CSE 3241 Project Checkpoint 04 – Functional Dependencies and Normal Forms

Names: Jeanie Chen, Yuanzhe Li, Zhe Luan, Haoyan Xiang Date: 6/21/2017

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.**

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).

   CUSTOMER_has_ADDRESS: 3NF
   Zipcode -> street
   Email -> Zipcode

   ADDRESS: 3NF
   Zipcode -> street

   CUSTOMER_has_PAYMENT: 3NF
   {Email, credit_card} -> Email
   {Email, credit_card} -> credit_card

   CUSTOMER: 3NF
   Email -> {Phone#, Fname, Lname}

   PAYMENT: 3NF
   credit_card -> {expired_date, security_code, Zipcode}

   PUBLISHER: 3NF
   Name -> Phone#

   REVIEW: 3NF
   R_id -> {rating, R_content, Cmail, ISBN}

   INVENTORY_store_BOOK: 3NF
   {ISBN, in_id} -> amount

   INVENTORY: 3NF
   inventory_id -> {Zipcode, amount,street}

AUTHOR: 3NF
Phone# -> {Fname,Lname,Mname}

AUTHOR_write_BOOK: 3NF
{ISBN,Aphone_num} -> ISBN
{ISBN,Aphone_num} -> Aphone#

BOOK: 3NF
ISBN -> {Pname, title, price, edition, publish_date, sales_amount}

TRANSACTION_HISTORY: 3NF
transaction_id -> {Cmail, date, Cphone#, sold, return_date, Returned, amount}

BOOK_has_TRANSACTION: 3NF
{ISBN,transaction_id} -> ISBN
{ISBN,transaction_id} -> transaction_id

POST_COURSE: 3NF
{prerequisite#,post_course#} -> {prerequisite#,post_course#}

COURSE_use_BOOK: 3NF
{course#,ISBN} -> ISBN
{course#,ISBN} -> course#

BOOKS_belongs_to_CATEGORY: 3NF
{ISBN,category_name} -> {ISBN,category_name}

CATEGORY:
name

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.

   They are all in 3NF.

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 in 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.

(1) The list of number of courses every customer might take in the next semester:

```
CREATE VIEW FutureCourseAmount (AccountEmail, NumPostCourse)

AS SELECT Email, count(Post_course_num)

FROM  CUSTOMER, TRANSACTION_HISTORY, BOOK_has_TRANSACTION, COURSE_use_BOOK, POST_COURSE

WHERE CUSTOMER.Email = TRANSACTION_HISTORY.Cmail and TRANSACTION_HISTORY.transaction_id =
BOOK_has_TRANSACTION.transaction_id and BOOK_has_TRANSACTION.ISBN = COURSE_use_BOOK.ISBN and
COURSE_use_BOOK.Course_num = POST_COURSE.Prerequisite_num

GROUP BY Email;
```

(2) The list of book's title with best average rating:

```
CREATE VIEW BestRatedBook(Title, Rating)

AS SELECT  title, max(Rating)

FROM REVIEW, BOOK

WHERE REVIEW.ISBN = BOOK.ISBN and

        (select title, avg(Rating)

        FROM REVIEW, BOOK

        WHERE REVIEW.ISBN = BOOK.ISBN

        GROUP BY BOOK.ISBN)

GROUP BY BOOK.ISBN;
```