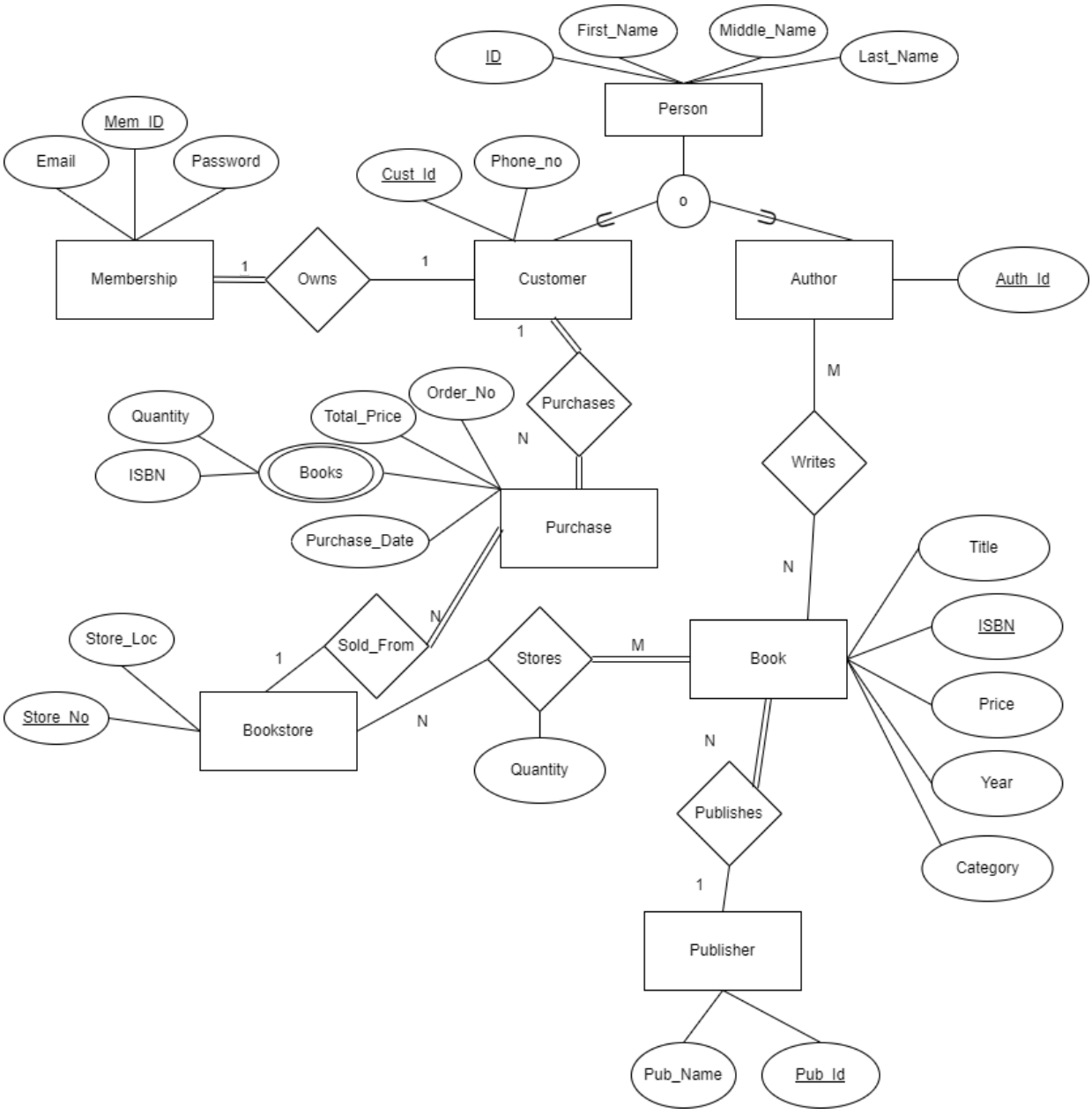
CSE 3241 Project Checkpoint 04 – Functional Dependencies and Normal Forms

Names: Tyra Shin, Sean Kelley, Trey Thomas, Jack Kern, Justin Sparacino	Date

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

1. Provide a current version of your ER Diagram and Relational Model as per Project Checkpoint 03. **If you were instructed to change the model for Project Checkpoint 03, make sure you use the revised versions of your models.**



Membership(Password, Mem_ID(FK))

Person(ID, First_Name, Middle_Name, Last_Name, Email)

Author(Auth_ID)

Writes(Author_ID, ISBN)

Purchase(Order_No, Cust_ID(FK), Total_Price, Purchase_Date, Store_no(FK))

Publisher(Pub_ID, Pub_Name)

Customer(Cust_Id(FK), Phone_no)

Book(ISBN, Pub_ID(FK), Year, Price, Title, Category)

Books(ISBN**,** Order_no, quantity)

Bookstore(Store_no, store_loc)

Stored(ISBN, Store_no, Quantity)

2. For each relation schema in your model, indicate the functional dependencies. Think carefully about what you are modeling here - make sure you consider all the possible dependencies in each relation and not just the ones from your primary keys. For example, a customer's credit card number is unique, and so will uniquely identify a customer even if you have another key in the same table (in fact, if the customer can have multiple credit card numbers, the dependencies can get even more involved).

   Person:
   ID -> (First_Name, Middle_Name, Last_Name, Email)

   Book:
   ISBN -> (Pub_ID, Year, Price, Title, Category)

   Bookstore:
   Store_no -> Store_loc

   Customer:
   Cust_id -> Phone_no

   Books:
   (ISBN, Order_No) -> quantity

   Membership:
   Mem_ID -> Password

Author:
Auth_ID -> None (since there are no other attributes)

Writes:
(Author_ID, ISBN) -> None (since there are no other attributes)

Publisher:
Pub_ID -> Pub_Name

Stored:
(ISBN, Store_no) -> Quantity

Purchase:
Order_no -> (Cust_Id, Store_no, Total_price, Purchase_date)

3. For each relation schema in your model, determine the highest normal form of the relation. If the relation is not in 3NF, rewrite your relation schema so that it is in at least 3NF.

   Person: BCNF

   Book: BCNF

   Bookstore: BCNF

   Customer: BCNF

   Books: BCNF

   Membership: BCNF

   Author: BCNF

   Writes: BCNF

   Publisher: BCNF

   Stored: BCNF

4. For each relation schema in your model that is in 3NF but not in BCNF, either rewrite the relation schema to BCNF or provide a short justification for why this relation should be an exception to the rule of putting relations into BCNF.

   They are all already BCNF

5. For your database, propose at least two interesting views that can be built from your relations. These views must involve joining at least two tables together each and must include some kind of aggregation in the view. Each view must also be able to be described by a one or two sentence description in plain English. Provide the code for constructing your views along with the English language description of what the view is supposed to be providing.

   View 1: Customer Purchases. This view could show each customer's total expenditure on books. It would involve joining the Customer and Purchase tables and aggregating on Total_Price.

   ```
   CREATE VIEW Customer_Purchases AS

   SELECT Cust_id, SUM(Total_Price) as Total_Expenditure

   FROM Customer JOIN Purchase ON Customer.Cust_id = Purchase.Cust_id

   GROUP BY Cust_id;
   ```

   View 2: Author Book Inventory. This view would show the total quantity of all of an author's books across all bookstores. It would involve joining the Book, Stored, and Writes tables and aggregating on Quantity.

   ```
   CREATE VIEW Book_Inventory AS

   SELECT Title, SUM(Quantity) as Total_Quantity

   FROM Writes JOIN (Book JOIN Stored ON Book.ISBN = Stored.ISBN) ON Writes.Auth_id = Book. Auth_id

   WHERE Auth_id = <author id>

   GROUP BY Title;
   ```

Feedback: There needs to be a direct relationship between Purchase and Book, not thought Bookstore. You don't include "Books" as a multivalued attribute on Purchase, that should be the relationship.

How we addressed it: We created a direct relationship between Purchase and Book. We also removed the Books as a multivalued attribute on Purchase.