RIGA TECHNICAL UNIVERSITY
FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
INSTITUTE OF APPLIED COMPUTER SYSTEMS

Practical Assignment #5
“Database Management Systems”
PL/SQL

Author: Emir Oğuz
Course, Group: DSP201, Group 1
Student Card No: 230ADB011

Checked: Andrejs Gaidukovs

2022 / 2023 Study Year

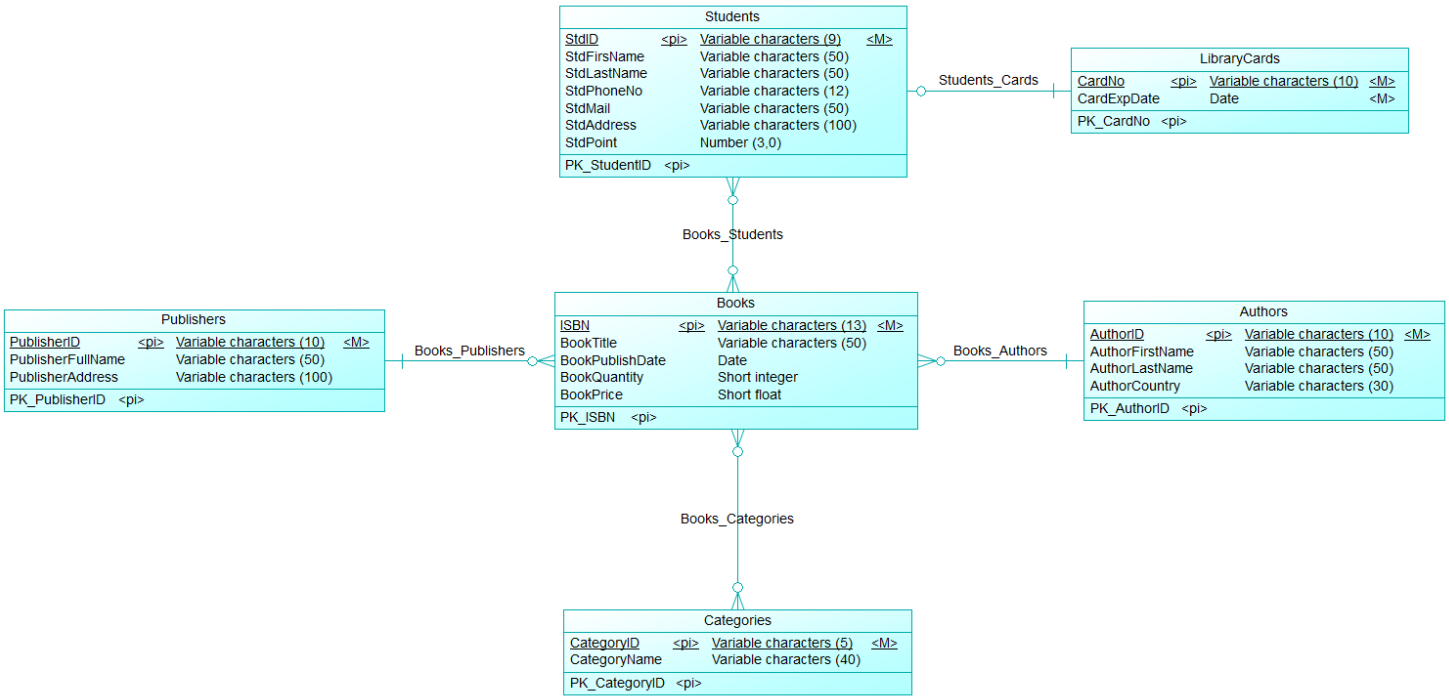
Content

1 Task

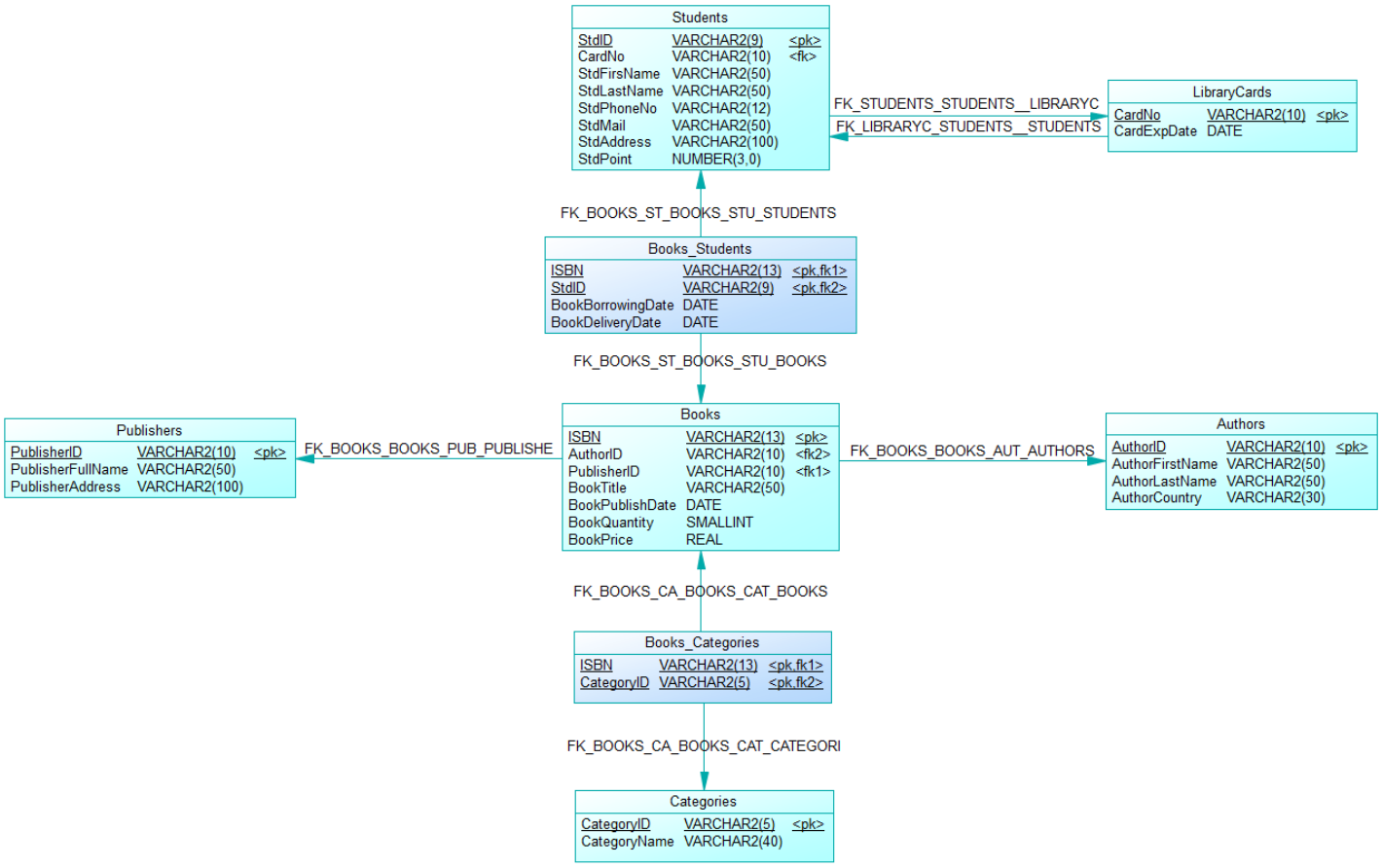
1. Write a PL/SQL program, which overwrites database table data using PL/SQL cursor.
2. Create a database trigger and test its operation. Visualize the test results to show that the trigger is operating and performing the specified functions.
3. Create function and procedure into SQL WITH statement. Test their operations and show results.
4. Create a PL/SQL program that demonstrates the use of dynamic SQL and dynamic PL/SQL.
5. Conclusions
6. Submit to Ortus:
    a. Report file in MS Word format names: DBMS_6_Surname.docx
    b. SQL script: DBMS_6_Surname.sql

2 Database Description

Logical



Physical



3 Main Sections of Practical Work

1. Write a PL/SQL program, which overwrites database table data using PL/SQL cursor.

Code

```
DECLARE
  CURSOR c_books IS
    SELECT * FROM BOOKS;

  TYPE t_books IS TABLE OF c_books%ROWTYPE;
  l_books t_books;

BEGIN
  OPEN c_books;
  FETCH c_books BULK COLLECT INTO l_books;
  CLOSE c_books;

  FOR i IN 1..l_books.COUNT LOOP
    l_books(i).BOOKTITLE := 'New Title ' || i;
  END LOOP;

  FORALL i IN 1..l_books.COUNT
    UPDATE BOOKS
    SET BOOKTITLE = l_books(i).BOOKTITLE
    WHERE ISBN = l_books(i).ISBN;

  COMMIT;
END;
/

SELECT * FROM BOOKS;
```

```
PL/SQL procedure successfully completed.

>>Query Run In:Query Result
```

"Books" Table Before the Code Above

	ISBN	AUTHORID	PUBLISHERID	BOOKTITLE	BOOKPUBLISHDATE	BOOKQUANTITY	BOOKPRICE
1	9780545010221	A01	PUB01	Harry Potter and the Deathly Hallows	21-07-2007	100	19.99
2	9781501142970	A02	PUB02	The Outsider	22-05-2018	50	18.99
3	9780060935467	A03	PUB03	To Kill a Mockingbird	11-07-1960	200	10.99
4	9780307389733	A04	PUB04	One Hundred Years of Solitude	30-05-1967	75	14.99
5	9780385490818	A05	PUB05	The Handmaid's Tale	14-06-1985	150	12.99
6	9780807610664	A06	PUB06	Things Fall Apart	17-06-1958	125	8.99
7	9781400033423	A07	PUB07	Beloved	02-09-1987	100	11.99
8	9780679745587	A08	PUB08	The God of Small Things	04-04-1997	80	9.99
9	9781594480003	A09	PUB09	The Kite Runner	29-05-2003	90	13.99
10	9780007548699	A10	PUB10	The House of the Spirits	22-06-1982	70	10.99

"Books" Table After the Code Above

	ISBN	AUTHORID	PUBLISHERID	BOOKTITLE	BOOKPUBLISHDATE	BOOKQUANTITY	BOOKPRICE
1	9780545010221	A01	PUB01	New Title 1	21-07-2007	100	19.99
2	9781501142970	A02	PUB02	New Title 2	22-05-2018	50	18.99
3	9780060935467	A03	PUB03	New Title 3	11-07-1960	200	10.99
4	9780307389733	A04	PUB04	New Title 4	30-05-1967	75	14.99
5	9780385490818	A05	PUB05	New Title 5	14-06-1985	150	12.99
6	9780807610664	A06	PUB06	New Title 6	17-06-1958	125	8.99
7	9781400033423	A07	PUB07	New Title 7	02-09-1987	100	11.99
8	9780679745587	A08	PUB08	New Title 8	04-04-1997	80	9.99
9	9781594480003	A09	PUB09	New Title 9	29-05-2003	90	13.99
10	9780007548699	A10	PUB10	New Title 10	22-06-1982	70	10.99

2. Create a database trigger and test its operation. Visualize the test results to show that the trigger is operating and performing the specified functions.

Code

```
CREATE OR REPLACE TRIGGER trg_modify_book_title
BEFORE INSERT ON BOOKS
FOR EACH ROW
BEGIN
  :NEW.BOOKTITLE := 'Alias Grace (Trigger Modified Title)';
END;
/

INSERT INTO BOOKS (ISBN, AUTHORID, PUBLISHERID, BOOKTITLE, BOOKPUBLISHDATE,
BOOKQUANTITY, BOOKPRICE)
VALUES ('9780771008351', (SELECT AUTHORID FROM AUTHORS WHERE AUTHORID = 'A05'),
(SELECT PUBLISHERID FROM PUBLISHERS WHERE PUBLISHERID = 'PUB05'), 'Original Title',
TO_DATE('1997/02/27', 'YYYY/MM/DD'), 95, 8.99);

SELECT * FROM BOOKS;
```

```
Trigger TRG_MODIFY_BOOK_TITLE compiled


1 row inserted.

>>Query Run In:Query Result
```

"Books" Table Before the Code Above

	ISBN	AUTHORID	PUBLISHERID	BOOKTITLE	BOOKPUBLISHDATE	BOOKQUANTITY	BOOKPRICE
1	9780545010221	A01	PUB01	Harry Potter and the Deathly Hallows	21-07-2007	100	19.99
2	9781501142970	A02	PUB02	The Outsider	22-05-2018	50	18.99
3	9780060935467	A03	PUB03	To Kill a Mockingbird	11-07-1960	200	10.99
4	9780307389733	A04	PUB04	One Hundred Years of Solitude	30-05-1967	75	14.99
5	9780385490818	A05	PUB05	The Handmaid's Tale	14-06-1985	150	12.99
6	9780807610664	A06	PUB06	Things Fall Apart	17-06-1958	125	8.99
7	9781400033423	A07	PUB07	Beloved	02-09-1987	100	11.99
8	9780679745587	A08	PUB08	The God of Small Things	04-04-1997	80	9.99
9	9781594480003	A09	PUB09	The Kite Runner	29-05-2003	90	13.99
10	9780007548699	A10	PUB10	The House of the Spirits	22-06-1982	70	10.99

"Books" Table After the Code Above

	ISBN	AUTHORID	PUBLISHERID	BOOKTITLE	BOOKPUBLISHDATE	BOOKQUANTITY	BOOKPRICE
1	9780545010221	A01	PUB01	Harry Potter and the Deathly Hallows	21-07-2007	100	19.99
2	9781501142970	A02	PUB02	The Outsider	22-05-2018	50	18.99
3	9780060935467	A03	PUB03	To Kill a Mockingbird	11-07-1960	200	10.99
4	9780307389733	A04	PUB04	One Hundred Years of Solitude	30-05-1967	75	14.99
5	9780385490818	A05	PUB05	The Handmaid's Tale	14-06-1985	150	12.99
6	9780807610664	A06	PUB06	Things Fall Apart	17-06-1958	125	8.99
7	9781400033423	A07	PUB07	Beloved	02-09-1987	100	11.99
8	9780679745587	A08	PUB08	The God of Small Things	04-04-1997	80	9.99
9	9781594480003	A09	PUB09	The Kite Runner	29-05-2003	90	13.99
10	9780007548699	A10	PUB10	The House of the Spirits	22-06-1982	70	10.99
11	9780771008351	A05	PUB05	Alias Grace (Trigger Modified Title)	27-02-1997	95	8.99

3. Create function and procedure into SQL WITH statement. Test their operations and show results.

Code

```
CREATE OR REPLACE FUNCTION print_book_price(p_isbn IN VARCHAR2) RETURN REAL IS
  l_price REAL;
BEGIN
  SELECT BOOKPRICE INTO l_price
  FROM BOOKS
  WHERE ISBN = p_isbn;

  RETURN l_price;
END;
/

CREATE OR REPLACE PROCEDURE display_book_details(p_isbn IN VARCHAR2) IS
  l_isbn VARCHAR2(13);
  l_title VARCHAR2(50);
  l_publishdate DATE;
  l_price REAL;
BEGIN
  SELECT ISBN, BOOKTITLE, BOOKPUBLISHDATE, BOOKPRICE INTO l_isbn, l_title, l_publishdate, l_price
  FROM BOOKS
  WHERE ISBN = p_isbn;

  DBMS_OUTPUT.PUT_LINE('ISBN: ' || l_isbn);
  DBMS_OUTPUT.PUT_LINE('Title: ' || l_title);
  DBMS_OUTPUT.PUT_LINE('Publish Date: ' || l_publishdate);
  DBMS_OUTPUT.PUT_LINE('Price: ' || l_price);
END;
/

WITH
  book_data AS (
    SELECT '9780545010221' AS ISBN, print_book_price('9780545010221') AS book_price FROM DUAL
  )
SELECT ISBN, book_price FROM book_data;

BEGIN
  display_book_details('9780545010221');
END;
/
```

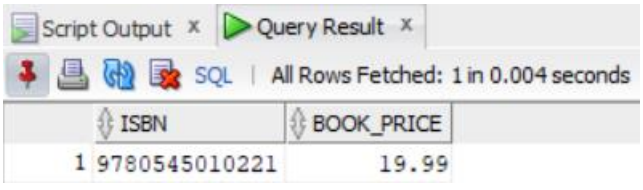
Results

```
Function PRINT_BOOK_PRICE compiled


Procedure DISPLAY_BOOK_DETAILS compiled

>>Query Run In:Query Result
ISBN: 9780545010221
Title: Harry Potter and the Deathly Hallows
Publish Date: 21-07-2007
Price: 19.99


PL/SQL procedure successfully completed.
```



4. Create a PL/SQL program that demonstrates the use of dynamic SQL and dynamic PL/SQL.

Code

```
DECLARE
  l_sql_stmt VARCHAR2(500);
  l_result NUMBER;

BEGIN
  l_sql_stmt := 'SELECT COUNT(*) FROM BOOKS WHERE AUTHORID = :author_id';
  EXECUTE IMMEDIATE l_sql_stmt INTO l_result USING 'A01';
  DBMS_OUTPUT.PUT_LINE('Number of books by author: ' || l_result);

  l_sql_stmt := 'CREATE OR REPLACE PROCEDURE dynamic_procedure IS
                   l_book_count NUMBER;
                 BEGIN
                   SELECT COUNT(*) INTO l_book_count FROM BOOKS;
                   DBMS_OUTPUT.PUT_LINE(''Total books: '' || l_book_count);
                   DBMS_OUTPUT.PUT_LINE(''Hello from Dynamic PL/SQL!'');
                 END;';
  EXECUTE IMMEDIATE l_sql_stmt;

  EXECUTE IMMEDIATE 'BEGIN dynamic_procedure; END;';

  EXECUTE IMMEDIATE 'DROP PROCEDURE dynamic_procedure';
END;
/
```

Results

```
Number of books by author: 1
Total books: 10
Hello from Dynamic PL/SQL!


PL/SQL procedure successfully completed.
```

4 Conclusions

During this task, I successfully completed various tasks related to the Oracle 12c database system. I started by writing a PL/SQL program that overwrites database table data using a PL/SQL cursor. By utilizing a cursor, I was able to fetch and update the table data efficiently, ensuring the desired changes were made.

Next, I created a database trigger based on the provided SQL script. The trigger was designed to execute before inserting a new row into the BOOKS table, modifying the BOOKTITLE column. Testing the trigger confirmed its functionality, as the inserted row's BOOKTITLE was automatically updated according to the trigger's logic.

I then proceeded to create a function and a procedure within a SQL WITH statement. Testing these operations validated their successful execution, demonstrating the accurate retrieval and modification of data.

Lastly, I showcased the usage of dynamic SQL and dynamic PL/SQL by creating a PL/SQL program. The program utilized dynamic SQL to query the count of books by a specific author and dynamically created a procedure that retrieved and displayed the total number of books. The dynamic nature of these constructs allowed for flexibility and adaptability in working with SQL statements and PL/SQL blocks.

Overall, these tasks provided a comprehensive understanding of various PL/SQL concepts and their practical implementation. By working through these exercises, I gained valuable experience in writing efficient code, utilizing triggers, functions, and procedures, and harnessing the power of dynamic SQL and dynamic PL/SQL in an Oracle 12c database environment.

5 References

- Riga Technical University, Faculty of Computer Science and Information Technology, Institute of Applied Computer Systems, DSP201 – Database Management Systems, Presentations