DataBase Course

12 MAY, 2025

InkVerse

Group no. 5

Team Members

Habiba Walid Hussein
Sara Mohamed Fares
Abdelrahman Yahia Sayed
Mosab Mohamed Sayed

Under Supervision

DR. Manal Shoman

individual

contributions

Habiba Walid Hussein

- Wrote the project overview, objectives, and business description of the bookstore.
- Identified the problem/opportunity and listed the information needs.
- Defined the types of data to be stored and their sources.
- Designed and documented the ER diagram (entities, attributes, relationships, and constraints).
- Contributed to writing, editing, and formatting the final report.

Sara Mohamed Fares

- Created the attribute domain table (names, data types, business/technical constraints).
- Converted the ER model to relational schema.
- Designed table structures with appropriate field types, primary keys, and foreign keys.
- Verified the logical structure of the database and ensured it matched conceptual design.

Abdelrahman Yahia Sayed

- Wrote the CREATE TABLE and ALTER TABLE SQL statements to enforce constraints.
- Ensured foreign key dependencies were respected and tested referential integrity.
- Validated that enough sample data was provided to demonstrate the use of the database.
- Supported others by checking table contents and correcting any data entry issues.

Mosab Mohamed Sayed

- Prepared realistic sample data and inserted it using INSERT INTO SQL statements.
- Wrote basic SELECT queries to retrieve data from individual tables , and designed queries to support simple use cases (e.g., list all books, find books by author or category, etc.).
- Tested and formatted query results for inclusion in the report
- Documented the purpose and expected result of each query in the final report.

Project

Purpose

Our Purpose was to design and implement a relational database system for a bookstore that manages key operations such as book listings, user accounts, customer orders, cart and wishlist functionalities, and book categorization.

The system centralizes all business-related data in a structured format, enabling smooth and efficient operations, reliable data storage, and a foundation for building user-facing applications in the future (e.g., a website or app).
The bookstore business model includes customers browsing books, adding them to their cart or wishlist, placing orders, and receiving shipments.

On the administrative side, staff (admins) are responsible for managing book entries, prices, and tracking stock or availability. Without an organized system, managing these operations manually can lead to inefficiencies, data duplication, and poor customer service.

This database project addresses these challenges by offering a unified system that automates inventory management, tracks orders, categorizes books, links books with publishers and authors, and stores user-related actions like wishlist or cart activity. It also supports administrative functions while ensuring data consistency and security.

Stored Data

Information

To solve the operational problems and support business growth, the system identifies and stores the following types of information:

- **Users:** Data about bookstore customers, including username, email, phone, address, and login credentials.
- **Admins**: Admin account information to allow secure book and inventory management.
- **Books**: Comprehensive book data including price, rating, page count, description, language, and publication date.
- **Orders**: Data on customer orders stored in Book_Order (total amount, shipping address, order status, payment status, etc.) and linked to users via User_Order.

- **Cart & Wishlist:** Temporary storage for books a user intends to buy or save, implemented using Cart, Wishlist, and their respective item tables.
- **Authors:** Details like author name and nationality, associated with specific books.
- **Publishers**: Publisher name and contact info, linked to books via Book_Publisher.

- **Categories**: Book classification using categories such as genre or type, stored in Category and related through Book_Category.
- **Order Items:** Specific books and their quantities in each order, stored in Order_items.

Conceptual Model

Domain Model

InkVerse
Domain model

The domain table below outlines the structure of our bookstore database.

- It lists all attributes, their data types, technical constraints (e.g., keys), and business descriptions.
- This serves as the foundation for a clear and well-structured database design that supports key bookstore operations.

Table Name	Attribute Name	Data Type	Technical Constraints	Business Constraints / Description
User	User_id	INT	PRIMARY KEY, NOT NULL	Unique identifier for each user
	username	VARCHAR(60)		Name of the user
	email	VARCHAR(60)		User email, ideally unique
	password	VARCHAR(60)		User password
	phone	VARCHAR(60)		Contact number
	address	VARCHAR(60)		Shipping or billing address
Admin	Admin_id	INT	PRIMARY KEY, NOT NULL	Unique admin identifier
	Admin_name	VARCHAR(60)		Name of the admin
	password	VARCHAR(60)		Admin password
Book	Book_id	INT	PRIMARY KEY, NOT NULL	Unique identifier for books
	Price	INT		Price in currency unit
	rating	INT		Rating out of 5
	page_count	INT		Number of pages
	Admin_id	INT	FOREIGN KEY → Admin(Admin_id)	Admin who added/approved the book
	quantity	INT		Quantity in stock
	describtion	VARCHAR(500)		Book description
	Language	VARCHAR(60)		Language of the book
	date	DATE		Date of addition/publication
Book_Order	Order_id	INT	PRIMARY KEY, NOT NULL	Unique order ID
	total_amount	INT		Total amount of the order
	shipping_address	VARCHAR(60)		Address for delivery
	tracking_number	INT		Tracking number for shipment
	order_status	VARCHAR(30)		Status (e.g., Shipped, Delivered)
	payment_status	VARCHAR(60		Paid/Unpaid/Refund

Conceptual Model

Domain Model

InkVerse
Domain model

				order
	shipping_address	VARCHAR(60)		Address for delivery
	tracking_number	INT		Tracking number for shipment
	order_status	VARCHAR(30)		Status (e.g., Shipped, Delivered)
	payment_status	VARCHAR(60)		Paid/Unpaid/Refunded
	order_date	DATE		Date of the order
Category	Cat_id	INT	PRIMARY KEY, NOT NULL	Unique category ID
	Cat_name	VARCHAR(60)		Category name (e.g., Fiction)
Publisher	Publisher_name	VARCHAR(60)	PRIMARY KEY	Unique name of the publisher
	email	VARCHAR(60)		Publisher contact email
Author	Author_name	VARCHAR(60)		Author name
	nationality	VARCHAR(60)		Country or origin
	Book_id	INT	FOREIGN KEY → Book(Book_id)	Book written by the author
Wishlist	user_id	INT	PRIMARY KEY, FOREIGN KEY	Wishlist tied to one user
	Book_id	INT	PRIMARY KEY, NOT NULL	Unique identifier for books
Cart	user_id	INT	PRIMARY KEY, FOREIGN KEY	Cart tied to one user
	Book_id	INT	PRIMARY KEY, NOT NULL	Unique identifier for books
Order_items	Book_id	INT	PRIMARY KEY, FOREIGN KEY	Book in the order
	quantity	INT		Quantity
	quantity	INT		Quantity of this book ordered
Book_Category	Cat_id	INT	FOREIGN KEY	Category of the book
	Book_id	INT	FOREIGN KEY	Book in the category
Book_Publisher	Publisher_name	VARCHAR(60)	FOREIGN KEY	Publisher
	Book_id	INT	FOREIGN KEY	Published book
User_Order	Order_id	INT	PRIMARY KEY, FOREIGN KEY	Order placed
	user_id	INT	FOREIGN KEY	User who placed the order

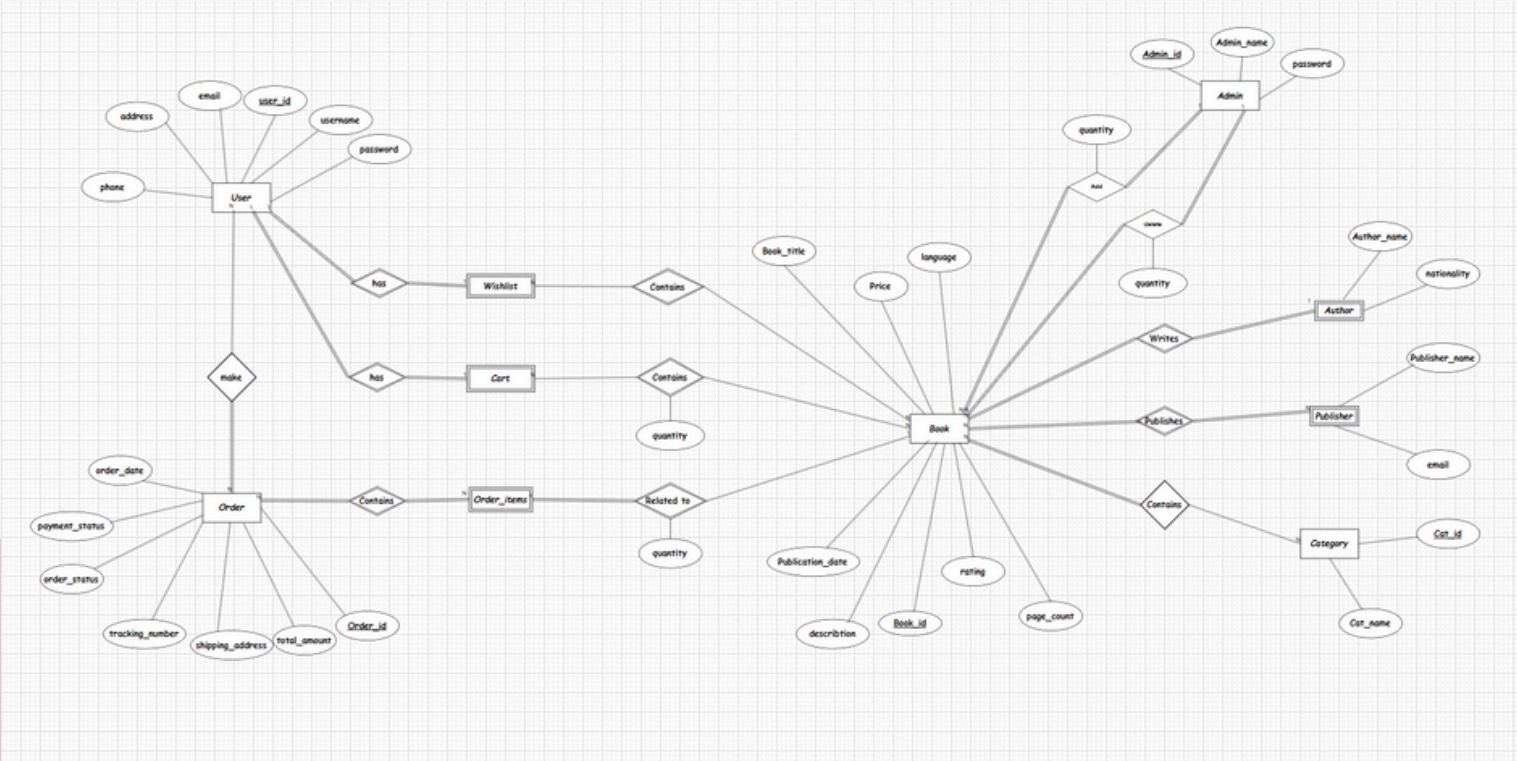
Conceptual Model

ER-Diagram

InkVerse ERD

This ERD represents a relational database for a bookstore system.

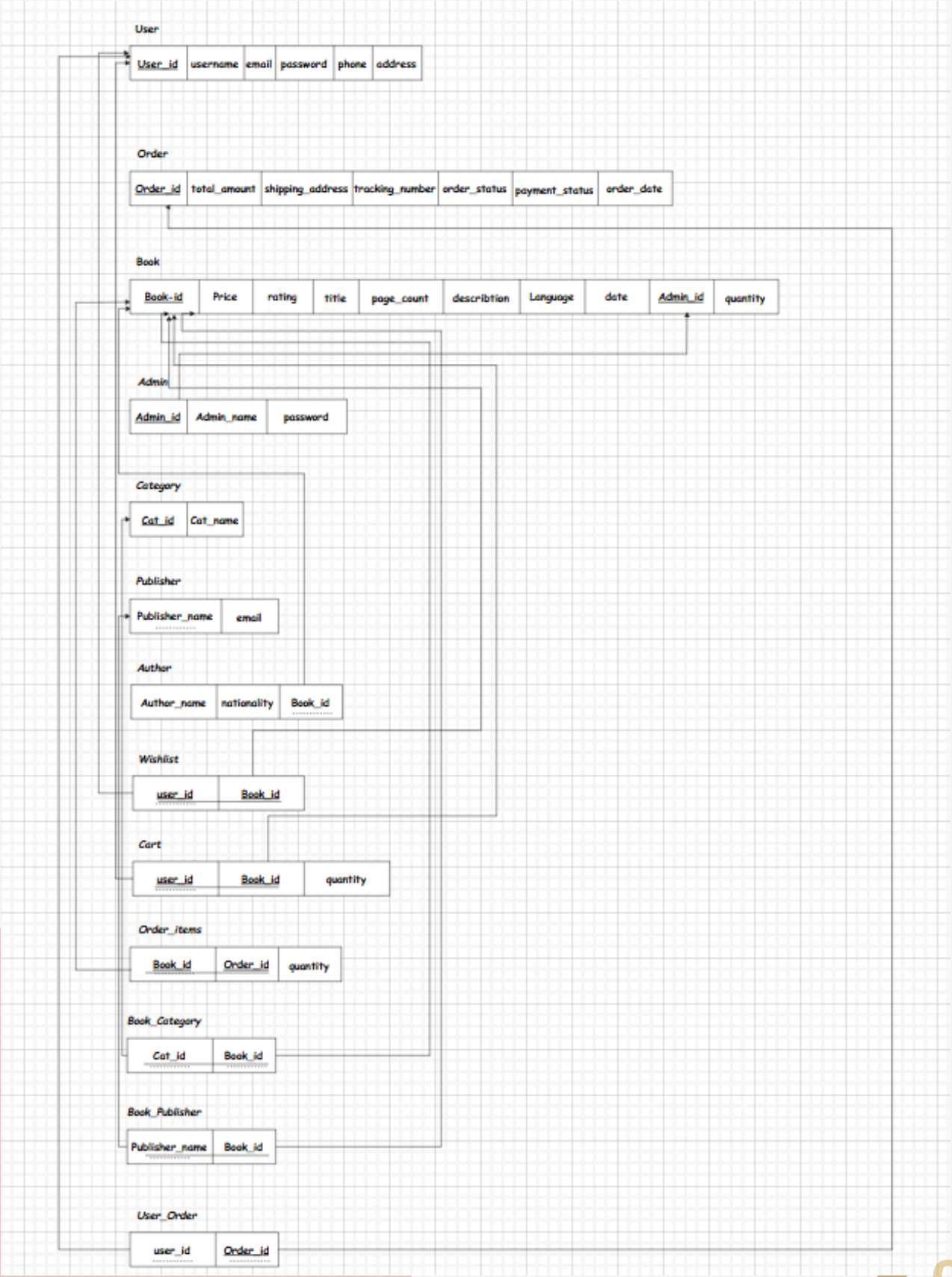
- It manages users, admins, books, orders, carts, wishlists, categories, authors, and publishers.
- The design supports essential operations like placing orders, managing inventory, and organizing books by category and publisher.



Physical Model

Relational Model

InkVerse Schema



DataBase

Implementation

InkVerse
SQL

- The process begins by creating the database itself:

```
DROP DATABASE IF EXISTS BOOk;
CREATE DATABASE BOOK;
USE BOOK;
```

- followed by the structure of each table using CREATE TABLE statements.

Examples:

```
CREATE TABLE User (
User_id INT NOT NULL,
username varchar(60),
email varchar(60),
password varchar(60),
phone varchar(60),
address varchar(60),
);
```

```
CREATE TABLE Book_Order (
Order_id INT NOT NULL,
total_amount INT,
shipping_address varchar(60),
tracking_number INT,
order_status varchar(30),
payment_status varchar(60),
order_date date,
);
```

```
CREATE TABLE Book (
Book_id INT NOT NULL,
Price INT,
rating INT,
page_count INT,
Admin_id INT,
quantity INT,
describtion varchar(500),
Language varchar(60),
date date,
);
```

```
CREATE TABLE Admin (
Admin_id INT NOT NULL,
Admin_name varchar(60),
password varchar(60),
);
```

DataBase

Implementation

InkVerse
SQL

Constraints such as primary keys and foreign keys are added separately using ALTER TABLE to clearly illustrate the relationships between entities.

Examples:

```
-- User Table
ALTER TABLE User
ADD CONSTRAINT PK_User PRIMARY KEY (User_id);

-- Book_Order Table
ALTER TABLE Book_Order
ADD CONSTRAINT PK_BookOrder PRIMARY KEY (Order_id);

-- Book Table
ALTER TABLE Book
ADD CONSTRAINT PK_Book PRIMARY KEY (Book_id),
    CONSTRAINT FK_Book_Admin FOREIGN KEY (Admin_id) REFERENCES Admin(Admin_id);

-- Admin Table
ALTER TABLE Admin
ADD CONSTRAINT PK_Admin PRIMARY KEY (Admin_id);
```

```
-- Category Table
ALTER TABLE Category
ADD CONSTRAINT PK_Category PRIMARY KEY (Cat_id);

-- Publisher Table
ALTER TABLE Publisher
ADD CONSTRAINT PK_Publisher PRIMARY KEY (Publisher_name);

-- Author Table
ALTER TABLE Author
ADD CONSTRAINT FK_Author_Book FOREIGN KEY (Book_id) REFERENCES Book(Book_id);

-- Wishlist Table
ALTER TABLE Wishlist
ADD CONSTRAINT PK_Wishlist PRIMARY KEY (user_id),
    CONSTRAINT FK_Wishlist_User FOREIGN KEY (user_id) REFERENCES [User](User_id);
```

DataBase

Implementation

Sample data is then inserted into each table using INSERT INTO statements to simulate real business operations, including books, users, orders, and more.

Examples:

```
-- Inserting Users
INSERT INTO User (User_id, username, email, password, phone, address) VALUES
(1, 'john_doe', 'john@example.com', 'pass123', 1234567890, '123 Elm St'),
(2, 'jane_smith', 'jane@example.com', 'secret456', 9876543210, '456 Oak St'),
(3, 'alice_wonder', 'alice@example.com', 'wonder123', 5556667777, '789 Pine St'),
(4, 'bob_builder', 'bob@example.com', 'canwefixit', 4443332222, '321 Maple St'),
(5, 'charlie_delta', 'charlie@example.com', 'delta123', 1231231234, '100 Cedar St'),
(6, 'eva_zen', 'eva@example.com', 'zenmode', 3213214321, '200 Birch St'),
(7, 'mike_hawk', 'mike@example.com', 'hawk789', 4567891234, '22 Walnut Rd'),
(8, 'lucy_heart', 'lucy@example.com', 'heart456', 7891234560, '77 Ivy Ln'),
(9, 'tony_stark', 'tony@example.com', 'ironman', 9998887776, '10880 Malibu Point'),
(10, 'bruce_wayne', 'bruce@example.com', 'batman', 8887776665, 'Wayne Manor');
```

```
-- Inserting Admins
INSERT INTO Admin (Admin_id, Admin_name, password) VALUES
(1, 'tech_admin', 'adminpass');
-- Inserting Categories
INSERT INTO Category (Cat_id, Cat_name) VALUES
(1, 'Fiction'),
(2, 'Science'),
(3, 'History'),
(4, 'Technology'),
(5, 'Biography'),
(6, 'Philosophy'),
(7, 'Self-Help'),
(8, 'Poetry'),
(9, 'Drama'),
(10, 'Comics');
-- Inserting Publishers
INSERT INTO Publisher (Publisher_name, email) VALUES
('Penguin Books', 'contact@penguin.com'),
('OReilly Media', 'info@oreilly.com'),
('HarperCollins', 'info@harpercollins.com'),
('MIT Press', 'contact@mitpress.org'),
('Random House', 'random@house.com'),
('Springer', 'info@springer.com'),
('Bloomsbury', 'books@bloomsbury.com'),
('Oxford Press', 'oxford@press.com'),
('Marvel Comics', 'info@marvel.com'),
('DC Comics', 'contact@dc.com');
```

Application

Implementation

The application layer of this project was implemented using SQL queries that interact directly with the underlying bookstore database.
we focused on writing and executing meaningful SQL statements to simulate the application's core functionalities.
These queries demonstrate how the database supports key operations for users, admins, and the bookstore system overall.

Examples:

Get all users:

```
SELECT * FROM User;
```

User_id	username	email	password	phone	address
1	john_doe	john@example.com	pass123	1234567890	123 Elm St
2	jane_smith	jane@example.com	secret456	9876543210	456 Oak St
3	alice_wonder	alice@example.com	wonder123	5556667777	789 Pine St
4	bob_builder	bob@example.com	canwefixit	4443332222	321 Maple St
5	charlie_delta	charlie@example.com	delta123	1231231234	100 Cedar St
6	eva_zen	eva@example.com	zenmode	3213214321	200 Birch St
7	mike_hawk	mike@example.com	hawk789	4567891234	22 Walnut Rd
8	lucy_heart	lucy@example.com	heart456	7891234560	77 Ivy Ln
9	tony_stark	tony@example.com	ironman	9998887776	10880 Malibu Point
10	bruce_wayne	bruce@example.com	batman	8887776665	Wayne Manor

Get all books along with their categories and publishers

```
SELECT b.Book_id, b.describtion, c.Cat_name, p.Publisher_name
FROM Book b, Category c, Publisher p, Book_Category bc, Book_Publisher bp
WHERE b.Book_id = bc.Book_id
 AND bc.Cat_id = c.Cat_id
 AND b.Book_id = bp.Book_id
 AND bp.Publisher_name = p.Publisher_name;
```

Book_id	describtion	Cat_name	Publisher_name
101	Thrilling fiction novel.	Fiction	Penguin Books
111	Improve your daily habits.	Fiction	Penguin Books
121	Ancient history insights.	Fiction	Penguin Books
131	Climate change explained.	Fiction	Penguin Books
141	Quantum computing basics.	Fiction	Penguin Books
102	Advanced science book.	Science	OReilly Media
109	Science for curious minds.	Science	Springer
112	Travel across continents.	Science	OReilly Media
119	Cybersecurity for beginners.	Science	Springer
122	Love and loss story.	Science	OReilly Media
129	Futuristic sci-fi saga.	Science	Springer
132	Biography of a great mind.	Science	OReilly Media
139	Modern web development.	Science	Springer
142	World War history.	Science	OReilly Media
149	Deep space exploration.	Science	Springer
107	Ancient history deep dive.	History	Bloomsbury
117	Guide to meditation.	History	Bloomsbury
127	Space colonization future.	History	Bloomsbury
137	Time travel adventure.	History	Bloomsbury
147	Thrilling fiction novel.	History	Bloomsbury
104	Tech innovations.	Technology	MIT Press
113	Classic literature analysis.	Technology	HarperCollins
123	Fantasy world building.	Technology	HarperCollins
133	Robot uprising tale.	Technology	HarperCollins
143	Classic literature analysis.	Technology	HarperCollins
103	Biography of inventor.	Biography	HarperCollins
116	Poetry collection.	Biography	Springer

Get users who added a specific book to their wishlist (e.g., Book_id = 101):

```
SELECT u.username, u.email
FROM User u JOIN Wishlist w ON u.user_id = w.user_id
WHERE w.book_id = 101;
```

username	email
john_doe	john@example.com

Application

Implementation

Get all orders placed by a specific user (e.g., user with User_id = 1):

```
SELECT bo.Order_id, bo.total_amount,
bo.order_date, bo.order_status
FROM Book_Order bo, User_Order uo
WHERE bo.Order_id = uo.Order_id
 AND uo.user_id = 1;
```

Order_id	total_amount	order_date	order_status
201	150	2024-01-10	Shipped
211	300	2025-03-20	Shipped

Get all books written by a specific author (e.g., 'Ray Kurzweil'):

```
SELECT b.Book_id, b.describtion,
a.Author_name
FROM Book b, Author a
WHERE b.Book_id = a.Book_id
   AND a.Author_name = 'Ray
Kurzweil';
```

Book_id	describtion	Author_name
104	Tech innovations.	Ray Kurzweil
114	Famous battles reimagined.	Ray Kurzweil
122	Love and loss story.	Ray Kurzweil
132	Biography of a great mind.	Ray Kurzweil
142	World War history.	Ray Kurzweil

Get books in a specific category (e.g., 'Science'):

```
SELECT b.Book_id, b.describtion,
c.Cat_name
FROM Book b, Category c,
Book_Category bc
WHERE b.Book_id = bc.Book_id
 AND bc.Cat_id = c.Cat_id
 AND c.Cat_name = 'Science';
```

Book_id	describtion	Cat_name
102	Advanced science book.	Science
109	Science for curious minds.	Science
112	Travel across continents.	Science
119	Cybersecurity for beginners.	Science
122	Love and loss story.	Science
129	Futuristic sci-fi saga.	Science
132	Biography of a great mind.	Science
139	Modern web development.	Science
142	World War history.	Science
149	Deep space exploration.	Science