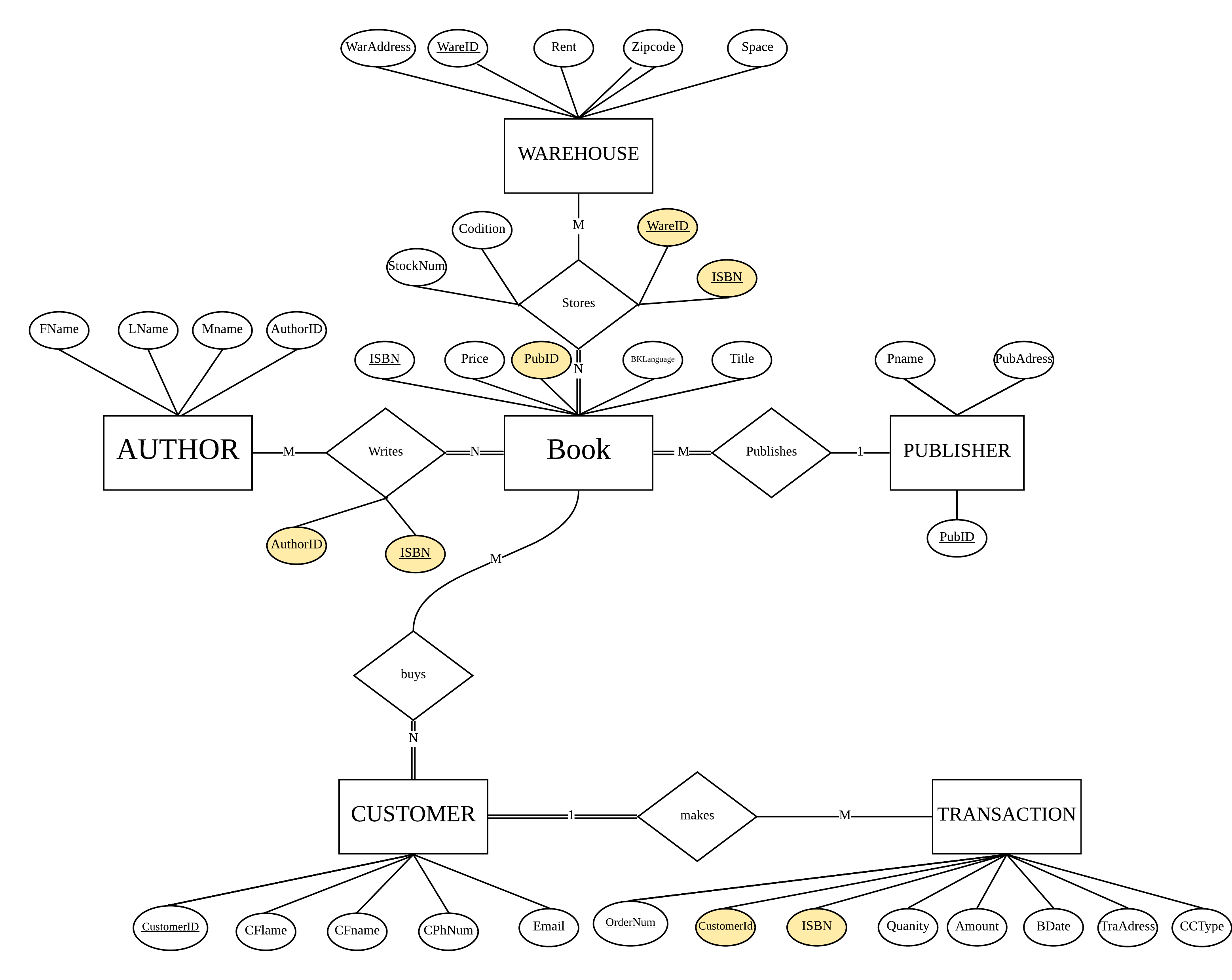
**CSE 3241 Project Checkpoint 04 – Functional Dependencies and Normal Forms**

Names: Lance Chen; Han Han, Yuliang Fu, Tongshuai Zhang Date: 11/14/2018

**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.

****

* **AUTHOR:** AuthorID , Fname, Lname, Mname
* **BOOK**: ISBN, **PubID,** Price, Title, BKLanguage
* **WRITTENBY: ISBN, AuthorID (Note: The red color represents the foreign key)**
* **PUBLISHER**:  PubID, PubAddress, PName
* **WAREHOUSE:** WareID**,** WarAddress, Rent, ZipCode, Space
* **STORES: ISBN ,WareID,** BKCondition, StockNum
* **CUSTOMER:** CustomerID,CFname,CLname,CPhNum, Email
* **TRANSACTION:** OrderNum, **CustomerID**, **ISBN,** Quantity, Amount, BDate, TraAddress, CCType

**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).

* FD1: ***AUTHOR***:  **{** **AuthorID } → {** Fname, Lname, Mname **}**
* FD2: ***BOOK*: { ISBN } →** **{** Price, Title, BKLanguage, PubID, PubAddress, PName **}**
* FD3: ***PUBLISHER*: { PubID } →** **{** PubAddress, PName **}**
* FD4: ***PUBLISHER*: { PubAddress }→{** PubID, PName **}**
* FD5: ***WAREHOUSE:* { WareID }→ {** WarAddress, Rent, Space, ZipCode, **}**
* FD6:***WAREHOUSE:* { WarAddress }→ {** WareID, Rent, Space, ZipCode **}**
* FD7: ***STORES:* { ISBN, WareID } → {** Condition, StockNum **}**
* FD8:***CUSTOMER:* { CustomerID } → {** Email,CFname,CLname,CPhNum **}**
* FD9: ***CUSTOMER:***  **{ Email** **} → {** CustomerID,CFname,CLname,CPhNum **}**
* FD10: ***TRANSACTION:* { OrderNum** **} → {** CustomerID, ISBN, Quantity, Amount, BDate, TraAddress, CCType **}**

**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.

**it’s necessary to build a 2NF for further demonstration and usage.**

**For the AUTHOR relation schema, the AuthorID attribute can determine the other three non-prime  attributes that are all *fully functionally dependent* the primary key *AuthorID.* For the BOOK relation schema, the ISBN attribute can determine the other three non-prime  attributes that are all *fully functionally dependent* the primary key *ISBN.***

**For the WRITTENBY schema, the combination of ISBN, AuthorID forming the primary key can determine the other non-prime attributes that  are all *fully functionally dependent* them*.***

**For the PUBLISHER relation schema, the PubID or PubAddress or both two attributes can determine the other non-prime attribute that is all *fully functionally dependent* the primary key *PubID or the* candidate key *PubAddress or the* super key *PubID* and *PubAddress.* So, the PUBLISHER schema should be decomposed into two relations: new relation *PubAddress* can fully determines the non-prime attributes lossless. Same analysis for the WAREHOUSE schema.**

**For the STORES relation schema, the primary key: ISBN ,WareID can determine the other two non-prime attributes that are all *fully functionally dependent* them*.***

**For the TRANSACTION relation schema, the primary key OrderNum can determine the other two non-prime attributes that are all *fully functionally dependent* them*.*So the CUSTOMER schema should be decomposed into two relations.**

**Besides, no non-prime attributes of the ten relations are transitively dependent on the primary key except the CUSTOMER schema as the CPhNum attribute (CPhNum** **→** CFname,CLname **). After breaking that relation to two relations, all the relations are in 3NF.**

* ***AUTHOR***:  **{** **AuthorID,** Fname, Lname, Mname **}**
* ***BOOK*: { ISBN,** Price, Title, BKLanguage **}**
* **WRITTENBY: ISBN, AuthorID**
* **PUBLISHER**:  PubID, PName
* **PUBLISHERADDR:** PubAddress, PName
* **WAREHOUSE:** WareID**,** Rent, ZipCode, Space
* **WAREHOUSEADDR:** WarAddress, Rent, ZipCode, Space
* **STORES: ISBN ,WareID,** Condition, StockNum
* **CUSTOMER:** CustomerID,CFname,CLname
* **CUSTOMEREMAIL:** Email,CFname,CLname
* **CUSTOMEREPHO: CPhNum**, CFname,CLname
* **TRANSACTION:** OrderNum, **CustomerID**, **ISBN,** Quantity, Amount, BDate, TraAddress, CCType

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.

**All the relation schemas are in BCNF, the only analysis is to demonstrate the STORES relation schema since it has the superKey. However, there are no dependency that the non-prime attributes can determine the prime attribute. So, the schema should in 3NF.**

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 01: Select how many books written by Pi Li the customerID is 23316331 has bought. Codes are below:**

**SELECT SUM(X.Quantity), X.Title, X.ISBN, C.CFname, C.CLname**

**From CUSTOMER C**

**Natural Join (SELECT  \***

**FROM TRANSACTION T, BOOK B, AUTHOR A,WRITTENBY W**

**WHERE T.ISBN = B.ISBN AND A. AuthorID = W.AuthorID AND**

**B.ISBN=W.ISBN AND A.Fname= Pi AND A.Lname = Li) X**

**on  C.CustomerID = X.CustomerID**

**WHERE C.CustomerID = 23316331**

**GROUP BY X.Title, X.ISBN, C.CFname, C.CLname ;**

**View 02: Select how many customers has bought the book where the title is “How to be Quarterfinals in S8 of League of Legends”. Codes are below:**

**SELECT COUNT( \* ), X.Title, X.ISBN**

**From BOOK B**

**Natural Join (SELECT  \***

**FROM TRANSACTION T, AUTHOR A,WRITTENBY W, CUSTOMER C**

**WHERE T.ISBN = B.ISBN AND A. AuthorID = W.AuthorID AND**

**B.ISBN=W.ISBN AND C.CustomerID = T.CustomerID) X**

**on  B.ISBN = X.ISBN**

**WHERE B.Title= “How to be Quarterfinals in S8 of League of Legends”**

**GROUP BY X.Title, X.ISBN**