**INVENTORY MANAGEMENT RELATIONAL**

**DATABASE DESIGN PROPOSAL**

BOOKS’R’US BOOKSTORE

**JULY 25, 2015**

**PROJECT NAME:** Books’R’Us Relational Database Design Project

**PRINCIPAL DESIGNER:** Shaun Borey

**PROJECT START DATE:** July 27, 2015

**PROJECT END DATE:** August 30, 2015

**PROJECT OVERVIEW**

Books’R’Us is a small local bookstore with two locations. The company currently manages inventory data using a simple system that utilizes data files. In order to facilitate the company’s continued growth and future plans, an updated data management system is required. This project is intended to deliver a relational database design implemented within Microsoft SQL Server. The completed design allows for efficient inventory control and accurate information reporting.

**PROJECT JUSTIFICATION**

The design of a new data management system is a large investment. Detailed information outlining the necessity of this investment is provided in this section.

**PROJECT BENEFITS**

The use of a relational database management system allows for greater data control and more efficient data access than is possible with file-based systems. The new system has the capability to fully support the company’s future plans to offer online sales direct through its website. The new system also allows efficient access to product information which enables the company to take greater advantage of publisher discounts. The ability to store and analyze customer information enhances marketing efforts.

**ISSUE RESOLUTION**

The new database management system resolves the following issues currently experienced by the company:

* ***High Risk of Data Inconsistency:*** Because the current system implements decentralized files to store data, there is a high risk of data inconsistencies developing. Reports based on inconsistent data are inaccurate, which can adversely affect business operations, operating costs, and profits. The use of a centralized relational database management system greatly reduces the risk of inconsistent data by removing unnecessary data redundancy.
* ***Inefficient Data Access:*** The current system does not allow the two company locations to easily access data from the other. The new system is centralized and stores data from both stores in one location, providing each store immediate access to all data.
* ***Inefficient Reporting:*** The current system does not enable efficient report generation. Having fast and accurate product information enables the company to take full advantage of bulk order discounts offered by publishers. Accurate customer information aids in developing more effective marketing plans. The new system enables the generation of fast, accurate information reports.

**CONSEQUENCES OF NON-IMPLEMENTATION**

If this project is not implemented, there are several consequences that result. The inability to access timely product information prevents the company from taking full advantage of publisher discount offers. Maintaining optimal inventory levels are extremely difficult using the current system because inventory information must be retrieved from each store location separately. Marketing efforts will not be as successful without the ability to store and retrieve customer data efficiently. Finally, it will be impossible to implement an e-commerce solution without a relational database.

**DELIVERABLE REQUIREMENTS**

The database design must implement the following functionality:

* Store book sales information that can be retrieved by publisher, author, genre, or customer. This requirement enables the following:
  + Ability to determine the most popular publishers
  + Ability to determine the most popular authors
  + Ability to determine the most popular genres
  + Ability to develop marketing based on sales data of various customer demographics
* Store detailed information for all products including books, magazines, café products, and gift products. This requirement enables the following:
  + Ability to obtain timely and accurate inventory levels
  + Ability to determine the most popular products
  + Ability to easily display product information in the company’s online store
* Store detailed customer information including name, address, and email address. This requirement enables the following:
  + Ability to process and ship online customer orders
  + Ability to develop marketing based on customer demographics
* Store information required for website e-commerce including payment details and shipping information. This requirement enables the following:
  + Ability to process payments for online orders
  + Ability to maintain shipping information for online orders

**PROPOSED DATABASE DESIGN**

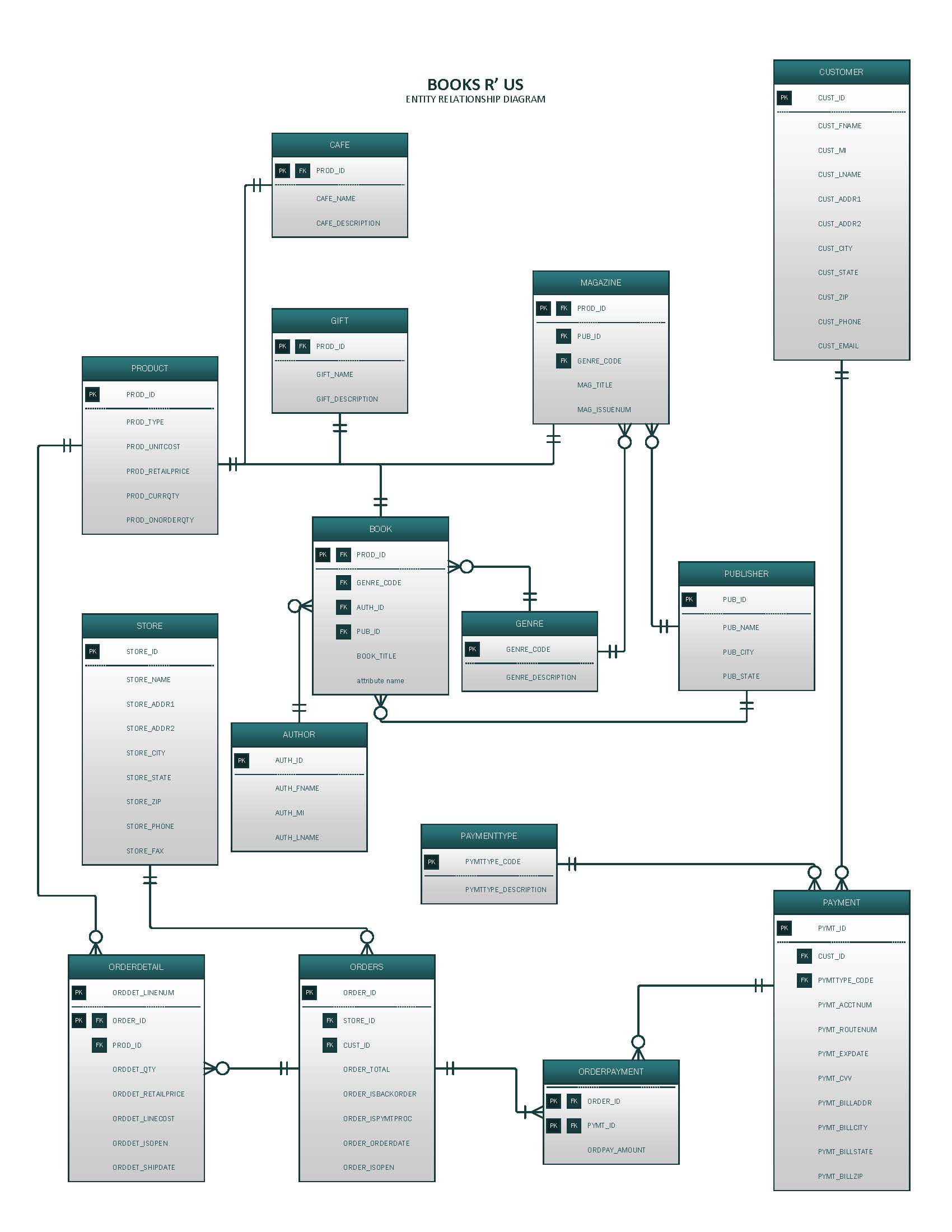
Based upon the company’s business rules and the project requirements outlined in this document, the database will be designed as follows:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **GIFT** |  | **BOOK** |  | **MAGAZINE** |  | **CAFE** |
| PROD\_ID **(PK) (FK)** |  | PROD\_ID **(PK) (FK)** |  | PROD\_ID **(PK) (FK)** |  | PROD\_ID **(PK) (FK)** |
| GIFT\_NAME |  | GENRE\_CODE **(FK)** |  | PUB\_ID **(FK)** |  | CAFE\_NAME |
| GIFT\_DESCRIPTION |  | AUTH\_ID **(FK)** |  | GENRE\_CODE **(FK)** |  | CAFE\_DESCRIPTION |
|  |  | PUB\_ID **(FK)** |  | MAG\_TITLE |  |  |
|  |  | BOOK\_TITLE |  | MAG\_ISSUENUM |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CUSTOMER** |  | **PRODUCT** |  | **PAYMENT** |  | **ORDERPAYMENT** |
| CUST\_ID **(PK)** |  | PROD\_ID **(PK)** |  | PYMT\_ID **(PK)** |  | ORDER\_ID **(PK) (FK)** |
| CUST\_FNAME |  | PROD\_TYPE |  | CUST\_ID **(FK)** |  | PYMT\_ID **(PK) (FK)** |
| CUST\_MI |  | PROD\_UNITCOST |  | PYMTTYPE\_CODE **(FK)** |  | ORDPAY\_AMOUNT |
| CUST\_LNAME |  | PROD\_RETAILPRICE |  | PYMT\_ACCTNUM |  |  |
| CUST\_ADDR1 |  | PROD\_CURRQTY |  | PYMT\_ROUTENUM |  |  |
| CUST\_ADDR2 |  | PROD\_ONORDERQTY |  | PYMT\_EXPDATE |  |  |
| CUST\_CITY |  |  |  | PYMT\_CVV |  |  |
| CUST\_STATE |  |  |  | PYMT\_BILLADDR |  |  |
| CUST\_ZIP |  |  |  | PYMT\_BILLCITY |  |  |
| CUST\_PHONE |  |  |  | PYMT\_BILLSTATE |  |  |
| CUST\_EMAIL |  |  |  | PYMT\_BILLZIP |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ORDERS** |  | **ORDER DETAIL** |  | **STORE** |  | **PAYMENTTYPE** |
| ORDER\_ID **(PK)** |  | ORDDET\_LINENUM **(PK)** |  | STORE\_ID **(PK)** |  | PYMTTYPE\_CODE **(PK)** |
| STORE\_ID **(FK)** |  | ORDER\_ID **(PK) (FK)** |  | STORE\_NAME |  | PYMTTYPE\_DESCRIPTION |
| CUST\_ID **(FK)** |  | PROD\_ID **(FK)** |  | STORE\_ADDR1 |  |  |
| ORDER\_TOTAL |  | ORDDET\_QTY |  | STORE\_ADDR2 |  |  |
| ORDER\_ISBACKORDER |  | ORDDET\_RETAILPRICE |  | STORE\_CITY |  |  |
| ORDER\_ISPYMTPROC |  | ORDDET\_LINECOST |  | STORE\_STATE |  |  |
| ORDER\_ORDERDATE |  | ORDDET\_ISOPEN |  | STORE\_ZIP |  |  |
| ORDER\_ISOPEN |  | ORDDET\_SHIPDATE |  | STORE\_PHONE |  |  |
|  |  |  |  | STORE\_FAX |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **GENRE** |  | **PUBLISHER** |  | **AUTHOR** |
| GENRE\_CODE **(PK)** |  | PUB\_ID **(PK)** |  | AUTH\_ID **(PK)** |
| GENRE\_DESCRIPTION |  | PUB\_NAME |  | AUTH\_FNAME |
|  |  | PUB\_CITY |  | AUTH\_MI |
|  |  | PUB\_STATE |  | AUTH\_LNAME |



**FUNCTIONAL RELATIONAL SCHEMAS/DEPENDENCIES DIAGRAMS**

The following relational schemas and dependency diagrams are based upon the Entity Relationship Diagram presented above and reflect table structures normalized to 3NF or above:

**Relational Schema (GIFT):**

GIFT(**PROD\_ID**, GIFT\_NAME, GIFT\_DESCRIPTION)

**Dependency Diagram (GIFT):**

|  |  |  |
| --- | --- | --- |
| **PROD\_ID** | **GIFT\_NAME** | **GIFT\_DESCRIPTION** |

**Relational Schema (BOOK):**

BOOK(**PROD\_ID**, GENRE\_CODE, AUTH\_ID, PUB\_ID, BOOK\_TITLE)

**Dependency Diagram (BOOK):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PROD\_ID** | **GENRE\_CODE** | **AUTH\_ID** | **PUB\_ID** | **BOOK\_TITLE** |

**Relational Schema (MAGAZINE):**

MAGAZINE(**PROD\_ID**, PUB\_ID, GENRE\_CODE, MAG\_TITLE, MAG\_ISSUENUM)

**Dependency Diagram (MAGAZINE):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PROD\_ID** | **PUB\_ID** | **GENRE\_CODE** | **MAG\_TITLE** | **MAG\_ISSUENUM** |

**Relational Schema (CAFE):**

CAFE(**PROD\_ID**, CAFE\_NAME, CAFE\_DESCRIPTION)

**Dependency Diagram (CAFE):**

|  |  |  |
| --- | --- | --- |
| **PROD\_ID** | **CAFE\_NAME** | **CAFE\_DESCRIPTION** |

**Relational Schema (CUSTOMER):**

CUSTOMER(**CUST\_ID**, CUST\_FNAME, CUST\_MI, CUST\_LNAME, CUST\_ADDR1, CUST\_ADDR2, CUST\_CITY, CUST\_STATE, CUST\_ZIP, CUST\_PHONE, CUST\_EMAIL)

**Dependency Diagram (CUSTOMER):**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CUST\_ID** | **CUST\_FNAME** | **CUST\_MI** | **CUST\_LNAM** | **CUST\_ADDR1** | **CUST\_ADDR2** | **CUST\_CITY** | **CUST\_STATE** | **CUST\_ZIP** | **CUST\_PHONE** | **CUST\_EMAIL** |

**Relational Schema (PRODUCT):**

PRODUCT(**PROD\_ID**, PROD\_TYPE, PROD\_UNITCOST, PROD\_RETAILPRICE, PROD\_CURRQTY, PROD\_ONORDERQTY)

**Dependency Diagram (PRODUCT):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PROD\_ID** | **PROD\_TYPE** | **PROD\_UNITCOST** | **PROD\_RETAILPRICE** | **PROD\_CURRQTY** | **PROD\_ONORDERQTY** |

**Relational Schema (PAYMENT):**

PAYMENT(**PYMT\_ID**, CUST\_ID, PYMTTYPE\_CODE, PYMT\_ACCTNUM, PYMT\_ROUTENUM, PYMT\_EXPDATE, PYMT\_CVV, PYMT\_BILLADDR, PYMT\_BILLCITY, PYMT\_BILLSTATE, PYMT\_BILLZIP)

**Dependency Diagram (PAYMENT):**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PYMT\_ID** | **CUST\_ID** | **PYMTTYPE\_CODE** | **PYMT\_ACCTNUM** | **PYMT\_ROUTENUM** | **PYMT\_EXPDATE** | **PYMT\_CVV** | **PYMT\_BILLADDR** | **PYMT\_BILLCITY** | **PYMT\_BILLSTATE** | **PYMT\_BILLZIP** |

**Relational Schema (ORDERPAYMENT):**

ORDERPAYMENT(**ORDER\_ID**, **PYMT\_ID**, ORDPAY\_AMOUNT)

**Dependency Diagram (ORDERPAYMENT):**

|  |  |  |
| --- | --- | --- |
| **ORDER\_ID** | **PYMT\_ID** | **ORDPAY\_AMOUNT** |

**Relational Schema (PAYMENTTYPE):**

PAYMENTTYPE(**PYMTTYPE\_CODE**, PYMTTYPE\_DESCRIPTION)

**Dependency Diagram (PAYMENTTYPE):**

|  |  |
| --- | --- |
| **PYMTTYPE\_CODE** | **PYMTTYPE\_DESCRIPTION** |

**Relational Schema (ORDER):**

ORDER(**ORDER\_ID**, CUST\_ID, STORE\_ID, PYMT\_ID, ORDER\_TOTAL, ORDER\_ISBACKORDER, ORDER\_ISPYMTPROC, ORDER\_ORDERDATE, ORDER\_ISOPEN)

**Dependency Diagram (ORDER):**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ORDER\_ID** | **CUST\_ID** | **STORE\_ID** | **PYMT\_ID** | **ORDER\_TOTAL** | **ORDER\_ISBACKORDER** | **ORDER\_ISPYMTPROC** | **ORDER\_ORDERDATE** | **ORDER\_ISOPEN** |

**Relational Schema (ORDERDETAIL):**

ORDERDETAIL(**ORDDET\_LINENUM**, **ORDER\_ID**, PROD\_ID, ORDDET\_QTY, ORDDET\_UNITPRICE, ORDDET\_LINECOST, ORDDET\_ISOPEN, ORDDET\_SHIPDATE)

**Dependency Diagram (ORDERDETAIL):**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ORDDET\_LINENUM** | **ORDER\_ID** | **PROD\_ID** | **ORDDET\_QTY** | **ORDDET\_UNITPRICE** | **ORDDET\_LINECOST** | **ORDDET\_ISOPEN** | **ORDDET\_SHIPDATE** |

**Relational Schema (STORE):**

STORE(**STORE\_ID**, STORE\_NAME, STORE\_ADDR1, STORE\_ADDR2, STORE\_CITY, STORE\_STATE, STORE\_ZIP, STORE\_PHONE, STORE\_FAX)

**Dependency Diagram (STORE):**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STORE\_ID** | **STORE\_NAME** | **STORE\_ADDR1** | **STORE\_ADDR2** | **STORE\_CITY** | **STORE\_STATE** | **STORE\_ZIP** | **STORE\_PHONE** | **STORE\_FAX** |

**Relational Schema (GENRE):**

GENRE(**GENRE\_CODE**, GENRE\_DESCRIPTION)

**Dependency Diagram (GENRE):**

|  |  |
| --- | --- |
| **GENRE\_CODE** | **GENRE\_DESCRIPTION** |

**Relational Schema (PUBLISHER):**

PUBLISHER(**PUB\_ID**, PUB\_NAME, PUB\_CITY, PUB\_STATE)

**Dependency Diagram (PUBLISHER):**

|  |  |  |  |
| --- | --- | --- | --- |
| **PUB\_ID** | **PUB\_NAME** | **PUB\_CITY** | **PUB\_STATE** |

**Relational Schema (AUTHOR):**

AUTHOR(**AUTH\_ID**, AUTH\_FNAME, AUTH\_MI, AUTH\_LNAME)

**Dependency Diagram (AUTHOR):**

|  |  |  |  |
| --- | --- | --- | --- |
| **AUTH\_ID** | **AUTH\_FNAME** | **AUTH\_MI** | **AUTH\_LNAME** |

**PROPOSED QUERIES**

The following list includes questions that were identified during an interview with the clients as being of particular importance, along with a proposed SQL query for each that would return the information requested in the question based upon the proposed relational database design:

1. How many books are sold each month by the publisher?

SELECT DATEPART(MM, ORDER\_ORDERDATE) AS ORDERMONTH, PUB\_NAME, SUM(ORDDET\_QTY) AS NUMBOOKS

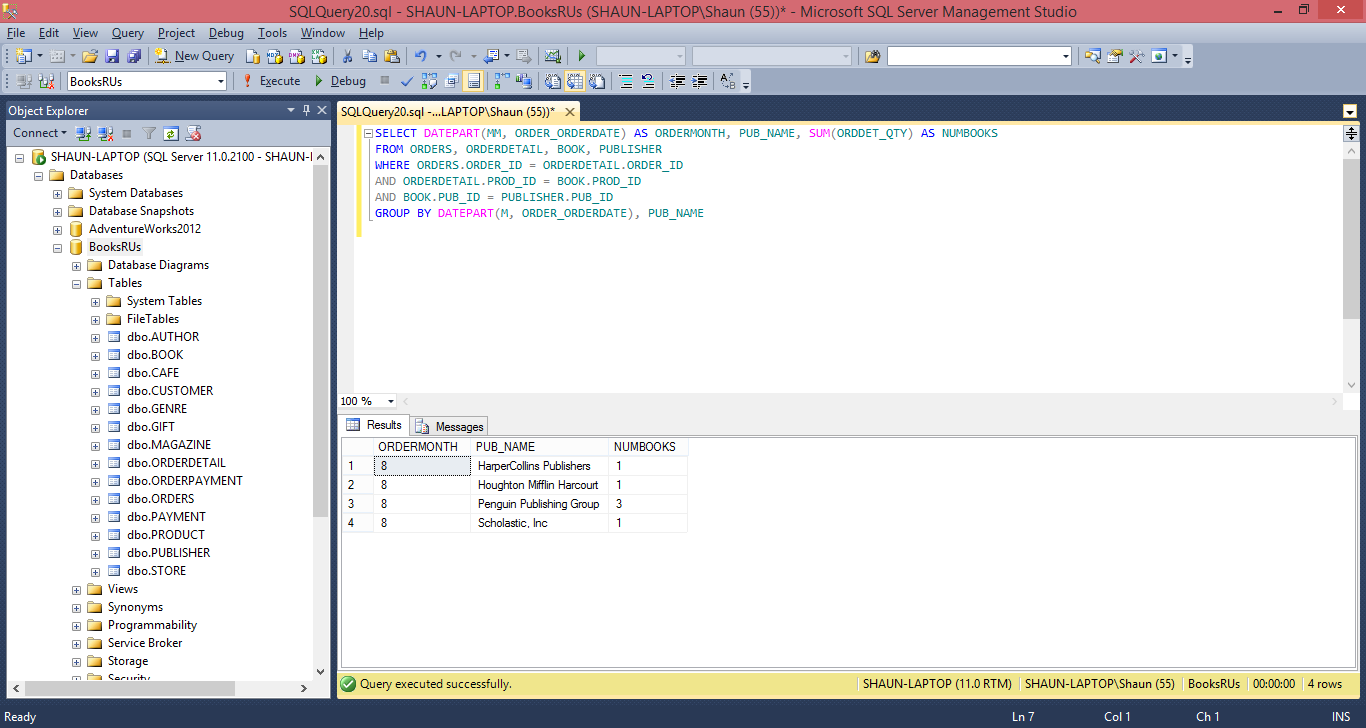
FROM ORDERS, ORDERDETAIL, BOOK, PUBLISHER

WHERE ORDERS.ORDER\_ID = ORDERDETAIL.ORDER\_ID

AND ORDERDETAIL.PROD\_ID = BOOK.PROD\_ID

AND BOOK.PUB\_ID = PUBLISHER.PUB\_ID

GROUP BY DATEPART(M, ORDER\_ORDERDATE), PUB\_NAME



1. Which authors are the biggest sellers of books in our stores?

SELECT AUTH\_LNAME, AUTH\_FNAME, AUTH\_MI, SUM(ORDDET\_RETAILPRICE) AS TOTALSALES

FROM ORDERS, ORDERDETAIL, BOOK, AUTHOR

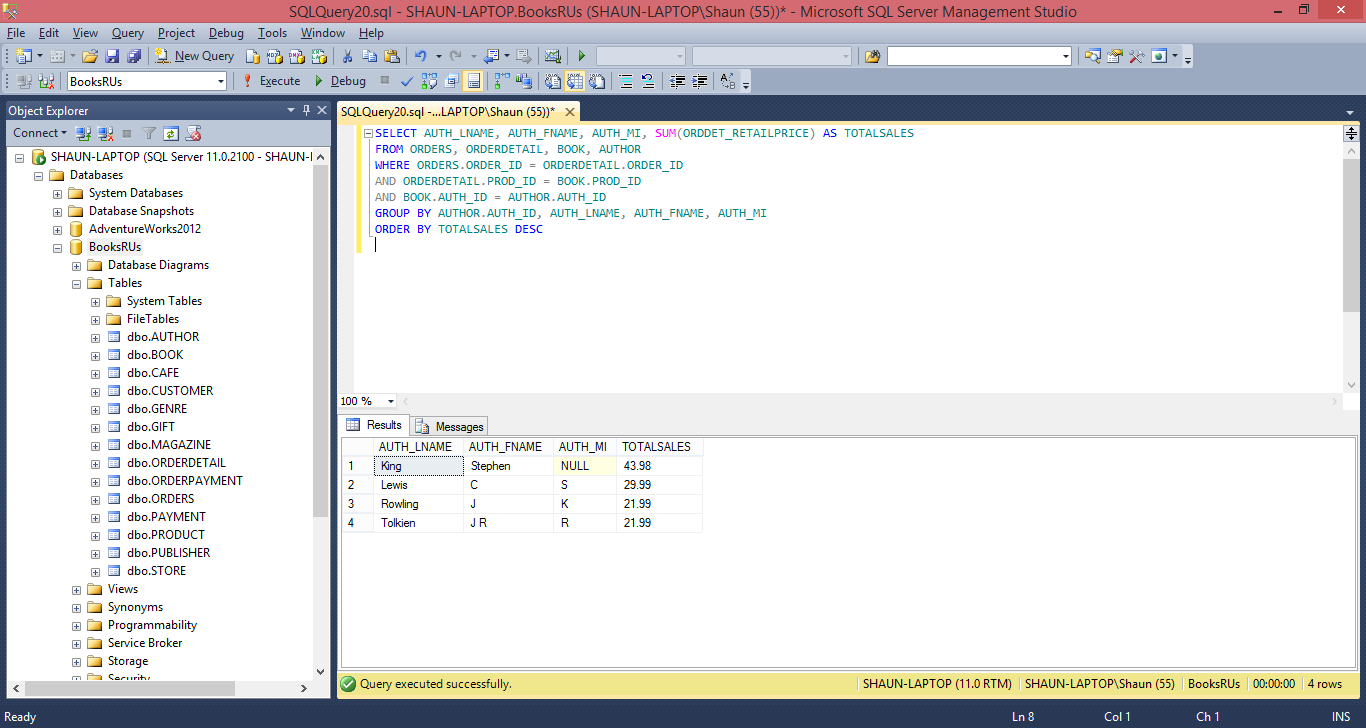
WHERE ORDERS.ORDER\_ID = ORDERDETAIL.ORDER\_ID

AND ORDERDETAIL.PROD\_ID = BOOK.PROD\_ID

AND BOOK.AUTH\_ID = AUTHOR.AUTH\_ID

GROUP BY AUTHOR.AUTH\_ID, AUTH\_LNAME, AUTH\_FNAME, AUTH\_MI

ORDER BY TOTALSALES DESC



1. What books are associated with each publisher?

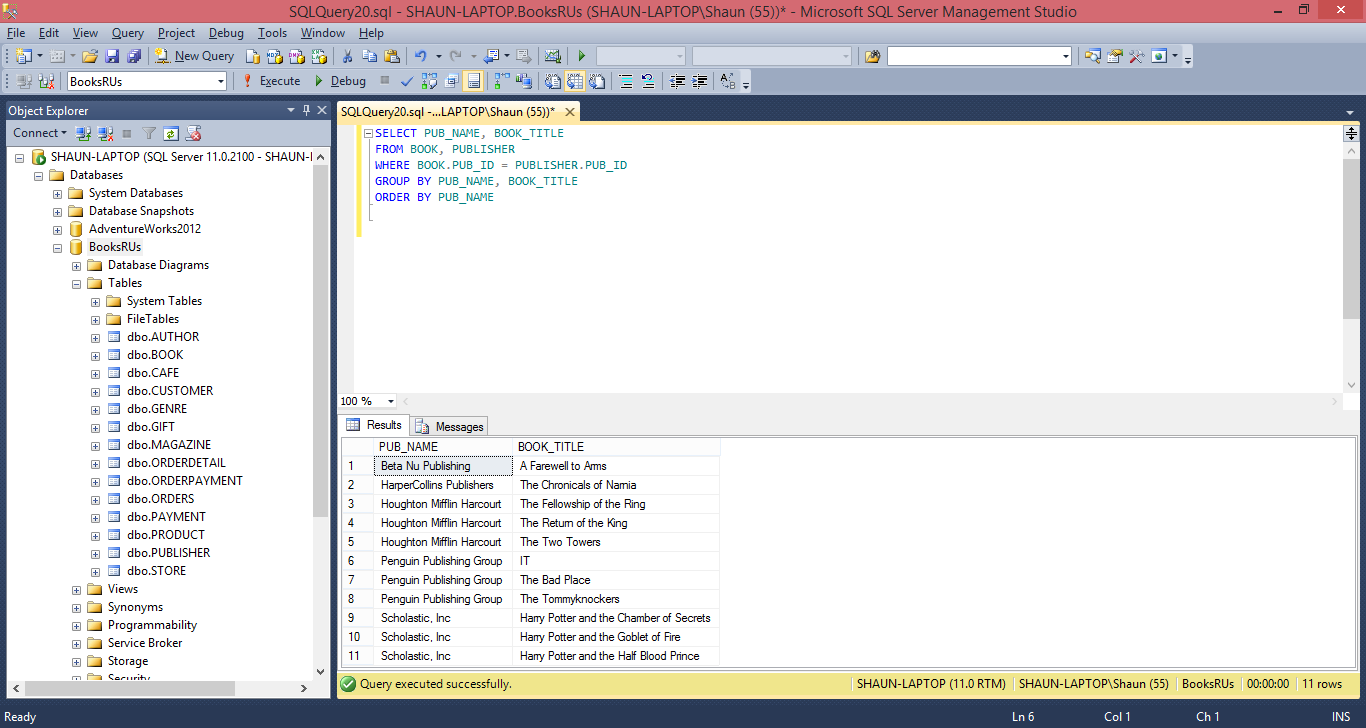
SELECT PUB\_NAME, BOOK\_TITLE

FROM BOOK, PUBLISHER

WHERE BOOK.PUB\_ID = PUBLISHER.PUB\_ID

GROUP BY PUB\_NAME, BOOK\_TITLE

ORDER BY PUB\_NAME



1. What are the most popular products besides books that are sold in each store?

SELECT STORE.STORE\_NAME, NONBOOKITEMS.PROD\_TYPE, SUM(NONBOOKITEMS.ORDDET\_RETAILPRICE) AS TOTALSALES

FROM (SELECT ORDERDETAIL.ORDER\_ID, ORDERDETAIL.ORDDET\_RETAILPRICE, PRODUCT.PROD\_TYPE

FROM ORDERDETAIL INNER JOIN PRODUCT

ON ORDERDETAIL.PROD\_ID = PRODUCT.PROD\_ID

WHERE PRODUCT.PROD\_TYPE <> 'BOOK') AS NONBOOKITEMS

INNER JOIN ORDERS

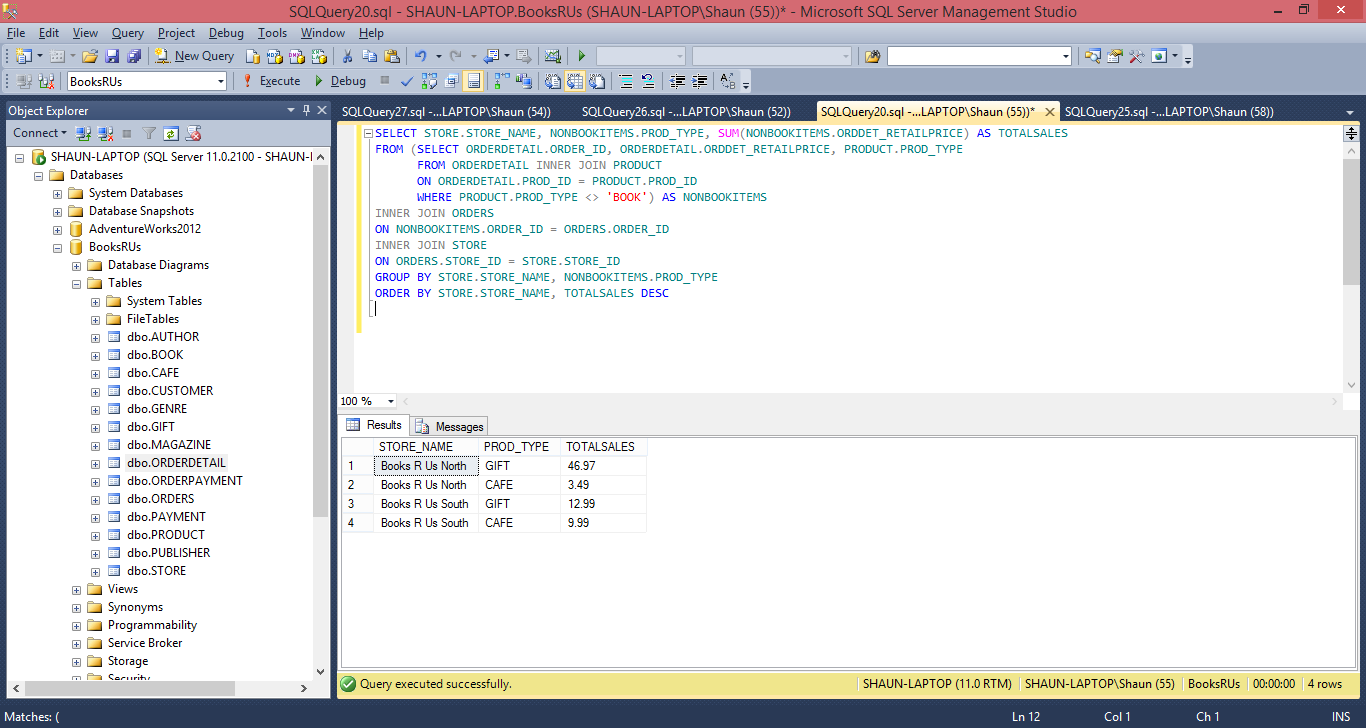
ON NONBOOKITEMS.ORDER\_ID = ORDERS.ORDER\_ID

INNER JOIN STORE

ON ORDERS.STORE\_ID = STORE.STORE\_ID

GROUP BY STORE.STORE\_NAME, NONBOOKITEMS.PROD\_TYPE

ORDER BY STORE.STORE\_NAME, TOTALSALES DESC

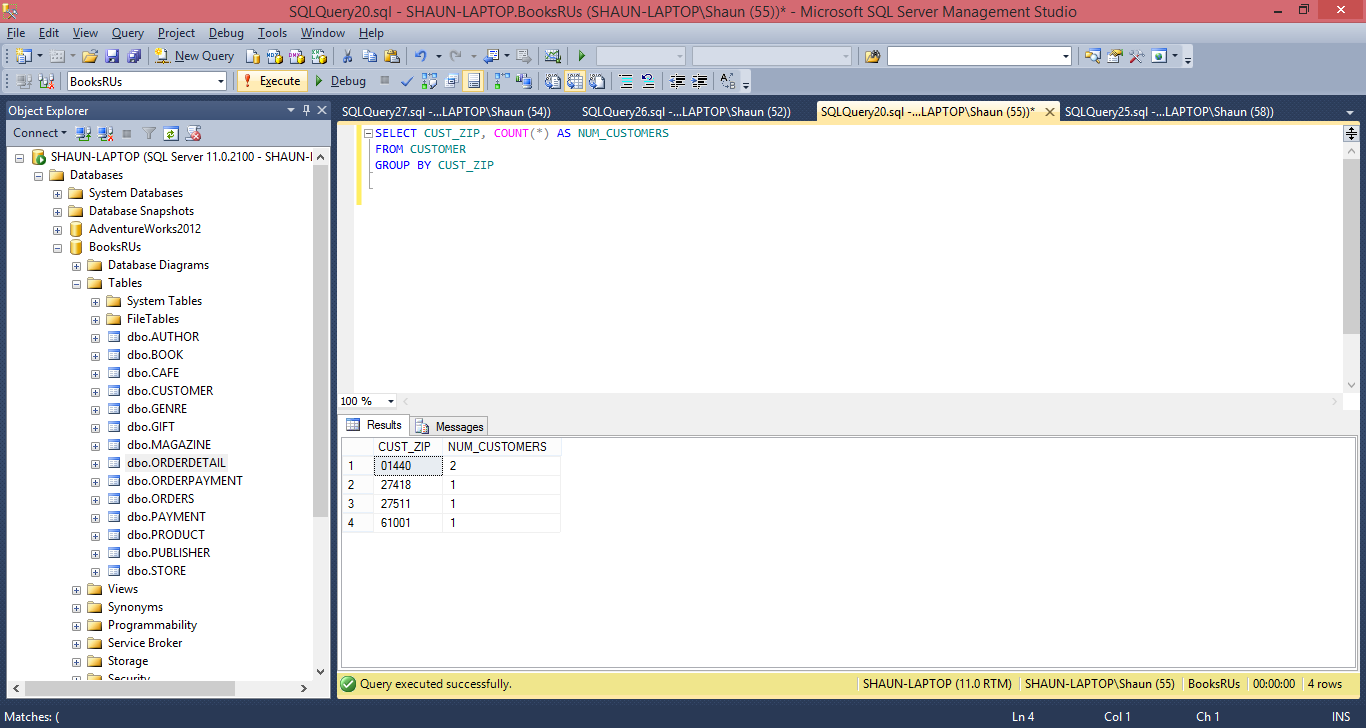


1. From what region(s) (by ZIP code) do customers visit our stores?

SELECT CUST\_ZIP, COUNT(\*) AS NUM\_CUSTOMERS

FROM CUSTOMER

GROUP BY CUST\_ZIP



1. What customer data must be stored for the e-commerce portion of the website?

INSERT INTO CUSTOMER

(CUST\_FNAME, CUST\_MI, CUST\_LNAME, CUST\_ADDR1, CUST\_CITY, CUST\_STATE, CUST\_ZIP, CUST\_PHONE, CUST\_EMAIL)

VALUES

('John', 'C', 'Smith', '123 Any Street', 'Somewhere', 'MA', '01440', '508-555-1234', 'jsmith@something.com')