**IST 220 – Intro to DB**

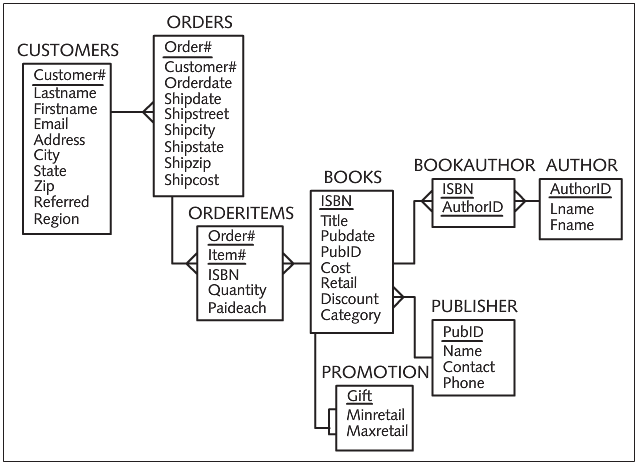
**Project 1 – Retrieving Info from One or More Tables**

Finish the following queries using the JustLee Bookstore database (JLBookstore), as described on the next 2 pages.

1. List the customer number for each customer in the CUSTOMERS table, along with the city and state in which the customer lives.
2. Create a list of authors that displays the last name followed by the first name for each author. The last names and first names should be separated by a comma and a blank space.
3. Modify the query from the last problem: rename the column label as Authors and sort the names in alphabetical order.
4. Create a list that displays the title of each book and the name and phone number of the contact at the publisher‘s office for reordering each book.
5. Determine which orders haven‘t yet shipped and the name of the customer who placed the order. Please sort the results by the date on which the order was placed.
6. Produce a list of all customers who live in the state of Florida and have ordered books about computers. [Hint: use values in the Category column to determine book topics.]
7. Determine which books customer Jake Lucas has purchased. Perform the search using the customer name, not the customer number.
8. Modify the query from the last problem: If the customer (Jake Lucas) has purchased multiple copies of the same book, just show each title once, i.e. display distinct titles only.

Please format your file the same way as the labs and upload it onto Canvas system.

**Description of the *JustLee Books* Database**



**Basic Assumptions**

Three assumptions were made when designing the JustLee Books database:

* An order isn’t shipped until all items for the order are available. (In other words, there are no back orders or partial order shipments.)
* All addresses are in the United States; otherwise, the structure of the Address/Zip Code fields would need to be altered because many countries use different address information, such as province names.
* Only orders placed in the current month or orders placed in previous months that didn’t ship are stored in the ORDERS table. (At the end of each month, all completed orders are transferred to an annual SALES table.)

**CUSTOMERS** table: Notice that the CUSTOMERS table is the first table in the DB schema diagram above. It serves as a master table for storing basic data related to any customer who has placed an order with JustLee Books. It stores the customer’s name, e-mail address, and mailing address, plus the CustomerNo of the person who referred that customer to the company.

**BOOKS** table: The BOOKS table stores each book’s ISBN, title, publication date, publisher ID, wholesale cost, and retail price. The table also stores a category name for each book (for example, Fitness, Children, Cooking) to track customers’ purchasing patterns, as mentioned.

**AUTHOR** and **BOOKAUTHOR** tables: As shown in Figure 1-5, the AUTHOR table maintains a list of authors’ names. Because a many-to-many relationship originally existed between the books entity and the Authors entity, the BOOKAUTHOR table was created as a bridging table between these two entities. The BOOKAUTHOR table stores each book’s ISBN and author ID. If you need to know who wrote a particular book, you have the DBMS look up the book’s ISBN in the BOOKS table, then look up each entry of the ISBN in the BOOKAUTHOR table, and finally trace the author’s name back to the AUTHORS table through the AuthorID field.

**ORDERS** and **ORDERITEMS** tables: Data about a customer’s order is divided into two tables: ORDERS and ORDERITEMS. The ORDERS table identifies which customer placed each order, the date the order was placed, the date it was shipped, and the shipping cost charged. Because the shipping address might be different from a customer’s billing address, the shipping address is also stored in the ORDERS table. If a customer’s order includes two or more books, the ORDERS table could contain a repeating group. Therefore, the items purchased for each order are stored separately in the ORDERITEMS table.

The ORDERITEMS table records the order number, the ISBN of the book being purchased, and the quantity for each book. To uniquely identify each item in an order when multiple items are purchased, the table includes an ItemNo field that corresponds to the item’s position in the sequence of products ordered. For example, if a customer places an order for three different books, the first book listed in the order is assigned ItemNo 1, the second book listed is ItemNo 2, and so on. A variation of this table could use the combination of the OrderNo and the book’s ISBN to identify each product for an order. However, the concept of itemNo or lineNo is widely used in the industry to identify line items on an invoice or in a transaction, so it has been included in this table to familiarize you with the concept.

The Paideach field in the ORDERITEMS table records the price the customer actually paid per copy for a specific book. This price is recorded because the Retail field in the BOOKS table is modified as book prices change, and the current database doesn’t maintain a historical book price list.

**PUBLISHER** table: The PUBLISHER table contains the publisher’s ID code, name, contact person, and telephone number. The PUBLISHER table can be joined to the BOOKS table through the PubID field, which is the common field. This linked data from the PUBLISHER and BOOKS table enables you to determine which publisher to contact when you need to reorder books by identifying which books you obtained from each publisher.

**PROMOTION** table: The last table in Figure 1-5 is the PROMOTION table. JustLee Books has an annual promotion that includes a gift with each book purchased. The gift is based on the book’s retail price. Customers ordering books that cost less than $12 receive a certain gift, and customers buying books costing between $12.01 and $25 receive a different gift. The PROMOTION table identifies the gift and the minimum and maximum retail values of the range. There’s no exact value that matches the Retail field in the BOOKS table; therefore, to determine the correct gift, you need to determine whether a retail price falls within a particular range.