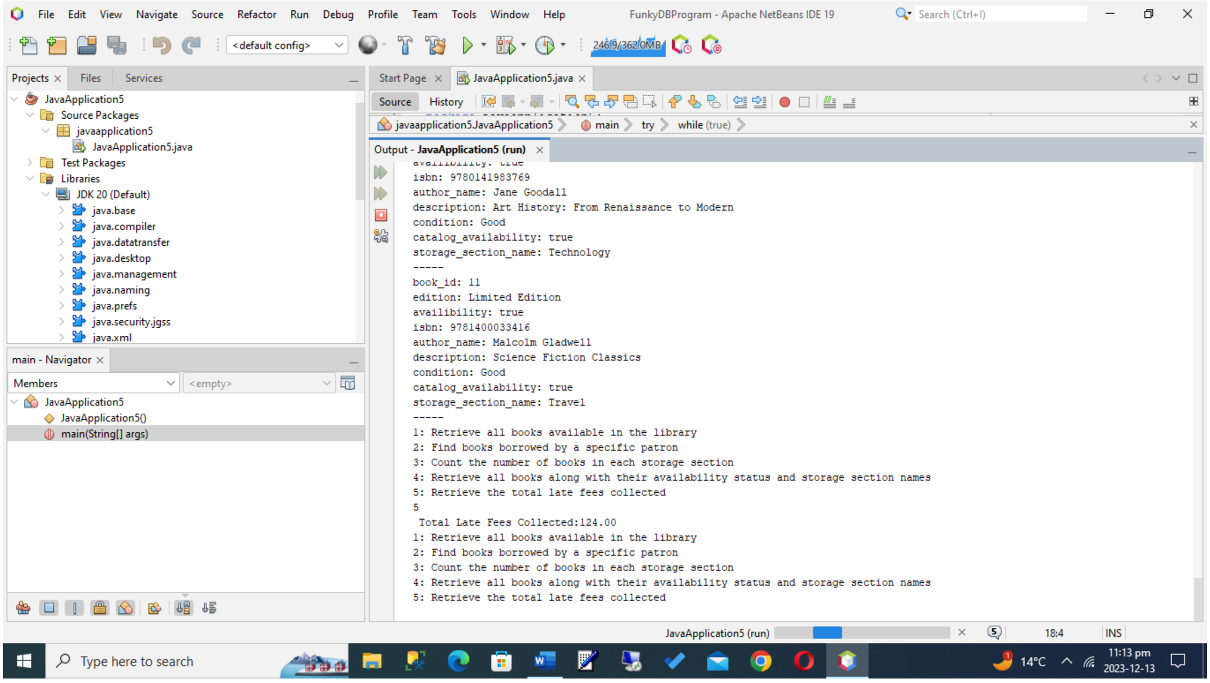
```
isbn: 9781400063416
author_name: Malcolm Gladwell
catalog_id: 10
storage_section_id: 10
patron_id: 10
reservation_id: 10
loan_id: 10
-----
1: Retrieve all books available in the library
2: Find books borrowed by a specific patron
3: Count the number of books in each storage section
4: Retrieve all books along with their availability status and storage section names
5: Retrieve the total late fees collected
2
book_id: 2
edition: Paperback
availibility: true
isbn: 9781593275846
author_name: Jane Austen
catalog_id: 1
storage_section_id: 1
patron_id: 1
reservation_id: 1
loan_id: 1
-----
1: Retrieve all books available in the library
2: Find books borrowed by a specific patron
3: Count the number of books in each storage section
4: Retrieve all books along with their availability status and storage section names
5: Retrieve the total late fees collected
```



```
3
section_name: Sports
book_count: 0
-----
section_name: Biography
book_count: 1
-----
section_name: Technology
book_count: 1
-----
section_name: Self-Help
book_count: 1
-----
section_name: Non-Fiction
book_count: 1
-----
section_name: History
book_count: 1
-----
section_name: Philosophy
book_count: 0
-----
section_name: Cooking
book_count: 0
-----
section_name: Poetry
book_count: 0
-----
section_name: Health
book_count: 0
-----
```



```
5: Retrieve the total late fees collected
4
book_id: 2
edition: Paperback
availibility: true
isbn: 9781593275846
author_name: Jane Austen
description: Introduction to Programming
condition: Good
catalog_availability: true
storage_section_name: Fiction
-----
book_id: 3
edition: Hardcover
availibility: true
isbn: 9780061120084
author_name: Leo Tolstoy
description: History of Ancient Civilizations
condition: Excellent
catalog_availability: true
storage_section_name: Non-Fiction
-----
book_id: 4
edition: Digital
availibility: true
isbn: 9781492047846
author_name: Agatha Christie
description: Machine Learning Essentials
condition: Fair
catalog_availability: false
storage_section_name: Science
```



```
availibility: true
isbn: 9780141983769
author_name: Jane Goodall
description: Art History: From Renaissance to Modern
condition: Good
catalog_availability: true
storage_section_name: Technology
-----
book_id: 11
edition: Limited Edition
availibility: true
isbn: 9781400033416
author_name: Malcolm Gladwell
description: Science Fiction Classics
condition: Good
catalog_availability: true
storage_section_name: Travel
-----
1: Retrieve all books available in the library
2: Find books borrowed by a specific patron
3: Count the number of books in each storage section
4: Retrieve all books along with their availability status and storage section names
5: Retrieve the total late fees collected
5
 Total Late Fees Collected:124.00
1: Retrieve all books available in the library
2: Find books borrowed by a specific patron
3: Count the number of books in each storage section
4: Retrieve all books along with their availability status and storage section names
5: Retrieve the total late fees collected
```