CSE 3241 Project Checkpoint 04 – Functional Dependencies and Normal Forms

Names 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.**

```
CREATE TABLE Book (
        isbn CHAR(10) NOT NULL,
        price INT NOT NULL,
        title VARCHAR(254) NOT NULL,
        pid INT,
        year CHAR(4),
        length INT,

        FOREIGN KEY(pid) REFERENCES Publisher(key),
        PRIMARY KEY(isbn)
);

CREATE TABLE BookDescription (
        isbn CHAR(10) NOT NULL,
        description VARCHAR(4095),

        FOREIGN KEY(isbn) REFERENCES Book(isbn),
        PRIMARY KEY(isbn)
);

CREATE TABLE BookCategory (
        isbn CHAR(10) NOT NULL,
        category VARCHAR(254),

        FOREIGN KEY(isbn) REFERENCES Book(isbn),
        PRIMARY KEY(isbn)
);


CREATE TABLE BookAuthor (
        isbn INT NOT NULL,
        nameid INT NOT NULL,

        FOREIGN KEY(isbn) REFERENCES Book(isbn),
        FOREIGN KEY(nameid) REFERENCES Names(id)

);

CREATE TABLE Customer (
        eid INT NOT NULL,
```

```
        PRIMARY KEY(eid),
        FOREIGN KEY(eid) REFERENCES Entity(id)
);

CREATE TABLE CreditCard (
        cc_num INT NOT NULL,
        eid INT NOT NULL,

        PRIMARY KEY(eid),
        FOREIGN KEY(eid) REFERENCES Entity(id)
);


CREATE TABLE Employee (
        eid INT NOT NULL,
        salary INT,
        wage INT,
        emanager INT,
        title VARCHAR(254),
        start_date DATE,
        wid INT,

        FOREIGN KEY(emanager) REFERENCES Employee(eid),
        FOREIGN KEY(wid) REFERENCES Warehouse(eid),
        FOREIGN KEY(eid) REFERENCES Entity(id),
        PRIMARY KEY(eid)
);

CREATE TABLE Names (
        id INT NOT NULL,
        fname VARCHAR(50) NOT NULL,
        lname VARCHAR(50),
        mname VARCHAR(50),

        PRIMARY KEY(nameid)
);

CREATE TABLE Entity (
        id INT NOT NULL,
        nameid INT NOT NULL,
        phone VARCHAR(15),
        email VARCHAR(254),
        po_box VARCHAR(10),
        city VARCHAR(50),
        state VARCHAR(2),
        postalcode VARCHAR(50),
        country VARCHAR(255)
        address VARCHAR(255),

        PRIMARY KEY(id)
        FOREIGN KEY(nameid) REFERENCES Name(id)
);
```

```
CREATE TABLE Inventory (
        bid INT NOT NULL,
        wid INT NOT NULL,
        quantity INT NOT NULL DEFAULT(0),

        FOREIGN KEY(bid) REFERENCES Book(isbn),
        FOREIGN KEY(wid) REFERENCES Warehouse(eid)
);

CREATE TABLE MonetaryTransaction (
        id INT NOT NULL,
        timestamp TIMESTAMP,
        amount INT NOT NULL DEFAULT(0),
        epayer INT NOT NULL,
        epayee INT NOT NULL,

        FOREIGN KEY(epayer) REFERENCES Entity(id),
        FOREIGN KEY(epayee) REFERENCES Entity(id),
        FOREIGN KEY(oid) REFERENCES CustomerOrder(id),
        PRIMARY KEY(id));

CREATE TABLE OrderContents (
        oid INT NOT NULL,
        bid INT NOT NULL,
        quantity INT NOT NULL DEFAULT(1),

        FOREIGN KEY(bid) references Book(isbn),
        FOREIGN KEY(oid) references CustomerOrder(id),
        FOREIGN KEY(transactionid) references MonetaryTransaction(id),
        PRIMARY KEY (oid, bid)
);

CREATE TABLE CustomerOrder (
        id INT NOT NULL ,
        efrom INT NOT NULL,
        departure TIMESTAMP,
        eto INT NOT NULL,
        arrival TIMESTAMP,
        transactionid NOT NULL,

        FOREIGN KEY(efrom) REFERENCES Entity(id),
        FOREIGN KEY(eto) REFERENCES Entity(id),
        PRIMARY KEY(id)
);

CREATE TABLE Publisher (
        eid INT NOT NULL,
        slogan VARCHAR(254),

        FOREIGN KEY(eid) REFERENCES Entity(id),
        PRIMARY KEY(eid)
);
```

```
CREATE TABLE Rating (
        cid INT NOT NULL,
        bid INT NOT NULL,
        rating INT,

        FOREIGN KEY(bid) REFERENCES Book(isbn),
        FOREIGN KEY(cid) REFERENCES Customer(eid)
);

CREATE TABLE Warehouse (
        eid INT NOT NULL,

        PRIMARY KEY(eid),
        FOREIGN KEY(eid) REFERENCES Entity(id)
);
```

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).

Book: isbn > pid, price, title, year, length
BookDescription: isbn > desciption
BookCategory: isbn > category
CreditCard: cc_num > eid
BookAuthor: isbn > nameid
Employee: eid > salary, wage, emanager, title, start_date, wid
Names: id > fname, mname, lname
Entity: id > nameid, phone, email, po_box, country, city, state, postalcode, address
Inventory: bid, wid > quantity
CustomerOrder: id > efrom, eto, transactionid
MonetaryTransaction: id > timestamp, amount, tax, epayer, epayee
OrderContents: oid > bid, quantity
Publisher: eid > slogan
Rating: cid, bid > rating

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.

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.

   It's in BCNF.

   Entity: Apparently zip code isn't enough to determine city and zip codes aren't strictly contained within cities. You could technically say it determines country but since it's trivial and NULL maps to any country without a zip codes we want to say it's in BCNF.

```
CREATE TABLE book
  (
     isbn   CHAR(10) NOT NULL,
     price  DOUBLE NOT NULL,
     title  VARCHAR(254) NOT NULL,
     pid    INT,
     year   CHAR(4),
     length INT,
     FOREIGN KEY(pid) REFERENCES publisher(KEY),
     PRIMARY KEY(isbn)
  );

CREATE TABLE bookauthor
  (
     isbn   INT NOT NULL,
     nameid INT NOT NULL,
     FOREIGN KEY(isbn) REFERENCES book(isbn),
     FOREIGN KEY(nameid) REFERENCES names(id)
  );

CREATE TABLE bookcategory
  (
     isbn     CHAR(10) NOT NULL,
     category VARCHAR(254),
     FOREIGN KEY(isbn) REFERENCES book(isbn),
     PRIMARY KEY(isbn)
  );

CREATE TABLE bookdescription
  (
     isbn        CHAR(10) NOT NULL,
     description VARCHAR(4095),
     FOREIGN KEY(isbn) REFERENCES book(isbn),
     PRIMARY KEY(isbn)
  );

CREATE TABLE creditcard
```

```
  (
     cc_num INT NOT NULL,
     eid    INT NOT NULL,
     PRIMARY KEY(eid),
     FOREIGN KEY(eid) REFERENCES entity(id)
  );

CREATE TABLE customer
  (
     eid INT NOT NULL,
     PRIMARY KEY(eid),
     FOREIGN KEY(eid) REFERENCES entity(id)
  );

CREATE TABLE customerorder
  (
     id            INT NOT NULL,
     efrom         INT NOT NULL,
     departure     TIMESTAMP,
     eto           INT NOT NULL,
     arrival       TIMESTAMP,
     transactionid NOT NULL,
     FOREIGN KEY(transactionid) REFERENCES monetarytransaction(id),
     FOREIGN KEY(efrom) REFERENCES entity(id),
     FOREIGN KEY(eto) REFERENCES entity(id),
     PRIMARY KEY(id)
  );

CREATE TABLE employee
  (
     eid        INT NOT NULL,
     salary     INT,
     wage       INT,
     emanager   INT,
     title      VARCHAR(254),
     start_date DATE,
     wid        INT,
     FOREIGN KEY(emanager) REFERENCES employee(eid),
     FOREIGN KEY(wid) REFERENCES warehouse(eid),
     FOREIGN KEY(eid) REFERENCES entity(id),
     PRIMARY KEY(eid)
  );

CREATE TABLE entity
  (
     id         INT NOT NULL,
     nameid     INT NOT NULL,
     phone      VARCHAR(15),
```

```
    email      VARCHAR(254),
    po_box     VARCHAR(10),
    city       VARCHAR(50),
    state      VARCHAR(2),
    postalcode VARCHAR(50),
    country    VARCHAR(255),
    address    VARCHAR(255),
    PRIMARY KEY(id)
    FOREIGN KEY(nameid) REFERENCES name(id)
  );

CREATE TABLE inventory
  (
    bid      INT NOT NULL,
    wid      INT NOT NULL,
    quantity INT NOT NULL DEFAULT(0),
    FOREIGN KEY(bid) REFERENCES book(isbn),
    FOREIGN KEY(wid) REFERENCES warehouse(eid)
  );

CREATE TABLE monetarytransaction
  (
    id        INT NOT NULL,
    timestamp TIMESTAMP,
    amount    INT NOT NULL DEFAULT(0),
    epayer    INT NOT NULL,
    epayee    INT NOT NULL,
    FOREIGN KEY(epayer) REFERENCES entity(id),
    FOREIGN KEY(epayee) REFERENCES entity(id),
    PRIMARY KEY(id)
  );

CREATE TABLE names
  (
    id    INT NOT NULL,
    fname VARCHAR(50) NOT NULL,
    lname VARCHAR(50),
    mname VARCHAR(50),
    PRIMARY KEY(id)
  );

CREATE TABLE ordercontents
  (
    oid      INT NOT NULL,
    bid      INT NOT NULL,
    quantity INT NOT NULL DEFAULT(1),
    FOREIGN KEY(bid) REFERENCES book(isbn),
    FOREIGN KEY(oid) REFERENCES customerorder(id),
```

```
    PRIMARY KEY (oid, bid)
  );

CREATE TABLE publisher
  (
    eid    INT NOT NULL,
    slogan VARCHAR(254),
    FOREIGN KEY(eid) REFERENCES entity(id),
    PRIMARY KEY(eid)
  );

CREATE TABLE rating
  (
    cid    INT NOT NULL,
    bid    INT NOT NULL,
    rating INT,
    FOREIGN KEY(bid) REFERENCES book(isbn),
    FOREIGN KEY(cid) REFERENCES customer(eid)
    PRIMARY KEY(cid, bid)
  );

CREATE TABLE warehouse
  (
    eid INT NOT NULL,
    PRIMARY KEY(eid),
    FOREIGN KEY(eid) REFERENCES entity(id)
  );
```

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.

Total spent by every customer.

```
CREATE VIEW CustomerTotalSpent AS
     SELECT N.id, N.fname, N.lname, Sum(M.amount)
       FROM Names N, Customer C, MonetaryTransaction M
      WHERE N.id = C.eid
        AND M.epayer = C.eid
   GROUP BY N.id
```

Author profit

```
CREATE VIEW AuthorProfitQuantity AS
      SELECT A.fname, A.mname, A.Lname, P.profit, T.numberSold
        FROM Names A,
             (SELECT O.bid, Sum(O.quantity) AS numberSold
              FROM OrderContents O
              GROUP BY O.bid) T,
            (SELECT T.Sum * B.price AS profit
                 From
                       (SELECT O.bid AS isbn, Sum(O.quantity) AS Sum
                       FROM OrderContents O
                       GROUP BY O.bid) T,
                       Book B
                 Where B.isbn = T.isbn
            ) P,
            BookAuthor BA
      WHERE BA.nameid = A.id
```