**Saajid Aliyar**

**101153376**

**Comp 3005 Project**

**2.1**

In this project, the design I had chosen allows the user to become its own entity where they only have control on checking out the book and must be a registered member of the bookstore. The user and book depend on the checkout relation, where a book will not be checked out if a user does not place an order. A user is dependent on the book. A book’s primary key would be the ISBN where a user can uniquely identify a book. Each book must be dependent on having a publisher, and a book cannot exist without a publisher, and we are assuming a publisher does not exist if they do not have at least 1 book. The bookstore contains books and shares a contains relationship, where a bookstore must have any number of books to become a bookstore. (A bookstore can still be a bookstore without any books, for reasons such as books being sold out etc.). A bookstore can be uniquely identified with a bookstore name which is a primary key. A bookstore must contain an order for it to become a bookstore as it requires an order and sales for it to become a bookstore. A bookstore would contain orders which has tracking information and the order number for when a user places an order. There can be multiple owners of the bookstore and have reports discussing the sales and expenditures where there is a report id to uniquely identify each report. There is also a computed salesVsX report where we can compute the salesVsExpenditures or any sales per genre etc.

**ASSUMPTIONS**

Assume one bookstore can have many owners

Assume one publisher can have many books

Assume any number of owners can have any number of reports

Assume that a user can’t checkout books without a registration ID

Assume that any report metric can be computed using other stored data

Assume that there is only one bookstore

Assume a book must have a publisher

Assume a publisher must have at least one book in the system

Assume a book is always available to be checked out by a user and a registered user must checkout a book

Diagram

Description automatically generated

**2.2**

Diagram

Description automatically generated

**2.3. Normalization of Relation Schemas (20%)**

**Registration\_ID = R**

**ownerID = O**

**reportID = P**

**ISBN = I**

**Email = E**

**orderNumber = D**

**bookStoreName = N**

**O->N,P**

**N->D,I,R**

**R->I**

**I->E**

**N->D**

Decomposing (N->D is removed because it’s redundant)

**O->N**

**O->P**

**N->D**

**N->I**

**N->R**

**R->I**

**I->E**

Removing N->I because N->R, and R->I so it becomes redundant

**O->N**

**O->P**

**N->D**

**N->R**

**R->I**

**I->E**

UNION RULE: A->B, A->C => A->B,C

**O->N,P**

**N->D,R**

**R->I**

**I->E**

**This is the normalization of the relational schema. No dependencies were lost so it’s a lossless decomposition. Therefore, it is in good form.**

**2.4. Database Schema Diagram (10%)**

Diagram

Description automatically generated

**2.4**