

LIBRARY MANAGEMENT SYSTEM

DATABASE DESIGN & SQL

FSDM 2023S

Student

Piyumika Samarasuriyage
C0900440

Submitted to

Prof. Sagara Samarawickrama

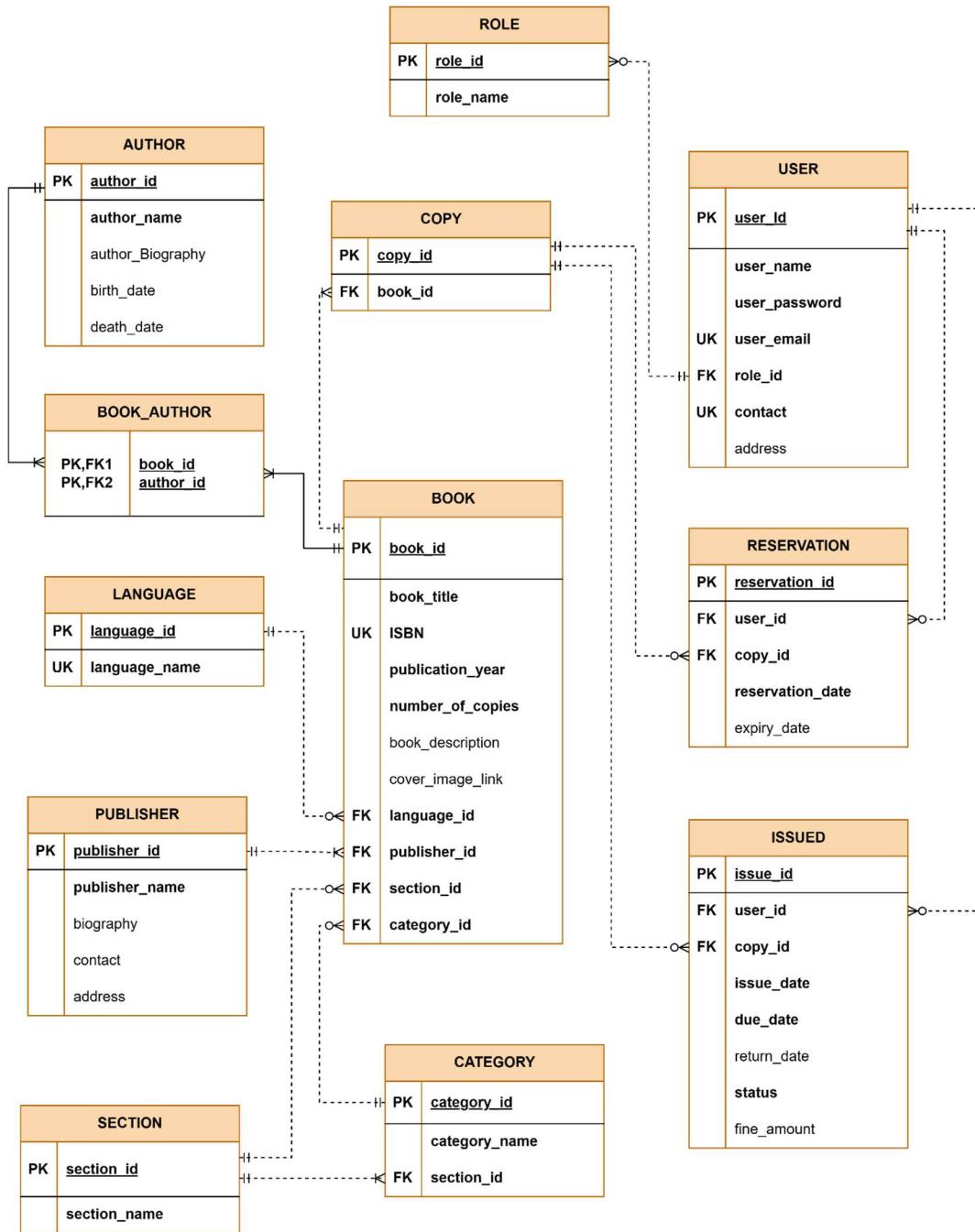
Table of Contents

1.	INTRODUCTION	3
2.	Created database tables	4
2.1	AUTHORS_440	4
2.2	LANGUAGES_440	4
2.3	PUBLISHERS_440	5
2.4	SECTIONS_440.....	5
2.5	CATEGORIES_440	6
2.8	BOOKS_440.....	7
2.9	BOOK_AUTHORS_440	8
2.11	RESERVATIOS_440	9
2.12	ISSUES_440	9
3.	SQL QUERIES	10
3.1	Category A	10
3.2	Category B	12
3.3	Category C	14
3.4	Category D	17
3.5	Category E	21
3.6	Category F.....	23
3.7	Advance Category	27

1. INTRODUCTION

This project report presents the implementation of physical database related to the Library Management System created in the group project. I have added sample records to all the tables and this report contains how SQL queries can be used to query data from those tables based on the knowledge gained in the course. All the scripts used in this individual project is presented in the attached text file.

The Entity Relationship Diagram during the group project is as follows.



2. Created database tables

The scripts used for creating the database is presented in the text file – database_creation.txt.

2.1 AUTHORS_440

The screenshot shows the Oracle SQL Developer interface. On the left, the Object Navigator displays the schema structure under the PiyumikaDB connection. In the center, the SQL Worksheet pane contains the following SQL code:

```
// Display all the tables
SELECT * FROM AUTHORS_440;
```

Below the code, the Query Result pane shows the data from the AUTHORS_440 table:

AUTHOR_ID	AUTHOR_NAME	AUTHOR_BIOGRAPHY	BIRTH_DATE	DEATH_DATE
N00001	SHAKESPEARE	Greatest writer in the English language.	(null)	(null)
N00002	JANE AUSTEN	English novelist known for her six major novels.	75-07-01	17-02-12
F00001	J.K.ROWLING	British author best known for writing the Harry Potter series.	(null)	(null)
N00003	LEO TOLSTOY	A Russian novelist famous for epic novels in the exploration of human nature and society.	(null)	(null)
S00013	CHARLES DARWIN	Darwin theory of evolution by natural selection revolutionized biology and life.	(null)	(null)
S00133	Stephen Hawking	Introduced complex cosmological concepts to the general public.	42-08-01	18-08-01
S03433	Oliver Sacks	A neurologist and author who explored human brain.	33-03-21	15-04-05
H05969	Desmond Morton	Canadian historian known for Canadian military and political history.	37-08-01	22-08-01

2.2 LANGUAGES_440

The screenshot shows the Oracle SQL Developer interface. On the left, the Object Navigator displays the schema structure under the PiyumikaDB connection. In the center, the SQL Worksheet pane contains the following SQL code:

```
SELECT * FROM AUTHORS_440;
SELECT * FROM LANGUAGES_440;
SELECT * FROM PUBLISHERS_440;
SELECT * FROM SECTIONS_440;
SELECT * FROM CATEGORIES_440;
SELECT * FROM ROLES_440;
SELECT * FROM USERS_440;
SELECT * FROM BOOKS_440;
```

Below the code, the Query Result pane shows the data from the LANGUAGES_440 table:

LANGUAGE_ID	LANGUAGE_NAME
1	ENGLISH
2	FRENCH
3	MANDARIN
4	PANJABI
5	SPANISH
6	TAMIL
7	RUSSIAN

2.3 PUBLISHERS_440

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the database structure under 'Assignments' for the 'PiyumikaDB' connection, including Tables, Views, Indexes, Packages, Procedures, Functions, Operators, Queues, Queues Tables, Triggers, Types, and Sequences. Below this is a 'Reports' section with options like All Reports, Analytic View Reports, Data Dictionary Reports, and Data Modeler Reports. The main workspace shows a worksheet with the following SQL code:

```
SELECT * FROM AUTHORS_440;
SELECT * FROM LANGUAGES_440;
SELECT * FROM PUBLISHERS_440;
SELECT * FROM SECTIONS_440;
```

The results pane displays the data from the PUBLISHERS_440 table:

PUBLISHER_ID	PUBLISHER_NAME	BIOGRAPHY	CONTACT	ADDRESS
1 1466	Penguin Random House	PUBLISHERS TO WORLD OVER 90 YEARS	8679055509 13,HEART RD,NEWSOUTH WALES	
2 2246	McClelland And Stewart	PUBLISHERS TO NATION OVER DECADES.	9059999999 123,BRUNEL RD,MISSISSAUGA	
3 2324	NOVEL PRINTERS	READERS FRIEND FOR 3 DECADES.	6474327890 96,BAKER ST,TORONTO	
4 3256	John Murray	Publisher OF scholarly and educational books.	4359687770 89,OXFORD ST, UK	
5 3456	Bloomsbury	FAMOUS FOR FICTION, NON-FICTION AND ACADEMIC TITLES	9059999999 12,CART RD,TORONTO	
6 3457	Simon And Schuster	Wide range of genres, including fiction, non-fiction, and academic titles	6547892345 2A,BEACHSURF RD,BRAMPTON	
7 7865	HarperCollins	GLOBAL PUBLISHER FAMOUS FOR CHILDREN PUBLISHES	437859400 3B,HEARTLAND RD,CALIFORNIA	

2.4 SECTIONS_440

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the database structure under 'Assignments' for the 'PiyumikaDB' connection, including Tables, Views, Indexes, Packages, Procedures, Functions, Operators, Queues, Queues Tables, Triggers, Types, and Sequences. Below this is a 'Reports' section with options like All Reports, Analytic View Reports, Data Dictionary Reports, and Data Modeler Reports. The main workspace shows a worksheet with the following SQL code:

```
SELECT * FROM LANGUAGES_440;
SELECT * FROM PUBLISHERS_440;
SELECT * FROM SECTIONS_440;
SELECT * FROM CATEGORIES_440;
```

The results pane displays the data from the SECTIONS_440 table:

SECTION_ID	SECTION_NAME
1 11H	WORLD HISTORY
2 1	SCIENCE
3 2	FICTION
4 3	FANTASY
5 4	NOVEL
6 5A	PICTURE BOOKS
7 5B	BOARD BOOKS
8 5C	EARLY READERS

2.5 CATEGORIES_440

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections pane lists an Oracle connection named 'PiyumikaDB'. The Object Navigator pane shows various database objects under 'Assignments' for this connection, including Tables, Views, Indexes, Packages, Procedures, Functions, Operators, Queues, Queue Tables, Triggers, Types, Sequences, and Materialized Views. Below these are Reports, Analytic View Reports, Data Dictionary Reports, and Data Modeler Reports. The Workflow Jobs pane is empty. In the center, the SQL Worksheet pane displays a query in the Worksheet tab:

```
SELECT * FROM PUBLISHERS_440;
SELECT * FROM SECTIONS_440;
SELECT * FROM CATEGORIES_440;
SELECT * FROM ROLES_440;
```

The Query Result 6 tab shows the data for the CATEGORIES_440 table:

CATEGORY_ID	CATEGORY_NAME	SECTION_ID
1 PH678	PHYSICS	1
2 CH123	CHEMISTRY	1
3 HF001	HISTORICAL FICTION	2
4 RF005	REALIST FICTION	2
5 CF113	CHILDREN FICTION	2
6 CH145	CHILDREN	3
7 SC345	SOCIETY	3
8 N001	ROMANCE	4
9 N002	TRAGEDY	4
10 HM768	HUMAN EVOLUTION	1
11 H0234	CANADIAN HISTORY	11H

2.6 ROLES_440

The screenshot shows the Oracle SQL Developer interface, identical to the previous one but with a different query in the Worksheet tab:

```
SELECT * FROM SECTIONS_440;
SELECT * FROM CATEGORIES_440;
SELECT * FROM ROLES_440;
SELECT * FROM USERS_440;
```

The Query Result 6 tab shows the data for the ROLES_440 table:

ROLE_ID	ROLE_NAME
1 1	LIBRARIAN
2 2	MEMBER READER
3 3	NON-MEMBER READER
4 4	CATALOGER
5 5	SYSTEM ADMINISTRATOR
6 6	CIRCULATION CLERK
7 7	LIBRARY ASSISTANT

2.7 USERS_440

The screenshot shows the SQL Developer interface with the 'Copies' query results for the USERS_440 table. The 'Copies' tab is selected in the top navigation bar. The 'Worksheet' tab is active, displaying the following SQL code:

```

SELECT * FROM CATEGORIES_440;
SELECT * FROM ROLES_440;
SELECT * FROM USERS_440;
SELECT * FROM BOOKS_440;

```

The 'Query Result 6' tab is selected, showing the following data:

USER_ID	USER_NAME	USER_PASSWORD	USER_EMAIL	ROLE_ID	CONTACT	ADDRESS
1 220005	GILLIAND JOSEPH	12341234	gill.joseph@gmail.com	1	607678566 (null)	
2 200045	PIYUMIKA BANDULA	Piyumikal994	piyumiakabandula@gmail.com	2	6474045667 2, LAKE RD, BRAMPTON	
3 200675	HELEN PEIRIS	hELEN145	helen.peiris@gmail.com	2	6547890345 (null)	
4 200300	KRITHIK KUMAR	krithikl	krithik.kumar@gmail.com	3	9085670999 (null)	
5 210100	BEULA PERERA	beulal234	beula.perera@hotmail.com	5	6475036767 (null)	
6 230745	WINDY SHENON	windy1998	windy.shenon@gmail.com	7	5467897890 2, TRISTAR RD, MISSISSAUGA	

All rows were fetched in 0.002 seconds.

2.8 BOOKS_440

The screenshot shows the SQL Developer interface with the 'Borrowed' query results for the BOOKS_440 table. The 'Borrowed' tab is selected in the top navigation bar. The 'Worksheet' tab is active, displaying the following SQL code:

```

SELECT * FROM ROLES_440;
SELECT * FROM USERS_440;
SELECT * FROM BOOKS_440;
SELECT * FROM BOOK_AUTHORS_440;
SELECT * FROM COPIES_440;

```

The 'Query Result 7' tab is selected, showing the following data:

BOOK_ID	BOOK_TITLE	ISBN	PUBLICATION_YEAR	NUMBER_OF_COPIES	BOOK_DESCRIPTION
1 1	Romeo and Juliet	1253679876	1597	10	Tragedy renowned for poetry, iconic characters, and timeless themes.
2 2	Macbeth	1267899876	1623	2	Tragedy renowned ambition, guilt, the corrupting influence of power, and the struggle for power.
3 1234	HARRY POTTER	98607475327	1997	10	SERIES OF SEVEN BOOKS OF FANTASY FOR CHILDREN
4 1235	HARRY POTTER	7866775345	1997	2	SERIES OF SEVEN BOOKS OF FANTASY FOR CHILDREN
5 2673	FRIE AND PREJUDICE	9780192827	1813	3	Fride And Prejudice
6 1365	On the Origin of Species	9780140439	1859	1	Means of Natural Selection for Life
7 1001	Anna Karenina	7835626288	1877	3	Themes of love, morality, and societal expectations
8 987	War and Peace	5463733738	1869	2	Lives of several aristocratic families in Russia during the Napoleonic Wars
9 23	A Short History of Canada	9780771065082	2006	10	Cultural shifts in Canadian History
10 24	A Short History of Canada	9780771065182	2006	5	Cultural shifts in Canadian History

All rows were fetched in 0.012 seconds.

2.9 BOOK_AUTHORS_440

The screenshot shows the Oracle Database Navigator in SQL Developer. The left sidebar displays the database schema for the 'PiyumikaDB' assignment, including tables, views, indexes, packages, procedures, functions, operators, queues, queue tables, triggers, types, and sequences. The 'Copies' table is selected in the 'Tables (Filtered)' section. The main workspace shows a query worksheet with the following SQL code:

```
SELECT * FROM BOOKS_440;
SELECT * FROM BOOK_AUTHORS_440;
SELECT * FROM COPIES_440;
```

The 'Query Result 9' tab shows the data for the 'COPIES' table:

BOOK_ID	AUTHOR_ID
1 1	N00001
2 2	N00001
3 1234	F00001
4 1235	F00001
5 2673	N00002
6 1365	S00013
7 1001	N00003
8 987	N00003
9 23	H05969

2.10 COPIES_440

The screenshot shows the Oracle Database Navigator in SQL Developer. The left sidebar displays the database schema for the 'PiyumikaDB' assignment, including tables, views, indexes, packages, procedures, functions, operators, queues, queue tables, triggers, types, and sequences. The 'Copies' table is selected in the 'Tables (Filtered)' section. The main workspace shows a query worksheet with the following SQL code:

```
SELECT * FROM BOOK_AUTHORS_440;
SELECT * FROM COPIES_440;
SELECT * FROM RESERVATIONS_440;
```

The 'Query Result 10' tab shows the data for the 'COPIES' table:

COPY_ID	BOOK_ID
1 001-1	1
2 001-2	1
3 002-1	2
4 1234-1	1234
5 1234-2	1234
6 1235-1	1235
7 2673-1	2673
8 1365-1	1365
9 1001-1	1001
10 987-1	987
11 023-1	23

2.11 RESERVATIOS_440

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays connections and reports. The main area has three tabs: 'SQL Worksheet' (active), 'History', and 'Category Asql'. The 'Worksheet' tab contains the following SQL code:

```
SELECT * FROM COPIES_440;
SELECT * FROM RESERVATIONS_440;
SELECT * FROM ISSUES_440;
```

The 'Query Result 10' tab shows the results for the first query:

	RESERVATION_ID	USER_ID	COPY_ID	RESERVATION_DATE	EXPIRY_DATE
1	200257	200045	023-1	20-02-05	20-02-12
2	200923	200300	1234-2	20-09-18	20-09-25
3	210612	200675	1235-1	21-06-17	20-06-24
4	220345	230745	2673-1	20-03-25	22-04-01
5	230732	210100	1001-1	23-07-13	23-07-20
6	230857	220005	023-1	23-08-10	23-02-17

2.12 ISSUES_440

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays connections and reports. The main area has three tabs: 'SQL Worksheet' (active), 'History', and 'Category Asql'. The 'Worksheet' tab contains the following SQL code:

```
SELECT * FROM RESERVATIONS_440;
SELECT * FROM ISSUES_440;
```

The 'Query Result 11' tab shows the results for the second query:

	ISSUE_ID	USER_ID	COPY_ID	ISSUE_DATE	DU_DATE	RETURN_DATE	STATUS	FINE_AMOUNT
1	200240	200045	023-1	20-02-07	20-02-21	20-02-17	RETURNED	0
2	200934	200300	1234-2	20-09-20	20-10-05	20-10-10	RETURNED	50
3	210620	200675	1235-1	21-06-21	20-07-04	20-07-10	RETURNED	60
4	220350	230745	2673-1	20-04-01	22-04-15	22-04-12	RETURNED	0
5	230736	210100	1001-1	23-07-14	23-07-28	(null)	BORROWED	(null)
6	230812	220005	023-1	23-08-11	23-08-25	(null)	BORROWED	(null)

3. SQL QUERIES

3.1 Category A

1. Select all the data from ROLES_440 table.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections pane shows an Oracle connection named 'PiyumikaDB'. The Tables node under it contains several objects: Tables (Filtered), Views, Indexes, Packages, Procedures, Functions, Operators, Queues, Queues Table, Triggers, and Types. The Reports pane shows 'All Reports', 'Analytic View Reports', and 'Data Dictionary Reports'. In the center, the SQL Worksheet tab is active, displaying the query: `// Select all the data from ROLES_440
SELECT * FROM ROLES_440;`. Below the worksheet is the Query Result tab, which displays the results in a grid:

ROLE_ID	ROLE_NAME
1	LIBRARIAN
2	MEMBER READER
3	NON-MEMBER READER
4	CATALOGER
5	SYSTEM ADMINISTRATOR
6	CIRCULATION CLERK
7	LIBRARY ASSISTANT

2. Select all records from selected columns - authors and biographies from AUTHORS_440 table. Here I have used aliases to rename the Author name column.

The screenshot shows the Oracle SQL Developer interface. The Connections and Reports panes are identical to the previous screenshot. In the center, the SQL Worksheet tab is active, displaying the query: `// Select all authors and biographies from AUTHORS_440 table
SELECT UPPER(AUTHOR_NAME) AS AUTHOR_NAME_440, AUTHOR_BIOGRAPHY FROM AUTHORS_440;`. Below the worksheet is the Query Result tab, which displays the results in a grid:

AUTHOR_NAME_440	AUTHOR_BIOGRAPHY
1 SHAKESPEARE	Greatest writer in the English language.
2 JANE AUSTEN	English novelist known for her six major novels.
3 J. K. ROWLING	British author best known for writing the Harry Potter series.
4 LEO TOLSTOY	A Russian novelist famous for epic novels in the exploration of human nature and society.
5 CHARLES DARWIN	Darwin theory of evolution by natural selection revolutionized biology and life.
6 STEPHEN HAWKING	Introduced complex cosmological concepts to the general public.
7 OLIVER SACKS	A neurologist and author who explored human brain.
8 DESMOND MORTON	Canadian historian known for Canadian military and political history.

3. Select DISTINCT that means unique book title and book descriptions from BOOKS_440 table

The screenshot shows the Oracle SQL Developer interface. On the left, the Object Navigator displays the schema structure under 'PiyumikaDB' (Tables, Views, Indexes, Procedures, Functions, Operators, Queues, Triggers, Types). The central workspace contains a SQL Worksheet tab with the following code:

```
// Select all book title and book descriptions from BOOKS_440 table
SELECT DISTINCT UPPER(BOOK_TITLE), BOOK_DESCRIPTION FROM BOOKS_440;
```

Below the worksheet is a 'Query Result' window showing the output:

UPPER(BOOK_TITLE)	BOOK_DESCRIPTION
1 ROMEO AND JULIET	Tragedy renowned for poetry, iconic characters, and timeless themes.
2 MACBETH	Tragedy renowned ambition, guilt, the corrupting influence of power, and the supernatural.
3 HARRY POTTER	SERIES OF SEVEN BOOKS OF FANTASY FOR CHILDREN
4 PRIDE AND PREJUDICE	Pride And Prejudice
5 ON THE ORIGIN OF SPECIES	Means of Natural Selection for Life
6 ANNA KARENINA	Themes of love, morality, and societal expectations
7 WAR AND PEACE	Lives of several aristocratic families in Russia during the Napoleonic Wars
8 A SHORT HISTORY OF CANADA	Cultural shifts in Canadian History

4. List user name and address as a single string for those who have address recorded in the database.

The screenshot shows the Oracle SQL Developer interface. The schema structure is identical to the previous screenshot. The SQL Worksheet tab contains the following code:

```
// List user name and address as a single string for those who have address recorded in the database
SELECT USER_NAME || ', ' || ADDRESS AS MAILING_ADDRESS
FROM USERS_440 WHERE ADDRESS IS NOT NULL;
```

The 'Query Result' window shows the output:

MAILING_ADDRESS
1 PIYUMIKA BANDULA, 2, LAKE RD, BRAMPTON
2 WINDY SHENON, 2, TRISTAR RD, MISSISSAUGA

5. List Users who Have Overdue Books

The screenshot shows the Oracle SQL Developer interface. The schema structure is identical to the previous screenshots. The SQL Worksheet tab contains the following code:

```
// List Users who Have Overdue Books
SELECT u.USER_NAME, i.ISSUE_DATE, i.DUE_DATE
FROM USERS_440 u
JOIN ISSUES_440 i ON u.USER_ID = i.USER_ID
WHERE i.RETURN_DATE IS NULL AND i.DUE_DATE < SYSDATE;
```

The 'Query Result' window shows the output:

USER_NAME	ISSUE_DATE	DUE_DATE
1 BEULA PERERA	23-07-14	23-07-28

3.2 Category B

1. List books published after the year 2000 using > operator.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows an Oracle connection named 'PiyumikaDB'. The 'Tables' node is expanded, showing 'Books_440'. In the central 'Worksheet' tab, a query is written:

```
// List books published after the year 2000 using > operator.  
SELECT DISTINCT BOOK_TITLE, PUBLICATION_YEAR  
FROM BOOKS_440  
WHERE PUBLICATION_YEAR > 2000;
```

The 'Query Result' tab at the bottom displays the results:

BOOK_TITLE	PUBLICATION_YEAR
A Short History of Canada	2006

2. Find expired reservations which were not borrowed at that time period.

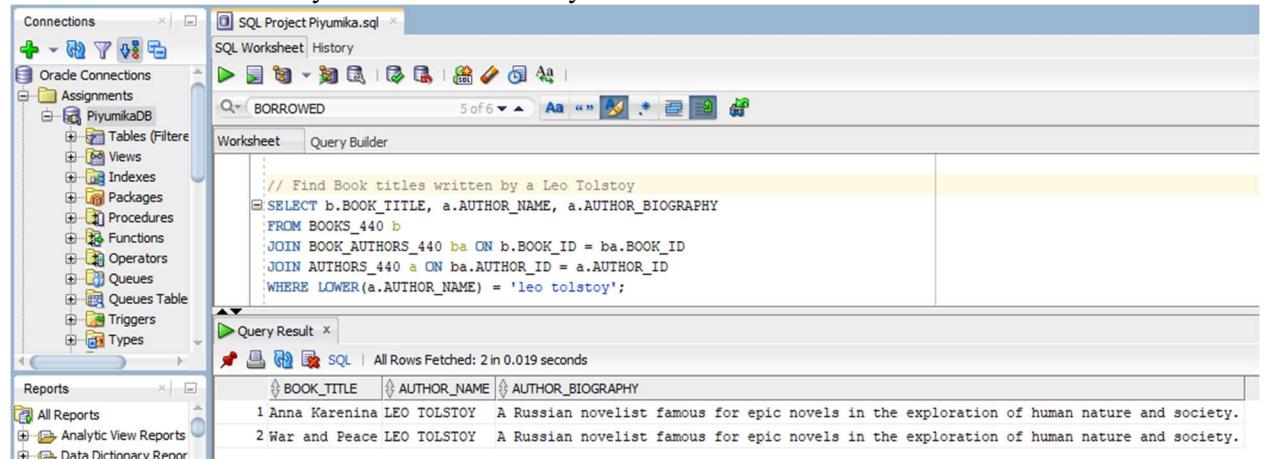
The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows an Oracle connection named 'PiyumikaDB'. The 'Tables' node is expanded, showing 'Reservations_440'. In the central 'Worksheet' tab, a query is written:

```
// Find expired reservations which were not borrowed at that time period.  
SELECT RESERVATION_ID, USER_ID, COPY_ID, EXPIRY_DATE  
FROM RESERVATIONS_440  
WHERE EXPIRY_DATE < SYSDATE;
```

The 'Query Result' tab at the bottom displays the results:

RESERVATION_ID	USER_ID	COPY_ID	EXPIRY_DATE
1 200257	200045	023-1	20-02-12
2 200923	200300	1234-2	20-09-25
3 210612	200675	1235-1	20-06-24
4 220345	230745	2673-1	22-04-01
5 230732	210100	1001-1	23-07-20
6 230857	220005	023-1	23-02-17

3. Find Book titles written by a Author Leo Tolstoy.



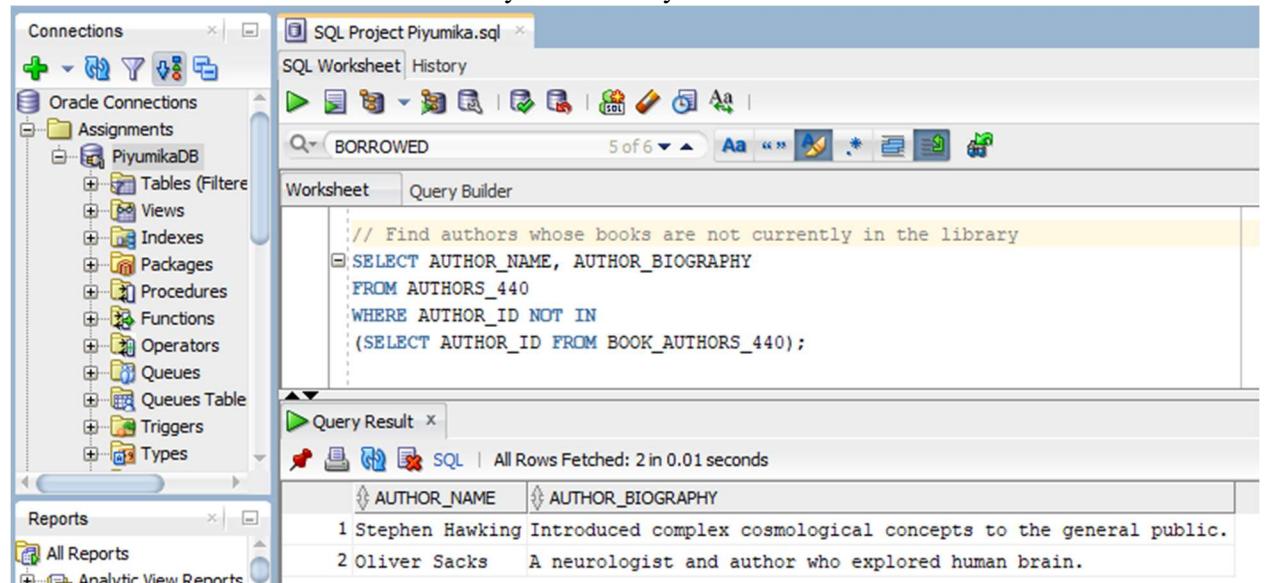
The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the database schema for 'PiyumikaDB' under 'Assignments'. The central workspace contains a 'SQL Worksheet' tab with the following SQL code:

```
// Find Book titles written by a Leo Tolstoy
SELECT b.BOOK_TITLE, a.AUTHOR_NAME, a.AUTHOR_BIOGRAPHY
FROM BOOKS_440 b
JOIN BOOK_AUTHORS_440 ba ON b.BOOK_ID = ba.BOOK_ID
JOIN AUTHORS_440 a ON ba.AUTHOR_ID = a.AUTHOR_ID
WHERE LOWER(a.AUTHOR_NAME) = 'leo tolstoy';
```

Below the worksheet is a 'Query Result' tab showing the output:

BOOK_TITLE	AUTHOR_NAME	AUTHOR_BIOGRAPHY
1 Anna Karenina	LEO TOLSTOY	A Russian novelist famous for epic novels in the exploration of human nature and society.
2 War and Peace	LEO TOLSTOY	A Russian novelist famous for epic novels in the exploration of human nature and society.

4. Find authors whose books are not currently in the library.



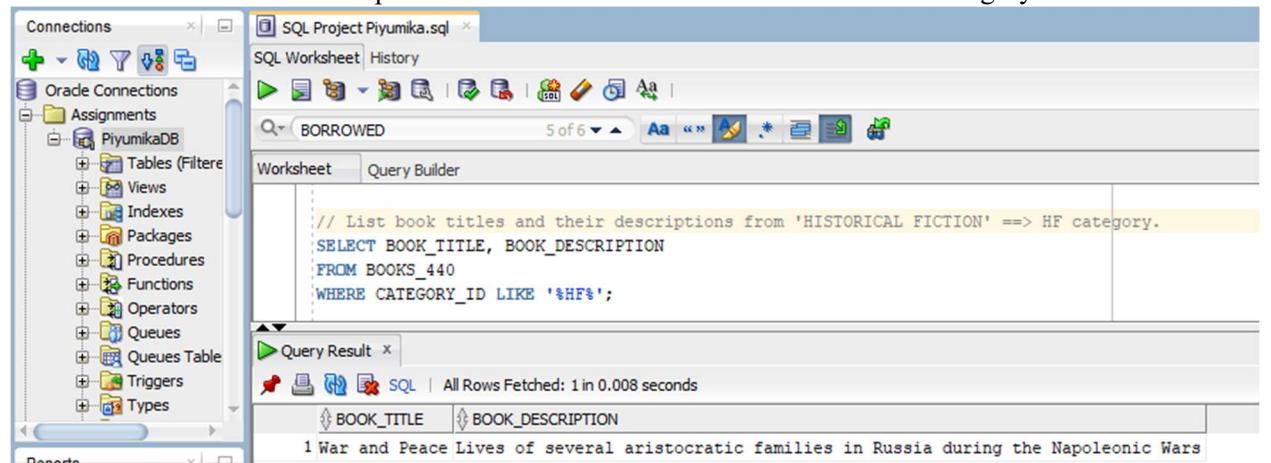
The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the database schema for 'PiyumikaDB' under 'Assignments'. The central workspace contains a 'SQL Worksheet' tab with the following SQL code:

```
// Find authors whose books are not currently in the library
SELECT AUTHOR_NAME, AUTHOR_BIOGRAPHY
FROM AUTHORS_440
WHERE AUTHOR_ID NOT IN
(SELECT AUTHOR_ID FROM BOOK_AUTHORS_440);
```

Below the worksheet is a 'Query Result' tab showing the output:

AUTHOR_NAME	AUTHOR_BIOGRAPHY
1 Stephen Hawking	Introduced complex cosmological concepts to the general public.
2 Oliver Sacks	A neurologist and author who explored human brain.

5. List book titles and their descriptions from 'HISTORICAL FICTION' ==> HF category.



The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the database schema for 'PiyumikaDB' under 'Assignments'. The central workspace contains a 'SQL Worksheet' tab with the following SQL code:

```
// List book titles and their descriptions from 'HISTORICAL FICTION' ==> HF category.
SELECT BOOK_TITLE, BOOK_DESCRIPTION
FROM BOOKS_440
WHERE CATEGORY_ID LIKE '%HF%';
```

Below the worksheet is a 'Query Result' tab showing the output:

BOOK_TITLE	BOOK_DESCRIPTION
1 War and Peace	Lives of several aristocratic families in Russia during the Napoleonic Wars

3.3 Category C

1. List users who will have to pay fines in future.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows an Oracle connection named 'PiyumikaDB' under 'Assignments'. The 'Tables' node is expanded, showing various table types like Filtered, Views, Indexes, Packages, Procedures, Functions, Operators, Queues, Queues Table, Triggers, and Types. In the center, the 'SQL Worksheet' tab is active, displaying the following SQL code:

```
// List users who will have to pay fines
SELECT u.USER_NAME,ROUND(SYSDATE-i.DUE_DATE) AS DAYS_OVERDUE FROM
USERS_440 u
JOIN ISSUES_440 i ON u.USER_ID = i.USER_ID
WHERE i.DUE_DATE < SYSDATE AND NOT i.STATUS = 'RETURNED';
```

Below the worksheet is the 'Query Result' tab, which shows the output of the query:

USER_NAME	DAYS_OVERDUE
1 BEULA PERERA	15

2. List the titles of books that are either in the "TRAGEDY" or in the "ROMANCE" category and published before the year 1990.

The screenshot shows the Oracle SQL Developer interface. The 'Connections' tree on the left is identical to the previous screenshot, showing the 'PiyumikaDB' connection. In the center, the 'SQL Worksheet' tab is active, displaying the following SQL code:

```
// List the titles of books that are either in the "TRAGEDY" or in the "ROMANCE" category
// and published before the year 1990
SELECT b.BOOK_TITLE
FROM BOOKS_440 b
JOIN CATEGORIES_440 c ON b.CATEGORY_ID = c.CATEGORY_ID
WHERE (UPPER(c.CATEGORY_NAME) = 'TRAGEDY' OR UPPER(c.CATEGORY_NAME) = 'ROMANCE') AND b.PUBLICATION_YEAR < 1990;
```

Below the worksheet is the 'Query Result' tab, which shows the output of the query:

BOOK_TITLE
1 Romeo and Juliet
2 Macbeth

3. Getting book titles and publication years from TRAGEDY category and ordering in the descending order of the publication year.

The screenshot shows the Oracle SQL Developer interface. On the left, there are three panels: 'Connections' (listing 'Assignments' and 'PiyumikaDB'), 'Reports' (listing 'All Reports' and other report types), and 'Workflow Jobs'. The main area consists of a 'SQL Worksheet' tab and a 'Query Result' tab. The 'SQL Worksheet' tab contains a query:// Getting book titles and publication years from TRAGEDY category and
// ordering in the descending order of the publication year
SELECT
 B.BOOK_TITLE,
 B.PUBLICATION_YEAR
FROM
 BOOKS_440 B
JOIN
 CATEGORIES_440 C ON B.CATEGORY_ID = C.CATEGORY_ID
WHERE
 C.CATEGORY_NAME = 'TRAGEDY'
ORDER BY
 B.PUBLICATION_YEAR DESC;The 'Query Result' tab shows the output of the query:| BOOK_TITLE | PUBLICATION_YEAR |
| --- | --- |
| 1 Macbeth | 1623 |
| 2 Romeo and Juliet | 1597 |

All Rows Fetched: 2 in 0.011 seconds

4. We will use the OFFSET clause and FETCH NEXT clause to above example.

OFFSET skips the first 1 row and the FETCH NEXT clause retrieves the next 5 rows resulting only 1 record in this example.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows an Oracle connection named 'PiyumikaDB' under 'Assignments'. The 'SQL Worksheet' tab is active, displaying the following SQL code:

```

// We will use the OFFSET clause and FETCH NEXT clause to above example.
// OFFSET skips the first 1 row and the FETCH NEXT clause retrieves the next 5 rows
// resulting only 1 record in this example.
SELECT
    B.BOOK_TITLE,
    B.PUBLICATION_YEAR
FROM
    BOOKS_440 B
JOIN
    CATEGORIES_440 C ON B.CATEGORY_ID = C.CATEGORY_ID
WHERE
    C.CATEGORY_NAME = 'TRAGEDY'
ORDER BY
    B.PUBLICATION_YEAR DESC
OFFSET 1 ROWS FETCH NEXT 5 ROWS ONLY;

```

The 'Query Result' tab at the bottom shows the output:

BOOK_TITLE	PUBLICATION_YEAR
Romeo and Juliet	1597

5. Total number of users associated with library.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree shows an Oracle connection named 'PiyumikaDB' under 'Assignments'. The 'SQL Worksheet' tab is active, displaying the following SQL code:

```

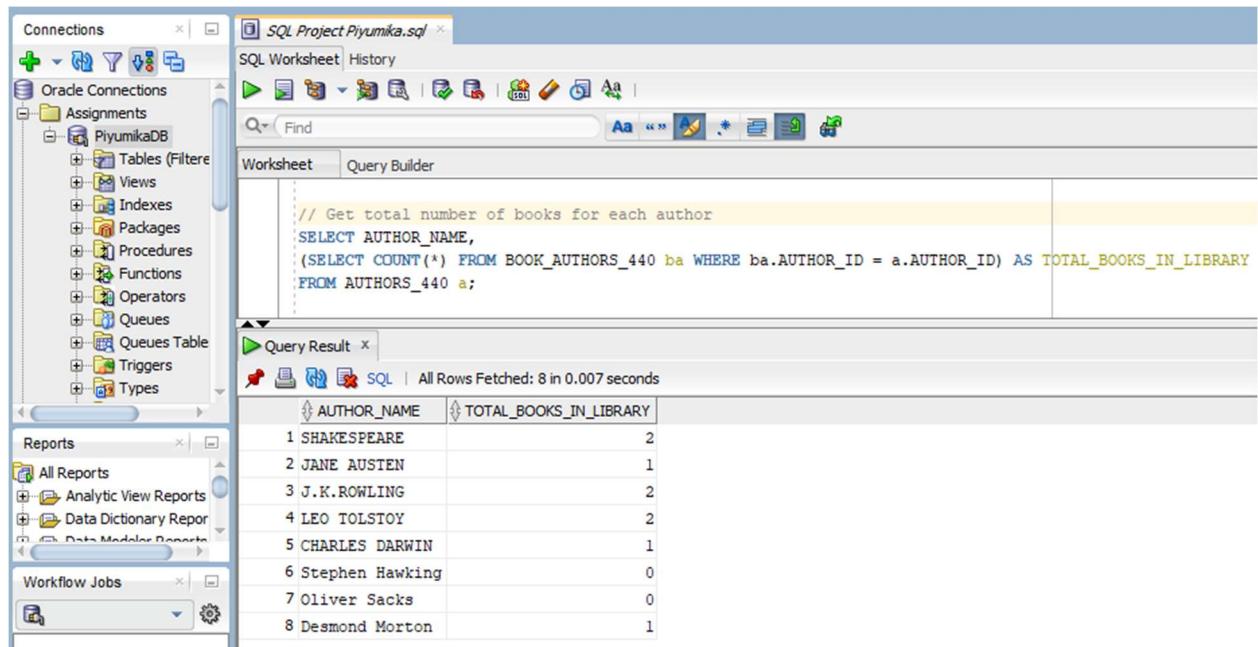
// Total number of users associated with library
SELECT COUNT(*) AS TOTAL_USERS FROM USERS_440;

```

The 'Query Result' tab at the bottom shows the output:

TOTAL_USERS
6

6. Get total number of books for each author.



The screenshot shows the Oracle SQL Developer interface. On the left, the Connections pane shows an Oracle connection named 'PiyumikaDB'. The central area has a 'SQL Worksheet' tab open with the following SQL code:

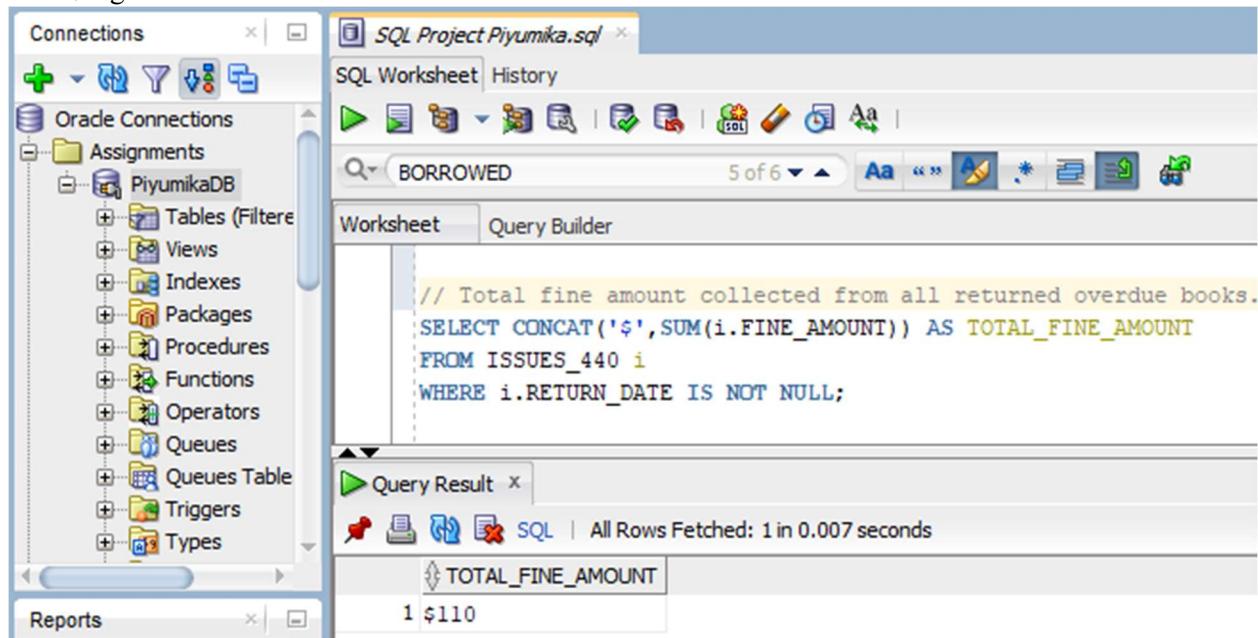
```
// Get total number of books for each author
SELECT AUTHOR_NAME,
(SELECT COUNT(*) FROM BOOK_AUTHORS_440 ba WHERE ba.AUTHOR_ID = a.AUTHOR_ID) AS TOTAL_BOOKS_IN_LIBRARY
FROM AUTHORS_440 a;
```

Below the worksheet is a 'Query Result' tab showing the output:

AUTHOR_NAME	TOTAL_BOOKS_IN_LIBRARY
1 SHAKESPEARE	2
2 JANE AUSTEN	1
3 J. K. ROWLING	2
4 LEO TOLSTOY	2
5 CHARLES DARWIN	1
6 Stephen Hawking	0
7 Oliver Sacks	0
8 Desmond Morton	1

3.4 Category D

1. Total fine amount collected from all returned overdue books and display the value concatenated with \$ sign.



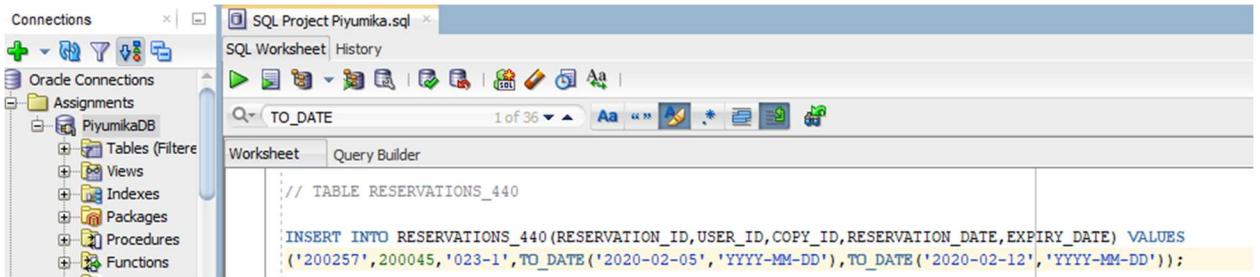
The screenshot shows the Oracle SQL Developer interface. On the left, the Connections pane shows an Oracle connection named 'PiyumikaDB'. The central area has a 'SQL Worksheet' tab open with the following SQL code:

```
// Total fine amount collected from all returned overdue books.
SELECT CONCAT('$',SUM(i.FINE_AMOUNT)) AS TOTAL_FINE_AMOUNT
FROM ISSUES_440 i
WHERE i.RETURN_DATE IS NOT NULL;
```

Below the worksheet is a 'Query Result' tab showing the output:

TOTAL_FINE_AMOUNT
1 \$110

2. TO_DATE function is used to convert a string representation of a date into an actual date data type. When I am using INSERT INTO for entering data into tables I am using it many times in this project. An example scenario is shown below.



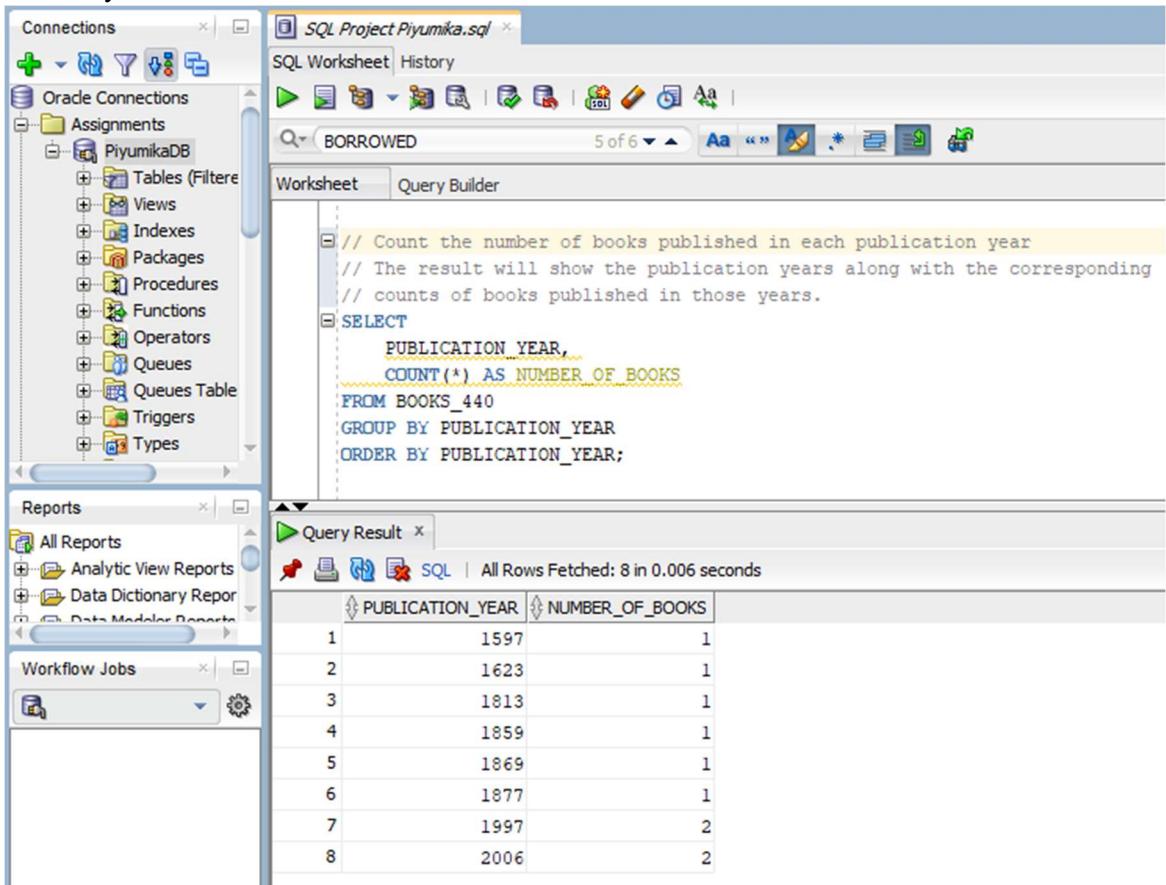
```

// TABLE RESERVATIONS_440
INSERT INTO RESERVATIONS_440(RESERVATION_ID,USER_ID,COPY_ID,RESERVATION_DATE,EXPIRY_DATE) VALUES
('200257',200045,'023-1',TO_DATE('2020-02-05','YYYY-MM-DD'),TO_DATE('2020-02-12','YYYY-MM-DD'));

```

3. Count the number of books published in each publication year.

The result will show the publication years along with the corresponding counts of books published in those years.



```

// Count the number of books published in each publication year
// The result will show the publication years along with the corresponding
// counts of books published in those years.
SELECT
    PUBLICATION_YEAR,
    COUNT(*) AS NUMBER_OF_BOOKS
FROM BOOKS_440
GROUP BY PUBLICATION_YEAR
ORDER BY PUBLICATION_YEAR;

```

PUBLICATION_YEAR	NUMBER_OF_BOOKS
1	1597
2	1623
3	1813
4	1859
5	1869
6	1877
7	1997
8	2006

4. List the books having MINIMUM of 2 language versions using GROUP BY and HAVING clauses.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections pane shows an Oracle connection named 'PiyumikaDB'. The Reports and Workflow Jobs panes are also visible. In the center, the Worksheet tab of the SQL Worksheet window contains the following SQL code:

```
// List the books having MINIMUM of 2 language versions using GROUP BY and HAVING clauses
SELECT
    b.BOOK_TITLE,
    COUNT(*) AS NUMBER_OF_LANGUAGES_AVAILABLE
FROM BOOKS_440 b
JOIN LANGUAGES_440 l ON b.LANGUAGE_ID = l.LANGUAGE_ID
GROUP BY b.BOOK_TITLE
HAVING COUNT(*) > 1
ORDER BY b.BOOK_TITLE;
```

Below the worksheet, the Query Result window displays the results:

BOOK_TITLE	NUMBER_OF_LANGUAGES_AVAILABLE
1 A Short History of Canada	2
2 HARRY POTTER	2

5. List languages available for each book using the LISTAGG function.

The screenshot shows the Oracle SQL Developer interface. The Connections, Reports, and Workflow Jobs panes are visible on the left. The Worksheet tab of the SQL Worksheet window contains the following SQL code:

```
// List languages available for each book using the LISTAGG function
SELECT b.BOOK_TITLE, LISTAGG(l.LANGUAGE_NAME, ', ') WITHIN GROUP (ORDER BY l.LANGUAGE_NAME) AS LANGUAGES
FROM BOOKS_440 b
JOIN LANGUAGES_440 l ON b.LANGUAGE_ID = l.LANGUAGE_ID
GROUP BY b.BOOK_TITLE;
```

Below the worksheet, the Query Result window displays the results:

BOOK_TITLE	LANGUAGES
1 A Short History of Canada	ENGLISH, FRENCH
2 Anna Karenina	ENGLISH
3 HARRY POTTER	ENGLISH, FRENCH
4 Macbeth	ENGLISH
5 On the Origin of Species	ENGLISH
6 PRIDE AND PREJUDICE	ENGLISH
7 Romeo and Juliet	ENGLISH
8 War and Peace	ENGLISH

6. List all users and put Null addresses as Brampton. NVL function is used in Oracle SQL to replace a null value with another value.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections pane shows an Oracle connection named 'PiyumikaDB'. The SQL Worksheet pane contains the following SQL code:

```
// List all users and put Null addresses as Brampton
SELECT USER_NAME, NVL(ADDRESS, 'Unknown')
FROM USERS_440;
```

The Query Result pane displays the results of the query:

USER_NAME	NVL(ADDRESS,'UNKNOWN')
1 GILLIAND JOSEPH	Unknown
2 PIYUMIKA BANDULA	2, LAKE RD, BRAMPTON
3 HELEN PEIRIS	Unknown
4 KRITHIK KUMAR	Unknown
5 BEULA PERERA	Unknown
6 WINDY SHENON	2, TRISTAR RD, MISSISSAUGA

7. List the addresses of users from the USERS_440 table, but if the address is null, we will use user email as a fallback.

The COALESCE function here is used to return the first non-null expression in a list of expressions. If all the expressions in the list are null, it returns null.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections pane shows an Oracle connection named 'PiyumikaDB'. The SQL Worksheet pane contains the following SQL code:

```
// List the addresses of users from the USERS_440 table, but if the address is null, we will use user email as a fallback.
// The COALESCE function here is used to return the first non-null expression in a list of expressions.
// If all the expressions in the list are null, it returns null.
SELECT USER_NAME, COALESCE(ADDRESS, USER_EMAIL) AS CONTACT_INFO
FROM USERS_440;
```

The Query Result pane displays the results of the query:

USER_NAME	CONTACT_INFO
1 GILLIAND JOSEPH	gill.joseph@gmail.com
2 PIYUMIKA BANDULA	2, LAKE RD, BRAMPTON
3 HELEN PEIRIS	helen.peiris@gmail.com
4 KRITHIK KUMAR	krithik.kumar@gmail.com
5 BEULA PERERA	beula.perera@hotmail.com
6 WINDY SHENON	2, TRISTAR RD, MISSISSAUGA

3.5 Category E

1. List Books by Category and Section.

The screenshot shows the Oracle SQL Developer interface. On the left, the Database Navigator displays the schema structure under the 'Assignments' folder, including 'PiyumikaDB' and its objects: Tables, Views, Indexes, Packages, Procedures, Functions, Operators, Queues, Queues Table, Triggers, and Types. Below the Database Navigator are the 'Reports' and 'Workflow Jobs' panes.

The central workspace contains a 'Worksheet' tab with the following SQL code:

```
// List Books by Category and Section
SELECT distinct b.BOOK_TITLE, c.CATEGORY_NAME, s.SECTION_NAME
FROM BOOKS_440 b
JOIN CATEGORIES_440 c ON b.CATEGORY_ID = c.CATEGORY_ID
JOIN SECTIONS_440 s ON b.SECTION_ID = s.SECTION_ID
ORDER BY s.SECTION_NAME, c.CATEGORY_NAME, b.BOOK_TITLE;
```

Below the Worksheet is a 'Query Result' tab showing the output of the query:

BOOK_TITLE	CATEGORY_NAME	SECTION_NAME
1 HARRY POTTER	CHILDREN FICTION	FANTASY
2 PRIDE AND PREJUDICE	SOCIETY	FANTASY
3 War and Peace	HISTORICAL FICTION	FICTION
4 Anna Karenina	REALIST FICTION	FICTION
5 Macbeth	TRAGEDY	NOVEL
6 Romeo and Juliet	TRAGEDY	NOVEL
7 On the Origin of Species	HUMAN EVOLUTION	SCIENCE
8 A Short History of Canada	CANADIAN HISTORY	WORLD HISTORY

2. Count the Number of Copies Available currently in the library for a given Book.

The screenshot shows the Oracle SQL Developer interface, similar to the previous one but with a different query. The Database Navigator and Reports panes are visible on the left.

The central workspace contains a 'Worksheet' tab with the following SQL code:

```
// Count the Number of Copies Available currently in the library for a given Book
SELECT b.BOOK_TITLE, b.NUMBER_OF_COPIES - COUNT(i.ISSUE_ID) AS AVAILABLE_COPIES, l.LANGUAGE_NAME
FROM BOOKS_440 b
LEFT JOIN COPIES_440 c ON b.BOOK_ID = c.BOOK_ID
LEFT JOIN ISSUES_440 i ON c.COPY_ID = i.COPY_ID
JOIN LANGUAGES_440 l ON b.LANGUAGE_ID = l.LANGUAGE_ID
WHERE LOWER(b.BOOK_TITLE) LIKE ('%history of canada%')
GROUP BY b.BOOK_TITLE, b.NUMBER_OF_COPIES, l.language_name;
```

Below the Worksheet is a 'Query Result' tab showing the output of the query:

BOOK_TITLE	AVAILABLE_COPIES	LANGUAGE_NAME
1 A Short History of Canada	8	ENGLISH
2 A Short History of Canada	5	FRENCH

3. Left Outer Join to list all the authors in the database despite the library has their books or not.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections pane shows an Oracle connection named 'PiyumikaDB'. The main workspace contains a SQL Worksheet tab with the following code:

```
// List all the authors in the database despite the library has their books or not.
SELECT DISTINCT a.AUTHOR_NAME, b.BOOK_TITLE
FROM AUTHORS_440 a
LEFT OUTER JOIN BOOK_AUTHORS_440 ba ON a.AUTHOR_ID = ba.AUTHOR_ID
LEFT OUTER JOIN BOOKS_440 b ON b.BOOK_ID = ba.BOOK_ID;
```

Below the worksheet is a Query Result tab displaying the results of the query:

AUTHOR_NAME	BOOK_TITLE
1 SHAKESPEARE	Romeo and Juliet
2 SHAKESPEARE	Macbeth
3 J.K.ROWLING	HARRY POTTER
4 JANE AUSTEN	PRIDE AND PREJUDICE
5 CHARLES DARWIN	On the Origin of Species
6 LEO TOLSTOY	Anna Karenina
7 LEO TOLSTOY	War and Peace
8 Desmond Morton	A Short History of Canada
9 Stephen Hawking	(null)
10 Oliver Sacks	(null)

4. We can get the same above result using Right Outer Join as well.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections pane shows an Oracle connection named 'PiyumikaDB'. The main workspace contains a SQL Worksheet tab with the following code:

```
// Using Right Outer Join to get the same result.
SELECT DISTINCT a.AUTHOR_NAME, b.BOOK_TITLE AS AVAILABLE_IN_LIBRARY
FROM BOOK_AUTHORS_440 ba
RIGHT OUTER JOIN AUTHORS_440 a ON a.AUTHOR_ID = ba.AUTHOR_ID
LEFT OUTER JOIN BOOKS_440 b ON b.BOOK_ID = ba.BOOK_ID;
```

Below the worksheet is a Query Result tab displaying the results of the query:

AUTHOR_NAME	AVAILABLE_IN_LIBRARY
1 SHAKESPEARE	Romeo and Juliet
2 SHAKESPEARE	Macbeth
3 J.K.ROWLING	HARRY POTTER
4 JANE AUSTEN	PRIDE AND PREJUDICE
5 CHARLES DARWIN	On the Origin of Species
6 LEO TOLSTOY	Anna Karenina
7 LEO TOLSTOY	War and Peace
8 Desmond Morton	A Short History of Canada
9 Stephen Hawking	(null)
10 Oliver Sacks	(null)

5. Let's use SELF JOIN to join the USERS_440 table with itself based on the ROLE_ID column. The condition $u1.USER_ID \neq u2.USER_ID$ ensures that we don't match a user with themselves. This query will return pairs of users who share the same role.

The screenshot shows the SQL Developer interface. The left sidebar displays database connections and assignments, with 'PiyumikaDB' selected. The central workspace contains a SQL Worksheet tab with the following code:

```

// Let's use SELF JOIN to join the USERS_440 table with itself based on the ROLE_ID column.
// The condition u1.USER_ID <> u2.USER_ID ensures that we don't match a user with themselves.
// This query will return pairs of users who share the same role.
SELECT u1.USER_NAME AS User1, u2.USER_NAME AS User2, r.ROLE_NAME
FROM USERS_440 u1
JOIN USERS_440 u2 ON u1.ROLE_ID = u2.ROLE_ID AND u1.USER_ID <> u2.USER_ID
JOIN ROLES_440 r ON u1.ROLE_ID = r.ROLE_ID;

```

Below the worksheet is a Query Result window showing the output:

USER1	USER2	ROLE_NAME
1 HELEN PEIRIS	PIYUMIKA BANDULA	MEMBER
2 PIYUMIKA BANDULA	HELEN PEIRIS	READER

3.6 Category F

1. Let's use INTERSECT set operator to get the names of those having fines greater than \$10 in their history.

The screenshot shows the SQL Developer interface. The left sidebar displays database connections and assignments, with 'PiyumikaDB' selected. The central workspace contains a SQL Worksheet tab with the following code:

```

// Let's use INTERSECT keyword to get the user names of those
// having fines greater than $10 in their history.
SELECT USER_NAME FROM
USERS_440 WHERE USER_ID IN
(SELECT USER_ID FROM USERS_440
INTERSECT
SELECT USER_ID FROM ISSUES_440 WHERE FINE_AMOUNT > 10);

```

Below the worksheet is a Query Result window showing the output:

USER_NAME
1 HELEN PEIRIS
2 KRITHIK KUMAR

2. Find books that do not have French translations in your database using MINUS set operator.
It will subtract the titles of books with French translations from the titles of all books.

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the database structure under 'Oracle Connections' and 'Assignments'. The central workspace contains a 'SQL Worksheet' tab with the following code:

```
// Find books that do not have French translations in your database using MINUS set operator.
// It will subtract the titles of books with French translations from the titles of all books.

SELECT BOOK_TITLE
FROM BOOKS_440
MINUS
SELECT DISTINCT b.BOOK_TITLE
FROM BOOKS_440 b
JOIN LANGUAGES_440 l ON b.LANGUAGE_ID = l.LANGUAGE_ID
WHERE l.LANGUAGE_NAME = 'FRENCH';
```

Below the worksheet is a 'Query Result' tab showing the output:

BOOK_TITLE
1 Romeo and Juliet
2 Macbeth
3 PRIDE AND PREJUDICE
4 On the Origin of Species
5 Anna Karenina
6 War and Peace

3. UNION Operator retrieves book titles from the BOOKS_440 table and BOOK_AUTHORS table while remove duplicates.

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the database structure under 'Oracle Connections' and 'Assignments'. The central workspace contains a 'SQL Worksheet' tab with the following code:

```
// UNION Operator retrieves book titles from the BOOKS_440
// table and BOOK_AUTHORS table while remove duplicates.

SELECT BOOK_TITLE FROM BOOKS_440
UNION
SELECT b.BOOK_TITLE
FROM BOOK_AUTHORS_440 ba
JOIN BOOKS_440 b ON ba.BOOK_ID = b.BOOK_ID
ORDER BY BOOK_TITLE;
```

Below the worksheet is a 'Query Result' tab showing the output:

BOOK_TITLE
1 A Short History of Canada
2 Anna Karenina
3 HARRY POTTER
4 Macbeth
5 On the Origin of Species

4. Let's use single-row subquery to retrieve information about a book and its author using the BOOK_ID from the COPIES_440 table.

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the database schema for 'PiyumikaDB' under 'Assignments'. The central area is a 'Worksheet' tab where a query is being written:

```

// Let's use single-row subquery to retrieve information about a
// book and its author using the BOOK_ID from the COPIES_440 table.

SELECT
    C.COPY_ID,
    B.BOOK_TITLE,
    (SELECT A.AUTHOR_NAME FROM AUTHORS_440 A WHERE A.AUTHOR_ID =
        (SELECT BA.AUTHOR_ID FROM BOOK_AUTHORS_440 BA WHERE BA.BOOK_ID = B.BOOK_ID))
        AS AUTHOR_NAME
FROM
    COPIES_440 C
JOIN
    BOOKS_440 B ON C.BOOK_ID = B.BOOK_ID
WHERE
    C.COPY_ID = '001-1';

```

Below the worksheet is a 'Query Result' window showing the output:

COPY_ID	BOOK_TITLE	AUTHOR_NAME
1 001-1	Romeo and Juliet	SHAKESPEARE

5. Let's find books that have been borrowed by users using EXISTS correlated subquery.

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the database schema for 'PiyumikaDB' under 'Assignments'. The central area is a 'Worksheet' tab where a query is being written:

```

// Let's find books that have been borrowed by users using EXISTS correlated subquery.

SELECT
    B.BOOK_TITLE,
    U.USER_NAME,
    I.ISSUE_DATE,
    I.DUE_DATE
FROM
    BOOKS_440 B
JOIN
    COPIES_440 C ON B.BOOK_ID = C.BOOK_ID
JOIN
    ISSUES_440 I ON C.COPY_ID = I.COPY_ID
JOIN
    USERS_440 U ON I.USER_ID = U.USER_ID
WHERE
    EXISTS (
        SELECT 1
        FROM
            ISSUES_440 I2
        WHERE
            I2.COPY_ID = C.COPY_ID
    );

```

Below the worksheet is a 'Query Result' window showing the output:

BOOK_TITLE	USER_NAME	ISSUE_DATE	DUE_DATE
1 Anna Karenina	BEULA PERERA	23-07-14	23-07-28
2 A Short History of Canada	GILLIAND JOSEPH	23-08-11	23-08-25

6. Find books that have not been borrowed by users.

The screenshot shows the Oracle SQL Developer interface. On the left, the Database Navigator displays the schema for 'PiyumikaDB' under the 'Assignments' node, including Tables, Views, Indexes, Packages, Procedures, Functions, Operators, Queues, Queues Table, Triggers, and Types. Below it are sections for Reports, All Reports, Analytic View Reports, Data Dictionary Reports, and Workflow Jobs. The central area contains a 'Worksheet' tab with the following SQL query:

```
// Find books that have not been borrowed by users.  
SELECT b.BOOK_TITLE  
FROM BOOKS_440 b  
WHERE NOT EXISTS (  
    SELECT 1  
    FROM ISSUES_440 i  
    WHERE i.COPY_ID IN (  
        SELECT c.COPY_ID  
        FROM COPIES_440 c  
        WHERE c.BOOK_ID = b.BOOK_ID  
    )  
);
```

Below the Worksheet is a 'Query Result' tab showing the output:

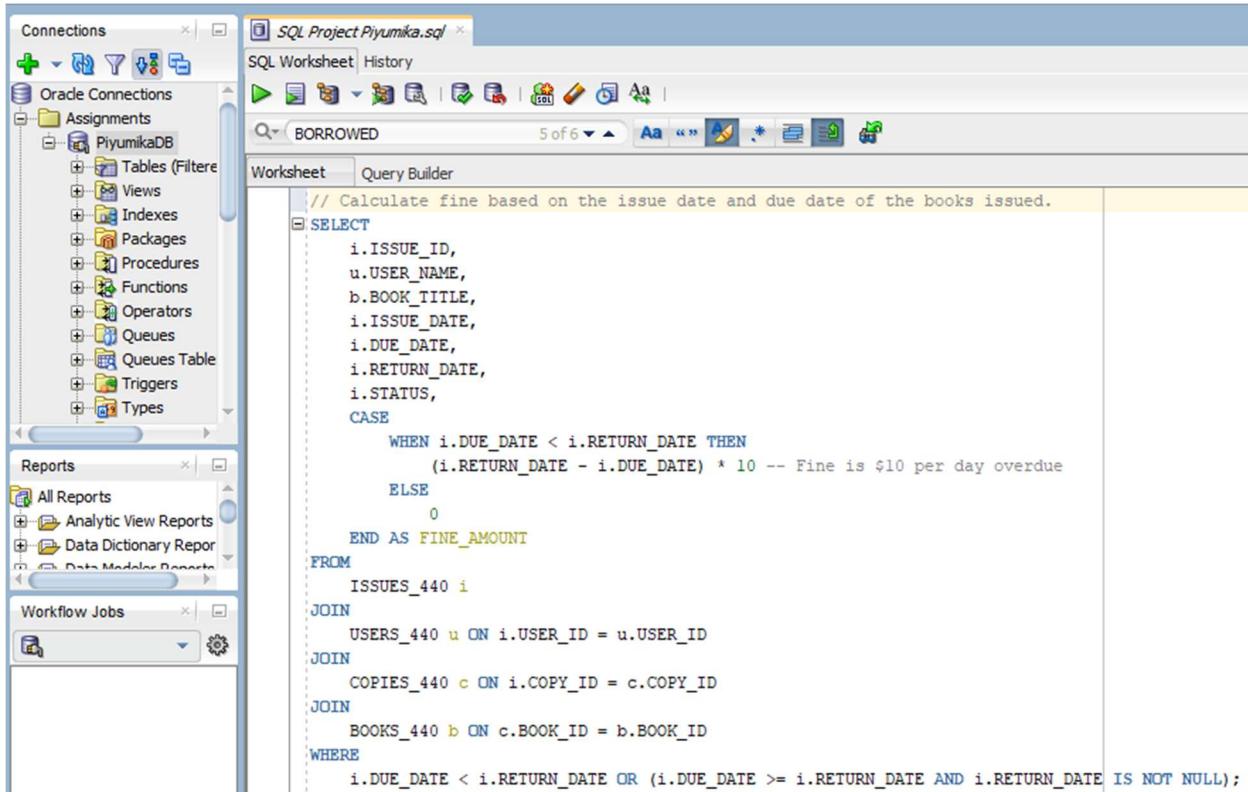
BOOK_TITLE
1 Romeo and Juliet
2 Macbeth
3 On the Origin of Species
4 War and Peace
5 A Short History of Canada

The status bar at the bottom right indicates "All Rows Fetched: 5 in 0.004 seconds".

3.7 Advance Category

Correlated subqueries refer to values from the outer query in the subquery.

1. Calculate fine based on the issue date and due date of the books issued.



The screenshot shows the Oracle SQL Developer interface. On the left, the Connections sidebar shows an Oracle connection named 'PiyumikaDB'. The main area is a 'SQL Worksheet' titled 'BORROWED' with the number '5 of 6' and various toolbar icons. The code in the worksheet is as follows:

```
// Calculate fine based on the issue date and due date of the books issued.
SELECT
    i.ISSUE_ID,
    u.USER_NAME,
    b.BOOK_TITLE,
    i.ISSUE_DATE,
    i.DUE_DATE,
    i.RETURN_DATE,
    i.STATUS,
    CASE
        WHEN i.DUE_DATE < i.RETURN_DATE THEN
            (i.RETURN_DATE - i.DUE_DATE) * 10 -- Fine is $10 per day overdue
        ELSE
            0
    END AS FINE_AMOUNT
FROM
    ISSUES_440 i
JOIN
    USERS_440 u ON i.USER_ID = u.USER_ID
JOIN
    COPIES_440 c ON i.COPY_ID = c.COPY_ID
JOIN
    BOOKS_440 b ON c.BOOK_ID = b.BOOK_ID
WHERE
    i.DUE_DATE < i.RETURN_DATE OR (i.DUE_DATE >= i.RETURN_DATE AND i.RETURN_DATE IS NOT NULL);
```

2. This query will provide a list of book titles and their corresponding author names for books published after the average publication year of all books in the BOOKS_440 table.

The main query retrieves book titles and author names from the BOOKS_440 and the nested subquery result (aliased as A).

The nested subquery retrieves book IDs and author names from the BOOK_AUTHORS_440 and AUTHORS_440 tables, respectively, using a JOIN to combine them.

The main query then joins the result of the nested subquery (A) with the BOOKS_440 table based on the BOOK_ID column.

The outer query's WHERE clause filters the results to include only books published after the average publication year of all books in the BOOKS_440 table.

Connections SQL Project Piyumika.sql

Oracle Connections
Assignments
PiyumikaDB
Tables (Filtered)
Views
Indexes
Packages
Procedures
Functions
Operators
Queues
Queues Table
Triggers
Types

Reports
All Reports
Analytic View Reports
Data Dictionary Report
Data Modeler Reports

Workflow Jobs

SQL Worksheet History

BORROWED 5 of 6 Aa “ ” A A * |

Worksheet Query Builder

```
SELECT DISTINCT B.BOOK_TITLE,
   A.AUTHOR_NAME
  FROM BOOKS_440 B
 JOIN (
    SELECT BA.BOOK_ID,
           AU.AUTHOR_NAME
      FROM BOOK_AUTHORS_440 BA
      JOIN AUTHORS_440 AU ON BA.AUTHOR_ID = AU.AUTHOR_ID
   ) A ON B.BOOK_ID = A.BOOK_ID
 WHERE B.PUBLICATION_YEAR > (
   SELECT AVG(PUBLICATION_YEAR)
     FROM BOOKS_440
   );

```

Query Result All Rows Fetched: 4 in 0.015 seconds

BOOK_TITLE	AUTHOR_NAME
1 HARRY POTTER	J.K. ROWLING
2 Anna Karenina	LEO TOLSTOY
3 War and Peace	LEO TOLSTOY

3. Let's get information about books, authors, publishers, languages, and available copies while considering borrowing status and sorting the results.

HAVING clause filters out books with no available copies.

The results are ordered by the number of available copies in descending order and then by book title.

The screenshot shows the SQL Developer interface. The left sidebar contains 'Connections' (with 'PiyumikaDB' selected), 'Assignments' (with 'Tables (Filtered)' selected), 'Reports' (with 'All Reports'), and 'Workflow Jobs'. The main area has tabs for 'Worksheet' and 'Query Builder'. The 'Worksheet' tab displays the following SQL code:

```
// Let's get information about books, authors, publishers, languages, and available copies while
// considering borrowing status and sorting the results
// HAVING clause filters out books with no available copies.
// The results are ordered by the number of available copies in descending order and then by book title.

SELECT
    B.BOOK_TITLE, A.AUTHOR_NAME, P.PUBLISHER_NAME, L.LANGUAGE_NAME, COUNT(C.COPY_ID) AS AVAILABLE_COPIES
FROM BOOKS_440 B
JOIN (
    SELECT BA.BOOK_ID, LISTAGG(AU.AUTHOR_NAME, ', ') WITHIN GROUP (ORDER BY AU.AUTHOR_NAME) AS AUTHOR_NAME
    FROM BOOK_AUTHORS_440 BA
    JOIN AUTHORS_440 AU ON BA.AUTHOR_ID = AU.AUTHOR_ID
    GROUP BY BA.BOOK_ID
) A ON B.BOOK_ID = A.BOOK_ID
JOIN PUBLISHERS_440 P ON B.PUBLISHER_ID = P.PUBLISHER_ID
JOIN LANGUAGES_440 L ON B.LANGUAGE_ID = L.LANGUAGE_ID
LEFT JOIN COPIES_440 C ON B.BOOK_ID = C.BOOK_ID
    AND C.COPY_ID NOT IN (SELECT COPY_ID FROM ISSUES_440 WHERE STATUS = 'BORROWED')
GROUP BY B.BOOK_TITLE, A.AUTHOR_NAME, P.PUBLISHER_NAME, L.LANGUAGE_NAME
HAVING COUNT(C.COPY_ID) > 0
ORDER BY
    AVAILABLE_COPIES DESC,
    B.BOOK_TITLE;
```

The 'Query Result' tab shows the output of the query:

BOOK_TITLE	AUTHOR_NAME	PUBLISHER_NAME	LANGUAGE_NAME	AVAILABLE_COPIES
1 HARRY POTTER	J.K. ROWLING	McClelland And Stewart	ENGLISH	2
2 Romeo and Juliet	SHAKESPEARE	Penguin Random House	ENGLISH	2
3 HARRY POTTER	J.K. ROWLING	McClelland And Stewart	FRENCH	1

4. Find users who have borrowed books using correlated subquery.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections and Reports panes are visible. The main area shows a SQL Worksheet with the following query:

```
//Find users who have borrowed books using correlated subquery.
SELECT USER_NAME
FROM USERS_440 U
WHERE EXISTS (
    SELECT 1
    FROM ISSUES_440 I
    JOIN COPIES_440 C ON I.COPY_ID = C.COPY_ID
    WHERE U.USER_ID = I.USER_ID
);
```

The Query Result pane displays the results:

USER_NAME
1 PIYUMIKA BANDULA
2 KRITHIK KUMAR
3 HELEN PEIRIS
4 WINDY SHENON
5 BEULA PERERA
6 GILLIAND JOSEPH

5. List authors whom the library has more books than the average number of books per author in the library.

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections and Reports panes are visible. The main area shows a SQL Worksheet with the following query:

```
// List authors whom the library has more books than the average number of books per author in the library.
SELECT AUTHOR_NAME
FROM AUTHORS_440 a
WHERE (SELECT COUNT(*) FROM BOOK_AUTHORS_440 ba WHERE ba.AUTHOR_ID = a.AUTHOR_ID) >
    (SELECT AVG(COUNT(*)) FROM BOOK_AUTHORS_440 GROUP BY AUTHOR_ID);
```

The Query Result pane displays the results:

AUTHOR_NAME
1 SHAKESPEARE
2 J.K. ROWLING
3 LEO TOLSTOY