Task 3 (4.0 marks)
Consider the following un-normalized relational schema holding information about a shopping basket submitted by a customer.

shoppingBasket(custNum, custName, custAddress, custContact, ordNum, ordDate, ordLineNum, itemDesc, quantity, totItemPrice, deliveryPerson, deliveryPersonContact, expectedDeliveryDate, creditCardNum, totOrdPrice)

In an online application, a customer may submit many orders online. Each order may contain multiple items. Once the orders are confirmed, checked out, and payment made, the respective items in the orders are then assigned to delivery personnel to deliver to the customers. Customer can make the payment with any of his/her credit cards.

The attributes of the relational schema shoppingBasket satisfy the following properties:

- $custNum \rightarrow custName, custAddress, custContact$
- $ordNum \rightarrow ordDate, totOrdPrice$
- $deliveryPerson \rightarrow deliveryPersonContact$
- $ordNum, ordLineNum \rightarrow itemDesc, quantity, totItemPrice$
- $ordNum, ordLineNum, custNum \rightarrow deliveryPerson, expectedDeliveryDate$
- $custNum, ordNum \rightarrow creditCardNum$
- $creditCardNum \rightarrow custNum$

i. Find all minimal-super-keys in the relational schema shoppingBasket. List the derivations of all minimal keys.
ii. Find the highest normal form the relational schema shoppingBasket. Provide justification for your answer.
iii. Decompose the relational schema shoppingBasket into a minimal number of relational schemas in BCNF. List all relational schemas obtained from the decompositions.

Given:
Relational Schema:
shoppingBasket(custNum, custName, custAddress, custContact, ordNum, ordDate, ordLineNum, itemDesc, quantity, totItemPrice, deliveryPerson, deliveryPersonContact, expectedDeliveryDate, creditCardNum, totOrdPrice)

Functional Dependencies (FDs):

1. custNum → custName, custAddress, custContact
2. ordNum → ordDate, totOrdPrice
3. deliveryPerson → deliveryPersonContact
4. ordNum, ordLineNum → itemDesc, quantity, totItemPrice
5. ordNum, ordLineNum, custNum → deliveryPerson, expectedDeliveryDate

6. custNum, ordNum → creditCardNum
7. creditCardNum → custNum

(i) Find all minimal-superkeys in the relational schema shoppingBasket. List the derivations of all minimal keys.

Key candidate: {custNum, ordNum, ordLineNum}
Closure of {custNum, ordNum, ordLineNum}:
custNum → custName, custAddress, custContact
ordNum → ordDate, totOrdPrice
ordNum, ordLineNum → itemDesc, quantity, totItemPrice
ordNum, ordLineNum, custNum → deliveryPerson, expectedDeliveryDate
deliveryPerson → deliveryPersonContact
custNum, ordNum → creditCardNum
creditCardNum → custNum (custNum already in closure)

$\{custNum, ordNum, ordLineNum\}^+$ = all attributes in the schema
Hence, {custNum, ordNum, ordLineNum} is a candidate key (minimal superkey).

check if any proper subset of it is a superkey:
custNum, ordNum → not enough (cannot derive itemDesc, quantity, etc.)
ordNum, ordLineNum → no way to get customer/card details
Others similarly fail

Therefore, the only minimal key is {custNum, ordNum, ordLineNum}.

(ii) Find the highest normal form of the relational schema shoppingBasket. Provide justification for your answer.

Assumption: The schema is in 1NF (atomic values only).

2NF Check:
The candidate key is {custNum, ordNum, ordLineNum}.
All non-prime attributes must be fully dependent on the whole key.

Check for partial dependencies:
custNum → custName, custAddress, custContact (partial dependency)

ordNum → ordDate, totOrdPrice

custNum, ordNum → creditCardNum

Partial dependencies found ⇒ NOT in 2NF

Highest normal form: 1NF

(iii) Decompose the schema into BCNF. List all resulting schemas.

We'll decompose using the violating FDs from part (ii).

Step 1: Decompose on custNum → custName, custAddress, custContact

R1(custNum, custName, custAddress, custContact)

R2 will have remaining attributes:
shoppingBasketR2(custNum, ordNum, ordLineNum, ordDate, itemDesc, quantity, totItemPrice, deliveryPerson, deliveryPersonContact, expectedDeliveryDate, creditCardNum, totOrdPrice)

Step 2: Decompose R2 on ordNum → ordDate, totOrdPrice

R3(ordNum, ordDate, totOrdPrice)

R4 remains:
R4(custNum, ordNum, ordLineNum, itemDesc, quantity, totItemPrice, deliveryPerson, deliveryPersonContact, expectedDeliveryDate, creditCardNum)

Step 3: Decompose R4 on deliveryPerson → deliveryPersonContact

R5(deliveryPerson, deliveryPersonContact)

R6 remains:
R6(custNum, ordNum, ordLineNum, itemDesc, quantity, totItemPrice, deliveryPerson, expectedDeliveryDate, creditCardNum)

Step 4: Decompose R6 on custNum, ordNum → creditCardNum

R7(custNum, ordNum, creditCardNum)

R8 remains:
R8(custNum, ordNum, ordLineNum, itemDesc, quantity, totItemPrice, deliveryPerson, expectedDeliveryDate)

Final BCNF Schemas:

1. R1(custNum, custName, custAddress, custContact)
2. R3(ordNum, ordDate, totOrdPrice)
3. R5(deliveryPerson, deliveryPersonContact)
4. R7(custNum, ordNum, creditCardNum)
5. R8(custNum, ordNum, ordLineNum, itemDesc, quantity, totItemPrice, deliveryPerson, expectedDeliveryDate)

All schemas are now in BCNF (Left-hand side of every FD is a superkey in its relation).