**Online Book Publishing and Sales Platform:**

create database onlineBookPublishing;

use onlineBookPublishing;

-- AUTHOR

CREATE TABLE Author (

author\_id INT PRIMARY KEY,

name VARCHAR(255),

biography TEXT

);

-- CUSTOMER

CREATE TABLE Customer (

customer\_id INT PRIMARY KEY,

name VARCHAR(255)

);

-- SHIPPING ADDRESS

CREATE TABLE ShippingAddress (

address\_id INT PRIMARY KEY,

customer\_id INT,

address TEXT,

FOREIGN KEY (customer\_id) REFERENCES Customer(customer\_id)

);

-- PUBLISHER

CREATE TABLE Publisher (

publisher\_id INT PRIMARY KEY,

name VARCHAR(255),

contact\_details TEXT

);

-- BOOK

CREATE TABLE Book (

book\_id INT PRIMARY KEY,

title VARCHAR(255),

ISBN VARCHAR(20)

);

-- EDITION

CREATE TABLE Edition (

edition\_id INT PRIMARY KEY,

book\_id INT,

edition\_number INT,

publication\_year YEAR,

price DECIMAL(10,2),

publisher\_id INT,

FOREIGN KEY (book\_id) REFERENCES Book(book\_id),

FOREIGN KEY (publisher\_id) REFERENCES Publisher(publisher\_id)

);

-- GENRE

CREATE TABLE Genre (

genre\_id INT PRIMARY KEY,

genre\_name VARCHAR(100)

);

-- BOOK\_AUTHOR (many-to-many)

CREATE TABLE BookAuthor (

book\_id INT,

author\_id INT,

PRIMARY KEY (book\_id, author\_id),

FOREIGN KEY (book\_id) REFERENCES Book(book\_id),

FOREIGN KEY (author\_id) REFERENCES Author(author\_id)

);

-- BOOK\_GENRE (many-to-many)

CREATE TABLE BookGenre (

book\_id INT,

genre\_id INT,

PRIMARY KEY (book\_id, genre\_id),

FOREIGN KEY (book\_id) REFERENCES Book(book\_id),

FOREIGN KEY (genre\_id) REFERENCES Genre(genre\_id)

);

-- ORDER

CREATE TABLE Order (

order\_id INT PRIMARY KEY,

customer\_id INT,

order\_date DATE,

shipment\_status VARCHAR(50),

FOREIGN KEY (customer\_id) REFERENCES Customer(customer\_id)

);

-- ORDER\_ITEM (edition in orders)

CREATE TABLE OrderItem (

order\_id INT,

edition\_id INT,

quantity INT,

discount DECIMAL(5,2),

PRIMARY KEY (order\_id, edition\_id),

FOREIGN KEY (order\_id) REFERENCES Order(order\_id),

FOREIGN KEY (edition\_id) REFERENCES Edition(edition\_id)

);

-- PAYMENT

CREATE TABLE Payment (

payment\_id INT PRIMARY KEY,

order\_id INT,

amount DECIMAL(10,2),

payment\_method VARCHAR(50),

payment\_date DATE,

FOREIGN KEY (order\_id) REFERENCES Order(order\_id)

);

-- WISHLIST (customer - edition)

CREATE TABLE Wishlist (

customer\_id INT,

edition\_id INT,

PRIMARY KEY (customer\_id, edition\_id),

FOREIGN KEY (customer\_id) REFERENCES Customer(customer\_id),

FOREIGN KEY (edition\_id) REFERENCES Edition(edition\_id)

);

**Data:**

-- Insert into Author

INSERT INTO Author (author\_id, name, biography)

VALUES (1, 'J.K. Rowling', 'Author of the Harry Potter series');

-- Insert into Customer

INSERT INTO Customer (customer\_id, name)

VALUES (1, 'John Doe');

-- Insert into ShippingAddress

INSERT INTO ShippingAddress (address\_id, customer\_id, address)

VALUES (1, 1, '123 Fictional Street, London, UK');

-- Insert into Publisher

INSERT INTO Publisher (publisher\_id, name, contact\_details)

VALUES (1, 'Penguin Books', 'Contact: +44 123 456 7890');

-- Insert into Book

INSERT INTO Book (book\_id, title, ISBN)

VALUES (1, 'Harry Potter and the Sorcerer\'s Stone', '9780439554930');

-- Insert into Edition

INSERT INTO Edition (edition\_id, book\_id, edition\_number, publication\_year, price, publisher\_id)

VALUES (1, 1, 1, 1997, 19.99, 1);

-- Insert into Genre

INSERT INTO Genre (genre\_id, genre\_name)

VALUES (1, 'Fantasy');

-- Insert into BookAuthor

INSERT INTO BookAuthor (book\_id, author\_id)

VALUES (1, 1);

-- Insert into BookGenre

INSERT INTO BookGenre (book\_id, genre\_id)

VALUES (1, 1);

-- Insert into Order

INSERT INTO `Order` (order\_id, customer\_id, order\_date, shipment\_status)

VALUES (1, 1, '2025-06-01', 'Shipped');

-- Insert into OrderItem

INSERT INTO OrderItem (order\_id, edition\_id, quantity, discount)

VALUES (1, 1, 2, 10.00);

-- Insert into Payment

INSERT INTO Payment (payment\_id, order\_id, amount, payment\_method, payment\_date)

VALUES (1, 1, 35.98, 'Credit Card', '2025-06-01');

-- Insert into Wishlist

INSERT INTO Wishlist (customer\_id, edition\_id)

VALUES (1, 1);

**EER Diagram:**

