CSE 3241 Project Checkpoint 02 – Relational Model and Relational Algebra

Ryan Bone, Brad Myers, David Baker, Joey Chagnon, Wade Duym - Mon July 7

In a **NEATLY TYPED** document, provide the following:

1. Provide a current version of your ER Model as per Project Checkpoint 01. If you were instructed to change the model for Project Checkpoint 01, make sure you use the revised version of your ER Model.

- some entities may not be necessary and some entities could be combined
- EMPLOYEE and TIMESHEET, may not relate too close to the sales operation in this bookstore.
- keep either BOOK or PRODUCT, since we are only considering book as the only product to sell. and many relations can be developed upon BOOK.
- the PUBLISHER could be the SUPPLIER,so maybe just keep one?
- SHIPPER, SHIPMENT may not be needed a lot later in the project, so I suggest you taking them out,
- just use the WAREHOUSE to keep track of inventories, and
- since I suggest taking out PRODUCT, INVENTORY is not necessary to be there as well.

2. Map your ER model to a relational schema. Indicate all primary and foreign keys.

- Publisher(ID, Name, Phone, Email, P.O. Box, Country, City, State, Zip, Address)
- Product (ISBN, Price, Title, Department, AuthorFName, AuthorLName, Category, Description, Length, PublisherID)
- Rating (CID, PID, Rating)
- Inventory (PID, Quantity, ProductID, WarehouseID)
- Warehouse(ID, Country, City, State, Zip, Address)
- Shipment(ID, PublisherID, WarehouseID, Departure, Arrival)
- Order (ID, WarehouseID, Departure, CustomerID, TransactionID)
- Contents (ProductID, PackageID) - PackageID is FK to a transaction, order or shipment ID
- Customer (ID, Name, Phone, Email, P.O. Box, Country, City, State, Zip, Address)
- Transaction (ID, type, datetime, amount, tax, PayerId, PayeeID, PaymentProcessorID)
- Employee(ID, Schedule, Salary, Wage, ManagerID, Title, Start Date, WarehouseID)
- Payment Processor(Name, Phone, Email, P.O. Box, Country, City, State, Zip, Address)

3. Given your relational schema, provide the relational algebra to perform the following queries. If your schema cannot provide answers to these queries, revise your ER Model and your relational schema to contain the appropriate information for these queries:
   a. Find the titles of all books by Pratchett that cost less than $10

   $\pi_{<Title>}(\sigma_{< Price < 10 AND Author = "Pratchett" >}(Product))$

   b. Give all the titles and their dates of purchase made by a single customer (you choose how to designate the customer)

   $\pi_{<Title, Datetime>}(\sigma_{<Payer = "Customer\_ID" >}(Transaction) \bowtie_{<T.id=TransactioniD>} Order \bowtie_{<O.id=C.PackageID>} Contents)$

c. Find the titles and ISBNs for all books with less than 5 copies in stock

$\pi_{<ISBN, Title>}(\sigma_{<Quantity < 5>}(Inventory) \bowtie Product)$

d. Give all the customers who purchased a book by Pratchett and the titles of Pratchett books they purchased

$\pi_{<CustomerID, Title>}(\sigma_{<AuthorLName="Pratchett">}(Product) \bowtie_{<ISBN=ProductID>} (Transaction \bowtie_{<Payer=CuatomerID>} Customer)$

e. Find the total number of books purchased by a single customer (you choose how to designate the customer)

$F_{COUNT<*>}(\sigma_{<Payer="CustomerID">}(Transaction))$

f. Find the customer who has purchased the most books and the total number of books they have purchased

$F_{MAX\ Count}(_{<Payer>}F_{COUNT<*>}\ (Transaction \bowtie_{<Payer=ID>} Customer))$

4. Come up with three additional interesting queries that your database can provide. Give what the queries are supposed to retrieve in plain English and then as relational algebra. Your queries should include joins and at least one should include an aggregate function. At least one of your queries should use "extra" entities you added to your model in Checkpoint 01.

Count the number of books we have by a particular author.

$F_{SUM<Quantity>}(\sigma_{<Author="J.K.">}(Product) \bowtie_{<ISBN=ProductID>} Inventory)$

Get average rating of a particular book.

$F_{AVG\ Rating}(\sigma_{<Title="War and Peace">}(Product) \bowtie_{<ISBN=PID>} Rating)$

Get the highest paid manager.

$F_{MAX\ Salary}\ (\pi_{<Manager>}(Employees)) \bowtie_{<Manager=ID>} Employee)$