DATABASE MANAGEMENT SYSTEMS
PROJECT REPORT

Group Members :

1. Divyanshu Talwar (2015028)
2. Shashwat Malik (2015092)

Project Description :

- **Objective**
  - To design a database application for an Online Book Store.
- **Features : The features of the Java application**
  - Lets the User to do the following:
    - Signup or Login
    - See Best Sellers.
    - Add Book to cart.
    - See Numbers of customers who bought a book.
    - Books in a particular price range.
    - See Most Popular Authors.
    - Search Book by a keyphrase.
    - Search book by a genre or a set of genres.
    - Search book by an author or a set of authors.
    - Search book by a publisher or a set of publishers.
  - Lets the Admin to do the following:
    - Login.
    - See Best Sellers.
    - Add Book to cart.
    - See Numbers of customers who bought a book.

- Books in a particular price range.
- See Most Popular Authors.
- Search Book by a keyphrase.
- Search book by a genre or a set of genres.
- Search book by an author or a set of authors.
- Search book by a publisher or a set of publishers.
- Total customers registered.
- Books that are more than ‘x’ in the stock.
- Stock Summary.
- Customers who bought all the books written by an Author “ABC”.
- Customers who bought all the books published by a Publisher “ABC”.

- **SQL Queries : The application caters to the following user queries**
  - **Simple Queries:**

1. Add a book to cart:

```
INSERT INTO Contains VALUES ('name', 'basketID ', ' quantity');

UPDATE Stocks SET stock_quantity = stock_quantity -  quantity WHERE ISBN = 'ISBN';
```

2. Best Sellers (top 5):

```
SELECT * FROM  ( SELECT SUM(basket_quantity) AS qty, ISBN FROM Contains GROUP BY(ISBN) ) AS A
NATURAL JOIN Book ORDER BY(quantity) DESC LIMIT 5;
```

3. Number of customers who bought a particular book:

```
SELECT COUNT(basketID) AS number FROM Contains GROUP BY (ISBN) HAVING ISBN = 'ISBN';
```

4. List all books whose price lies in a particular price range:

```
SELECT * FROM Book WHERE price BETWEEN a AND b;
```

5. Find a particular book by its name / keyphrase :

```
SELECT * FROM Book WHERE title LIKE '% phrase%;
```

6. Stock Summary:

```
SELECT S.stock_quantity, S.warehouse_code, B.title, B.ISBN FROM Stocks S NATURAL JOIN Book B ORDER BY(S.stock_quantity);
```

7. Books that are more than ‘x’ in the stock:

```
SELECT * FROM (SELECT sum(stock_quantity) AS qty , ISBN FROM Stocks GROUP BY(ISBN)) AS A NATURAL JOIN Book WHERE A.qty >='x';
```

8. Total number of customers registered :

```
SELECT COUNT(customer_email) FROM Customer;
```

- **Complex Queries:**

1. Find all books written by an author ABC or a set of authors :

```
for(int i = 0; i<Authors.length; i++){

	if(i == Authors.length -1) {

		query += "SELECT A.author_name, C.title, C.price, C.genre, C.year FROM Author A, WrittenBy W, Book C WHERE C.ISBN = W.ISBN AND W.author_name = A.author_name AND A.author_name = '"+Authors[i]+"';";

	}

	else{

		query += "SELECT A.author_name, C.title, C.price, C.genre, C.year FROM Author A, WrittenBy W, Book C WHERE C.ISBN = W.ISBN AND W.author_name = A.author_name AND A.author_name = '"+Authors[i]+"' UNION ";

	}

}
```

2. Find all books published by a publisher ABC or a set of publishers :

```
for(int i = 0; i<Publishers.length; i++){

	if(i == Publishers.length -1) {
```

```
            query += "SELECT A.publisher_name, C.title, C.price, C.genre, C.year FROM Publisher
            A,  PublishedBy W, Book C WHERE C.ISBN = W.ISBN AND W.publisher_name =
            A.publisher_name AND A.publisher_name = '"+Publishers[i]+"';";

        }

        else{

            query += "SELECT A.publisher_name, C.title, C.price, C.genre, C.year FROM Publisher
            A,  PublishedBy W, Book C WHERE C.ISBN = W.ISBN AND W.publisher_name =
            A.publisher_name AND A.publisher_name = '"+Publishers[i]+"' UNION ";

        }

}
```

3. List names of all customers who bought all the books written by ABC:

```
SELECT * FROM (SELECT basketID FROM Contains WHERE ISBN IN ( SELECT C.ISBN  FROM Author A,
WrittenBy W, Book C WHERE C.ISBN = W.ISBN AND A.author_name = W.author_name AND A.author_name =
'"+author+"') GROUP BY(Contains.basketID) HAVING COUNT(*) = (SELECT COUNT(*) FROM (Select C.ISBN
FROM Author A, WrittenBy W, Book C WHERE C.ISBN = W.ISBN AND A.author_name = W.author_name AND
A.author_name = '"+author+"') AS D)) AS E NATURAL JOIN ShoppingBasket NATURAL JOIN Customer;
```

4. List names of all customers who bought all the books published by ABC. :

```
SELECT * FROM (SELECT basketID FROM Contains WHERE ISBN IN ( SELECT C.ISBN  FROM Publisher A,
PublishedBy W, Book C WHERE C.ISBN = W.ISBN AND A.publisher_name = W.publisher_name AND
A.publisher_name = '"+publisher+"') GROUP BY(Contains.basketID) HAVING COUNT(*) = (SELECT COUNT(*)
FROM (SELECT C.ISBN  FROM Publisher A, PublishedBy W, Book C WHERE C.ISBN = W.ISBN AND
A.publisher_name = W.publisher_name AND A.publisher_name = '"+publisher+"') AS D)) AS E NATURAL
JOIN ShoppingBasket NATURAL JOIN Customer;
```

5. Find all books which belong to a specific genre or a set of authors :

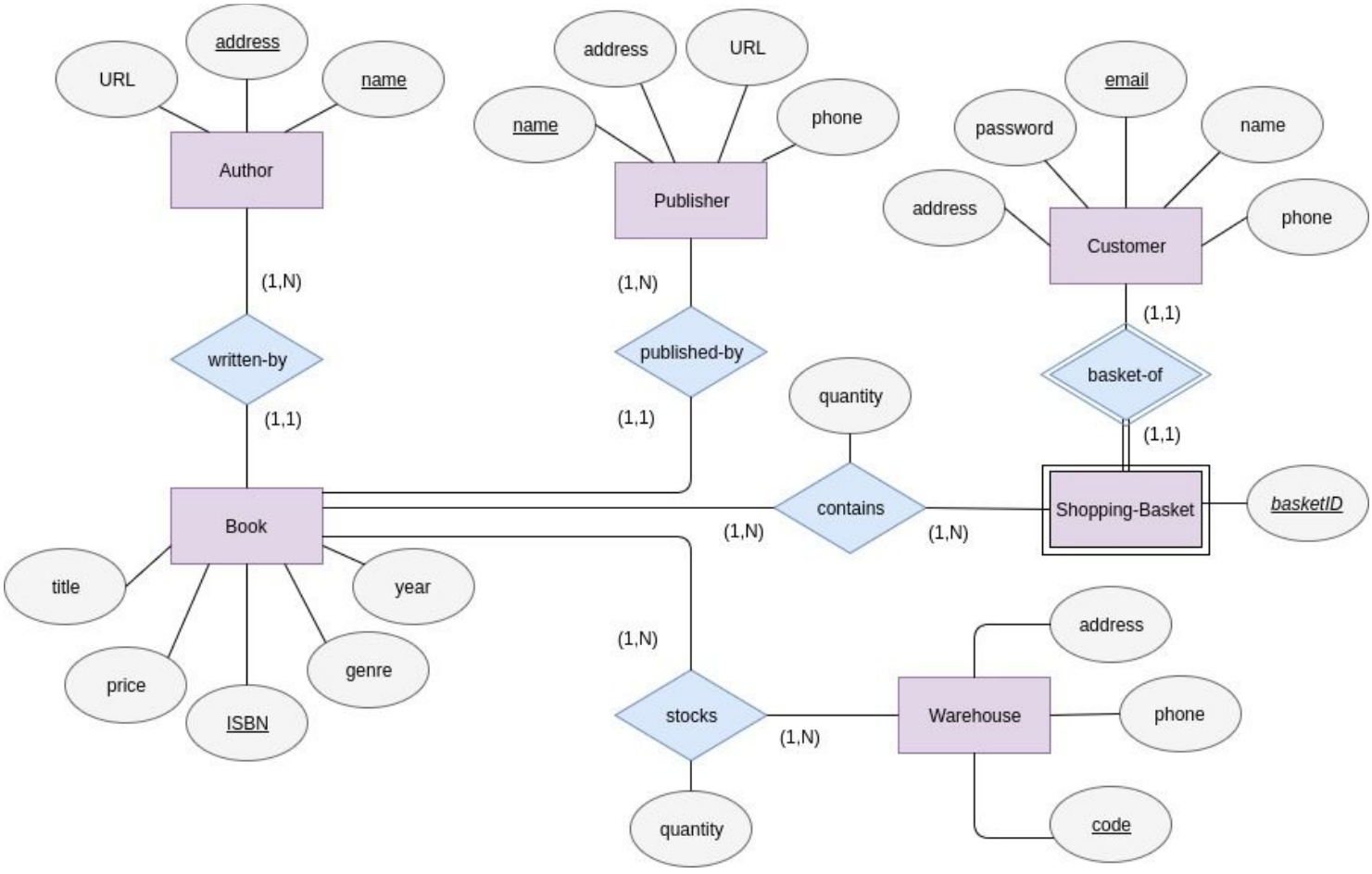
```
for(int i = 0; i<Genre.length; i++){

        if(i == Genre.length -1) {

            query += "SELECT C.title, C.price, C.genre, C.year FROM Book C WHERE C.genre LIKE
'%"+Genre[i]+"%';";

        }
```

```
        else{

                query += "SELECT C.title, C.price, C.genre, C.year FROM Book C where C.genre LIKE
'%"+Genre[i]+"%' UNION ";

        }

}
```

6. List the most popular authors :

```
SELECT D.name AS n, COUNT(D.quantity) AS qty FROM (SELECT A.author_name AS name, C.basket_quantity
AS quantity  FROM Author A,  WrittenBy W, Contains C WHERE C.ISBN = W.ISBN AND W.author_name =
A.author_name) AS D GROUP BY(n) ORDER BY(qty) DESC LIMIT 5;
```

- **ER-DIAGRAM:**



- **RELATIONAL SCHEMA:**

- Author (author_name, author_address, author_URL)
    - name, address(alphanumeric) and URL are all characters(VARCHAR).
- Publisher (publisher_name, publisher_address, publisher_URL, publisher_phone)
    - name, address(alphanumeric) and URL are all characters(VARCHAR).
    - phone is a number(NUMBER).
- Customer (customer_email, customer_name, customer_address, customer_phone, password)
    - name, address (alphanumeric) and URL are all characters(VARCHAR).
    - phone is a number(NUMBER).
    - password is alphanumeric(VARCHAR).
- Shopping-Basket (customer_email , basketID)
    - basketID is alphanumeric and customer_email is a character (VARCHAR).
- Book (title, price, genre, year, ISBN)
    - title and genre are characters (VARCHAR).
    - price, year and ISBN are numbers (NUMBERS).
- Warehouse (warehouse_code, warehouse_phone, warehouse_address)
    - phone is a number (NUMBERS).
    - code and address are alphanumeric (VARCHAR).
- Written-by (author_name, author_address, ISBN)
    - ISBN is a number (NUMBERS).
    - author_name and author_address(alphanumeric) are characters (VARCHAR).
- Published-by (publisher_name, ISBN)
    - ISBN is a number (NUMBERS).
    - publisher_name is a character (VARCHAR).
- Contains (ISBN, basketID, basket_quantity)
    - ISBN, quantity are numbers (NUMBERS).
    - basketID is alphanumeric (VARCHAR).
- Stocks (ISBN, warehouse_code, stock_quantity)
    - ISBN, quantity are numbers (NUMBERS).
    - code is alphanumeric (VARCHAR).