

Personal Library

1. What this project is

This project is a **database system for managing books**.

It stores information about:

- Students (users)
- Administrators
- Books
- Publishers
- Categories
- Reviews

The system also has **triggers** (rules that run automatically) and **views** (saved queries) for reports.

2. What you can do with it

Student login (features)

- View all books with details.
- Search books by title, author, category, or publisher.
- Download a book (download count goes up).
- Write a review for a book.
- See their own profile and reviews.

Administrator login (features)

- Add, edit, or remove books.
 - Manage publishers and categories.
 - Manage student accounts.
 - View reports, such as:
 - Students who downloaded more than average.
 - Books with no reviews.
 - Book details with publisher and category.
-

3. Tools you need

- **Oracle Database.**

- **Java JDK.**
 - **JDBC driver** for Oracle (ojdbc jar).
 - SQL tool like **SQL*Plus** or **SQL Developer**.
-

4. How to set it up

Step 1 — Create tables (schema structure)

Run these SQL statements in Oracle in this order:

-- 1. Categories

```
CREATE TABLE categories (  
    cat_id NUMBER PRIMARY KEY,  
    cat_name VARCHAR2(100) NOT NULL,  
    no_of_books NUMBER,  
    type VARCHAR2(50)  
);
```

-- 2. Publisher

```
CREATE TABLE publisher (  
    pub_id NUMBER PRIMARY KEY,  
    pub_name VARCHAR2(150) NOT NULL,  
    aut_name VARCHAR2(150)  
);
```

-- 3. Student

```
CREATE TABLE student (  
    stu_id NUMBER PRIMARY KEY,  
    name VARCHAR2(100) NOT NULL,  
    username VARCHAR2(50) UNIQUE NOT NULL,  
    password VARCHAR2(200) NOT NULL,  
    contact VARCHAR2(15),  
    email VARCHAR2(150),  
    branch VARCHAR2(50),
```

```
year NUMBER,  
downloaded_books NUMBER DEFAULT 0  
);
```

-- 4. Administrator

```
CREATE TABLE administrator (  
ad_id NUMBER PRIMARY KEY,  
ad_username VARCHAR2(50) UNIQUE NOT NULL,  
ad_password VARCHAR2(200) NOT NULL,  
name VARCHAR2(100)  
);
```

-- 5. Books

```
CREATE TABLE books (  
book_id NUMBER PRIMARY KEY,  
book_name VARCHAR2(200) NOT NULL,  
aut_name VARCHAR2(150),  
cat_id NUMBER REFERENCES categories(cat_id),  
isbn VARCHAR2(20),  
status VARCHAR2(30),  
pub_year NUMBER,  
pub_id NUMBER REFERENCES publisher(pub_id),  
rating NUMBER DEFAULT 5  
);
```

-- 6. Review

```
CREATE TABLE review (  
rev_id NUMBER PRIMARY KEY,  
book_id NUMBER REFERENCES books(book_id),  
stu_id NUMBER REFERENCES student(stu_id),  
rev_text VARCHAR2(1000),
```

```
rev_date DATE DEFAULT SYSDATE  
);
```

Step 2 — Insert sample data

You can add your own data or use examples, for instance:

```
INSERT INTO categories VALUES (1, 'Computer Science', 3, 'Technical');
```

```
INSERT INTO publisher VALUES (1, 'OReilly', 'Various');
```

```
INSERT INTO student VALUES (1, 'Alice Kumar', 'alice', 'alicepass', '9876543210',  
'alice@example.com', 'CSE', 3, 2);
```

```
INSERT INTO administrator VALUES (1, 'admin', 'adminpass', 'Library Admin');
```

```
INSERT INTO books VALUES (1, 'Intro to Algorithms', 'Cormen', 1, '9780262033848', 'Available',  
2009, 1, 9);
```

```
INSERT INTO review VALUES (1, 1, 1, 'Great algorithms book.', SYSDATE);
```

Step 3 — Run Java console app

1. Compile your Java program with Oracle's ojdbc driver. Example:
2. `javac -cp .:ojdbc8.jar Main.java`
3. `java -cp .:ojdbc8.jar Main`
4. Use the logins:
 - **Student:** username alice, password alicepass
 - **Admin:** username admin, password adminpass