1. (5 marks) Consider the following bank database. Please specify the primary keys and foreign keys according to our convention.

BRANCH(branch_name, branch_city, assets)
Primary Key(s): branch_name
Foreign Key(s): N/A

CUSTOMER(customer_name, customer_street, customer_city)
Primary Key(s): customer_name
Foreign Key(s): N/A

LOAN(loan_number, branch_name, amount)
Primary Key(s): loan_number
Foreign Key(s): branch_name

BORROWER(customer_name, loan_number)
Primary Key(s): customer_name, loan_number
Foreign Key(s): customer_name, loan_number

ACCOUNT(account_number, branch_name, balance)
Primary Key(s): account_number
Foreign Key(s): branch_name

DEPOSITOR(customer_name, account_number)
Primary Key(s): customer_name, account_number
Foreign Key(s): customer_name, account_number

We assume that customer_name is unique. We allow customers to have more than one account, and more than one loan.

2. (7 marks) Normalize the following relations to BCNF and 4NF. Please list each table split and clearly indicate its reasons. List the final set of relations at the end. Each relation in the final set should be verified. Indicate the primary keys and foreign keys in all the steps.

BOOK(accession_no, isbn, title, author, publisher)
USER(user_id, name, dept_id, dept_name)

Suppose the following dependencies hold:
accession_no → isbn
isbn → title
isbn → publisher
isbn →→ author
user_id → name
user_id → dept_id
dept_id → dept_name

BCNF:
USER relation can be split into two relations, USER and DEPARTMENT:

- USER(user_id, name, dept_id)
    - **Primary Key(s):** user_id
    - **Foreign Key(s):** dept_id
- DEPARTMENT(dept_id, dept_name)
    - **Primary Key(s):** dept_id
    - **Foreign Key(s):** N/A

Reasoning: the user_id determines the name and dept_id of a user, and the dept_id determines the dept_name therefore they should each be there own relation

BOOK relation can be split into two relations, BOOK and ISBN:

- BOOK(accession_no, isbn)
    - **Primary Key(s):** accession_no
    - **Foreign Key(s):** isbn
- ISBN(isbn, title, author, publisher)
    - **Primary Key(s):** isbn
    - **Foreign Key(s):** N/A

Reasoning: the accession_no of a book determines the isbn of the book, and the isbn determines title publisher and author

Final set of relations:
BOOK(accession_no, isbn) **Primary Key(s):** accession_no **Foreign Key(s):** isbn
ISBN(isbn, title, author, publisher) **Primary Key(s):** isbn **Foreign Key(s):** N/A
USER(user_id, name, dept_id) **Primary Key(s):** user_id **Foreign Key(s):** dept_id
DEPARTMENT(dept_id, dept_name) **Primary Key(s):** dept_id **Foreign Key(s):** N/A

4NF:

BOOK(accession_no, isbn) **Primary Key(s):** accession_no **Foreign Key(s):** isbn
ISBN(isbn, title, author, publisher) **Primary Key(s):** isbn **Foreign Key(s):** N/A
USER(user_id, name, dept_id) **Primary Key(s):** user_id **Foreign Key(s):** dept_id
DEPARTMENT(dept_id, dept_name) **Primary Key(s):** dept_id **Foreign Key(s):** N/A

Currently we have the above relation however we are missing the multivalued dependency:
isbn →→ author. Thus we are not in 4NF. To make the set of relations 4NF we must add the new relation:

- AUTHOR(isbn, author)
    - **Primary Key(s):** isbn, author
    - **Foreign Key(s):** isbn

We also need to remove author from ISBN since we created its own relation:

- ISBN(isbn, title, publisher)
    - **Primary Key(s):** isbn
    - **Foreign Key(s):** N/A

Final set of relations:

BOOK(accession_no, isbn) **Primary Key(s):** accession_no **Foreign Key(s):** isbn
ISBN(isbn, title, publisher) **Primary Key(s):** isbn **Foreign Key(s):** N/A
USER(user_id, name, dept_id) **Primary Key(s):** user_id **Foreign Key(s):** dept_id
DEPARTMENT(dept_id, dept_name) **Primary Key(s):** dept_id **Foreign Key(s):** N/A
AUTHOR(isbn, author) **Primary Key(s):** isbn, author **Foreign Key(s):** isbn

3. (8 marks) Normalize the following relation to BCNF and 4NF. Please list each table split and clearly indicate its reasons. List the final set of relations at the end. Each relation in the final set should be verified. Indicate the primary keys and foreign keys in all the steps.

INVOICE(CustomerNumber, FirstName, LastName, Phone, InvoiceNumber, DateIn, DateOut, ItemType, Quantity, ItemPrice, ExtendedPrice, SpecialInstructions)

Suppose the following dependencies and assumptions hold:
(1) CustomerNumber → (FirstName, LastName)
(2) CustomerNumber →→ Phone
(3) One customer could have many invoices, but each invoice is associated with only one customer.
(4) One invoice may have many ItemTypes, but each ItemType may occur only once in any invoice.
(5) Special instructions are associated with each item type, since different instructions may be given for different items.
(6) Order number is a number assigned to the orders themselves, without association with any particular customer. Thus we have such numbers as “123454”, 123455”, etc, rather than “Customer 101, Order 1”, Customer 101, Order 2”, Customer 102, Order 1”, etc.
(7) ItemPrice can vary with order so that ItemType cannot determine ItemPrice.

BCNF:
INVOICE can be split into CUSTOMER and INVOICE:

- CUSTOMER(CustomerNumber, FirstName, LastName, Phone)
    - **Primary Key(s):** CustomerNumber
    - **Foreign Key(s):** N/A
- INVOICE(CustomerNumber, InvoiceNumber, DateIn, DateOut, ItemType, Quantity, ItemPrice, ExtendedPrice, SpecialInstructions)
    - **Primary Key(s):** InvoiceNumber
    - **Foreign Key(s):** CustomerNumber

Reasoning: since CustomerNumber determines FirstName, LastName, and Phone we can make it its own relation

INVOICE can be split again into ITEM and INVOICE:

- ITEM(ItemType, SpecialInstructions)
    - **Primary Key(s):** ItemType
    - **Foreign Key(s):** N/A
- INVOICE(CustomerNumber, InvoiceNumber, DateIn, DateOut, ItemType, Quantity, ItemPrice, ExtendedPrice)
    - **Primary Key(s):** InvoiceNumber
    - **Foreign Key(s):** CustomerNumber, ItemType

Reasoning: ItemType determines the special instructions for that item so it can be made into its own relation

Final Set of Relations

CUSTOMER(CustomerNumber, FirstName, LastName, Phone)

- **Primary Key(s):** CustomerNumber
- **Foreign Key(s):** N/A

ITEM(ItemType, SpecialInstructions)

- **Primary Key(s):** ItemType
- **Foreign Key(s):** N/A

INVOICE(CustomerNumber, InvoiceNumber, DateIn, DateOut, ItemType, Quantity, ItemPrice, ExtendedPrice)

- **Primary Key(s):** InvoiceNumber
- **Foreign Key(s):** CustomerNumber, ItemType

4NF:

Currently we have the above relation however we are missing the multivalued dependencies:

- CustomerNumber →→ Phone
- One invoice may have many ItemTypes, but each ItemType may occur only once in any invoice.
- One customer could have many invoices, but each invoice is associated with only one customer.

Thus we are not in 4NF. To make the set of relations 4NF we must add three new relations:

- PHONE(CustomerNumber, Phone)
  - **Primary Key(s):** CustomerNumber, Phone
  - **Foreign Key(s):** CustomerNumber

- INVOICED_ITEM(InvoiceNumber, ItemType, Quantity, ItemPrice, ExtendedPrice)
  - **Primary Key(s):** InvoiceNumber, ItemType
  - **Foreign Key(s):** InvoiceNumber, ItemType

- CUSTOMER_INVOICE(InvoiceNumber, CustomerNumber)
  - **Primary Key(s):** InvoiceNumber, CustomerNumber
  - **Foreign Key(s):** InvoiceNumber, CustomerNumber

We also need to remove Phone from CUSTOMER since we made its own relation:

- CUSTOMER(CustomerNumber, FirstName, LastName)
  - **Primary Key(s):** CustomerNumber
  - **Foreign Key(s):** N/A

We also need to remove CustomerNumber, ItemType, Quantity, ItemPrice, and ExtendedPrice from INVOICE since we moved them to their own relations:

- INVOICE(InvoiceNumber, DateIn, DateOut)
  - **Primary Key(s):** InvoiceNumber
  - **Foreign Key(s):** N/A

Final Set of Relations

CUSTOMER(CustomerNumber, FirstName, LastName)

- **Primary Key(s):** CustomerNumber, **Foreign Key(s):** N/A

PHONE(CustomerNumber, Phone)

- **Primary Key(s):** CustomerNumber, Phone, **Foreign Key(s):** CustomerNumber

ITEM(ItemType, SpecialInstructions)

- **Primary Key(s):** ItemType, **Foreign Key(s):** N/A

INVOICE(InvoiceNumber, DateIn, DateOut)

- **Primary Key(s):** InvoiceNumber, **Foreign Key(s):** N/A

INVOICED_ITEM(InvoiceNumber, ItemType, Quantity, ItemPrice, ExtendedPrice)

- **Primary Key(s):** InvoiceNumber, ItemType, **Foreign Key(s):** InvoiceNumber, ItemType

CUSTOMER_INVOICE(InvoiceNumber, CustomerNumber)

- **Primary Key(s):** InvoiceNumber, CustomerNumber, **Foreign Key(s):** InvoiceNumber, CustomerNumber