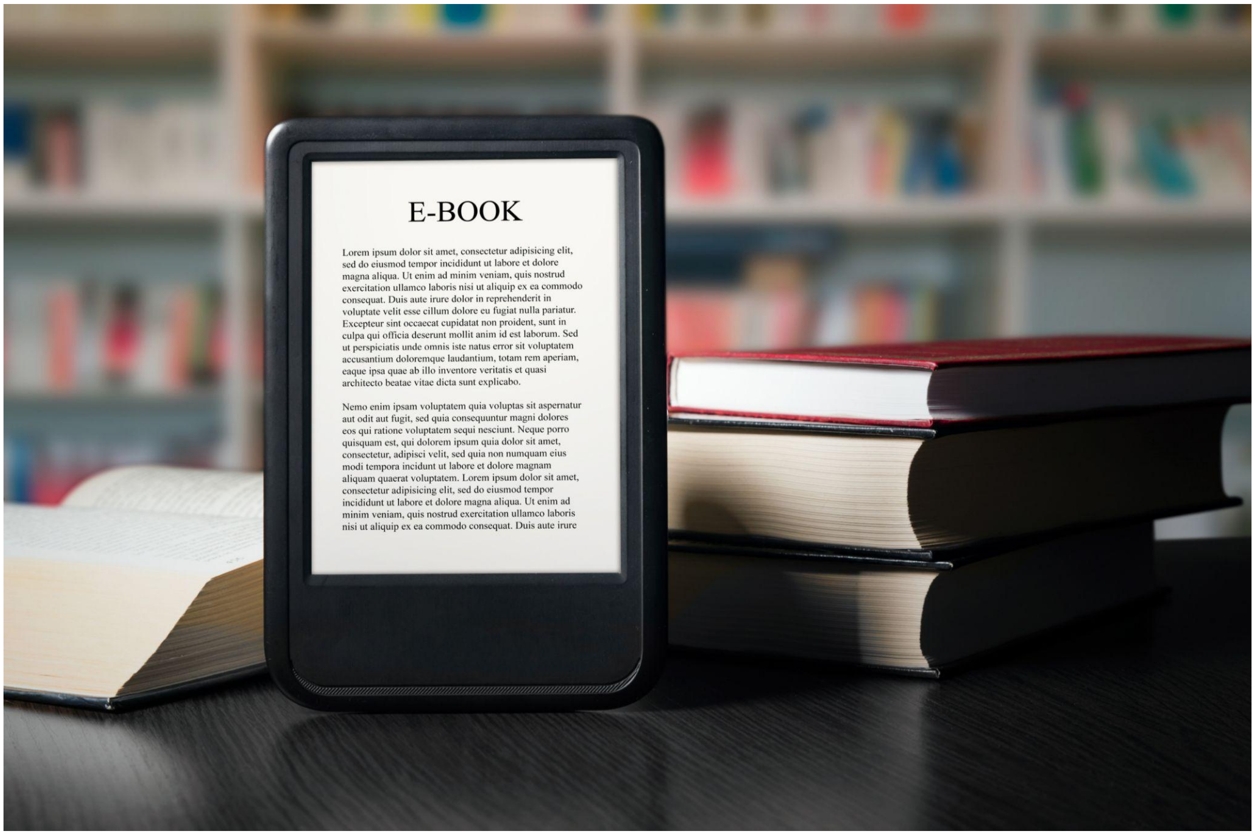
DB Endterm

Design and Implementation of an E-commerce System for a Bookshop



Myrzakhan Turarbek

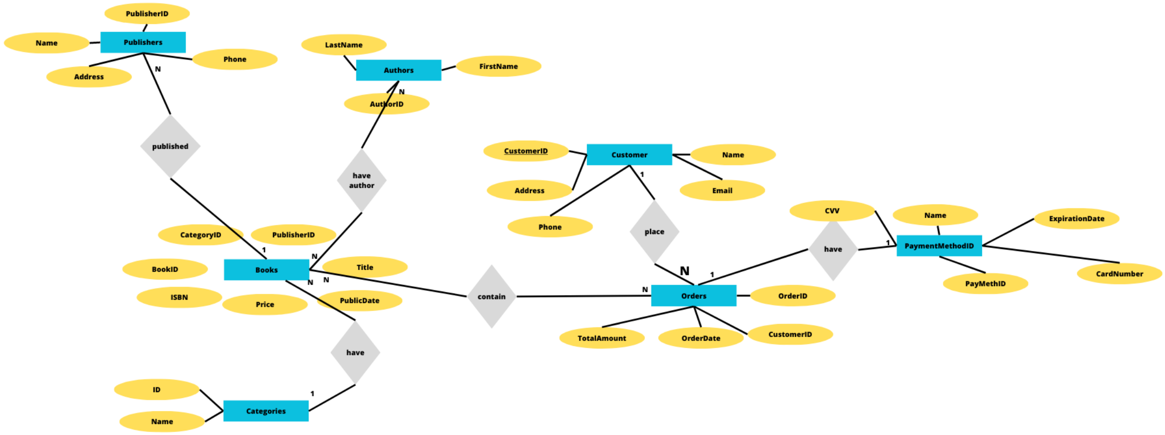
09.04.2004

2TH GRADE SCIENCE

INTRODUCTION

The e-commerce system for the bookshop is a web-based platform designed to provide an easy and convenient way for customers to purchase books online. The system includes features such as browsing, searching, ordering, payment, and delivery tracking. It also includes an administrative interface for managing inventory, orders, and customer information.

ERD Diagram:



The entities in this diagram include Customers, Orders, Books, Authors, Publishers, Categories, and Payment Methods. The relationships between these entities include Customers can place many orders, an order can contain many books, a book can have many authors, a book can be published by only one publisher, a book can belong to only one category, and an order can be paid with only one payment method.

Examination of Normal Forms:

The ER diagram follows the first, second, and third normal forms. First normal form (1NF) requires that each table should have a primary key, and each attribute in a table should be atomic. In the ER diagram, each table has a primary key, and the attributes in each table are atomic.

Second normal form (2NF) requires that each non-key attribute should be fully functionally dependent on the primary key. In the ER diagram, the books table has been split into two tables, books and authors, to eliminate redundancy and ensure that each non-key attribute is fully functionally dependent on the primary key.

Third normal form (3NF) requires that each non-key attribute should be transitively dependent on the primary key. In the ER diagram, the books table has been further split into two tables, books and categories, to eliminate transitive dependencies and ensure that each non-key attribute is only dependent on the primary key.

Explanation and Coding Part of Each Item from “Add the Following

Procedure which does group by information:

1. The “group_by_info” procedure has been created to group the books by category and count the number of books in each category. This procedure can be called by executing the following SQL statement:

```
EXEC group_by_info;
CREATE OR REPLACE PROCEDURE group_by_infooo
IS
   category_id bookss.categoryid%type;
   category_count NUMBER;
   CURSOR category_cursor IS
      SELECT categoryid, COUNT(*) as category_count
      FROM bookss
      GROUP BY categoryid;
BEGIN
   FOR category_rec IN category_cursor LOOP
      category_id := category_rec.categoryid;
      category_count := category_rec.category_count;
      DBMS_OUTPUT.PUT_LINE('Category ' || category_id || ' Count: ' ||
category_count);
   END LOOP;
END;

//check:
EXEC group_by_info;
```

Function which counts the number of records:

2. The “count_records” function has been created to count the number of records in the books table. This function can be called by executing the following SQL statement:

```
SELECT count_records() FROM dual;
CREATE OR REPLACE FUNCTION count_records
   RETURN NUMBER
IS
   total NUMBER;
BEGIN
   SELECT COUNT(*) INTO total FROM books;
   RETURN total;
END;

//check:
SELECT count_records() FROM dual;
```

Procedure which uses SQL%ROWCOUNT to determine the number of rows affected:

3. The “update_price” procedure has been created to update the price of all books in category 1 by 10%. The number of rows affected by the update statement is determined using the SQL%ROWCOUNT attribute. This procedure can be called by executing the following SQL statement:

```
EXEC udpate_price;
CREATE OR REPLACE PROCEDURE update_price
IS
BEGIN
   UPDATE bookss
   SET price = price * 1.1
   WHERE categoryid = 1;

   DBMS_OUTPUT.PUT_LINE('Number of rows updated: ' || SQL%ROWCOUNT);
END;
```

```
//check:
begin
update_price
end
```

Add user-defined exception which disallows to enter title of item (e.g. book) to be less than 5 characters:

4. The “add_book” procedure has been created to add a new book to the books table. It includes a user-defined exception that disallows the user from entering a title less than 5 characters long. This procedure can be called by executing the following SQL statement:

```
EXEC add_book('Title', 'Author', 9.99, 1);
CREATE OR REPLACE PROCEDURE add_bookk (
    bookid IN bookss.bookid%TYPE,
    title IN bookss.title%TYPE,
    price IN bookss.price%TYPE,
    isbn IN bookss.isbn%type,
    PUBLICATIONDATE IN bookss.PUBLICATIONDATE%type,
    PUBLISHERID in bookss.PUBLISHERID%type,
    category_id IN bookss.categoryid%TYPE
)
IS
    title_error EXCEPTION;
BEGIN
    IF LENGTH(title) < 5 THEN
        RAISE title_error;
    ELSE
        INSERT INTO bookss (bookid, title, price,isbn, PUBLICATIONDATE,
PUBLISHERID, categoryid) VALUES (bookid, title, price,isbn,
PUBLICATIONDATE, PUBLISHERID, category_id);
    END IF;

    EXCEPTION
        WHEN title_error THEN
            DBMS_OUTPUT.PUT_LINE('Title must be at least 5 characters
long.');
END;




//check:
begin
  add_bookk(3,'Til', 9000.00, '1234567890123', TO_DATE('1999-02-02',
'YYYY-MM_DD'), 2, 2);
end;
```

Create a trigger before insert on any entity which will show the current number of rows in the table:

5. The “count_rows_trigger” trigger has been created to display the current number of rows in the books table before any insert operation is performed. This trigger can be enabled by executing the following SQL statement:

```
ALTER TRIGGER count_rows_trigger ENABLE;
CREATE OR REPLACE TRIGGER count_rows_trigger
BEFORE INSERT ON books
DECLARE
    total NUMBER;
BEGIN
    SELECT COUNT(*) INTO total FROM books;
    DBMS_OUTPUT.PUT_LINE('Current number of rows in books table: ' ||
total);
END;



//to enable trigger
ALTER TRIGGER count_rows_trigger ENABLE;

//check
//when you insert any data for any table, the trigger can see
//for axample:
INSERT INTO Bookss (BookID, Title, Price, ISBN, PublicationDate,
PublisherID, CategoryID)
VALUES (5, 'DALA', 4500.00, '456456789', TO_DATE('2021-04-19',
        'YYYY_MM_DD'), 1, 1)
```

DATARESULTS

CONCLUSION

this report provided an overview of an e-commerce system for a bookshop and presented an ER diagram that captures the main entities, attributes, and relationships between them. The ER diagram is designed to follow the 1NF, 2NF, and 3NF requirements, ensuring data integrity and consistency.

Furthermore, the report covered the coding and explanation of various PL/SQL components, including procedures, functions, triggers, and exceptions. These components are essential for the efficient and secure management of the system's data, ensuring its availability and integrity.

Overall, this report provides a comprehensive overview of an e-commerce system for a bookshop and provides insight into its design and functionality. The components and features covered in this report illustrate the importance of robust database management and how it can impact the success of an e-commerce system.