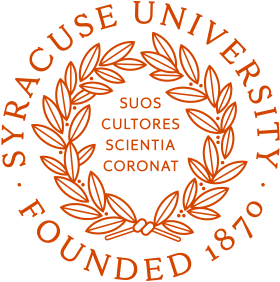
****

****

**BOOKSTORE DATABASE MANAGEMENT SYSTEM**

**PROJECT IMPLEMENTATION REPORT**

**Srinivas Reddy Pachika**

**IST-659: Data Admin Concepts & Db Mgmt.**

**Project summary**

The objective of the project is to build an automated database in the place of current manual entry system to better keep track of its customers, suppliers, order and stock. As bookstore has huge unorganized data and new data adding up every single day it is becoming hard for bookstore management to manage its customers data and keep a track of its supplies, order and inventory.

At present, the book store management system uses MS Excel, all records related to Sales, stock, Orders, Payment and Suppliers are stored in excel files. There is lot of duplicate and recurring data. When the records are changed they need to update each and every excel file. In case of Customer records, all information related to customers and the product which the customer has purchased is to be stored in the Customers excel files. Any changes or updates related to customer like his phone number, address, email etc. are manually updated in the excel.

To manage the whole data, the person maintaining records has to take great pain. Various excel files has to be maintained for each separate process. There is no option to find previous saved records. There is no security; anybody can access any report and sensitive data.

This system is designed to overcome all the above discussed problems of bookstore. This system is designed to record the customer’s purchase history, supplier’s incoming stock, Order history, current stock of books in the bookstore and all other transaction details between customer- bookstore and bookstore-supplier. Thus, this could be an inventory system which helps to keep a track of what you have in your store, informs you when you need to purchase more stock or supplies by category and maintains all customers data which can be used for analyzing the buying trends of customers. However, this project design is limited to maintain customer data base only and not used for analyzing customer buying trends.

This data base management system is designed to overcome data redundancy issues, automate data management and reduce manual effort. This system is designed to easily store, manage, update and retrieve data when needed. This report comprises of proposed bookstore

### Entities and their Attributes:

|  |  |
| --- | --- |
| **OBJECTS** | **DESCRIPTIONRIPTION** |
| * **Book** | **Information of a book in bookstore** |
| Name | Primary Key/ Uniquely identifies the book. |
| Author Id | Foreign Key/ Unique number allocated to every author |
| Publisher Id | Foreign Key/ Unique number allocated to every publisher |
| Shelf Number | Foreign Key/ Unique number of a shelf on which a particular book is placed |
| ISBN | Unique number allocated to every book |
| * **Author** | Information of the Author |
| Author Id | Unique Id number assigned to every author |
| Full Name | Full Name of the Author |
| Country | Country to which author belongs to |
| * **Publisher** | **Information of book publisher** |
| Publisher Id | Primary key/ Unique number given to every publisher. |
| Name | Primary key/ unique name of book publisher |
| Full address | Address of the book publisher |

|  |  |
| --- | --- |
| * **Shelf** | **Details of the shelf on which a book is placed in the bookstore** |
| Shelf Number | Primary Key/Number of a shelf in the bookstore on which a particular book is placed |
| Book Name | Foreign Key/Name of the book which is placed on that particular shelf |
| Quantity Available | Remaining books on the shelf |
| * BSProduct | Details of a product in the book store |
| Product Id | Primary key/ Unique Id assigned for every product in the book store |
| Product name | Name of the product |
| Shelf Number | Foreign Key**/** Number of a shelf in the bookstore on which a particular book is placed |
| Product Price | Price of the product |
| * BSOrderLine | Associate entity/ gives the details of product Id and order Id |
| Order Id | Foreign key/ Unique Id assigned for every Order |
| Product Id | Foreign key/ Unique Id assigned for every product in the book store |

|  |  |
| --- | --- |
| * **CusOrder** | **Details of an Order** |
| Order Id | Primary key/ Unique Id assigned for every Order |
| Customer Id | Foreign key/Unique identification number of a customer |
| Order Date | Date on which an order is made |
| Quantity | Quantitiy of a particular item in an order |
| Total Billing Amount | Total billing amount of the order |
| * **Customer** | **Information of the customer** |
| Customer Id | Primary key/Unique identification number of a customer |
| Name | Name of the customer |
| Address | Address of the customer |
| DOB | Date of birth of the customer |
| State | State of residency of the customer |
| City | City of the customer |

Entity Relationship Diagram



**Business Rules:**

* Every customer must have at least one order associated with the bookstore.
* Every order must have at least one item.
* Every item must be part of at least one order.
* Every item is stored in one and only one shelf and every shelf contains only one particular item.
* Shelf can have zero or more items.
* Every boeok is printed by one and only one publisher and every publisher must publish at least one book.
* Ever book is authored by one and only one author and every author must at least write one book.

SQL SCRIPTS FOR CREATING AND INSERTING SAMPLE DATA:

**Creating Tables:**

**CREATE TABLE BSCustomer**

**(**

**CustomerId VARCHAR(6) PRIMARY KEY NOT NULL,**

**Name CHAR(10) NOT NULL,**

**DOB DATE NOT NULL,**

**Address CHAR(15) NOT NULL,**

**city CHAR(5) NOT NULL,**

**State CHAR(5) NOT NULL,**

**);**

**Create Table Publisher**

**(**

**PublisherId VARCHAR(6) PRIMARY KEY NOT NULL,**

**Name VARCHAR(15) NOT NULL,**

**FullAddress VARCHAR(20) NOT NULL**

**);**

**Create Table Author**

**(**

**AuthorId VARCHAR(6) PRIMARY KEY NOT NULL,**

**FullName VARCHAR(15) NOT NULL,**

**Country VARCHAR(20) NOT NULL**

**);**

**CREATE TABLE BOOK**

**(**

**BookName VARCHAR(30) PRIMARY KEY NOT NULL,**

**ISBN VARCHAR(13) NOT NULL,**

**AuthorId VARCHAR(6) NOT NULL,**

**PublisherId VARCHAR(6) NOT NULL,**

**CONSTRAINT BOOK\_FK\_AUTHID FOREIGN KEY(AuthorId) REFERENCES Author(AuthorId),**

**CONSTRAINT BOOK\_FK\_PUBID FOREIGN KEY(PublisherId) REFERENCES Publisher(PublisherId)**

**);**

**DROP TABLE BOOK**

**CREATE TABLE Shelf**

**(**

**ShelfNumber VARCHAR(6) PRIMARY KEY NOT NULL,**

**BookName VARCHAR(30) NOT NULL,**

**QuantityAvailable INTEGER NOT NULL,**

**CONSTRAINT Shelf\_FK FOREIGN KEY(BookName) REFERENCES BOOK(BookName)**

**);**

**CREATE TABLE BSProduct**

**(**

**ProductId VARCHAR(6) PRIMARY KEY NOT NULL,**

**ProductName VARCHAR(30) NOT NULL UNIQUE,**

**ShelfNumber VARCHAR(6) NOT NULL,**

**ProductPrice Decimal(4,2) NOT NULL,**

**constraint BSProduct\_FK FOREIGN KEY(ShelfNumber) REFERENCES Shelf(ShelfNumber)**

**);**

**CREATE TABLE CUSORDER**

**(**

**OrderId VARCHAR(6) PRIMARY KEY NOT NULL,**

**OrderDate DATETIME DEFAULT getdate(),**

**CustomerId VARCHAR(6) NOT NULL,**

**Qunatity INT NOT NULL,**

**TotalBillingAmount DECIMAL(6,2) NOT NULL,**

**constraint order\_FK FOREIGN KEY(CustomerId) REFERENCES BSCustomer(CustomerId)**

**);**

**CREATE TABLE BSOrderLine**

**(**

**ProductId VARCHAR(6) NOT NULL,**

**OrderId VARCHAR(6) NOT NULL,**

**CONSTRAINT BSOrderLine\_PK PRIMARY KEY(ProductId,OrderId),**

**CONSTRAINT BSOrderLine\_FK\_ProductId FOREIGN KEY(ProductId) REFERENCES BSProduct(ProductId),**

**CONSTRAINT BSOrderLine\_FK\_OrderId FOREIGN KEY(OrderId) REFERENCES CUSORDER(OrderId)**

**);**

**Inserting Data:**

**INSERT INTO BSCustomer VALUES( '123546', 'srinivas', '5/6/1994', 'Lexigton','syra','Ny');**

**INSERT INTO BSCustomer VALUES( '123547', 'Ben', '7/6/1984', 'Westcott','syra','Ny');**

**INSERT INTO BSCustomer VALUES( '123548', 'ron', '8/6/1984', 'cherry','syra','Ny');**

**INSERT INTO BSCustomer VALUES( '123549', 'tim', '9/6/1984', 'southbeach','syra','Ny');**

**INSERT INTO BSCustomer VALUES( '123550', 'kim', '1/6/1984', 'dell','syra','Ny');**

**INSERT INTO BSCustomer VALUES( '123551', 'chan', '6/6/1984', 'eastgen','syra','Ny');**

**INSERT INTO BSCustomer VALUES( '123552', 'wang', '5/6/1984', 'columbus','syra','Ny');**

**INSERT INTO Publisher VALUES( '657675', 'mchill', 'texas');**

**INSERT INTO Publisher VALUES( '657676', 'pearson', 'hyderabad');**

**INSERT INTO Publisher VALUES( '657677', 'arihant', 'syracuse');**

**INSERT INTO Publisher VALUES( '657678', 'tata', 'warangal');**

**INSERT INTO Publisher VALUES( '657679', 'cingage', 'hanamkonda');**

**select \*from Publisher**

**INSERT INTO Author VALUES( '43523', 'reddy', 'India');**

**INSERT INTO Author VALUES( '93524', 'jim', 'UK');**

**INSERT INTO Author VALUES( '53943', 'RAM', 'India');**

**INSERT INTO Author VALUES( '12223', 'SHIVA', 'India');**

**INSERT INTO Author VALUES( '93523', 'DIN', 'USA');**

**INSERT INTO CUSORDER VALUES ( '275432', '11/27/2018', '123546','5', '50');**

**INSERT INTO CUSORDER VALUES ( '675632', '10/27/2018', '123547','1', '20');**

**INSERT INTO CUSORDER VALUES ( '976132', '11/27/2018', '123548','9', '100');**

**INSERT INTO BOOK VALUES ( 'intro to ds','123456789', '93523','657679');**

**INSERT INTO BOOK VALUES ( 'modern DBMS', '122237658','43523','657678');**

**INSERT INTO BOOK VALUES ( 'data analytics', '53943987651','12223','657676');**

**select\* from BSProduct**

**INSERT INTO shelf VALUES( '453', 'intro to ds', '10');**

**INSERT INTO shelf VALUES( '67', 'modern DBMS', '12');**

**INSERT INTO shelf VALUES( '912', 'data analytics', '18');**

**INSERT INTO BSProduct VALUES ( '11111', 'intro to ds', '453', '50');**

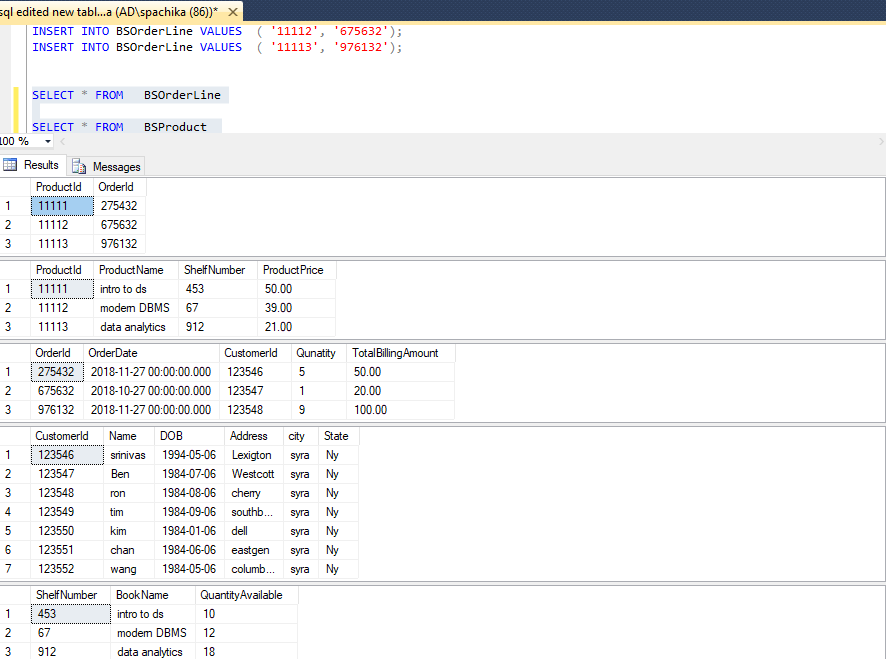
**INSERT INTO BSProduct VALUES ( '11112', 'modern DBMS', '67', '39');**

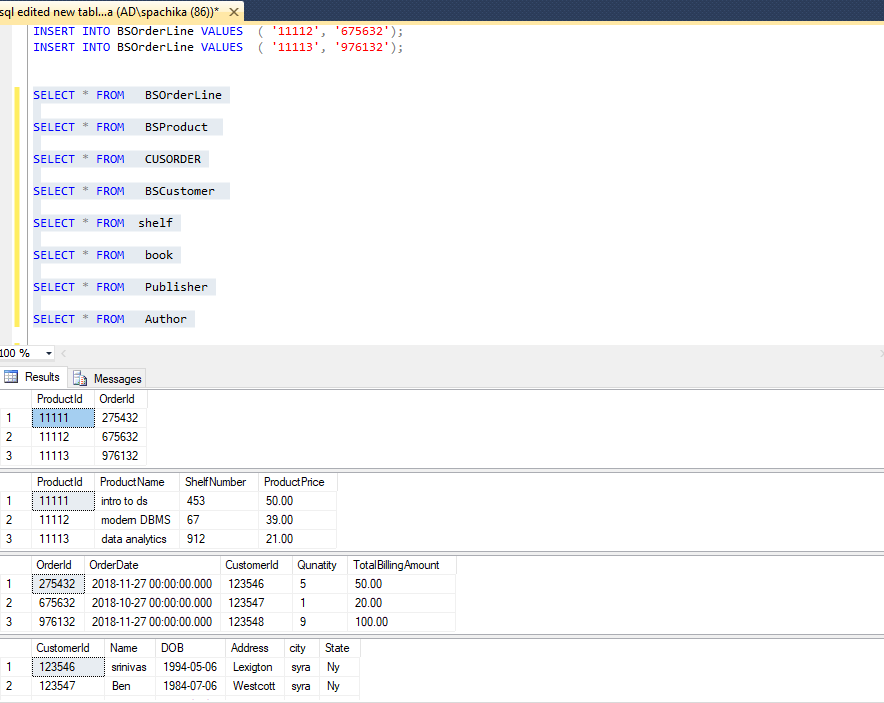
**INSERT INTO BSProduct VALUES ( '11113', 'data analytics', '912', '21');**

**INSERT INTO BSOrderLine VALUES ( '11111', '275432');**

**INSERT INTO BSOrderLine VALUES ( '11112', '675632');**

**INSERT INTO BSOrderLine VALUES ( '11113', '976132');**

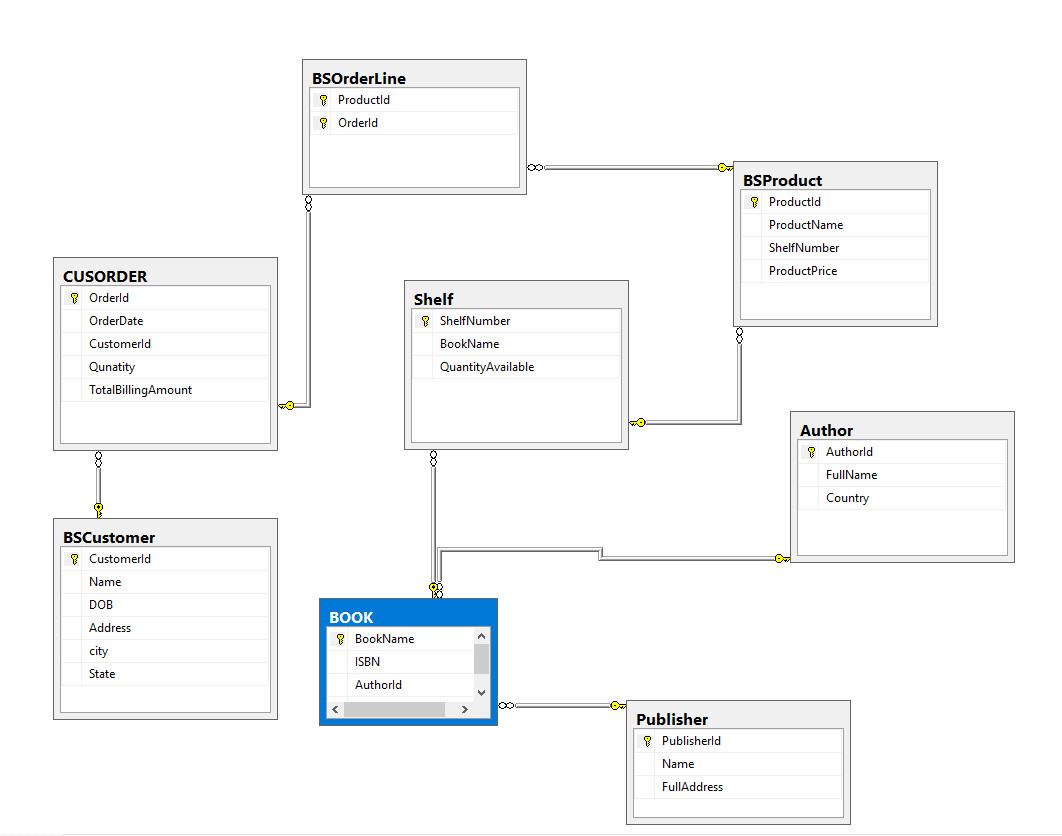




* **Major Data Questions**

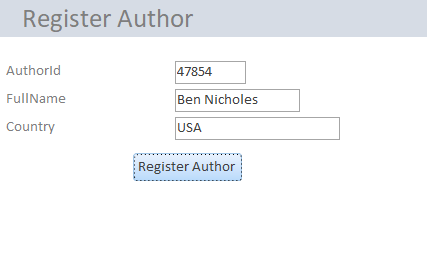
1. **Generating Daily sales Report**
2. **Customer purchase history**
3. **Checking the book availability**
4. **View Full book details**
5. **Checking the shelf number of a book**
6. **Which item/book is most sold during a particular time frame.**

**SQL ER-DIAGRAM**

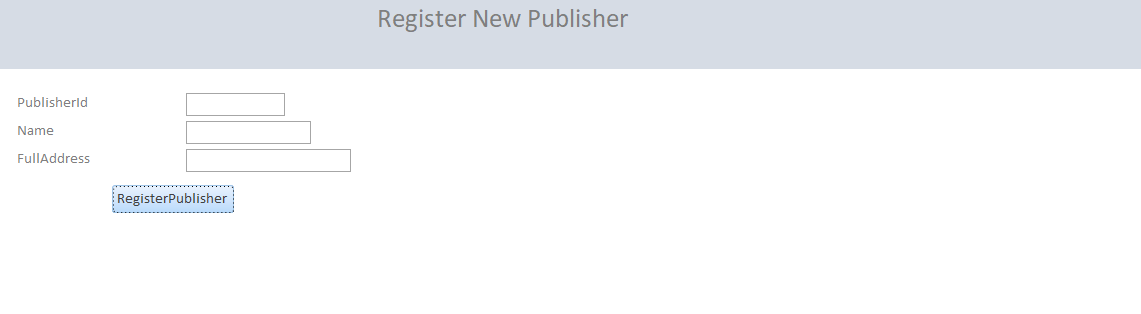
****

**FORMS**

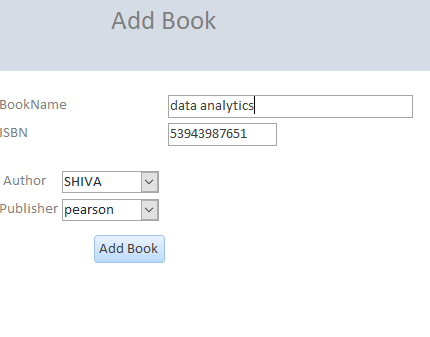
1. **Author form : Helps the store owner to register a Author**

****

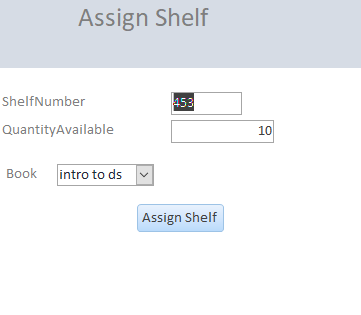
1. **Publisher Form : Helps the store owner to register a publisher**

****

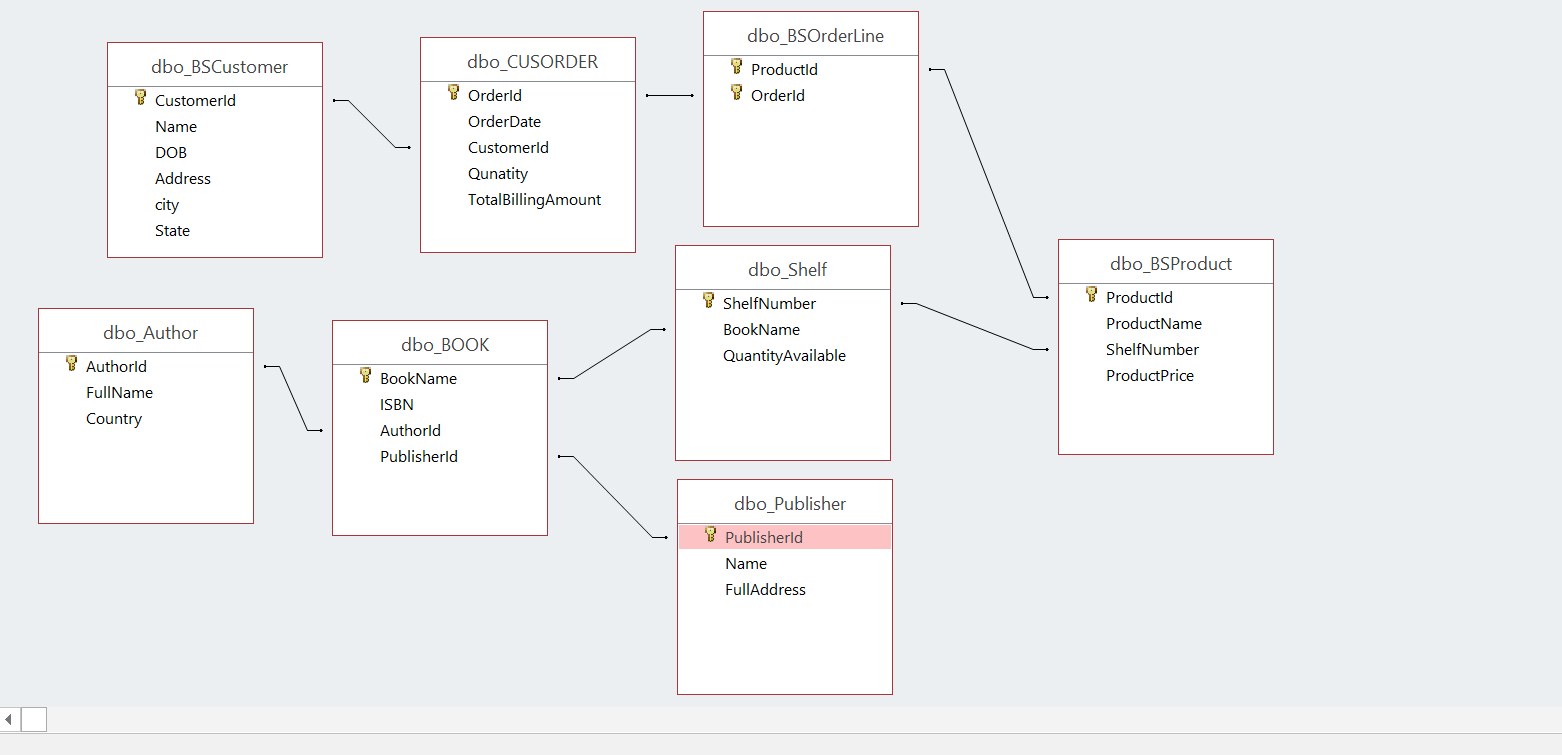
1. **Add book form : Helps the store owner to add a new book**

****

1. **Shelf form : helps the book store owner to assign a particular shelf number to every item in the book store**

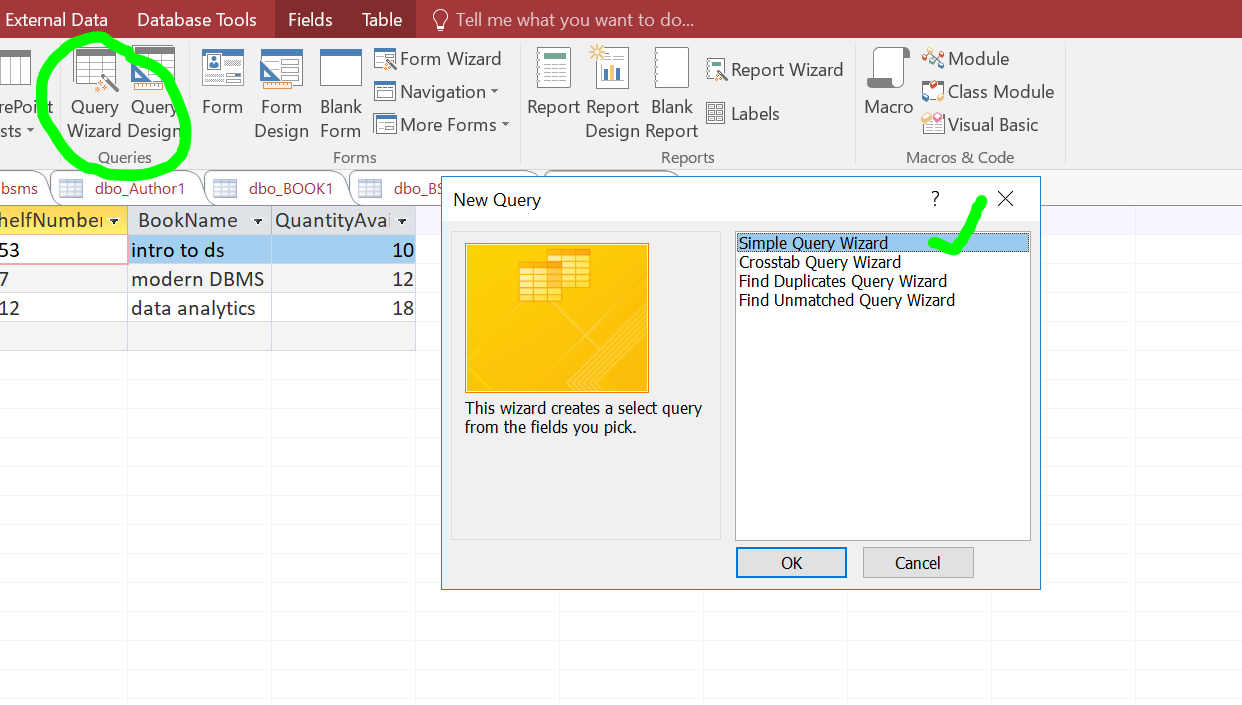
****

**RELATIONSHIP DIAGRAM:**



**Reports:**

I have used Access query wizard option to create queries and then used these queries to create reports.



1. Sales summary report

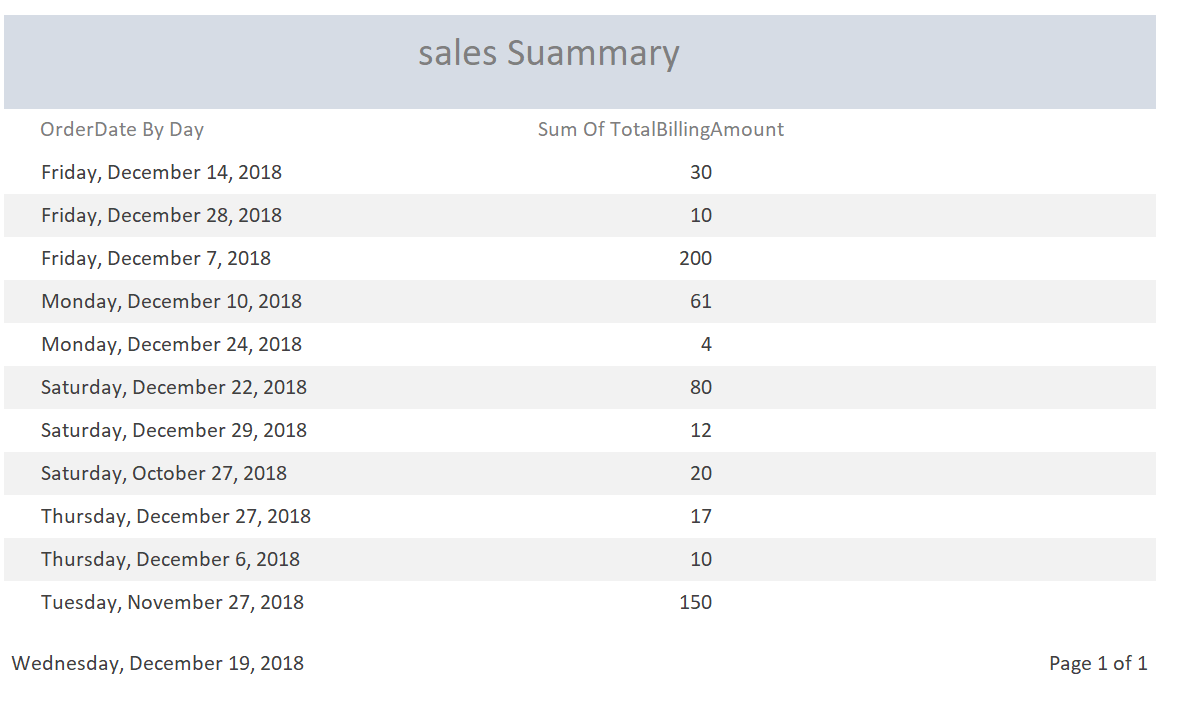
SQL query

**Data question : 1) Generating Daily sales Report**

**SELECT DISTINCTROW Format$([dbo\_CUSORDER].[OrderDate],'Long Date') AS [OrderDate By Day], Sum(dbo\_CUSORDER.TotalBillingAmount) AS [Sum Of TotalBillingAmount]**

**FROM dbo\_CUSORDER**

**GROUP BY Format$([dbo\_CUSORDER].[OrderDate],'Long Date');**



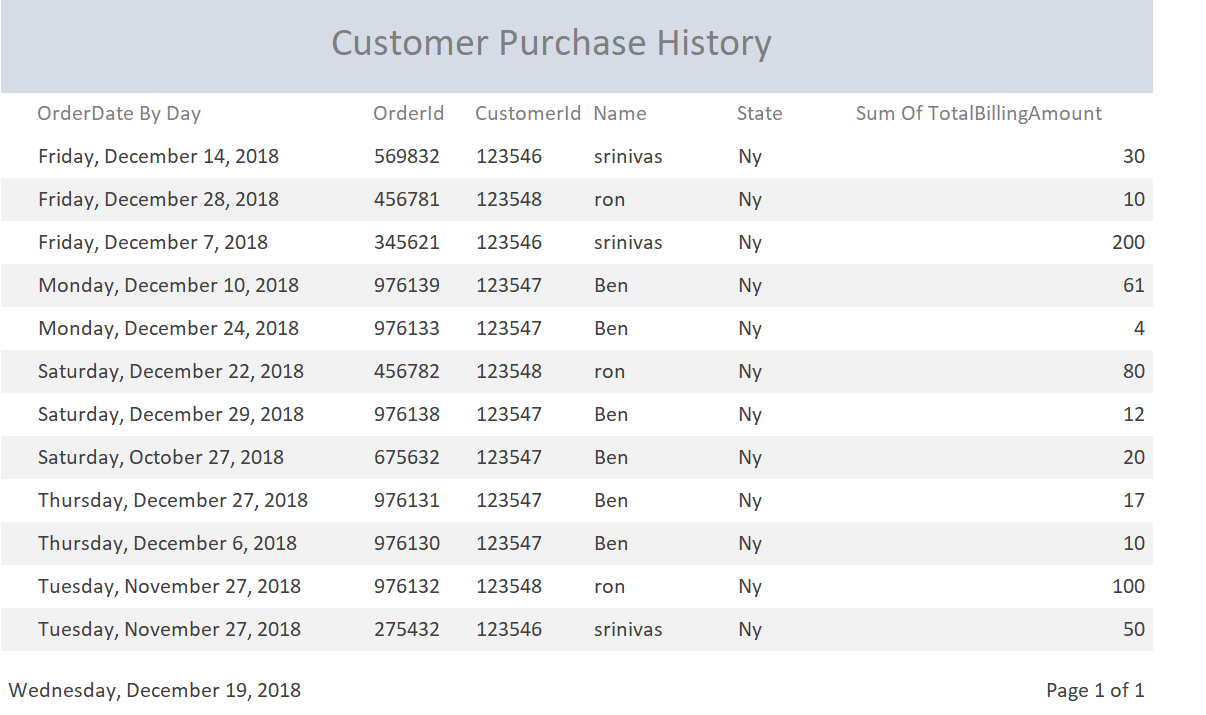
1. **Customer Purchase History Report**

**Sql Query**

**SELECT DISTINCTROW dbo\_CUSORDER.OrderId, Format$([dbo\_CUSORDER].[OrderDate],'Long Date') AS [OrderDate By Day], dbo\_CUSORDER.CustomerId, dbo\_BSCustomer.Name, dbo\_BSCustomer.State, Sum(dbo\_CUSORDER.TotalBillingAmount) AS [Sum Of TotalBillingAmount]**

**FROM dbo\_BSCustomer INNER JOIN dbo\_CUSORDER ON dbo\_BSCustomer.[CustomerId] = dbo\_CUSORDER.[CustomerId]**

**GROUP BY dbo\_CUSORDER.OrderId, Format$([dbo\_CUSORDER].[OrderDate],'Long Date'), dbo\_CUSORDER.CustomerId, dbo\_BSCustomer.Name, dbo\_BSCustomer.State;**

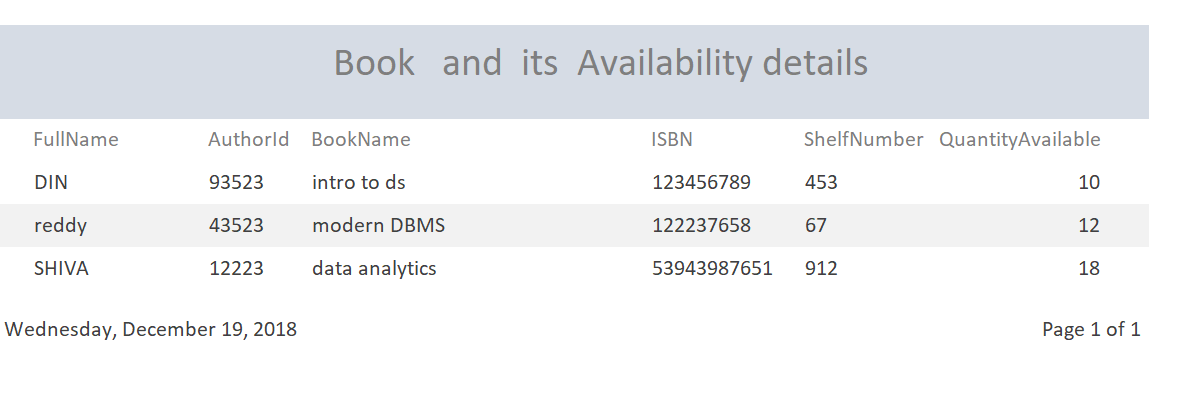


1. **Book and its availability**

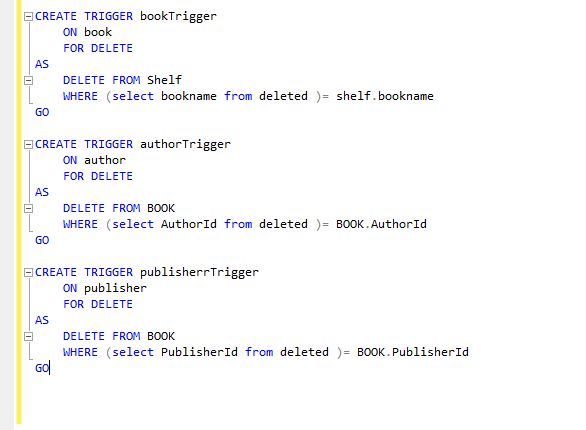
**Sql query**

**SELECT dbo\_Author.AuthorId, dbo\_Author.FullName, dbo\_BOOK.BookName, dbo\_BOOK.ISBN, dbo\_Shelf.ShelfNumber, dbo\_Shelf.QuantityAvailable**

**FROM (dbo\_Author INNER JOIN dbo\_BOOK ON dbo\_Author.[AuthorId] = dbo\_BOOK.[AuthorId]) INNER JOIN dbo\_Shelf ON dbo\_BOOK.[BookName] = dbo\_Shelf.[BookName];**



**Triggers**

****