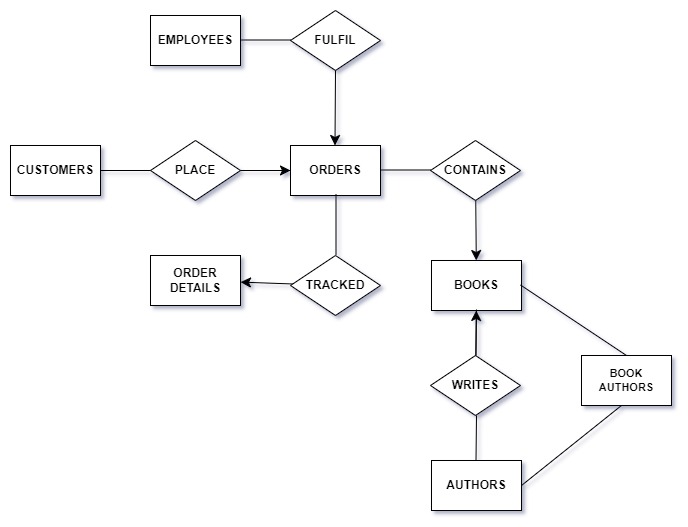
**XYLEMBD Online bookstore database management model design**

**Conceptual Data Model:**

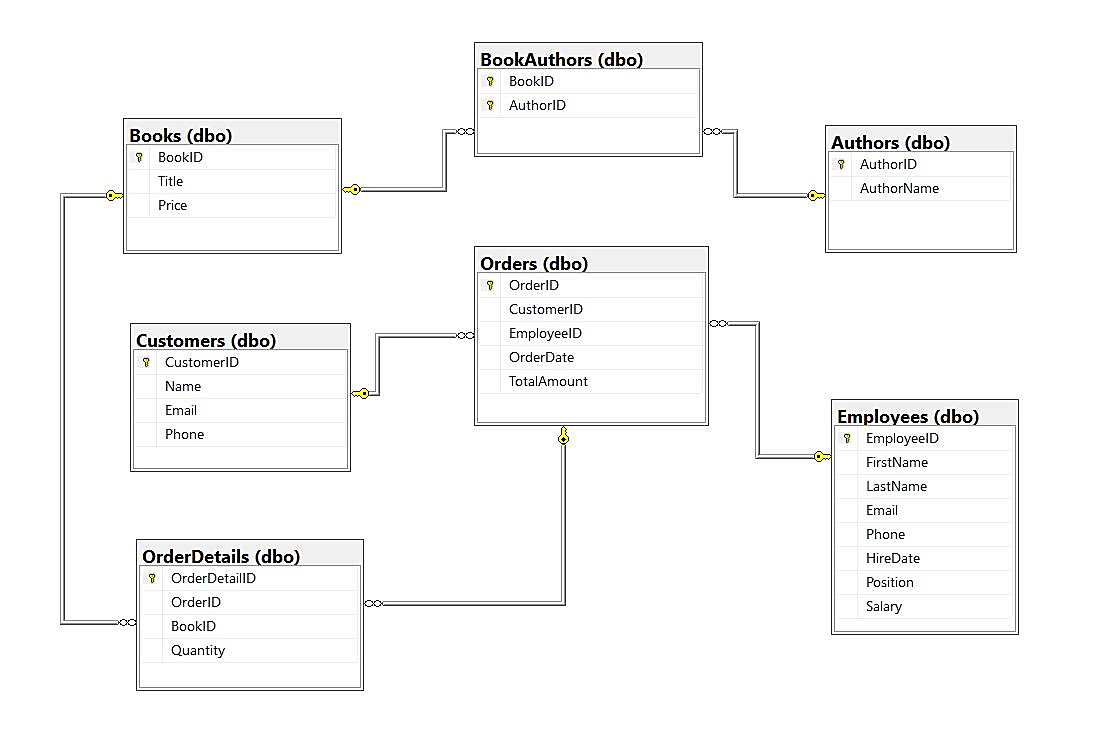
|  |  |  |
| --- | --- | --- |
| S/N | ENTITY | ATTRIBUTES |
| 1 | Customer | CustomerID, Name, Phone, Email |
| 2 | Orders | OrderID, CustomerID, OrderDate, TotalAmount, EmployeeID |
| 3 | OrderDetails | OrderDetailID(PK), OrderID(FK), BookID(FK), Quantity |
| 4 | Books | BookID(PK), Title, Price |
| 5 | Authors | AuthorID(PK), Name |
| 6 | BookAuthors | BookID(FK), AuthorID(FK) |
| 7 | Employees | EmployeeID, FirstName, LastName, Position, HireDate, Salary, Phone, Email |





**Logical Data Model:**

|  |  |  |
| --- | --- | --- |
| S/N | ENTITY | ATTRIBUTES |
| 1 | Customer | CustomerID INT (PK) Identity 1,1) NOT NULL,  Name NVARCHAR (100) NOT NULL,  Phone INT NOT NULL,  Email NVARCAHR (100) NOT NULL |
| 2 | Orders | OrderID INT (PK) Identity (1, 1) NOT NULL,  CustomerID INT FOREIGN KEY REFERENCES Customers(CustomerID),  EmployeeID INT FOREIGN KEY REFERENCES Employees(EmployeeID)  OrderDate DATE NOT NULL,  TotalAmount DECIMAL (10, 2) |
| 3 | Books | BookID INT (PK) Identity (1, 1) NOT NULL,  Title NVARCAHR (200) NOT NULL,  Price DECIMAL (10, 2) |
| 4 | Authors | MenuID INT (PK),  Name NVARCHAR (100),  Description NVARCAHR (100), |
| 5 | BookAuthors | BookID FOREIGH KEY REFERENCES Books (BookID)  AuthorID FOREIGH KEY REFERENCES Authors (AuthorID) |
| 6 | OrderDetails | OrderDetailID INT (PK) Identity (1, 1) NOT NULL,  OrderID INT FOREIGN KEY REFERENCES Orders (OrderID),  BookID FOREIGH KEY REFERENCES Books (BookID),  DeliveryDate DATE,  Status BOOLEAN |
| 7 | Employees | EmployeeID INT (PK) Identity (1, 1) NOT NULL,  First Name NVARCAHR (100),  Last Name NVARCAHR (100),  Phone NVARCAHR (100) NOT NULL,  Email NVARCHAR (100) NOT NULL  Position NVARCHAR (100),  Salary DECIMAL (10, 2),  HireDate DATE |



**Figure 2:ER Diagram**

**Entity-Relationship Diagram (ERD) Explanation:**

In the Xylem database, the relationships between tables are established through primary and foreign key constraints. Below is a brief explanation of the relationships;

1. Customers and Orders: This is a one-to-many relationship, where one customer can place multiple orders. The Orders table has a foreign key column referencing the CustomerID column in the Customers table.

2. Orders and OrderDetails: This is also a one-to-many relationship, where one order can have multiple order details. The OrderDetails table has a foreign key column referencing the OrderID column in the Orders table.

3. Books and OrderDetails: This is a many-to-many relationship, where multiple books can appear in multiple orders, and each order can contain multiple books. This OrderDetails table contains foreign key columns referencing both the BookID column in the Books table and the OrderID column in the OrderDetails table.

4. Authors and Books: This is also a many-to-many relationship, where multiple authors can write multiple books, and each book can have multiple authors. The BookAuthor table (weak entity) contains foreign key columns referencing both the AuthorID column in the Authors table and the BookID column in the Books table.

5. Employees and Orders: This is a one-to-many relationship, where one employee can handle multiple orders. The Orders table has a foreign key column referencing the EmployeeID column in the Employees table.

These relationships ensure data integrity and enable efficient querying and retrieval of information across related tables in the database.

CREATE TRIGGER trg\_insert\_bookauthor\_after\_book\_insert

ON Books

AFTER INSERT

AS

BEGIN

INSERT INTO BookAuthors (BookID, AuthorID)

SELECT BookID, 'default\_author\_id' FROM inserted;

END;

CREATE TRIGGER trg\_insert\_bookauthor\_after\_author\_insert

ON Authors

AFTER INSERT

AS

BEGIN

INSERT INTO BookAuthors (BookID, AuthorID)

SELECT 'default\_book\_id', AuthorID FROM inserted;

END;

|  |  |  |
| --- | --- | --- |
|  | Trigger on Books table | Trigger on Authors table |
| Trigger Name | trg\_insert\_bookauthor\_after\_book\_insert | trg\_insert\_bookauthor\_after\_author\_insert |
| Associated Table | Books | Authors |
| Trigger Type | AFTER INSERT | AFTER INSERT |
| Action | Inserts default author ID into BookAuthors table for each newly inserted book. | Inserts default book ID into BookAuthors table for each newly inserted author |
| These triggers automate the process of populating the `BookAuthors` table whenever new records are inserted into either the `Books` table or the `Authors` table. Each trigger ensures that the `BookID` and `AuthorID` columns in the `BookAuthors` table are appropriately populated with the corresponding values from the source table, while also providing default values where necessary. | | |