UCS 2411 DATABASE LAB

PHARMACY SUPPLY MANAGEMENT SYSTEM

SUBMITTED BY
SRI HARI VK (3122225001138)
R SKANTA SAMVARTAN (3122225001137)
SHRI VISHAAL (3122225001131)

Department of Computer Science and Engineering

Sri Sivasubramaniya Nadar College of Engineering
(An Autonomous Institution, Affiliated to Anna University)
Kalavakkam – 603110

2023 – 2024

INDEX:

Insights of the Project:

The project is pharmaceutical management system that tracks drugs from suppliers to pharmacies and customers. The system includes inventory management, order processing, supplier and manufacturer details, and customer information.

Tables and Their Attributes

1. Supplier

-Attributes:

- `supplier_id`: Unique identifier for the supplier.
- `supplier_name`: Name of the supplier.
- `phone_no`: Contact number of the supplier.
- `address`: Physical address of the supplier.
- `email`: Email address of the supplier.

-Relationships:

- Supplies drugs to the `Pharmacy`.

2. Pharmacy

-Attributes:

- `pharmacy_id`: Unique identifier for the pharmacy.
- `pharmacy_name`: Name of the pharmacy.
- `pharmacy_type`: Type of pharmacy (e.g., retail, hospital).
- `license_no`: License number of the pharmacy.
- `open_hours`: Operating hours of the pharmacy.
- `phone`: Contact number of the pharmacy.
- `address`: Physical address of the pharmacy.
- `administration_status`: Status of the administration (possibly regulatory compliance status).

-**Relationships:**

- Receives supplies from `Supplier`.

3. Inventory

- Attributes:

- `inventory_id`: Unique identifier for the inventory entry.
- `location`: Storage location within the inventory.
- `batch_number`: Batch number of the stored drugs.
- `invoice_date`: Date of the invoice for the inventory entry.
- `expiry_date`: Expiry date of the stored drugs.
- `quantity`: Quantity of the drugs in the inventory.
- `status`: Current status of the inventory entry.

-Relationships:

- Gets drugs from `Supplier`.
- Stores `Drug`.

4. Drug

-Attributes:

- \`drug_id\`: Unique identifier for the drug.
- \`drug_name\`: Name of the drug.
- \`drug_code\`: Code associated with the drug.
- \`storage_conditions\`: Conditions required for storing the drug.
- \`drug_category\`: Category or type of the drug.
- \`composition\`: Chemical composition of the drug.
- \`dosage\`: Dosage information for the drug.
- \`unit_price\`: Price per unit of the drug.

-Relationships:

- Stored in \`Inventory\`.
- Manufactured by \`Manufacturer\`.

5. Manufacturer

-Attributes:

- \`manufacturer_id\`: Unique identifier for the manufacturer.
- \`manufacturer_name\`: Name of the manufacturer.
- \`phone_no\`: Contact number of the manufacturer.
- \`address\`: Physical address of the manufacturer.
- \`email\`: Email address of the manufacturer.

-Relationships:

- Manufactures \`Drug\`.

6. Customer

-Attributes:

- \`cust_id\`: Unique identifier for the customer.
- \`cust_name\`: Name of the customer.
- \`phone\`: Contact number of the customer.
- \`email\`: Email address of the customer.
- \`gender\`: Gender of the customer.
- \`dob\`: Date of birth of the customer.
- \`address\`: Physical address of the customer.

-Relationships:

- Places \`Order\`.

7. Order

-Attributes:

- \`order_id\`: Unique identifier for the order.
- \`order_date\`: Date the order was placed.
- \`quantity\`: Quantity of drugs ordered.
- \`tot_amt\`: Total amount of the order.
- \`delivery_date\`: Date the order was delivered.

-Relationships:

- Contains \`Drug\`.
- Placed by \`Customer\`.

8. Payment

- **Attributes:**
  - `pay_id`: Unique identifier for the payment.
  - `pay_amt`: Amount paid.
  - `pay_date`: Date of the payment.
- **Relationships:**
  - Associated with `Customer`.

Relationships:

- Supplier to Inventory: Suppliers provide drugs to be stored in the inventory.
- Inventory to Drug: Inventory entries store specific drugs.
- Drug to Manufacturer: Drugs are produced by manufacturers.
- Customer to Order: Customers place orders for drugs.
- Order to Drug: Orders include specific drugs.
- Customer to Payment: Payments are made by customers.

ASSUMPTIONS:

Customer Registration:

New customers are added to the CUSTOMER table with unique USER_NAME.
Passwords are securely stored (assumed hashed and not plain text).
Placing Orders:

When a customer places an order, a record is created in the ORDERS table with a foreign key reference to USER_NAME.
Each order contains multiple entries in the ORDER_LIST table, detailing the DRUG_ID and quantity ordered.
Order History:

The system can query the ORDERS table to retrieve all orders placed by a specific customer using USER_NAME.
The ORDER_LIST table can be queried to find out all drugs ordered within a particular order.

Manufacturer and Drug:

Each drug (DRUG_ID) is produced by a single manufacturer (MANUFACTURER_ID).
The relationship ensures traceability and accountability for the drug's production.
Drug and Inventory:

Each drug can have multiple inventory records, indicating different batches or stocks.
This allows tracking of stock levels, expiry dates, and inventory locations.
Drug and Pharmacy:

The single pharmacy maintains a record for each drug, indicating the supplied quantity.
This relationship simplifies the tracking of drug availability in the pharmacy.
Drug and Orders:

Each drug can be ordered multiple times by different customers.
The ORDER_LIST table captures these details, linking each drug to its respective orders and quantities.

IDENTIFICATION OF FD's:

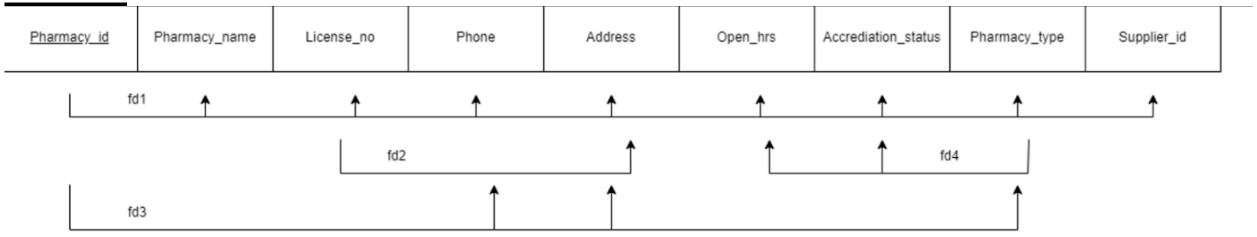
1)Pharmacy:

(A-Pharmacy_id,B → Pharmacy_name,C → License_no,D → Phone,E → address,F → open_hrs,
G → accreditation_status,H → Pharmacy_type,I → Supplier_id)
Pharmacy_id
→Pharmacy_name,license_no,Phone,address,open_hrs,accreditation_status,

Pharmacy_type.
License_no → address
Pharmacy_type → accrediation_status
Pharmacy_type → open_hrs
Pharmacy_id,Pharmacy_name → phone,Pharmarcy_type,address

- A → B,C,D,E,F,G,H,I
- C → E
- H → F,G
- A,B → D,H,E

Schema:



MINIMAL FDs:

1)

a)

With A → B
A+={A,B,C,D,H,E,F,G,I}
Without A → B
A+={A}
Therefore A→B is not redundant

b)

With A→C
A+={A,C,E}
Without A→C
A+={A}
Therefore A→C is not redundant

c)

With A→D
A+={A,B,C,D,E,F,G,H,I}
Without A→D
A+={A,B,C,E,F,G,H,I}
Therefore A→C is not redundant

d)

With A→E
A+={A,B,C,D,E,F,G,H,I}
Without A→E
A+={A,B,C,D,E,F,G,H,I}
Therefore the fd is not redundant

e)

With C → E
C={C,E}
Without C → E
C+={C}
Therefore C → E is not redundant

f)

With H → F,G
H+={H,F,G}
Without H → F,G
H+={H}
Therefore H → F,G is not redundant

g)

With A,B → D,H,E
AB+={A,B,D,H,E,C,F,G,I}
Without A,B → D,H,E
AB+={A,B,C,D,E,F,G,H,I}
Therefore A,B → D,H,E is redundant

- The minimal FDs are

{
A → B,C,D,H,I
C → E
H → F,G
}

- **The candidate key is {A}**
- **PA={A}**
- **NPA={B,C,D,E,F,G,H,I}**

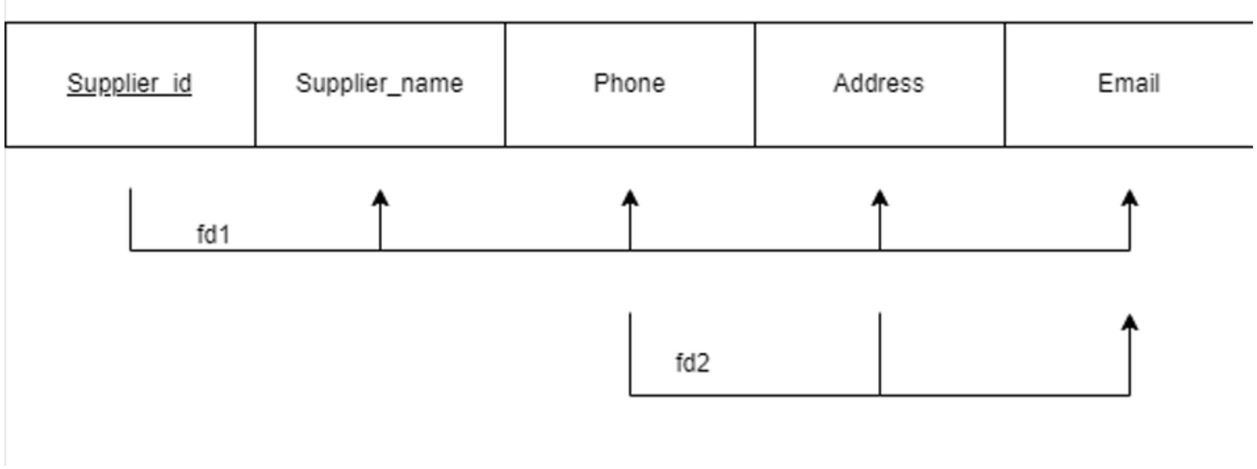
<u>2)Supplier:</u>

A → Supplier_id,B → Supplier_name,C → phone,D → address,E → email

<u>FDs:</u>

- A → B,C,D,E
- C,E → D

<u>Schema:</u>



Minimal FDs:

i)Decomposition:

- A → B
- A → C
- A → D
- A → E

I)A → B

With A → B
A+={A,B,C,D,E}
Without A → B
A+={A,C,D,E}
Therefore A → B Is not redundant

ii)A → C

With A → C
A+={A,B,C,D,E}
Without A → C
A+={A,B,D,E}
Therefore A → C is not redundant

iii)A → D

With A → D
A+={A,B,C,D,E}
Without A → D
A+={A,B,D,C,E}
Therefore A → D is redundant

IV)A → E

With A → E
A+={A,B,C,D,E}
Without A → E
A+={A,B,C,D}
Therefore A → E is not redundant

2)

C,E → D

With C,E → D
CE+={C,E,D}
Without C,E → D
CE+={C,E}
Therefore C,E → D is not redundant

MINIMAL FDs:

A → B,C,E
C,E → D

The candidate key for relation is {A}
PA={A}
NPA={B,C,D,E}

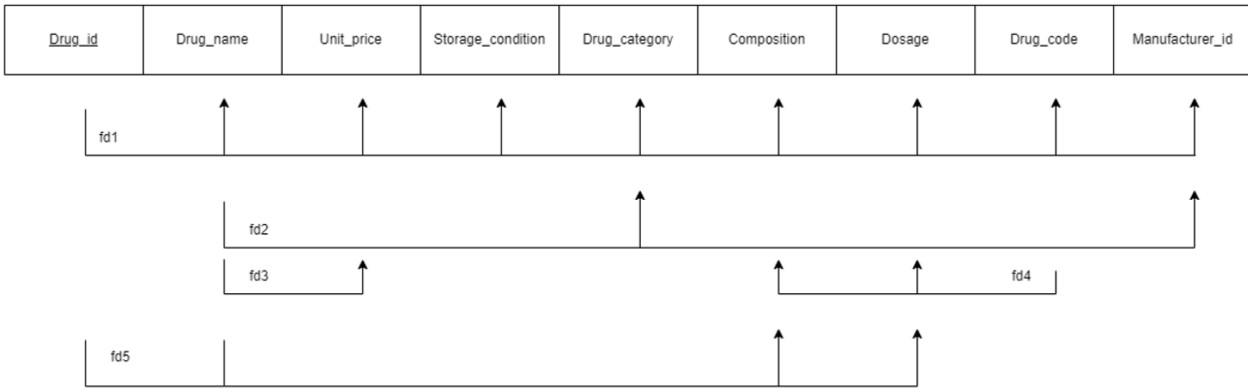
3)DRUG:

A-Drug_id,B → Drug_name,C → Unit_price,D → Storage_condition,E →Drug_category, F → Composition,G → Dosage,H → Drug_code,I → manufacturer_id

FDs:

- A → B,C,D,E,F,G,H,I
- B → E,I
- H → G,F
- B → C
- A,B → G,F

Schema:



MINIMAL FDs:

i)Decomposition:

A → B

A → C

A → D

A → E

A → F

A → G

A → H

A → I

B → E

B → I

H →G

H → F

B → C

1)

a)

With A → B

A+={A,B,C,D,E,F,G,H,I}

Without A → B

A+={A,C,D,E,F,G,H,I}

A → B is not redundant

b)

With A → C

A+={A,B,C,D,E,F,G,H,I}

Without A → C

A+={A,B,C,D,E,F,G,H,I}

A → C is redundant

c)

With A → D

A+={A,B,C,D,E,F,G,H,I}

Without A → D

A+={A,B,C,E,F,G,H,I}

A → D is not redundant

d)

With A → E

A+={A,B,C,D,E,F,G,H,I}

Without A → E

A+={A,B,C,D,E,F,G,H,I}

A → E is redundant

e)

With A → F
A+={A,B,C,D,E,F,G,H,I}
Without A → F
A+={A,B,C,D,E,F,G,H,I}
A → F is redundant

f)

With A → G
A+={A,B,C,D,E,F,G,H,I}
Without A → G
A+={A,B,C,D,E,F,G,H,I}
A → G is redundant

g)

With A → H
A+={A,B,C,D,E,F,G,H,I}

Without A → H
A+={A,B,C,D,E,F,G,I}
A → H is not redundant

g1)

With A → I
A+={A,B,C,D,E,F,G,H,I}
Without A → I
A+={A,B,C,D,E,F,G,H,I}
A → I is redundant

h)

With B → E
B+={B,E,C,I}
Without B → E
B+={B, C,I}
Therefore B → E is not redundant

i)

With B → C
B+={B,C,E,I}
Without B → C
B+={B,E,I}
Therefore B → C is not redundant

i1)

With B → I
B+={B,C,E,I}
Without B → I
B+={B,C,E}
Therefore B → I is not redundant

j)

With H → G
H+={H,G,F}
Without H → G
H+={H,F}
Therefore H → G is not redundant

k)

With H → F
H+={H,G,F}
Without H → F
H+={H,G}
Therefore H → F is not redundant

MINIMAL FDs:

- A → B,D,H
- H → F,G
- B → C,E,I

Candidate key of the relation is {A}
Therefore

- **PA={A}**
- **NPA={B,C,D,E,F,G,H,I}**

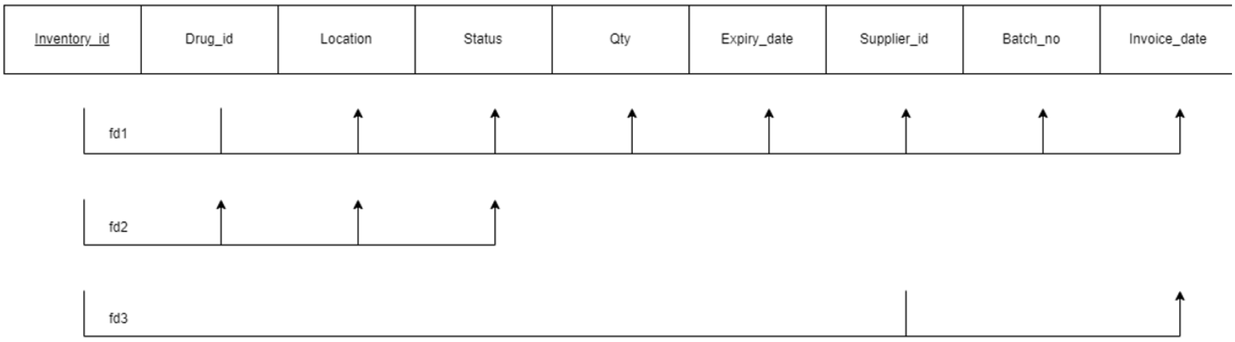
4)Inventory:

A-Inventory_id
B-Batch_no
C-Location
D-Status
E-Qty
F-Expiry_date
G-Supplier_id
H-Drug_id
I-Invoice_date

FDs:

- A,H → B,C,D,E,F,G,I
- A → C,D,B
- A,G → I

Schema:



Decomposing:

A,H → B
A,H → C
A,H → D
A,H → E
A,H → F
A,H → G
A,H → I
A→C
A,G → I

Minimal FDs:

a)A,H → B

With A,H → B
AH+={A,B,C,D,E,F,G,H,I}
Without AH → B
AH+={A,B,C,D,E,F,G,H,I}
Therefore the above FD is redundant

b)

With AH → C
AH+={A,B,C,D,E,F,G,H,I}
Without AH → C
AH+={A,B,C,D,E,F,G,H,I}
Therefore the above FD is redundant

c)

With AH → D
AD+={A,B,C,D,E,F,G,H,I}
Without AH → D
AD+={A,B,C,D,E,F,G,H,I}
Therefore the above FD is redundant

d)

With AH → E
AH+={A,B,C,D,E,F,G,H,I}
Without AH → E
AH+={A,B,C,D,F,G,H,I}
Therefore the above FD is not redundant

e)

With AH → F
AH+={A,B,C,D,E,F,G,H,I}
Without AH → F
AH+={A,B,C,D,E,G,H,I}
Therefore the above fd is not redundant

f)

With A,H → G
AH+={A,B,C,D,E,F,G,H,I}
Without AH → G
AH+={A,B,C,D,E,F,H,I}
Therefore the above fd is not redundant

g)

With AH → I
AH+={A,B,C,D,E,F,G,H,I}
Without AH → I
AH+={A,B,C,D,E,F,G,H,I}
Therefore the above fd is redundant

h)

With A,G → I
AG+={A,G,I}
Without A,G → I
AG+={A,G}

Therefore the above fd is not redundant

MINIMAL FDs:

- AH → EFG
- A→B,C,D
- AG → I

The candidate key for the relation is AH.
PA={A,H}
NPA={B,C,D,E,F,G,I}

5)CUSTOMER TABLE :

ATTRIBUTES:

A : CUST_ID
B : CUST_NAME
C : EMAIL
D : ADDRESS
E : PHONE
F : PAY_ID
G : PAY_AMT
H : PAY_DATE

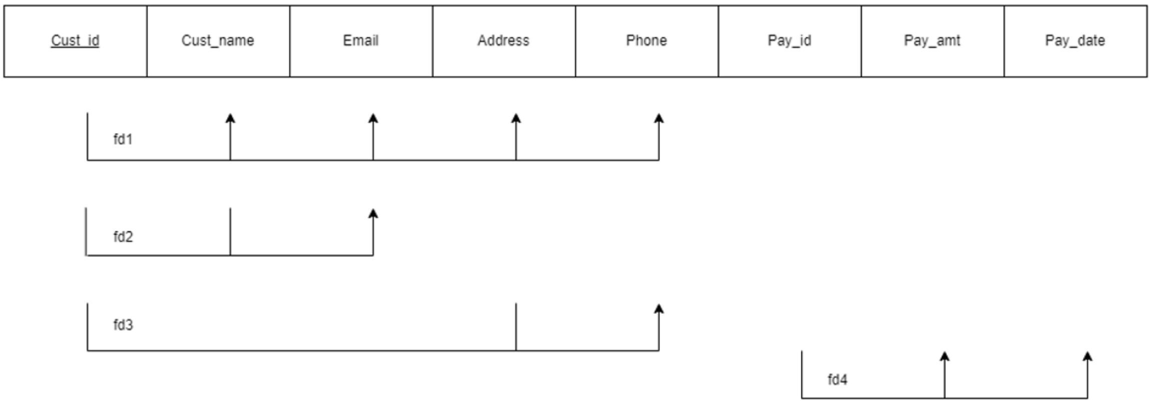
FUNCTIONALS DEPENDENCIES:

A → B , C , D , E
A , B → C
A , B → C
A , B → B , C
A , D → E
F → G , H

Minimal FD’s:

REMOVING TRIVIAL FD’s :
A , B →B, C

Schema:



AFTER DECOMPOSING :
A → B
A → C
A → D
A → E

A , B → C
A , D → E
F → G
F → H

CHECKING FOR EXTRANEOUS :
1) A , B → C
TAKING B AS EXTRANEOUS :
A → C
{A}+ = { A , B , C ,D ,E}
TAKING A AS EXTRANEOUS :
B → C
{B}+ = {B ,C}
SO , A→; C IS THE MINIMAL FD.

2) A , D → E
TAKING A AS EXTRANEOUS :
A → E
{A}+ = { A , B , C ,D ,E}
TAKING A AS EXTRANEOUS :
D → E
{D}+ = {D , E}
SO , A→ E IS THE MINIMAL FD.

NON - REDUNDANT AND MINIMAL FD’s :
{
A → B
A → C
A → D
A → E
F → G
F → H
}

FINDING CANDIDATE KEY :

{A}+ = {A ,B , C , D , E}
{F}+ = {F , G ,H}
{A, F} = {A ,B ,C ,D , E, F , G, H}
SO ,THEREFORE {A , F} IS THE CANDIDATE KEY .
PA = {A , F}
NPA = {B,C,D,E,G,H}

6)MANUFACTURERS:

Attributes:

A: MANUFACTURER_ID

B: MANUFACTURER_NAME

C: ADDRESS

D: EMAIL

E: PHONE_NO

F: INVENTORY_ID

Functional Dependencies:

Given the attributes, the likely functional dependencies can be inferred as follows:

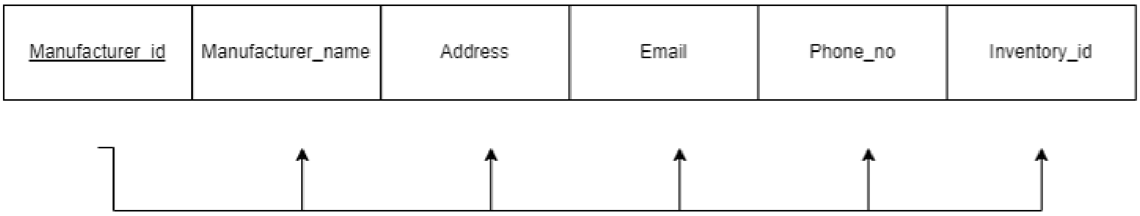
A → B, C, D, E, F

Minimal FD’s:

A → B
A → C
A → D
A → E
A → F

Since the given dependency is already minimal and there are no composite keys, we can proceed to normalization.

Schema:



Manufacturer_id	Manufacturer_name	Address	Email	Phone_no	Inventory_id

Candidate Key:

{A} (MANUFACTURER_ID)

Prime Attributes:

{A}

Non-Prime Attributes:

- {B, C, D, E, F}

7)ORDERS:

Let the attributes be:

A: ORDER_ID

B: ORDER_DATE

C: DELV_DATE

D: QUANTITY

E: TOT_AMT

F: CUST_ID

G: DRUG_ID

Functional Dependencies:

AG->DE

AF->BC

A->BCF

Minimal FD’s:

=>Splitting the FD’s:

A->B

A->C

A->F

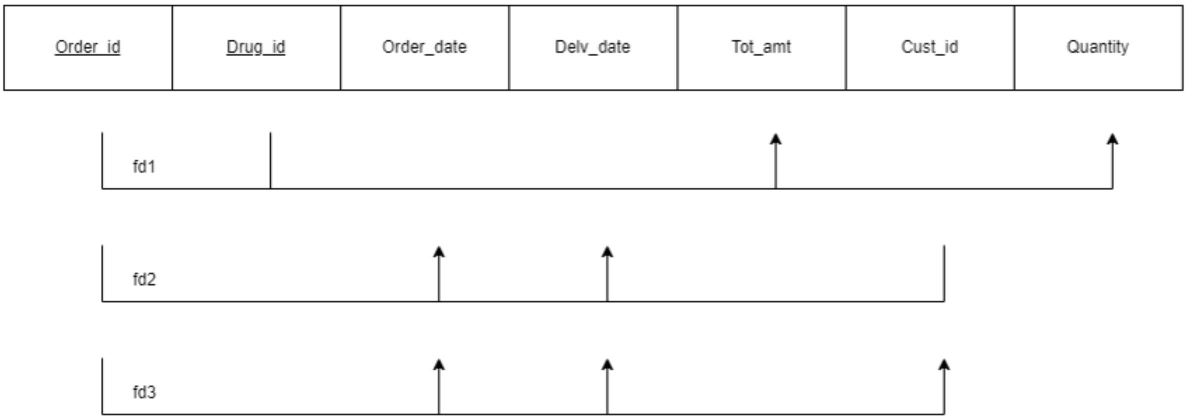
AF->B

AF->C

AG->D

AG->E

Schema:



=>Removing Extraneous Attributes:

Extraneous Attributes are:

AF->B, AF->C

Since,

A->B , A->C

=>Checking For Redundancy :

a) A->B

With A->B:

A+={B,C,D,E,F,G}

Without A->B:

A+={C,D,E,F,G}

A->B IS NOT REDUNDANT.

b) A->C

With A->C:

A+={B,C,D,E,F,G}

Without A->C:

A+={B,D,E,F,G}

A->C IS NOT REDUNDANT.

c) A->F

With A->F:

A+={B,C,D,E,F,G}

Without A->F:

A+={B,C,D,E,G}

A->F IS NOT REDUNDANT.

d) AG->D

With AG->D:

AG+={A,G,D,E}

Without AG->D:

AG+={A,G,E}

AG->D IS NOT REDUNDANT.

e) AG->E

With AG->E:

AG+={A,G,D,E}

Without AG->E:

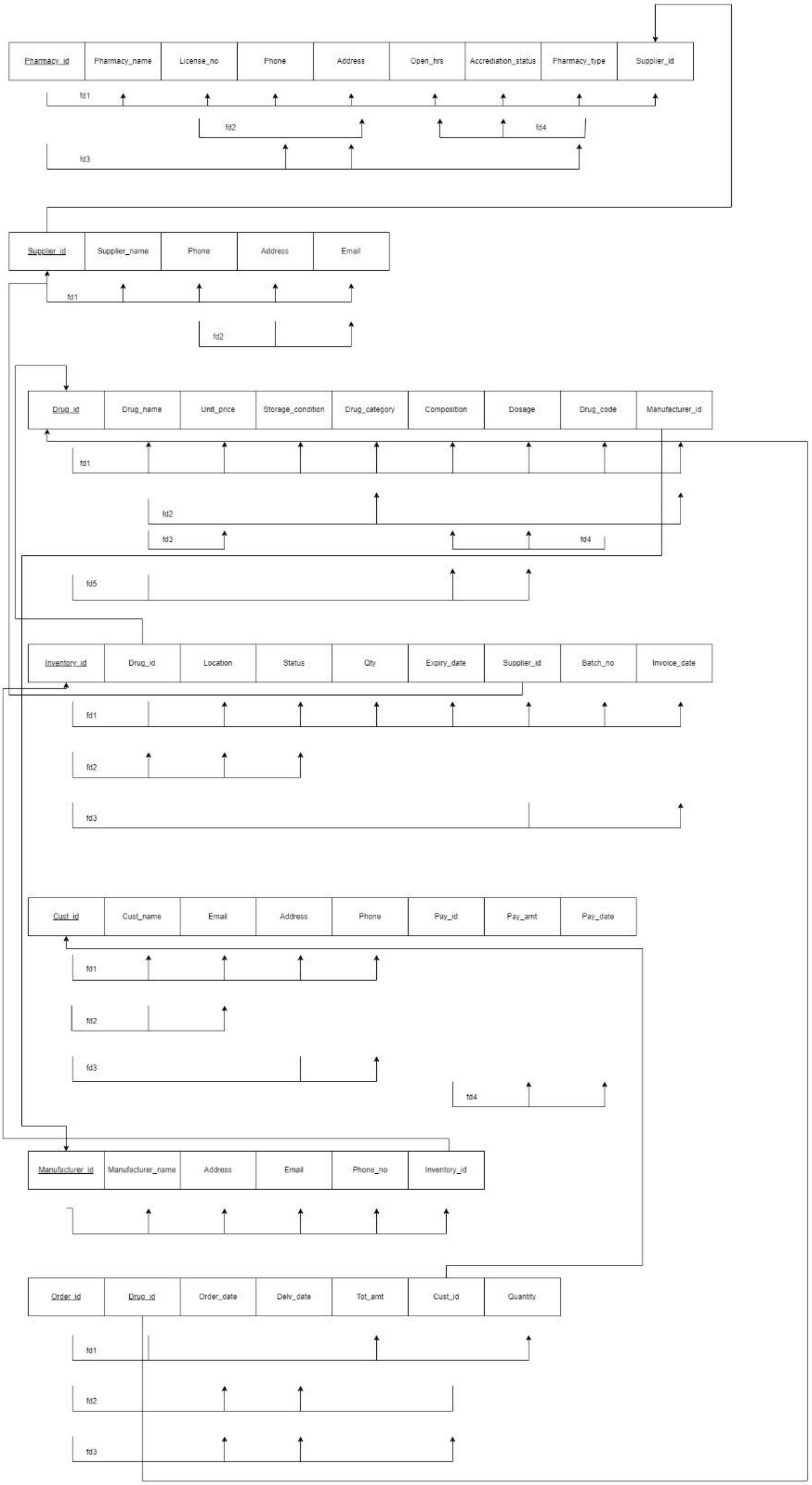
AG+={A,G,D}

AG->E IS NOT REDUNDANT.

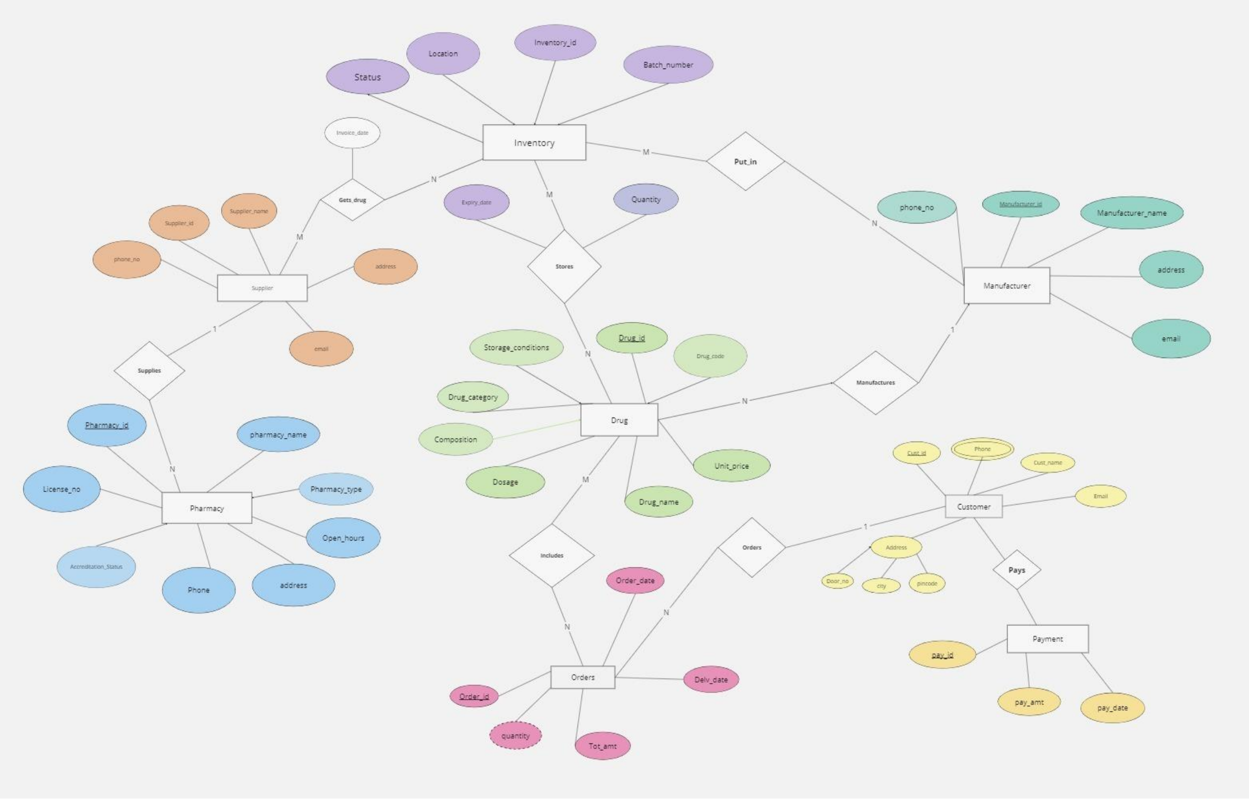
CANDIDATE KEY: {A,AG}

PRIME ATTRIBUTES: {A,G}

NON-PRIME ATTRIBUTES: {B,C,D,E,F}


Pharmacy_id
Pharmacy_name
License_no
Phone
Address
Open_hrs
Accrediation_status
Pharmacy_type
Supplier_id
fd1
fd2
fd4
fd3
Supplier_id
Supplier_name
Phone
Address
Email
fd1
fd2
Drug_id
Drug_name
Unit_price
Storage_condition
Drug_category
Composition
Dosage
Drug_code
Manufacturer_id
fd1
fd2
fd3
fd4
fd5
Inventory_id
Drug_id
Location
Status
Qty
Expiry_date
Supplier_id
Batch_no
Invoice_date
fd1
fd2
fd3
Cust_id
Cust_name
Email
Address
Phone
Pay_id
Pay_amt
Pay_date
fd1
fd2
fd3
fd4
Manufacturer_id
Manufacturer_name
Address
Email
Phone_no
Inventory_id
Order_id
Drug_id
Order_date
Delv_date
Tot_amt
Cust_id
Quantity
fd1
fd2
fd3

ER DIAGRAM:


Status
Location
Inventory_id
Batch_number
Inventory
M
Put_in
Invoice_date
N
Gets_drug
M
Expiry_date
Quantity
Supplier_id
Supplier_name
phone_no
address
Supplier
Stores
1
email
Supplies
N
Storage_conditions
Drug_id
Drug_code
Drug_category
Composition
Drug
Dosage
M
Unit_price
Drug_name
N
Manufactures
1
phone_no
Manufacturer_id
Manufacturer_name
address
Manufacturer
email
Pharmacy_id
pharmacy_name
License_no
Pharmacy
Pharmacy_type
Open_hours
Accreditation_Status
Phone
address
Includes
N
Order_date
Orders
N
1
Order_id
quantity
Tot_amt
Delv_date
Cust_id
Phone
Cust_name
Email
Customer
Address
Door_no
city
pincode
Pays
Payment
pay_id
pay_amt
pay_date

`

PHASE 2

NORMALIZATION OF RELATIONS INTO

BCNF,3NF

1)Pharmacy:

Checking for 1NF:

- Atomicity is satisfied in this relation relation

Therefore the following relation is in the 1NF

Checking for 2NF:

- No partial dependency should determine a NPA attribute of the relation
- Therefore the given relation satisfies the above condition
- The given relation is in the 2NF

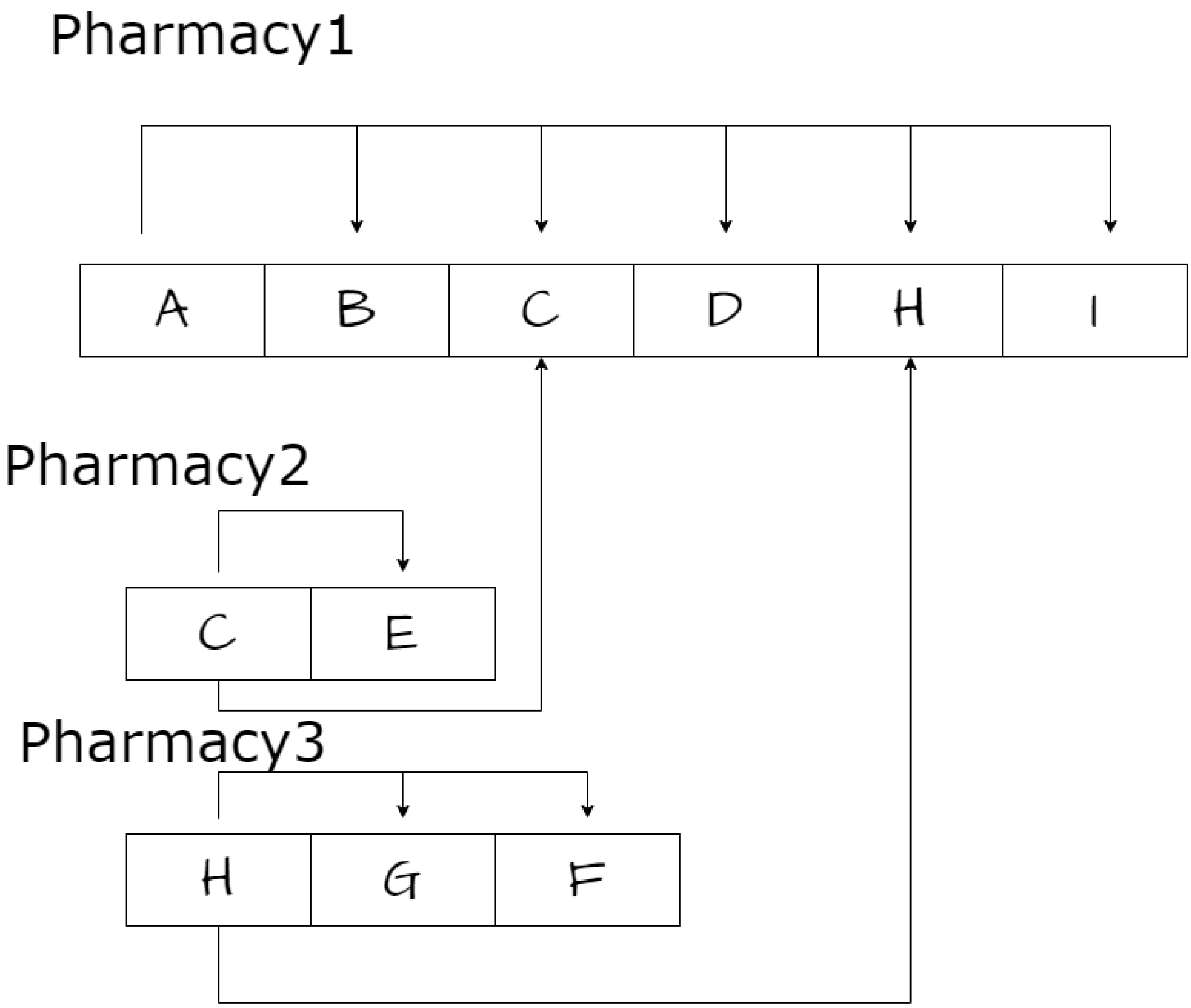
Checking for 3NF:

- No NPA → NPA in the relation
- C → E
- H → F,G
- The above FDs violate the 3NF conditions

Checking for BCNF:

The condition for BCNF is

- LHS must be a super key
- There is no FDs that violate BCNF



Pharmacy1:

Candidate key ={A}

FD’s={A → B,C,D,H,I}

Pharmacy2:

Candidate key={c}

FD’s={C → E}

Pharmacy3:

Candidate key={H}

FD’s={H → F,G}

Pharmacy4:

Candidate key={A}

FD’s ={A → C,H}

2)Supplier:

Checking for 1NF:

- The atomicity of the relation is satisfied.
- The following FDs are in 1NF relation.

Checking for 2NF:

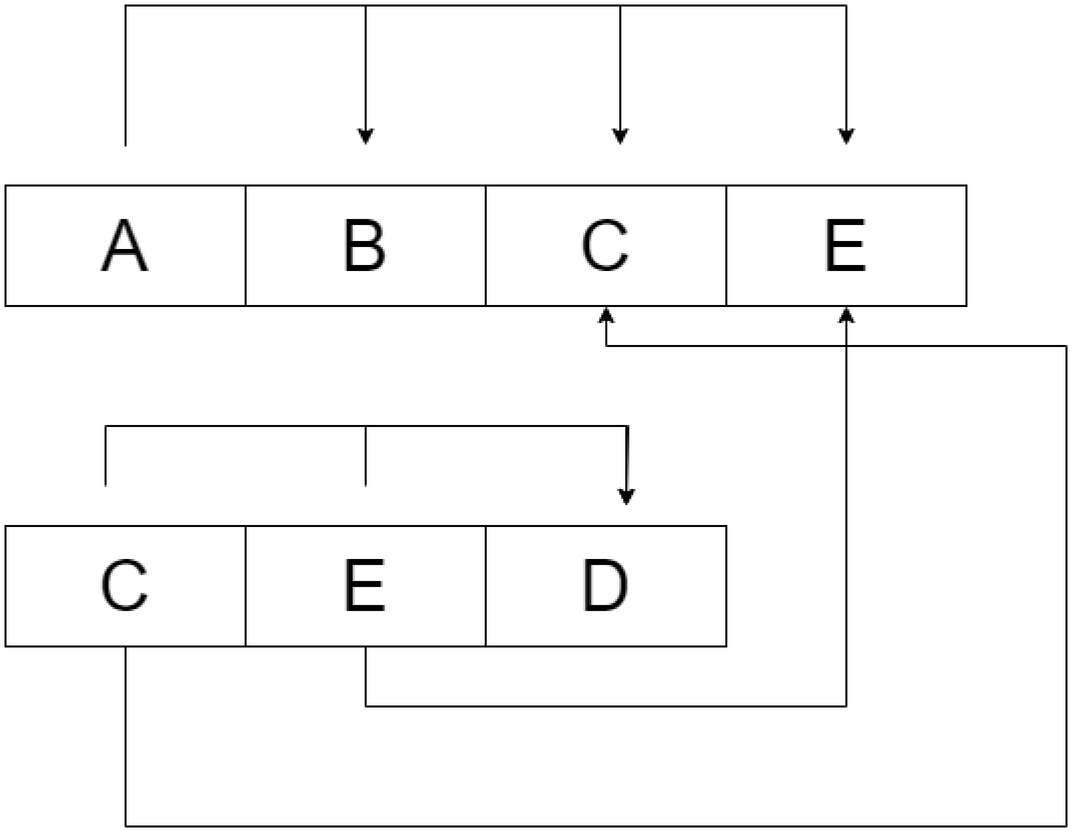
- No partial dependency should determine a NPA attribute of the relation
- Therefore the given relation satisfies the above condition
- The given relation is in the 2NF

Checking for 3NF:

- No transitive dependency in the relation

- NPA → NPA

The fd which violates the 3NF is C,E → D



Supplier1:

- Candidate key={A}
- PA={A}
- NPA={B,C,E}
- FDs:

  A→B,C,E

Supplier2:

- Candidate Key={C}
- PA={C}
- NPA={E,D}
- FDs:

C->E,D

Checking for BCNF:

- LHS must be a super key
- The relation satisfies BCNF

3)DRUG:

Checking for 1NF:

- The atomicity of the relation is satisfied.
- The above relation satisfies the condition.

Checking for 2NF:

- No partial dependency should exist in the relation
- The above relation satisfies the condition.

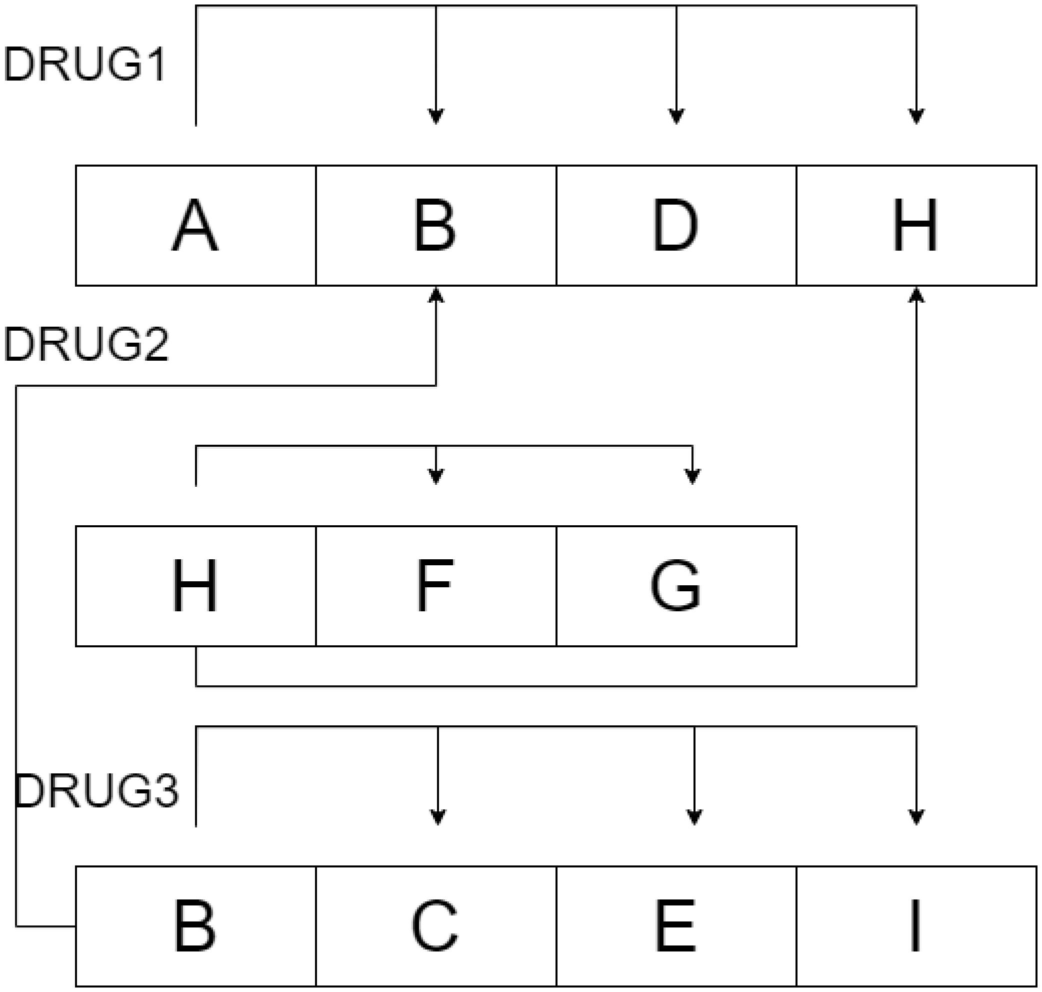
Checking for 3NF:

- No transitive dependency should exist in the relation.
- NPA → NPA

H → F,G

B → C,E,I

The above two FDs violate the 3NF condition.



The 3NF is satisfied .

Drug1:

- Candidate key={A}
- PA={A}
- NPA={B,D,H}
- FDs:

A→B,D,H

Drug2:

- Candidate key={H}
- PA={H}
- NPA={F,G}
- FDs:

H→F,G

Drug3:

- Candidate key={B}
- PA={B}
- NPA={C,E,I}
- FDs:

  B→C,E,I

Checking for BCNF:

- LHS must be a Super-key
- The above relation satisfies the BCNF condition

4)Inventory:

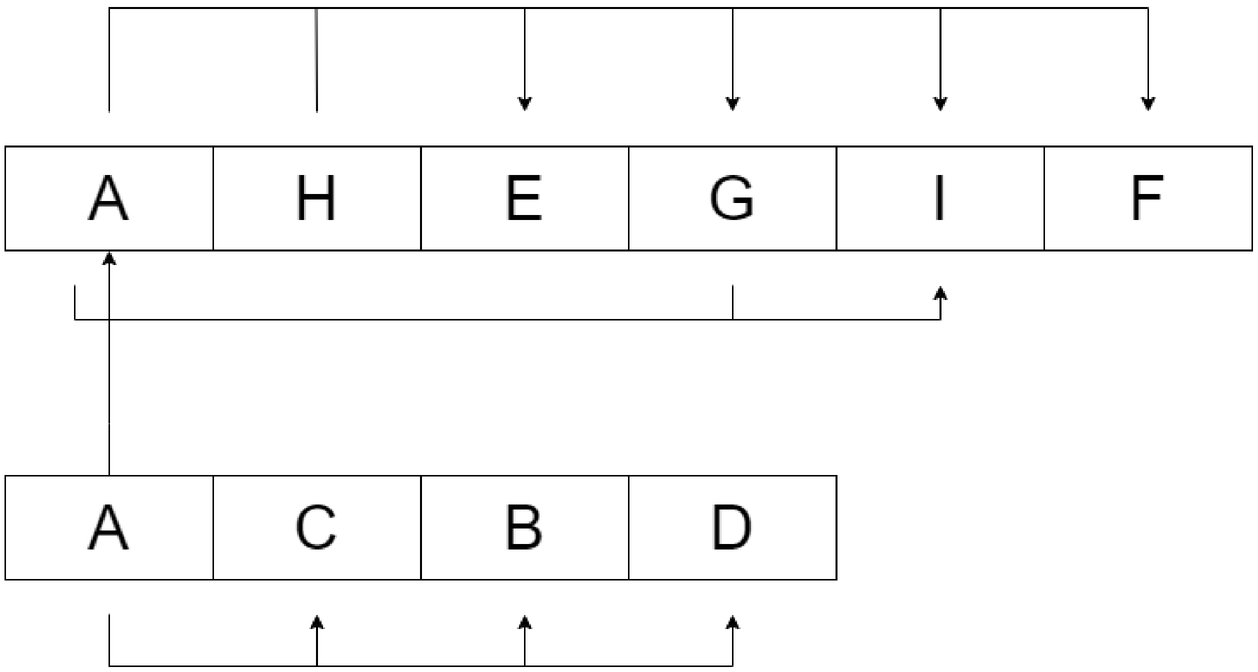
Checking for 1NF:

The atomicity of the relation is satisfied

Checking for 2NF:

No partial dependency in the relation

A→BCD Violates the 2NF condition



Inventory1:

- Candidate key={A,H}
- PA={A,H}
- NPA={E,F,G,I}
- FDs:

  AH→E,F,G,I

Inventory2:

- Candidate key={A}
- PA={A}
- NPA={CBD}
- FDs:

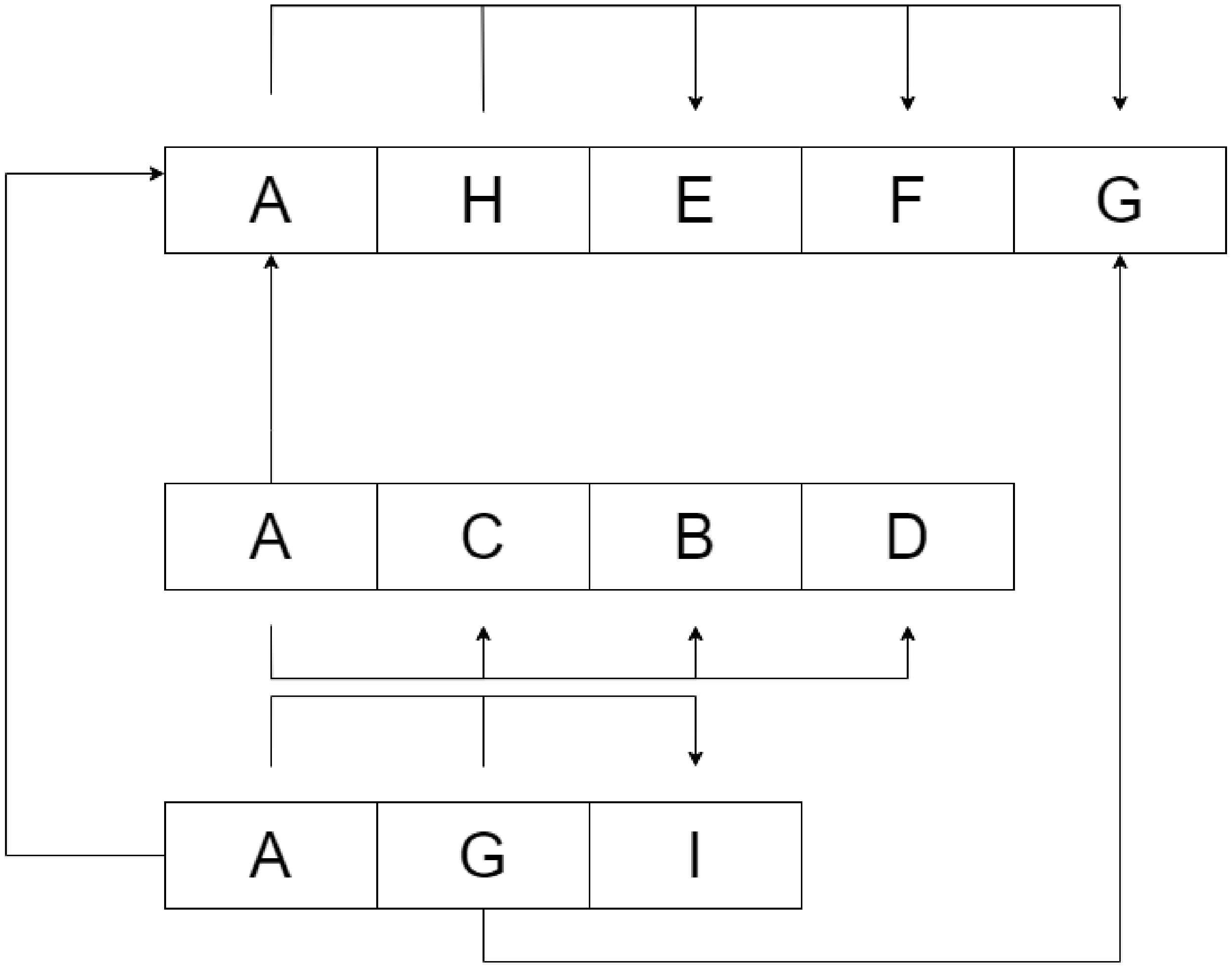
Checking for 3NF:

- PA={AH}
- NPA={E,G,I,F}

- Candidate key is AH
- No transitive dependency should exist in the relation
- NPA→NPA

AG→I

violates the condition for 3NF



Therefore 3CNF is satisfied

Checking for BCNF:

- LHS must be a super-key
- The above relation satisfies the condition.

5)CUSTOMER TABLE :

CHECKING FOR 1NF:

COMPOSITE AND MULTIVALUED ATTRIBUTE PRESENT

DECOMPOSING TO 1 NF:

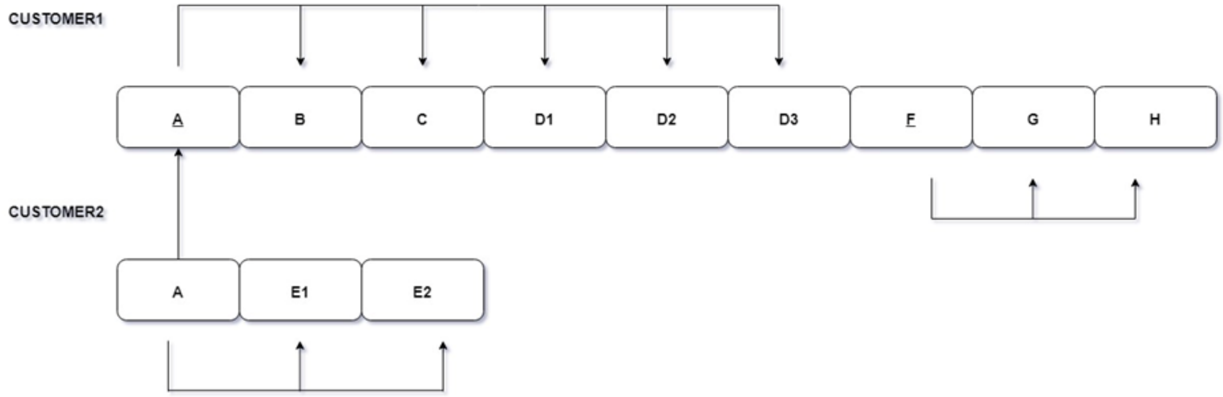
D1 : DOOR_NO

D2 : CITY

D3 : PINCODE

E1 : PHONE_NUM1

E2 : PHONE_NUM2



CUSTOMER1 TABLE FDS ARE :

{

A -> B , A -> C , A -> D1 , A -> D2 , A -> D3 , F -> G , F -> H

}

CUSTOMER2 TABLE FDS ARE:{A -> E1 , E2}

A IS THE CANDIDATE KEY FOR CUSTOMER1 TABLE .

CHECKING FOR 2NF :

PARTIAL DEPENDENCIES ARE PRESENT IN CUSTOMER1 TABLE .

THE FDS WITH PARTIAL DEPENDENCIES

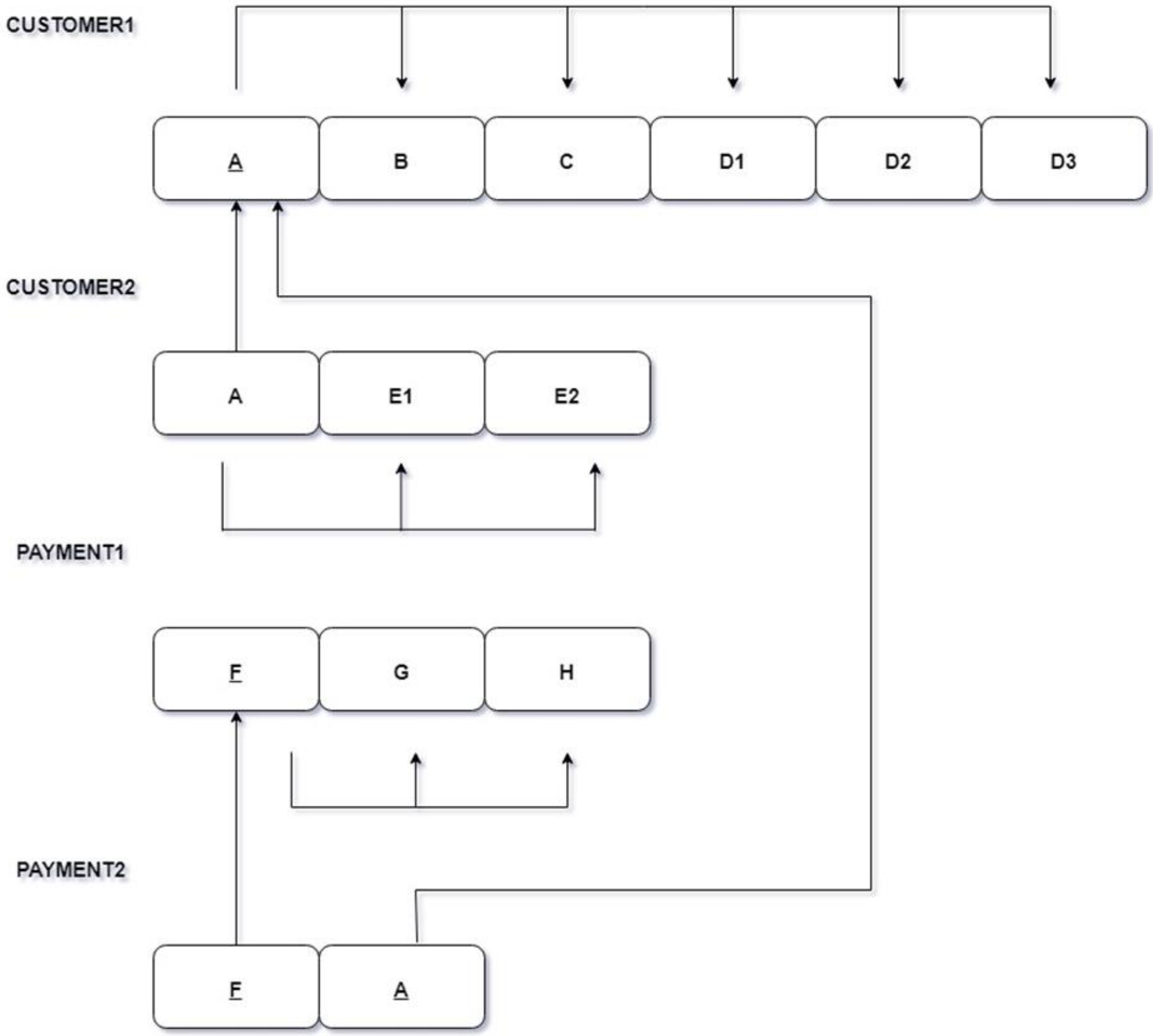
{

A -> B ,C ,D1 ,D2 , D3

F -> G,H

}

DECOMPOSING TO 2NF :



CHECKING FOR FURTHER PARTIAL DEPENDENCIES :

IN CUSTOMER1 TABLE THE CANDIDATE KEY IS {A}

THERE IS NO PARTIAL DEPENDENCIES IN CUSTOMER1.

IN PAYMENT1 TABLE THE CANDIDATE KEY ID {F}

THERE IS NO PARTIAL DEPENDENCIES IN PAYMENT1.

CHECKING FOR 3NF :

NO , TRANSITIVE DEPENDENCIES IS PRESENT .

THEREFORE 3NF IS SATISFIED .

CHECKING FOR BCNF :

ALL FDS HAVE SUPER KEY AS THE LEFT ATTRIBUTE .

THEREFORE BCNF IS SATISFIED .

6)MANUFACTURERS:

1NF (First Normal Form):

- Condition: The relation with the functional dependency A → B, C, D, E, F is in First Normal Form (1NF) because:
- All attributes contain atomic values.
- Each attribute holds a single value.
- Values in each column are of the same data type.
- Each row is uniquely identified by the primary key MANUFACTURER_ID, ensuring there are no repeating groups or arrays.

Conclusion: The given relation satisfies all the conditions of 1NF.

2NF (Second Normal Form):

- Condition: A relation is in 2NF if it is in 1NF and all non prim attributes are fully functionally dependent on the entire primary key.
- Explanation:

The relation is in 1NF because all attributes contain only atomic values.

The primary key is A (MANUFACTURER_ID).

- The functional dependency A → B, C, D, E, F shows that all non prime attributes (B, C, D, E, F) are fully functionally dependent on the primary key A.
- Since A is a single attribute key, there cannot be partial dependencies.

Conclusion: The relation is in 2NF because it is in 1NF and all non-prime attributes are fully functionally dependent on the primary key A.

3NF (Third Normal Form):

- Condition: A relation is in 3NF if it is in 2NF and for every functional dependency X → Y, either X is a super key or Y is a prime attribute.
- Explanation:

The relation is already in 2NF.

For the dependency A → B, C, D, E, F, the left-hand side A is a super key

(since A is the primary key).

There are no transitive dependencies (dependencies where a non-prime attribute depends on another non-prime attribute).

Conclusion: The relation is in 3NF because it is in 2NF and for every functional dependency, the left-hand side is a super key.

BCNF (Boyce-Codd Normal Form):

Condition: A relation is in BCNF if for every non-trivial functional dependency X → Y, X is a super key.

- Explanation:

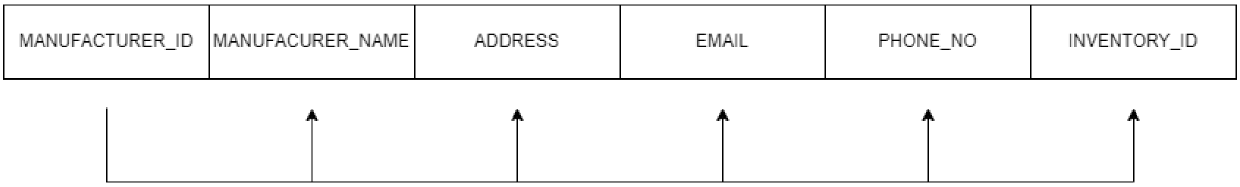
The relation is already in 3NF.

The given functional dependency A → B, C, D, E, F shows that the primary key A determines all other attributes.

In this case, A (MANUFACTURER_ID) is a super key because it uniquely identifies each tuple in the relation.

There are no other functional dependencies where a non-super key determines another attribute.

MANUFACTURER



MANUFACTURER_ID	MANUFACURER_NAME	ADDRESS	EMAIL	PHONE_NO	INVENTORY_ID

7)ORDERS:

- **1NF(First Normal Form):**

  The relations with the functional dependency A → B, C, F AND AG->D,E are in First Normal Form (1NF) because:

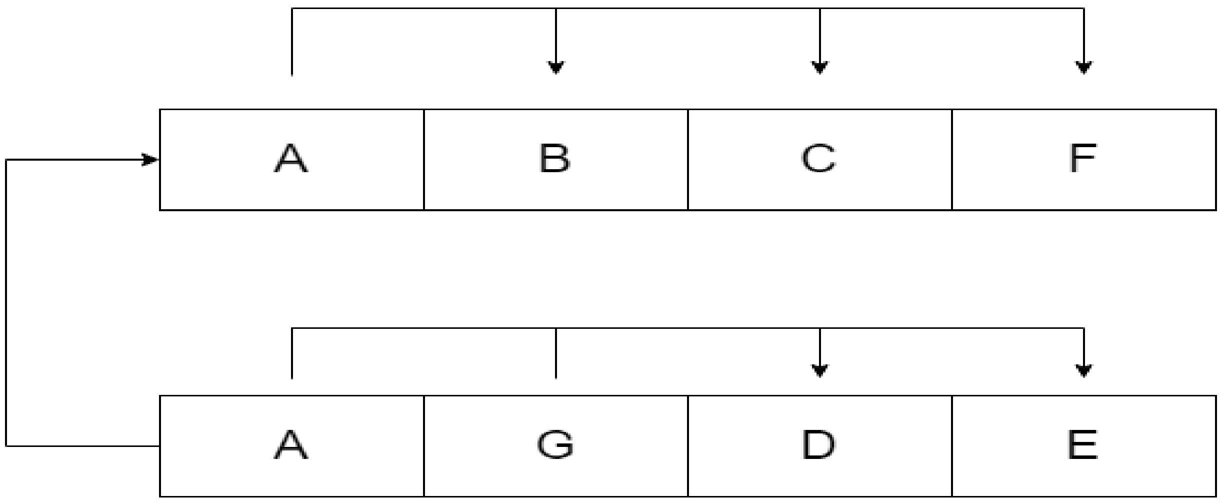
  - All attributes contain atomic values.
  - Each attribute holds a single value.
  - Values in each column are of the same data type.
  - Each row is uniquely identified by the primary key order_id, (order_id,drug_id) ensuring there are no repeating groups or arrays.

Therefore, the given relation satisfies all the conditions of 1NF.

- **2NF (Second Normal Form):**
  - To satisfy Second Normal Form (2NF), the relation must first be in 1NF and also meet the following criteria:

  - There should be no partial dependency, which means no non-prime attribute should be functionally dependent on any proper subset of any candidate key of the relation.
  - Given:
    - The candidate keys are {A, AG}.
    - Prime attributes are {A, G}.
    - Non-prime attributes are {B, C, D, E, F}.
    - From the given functional dependencies:
      - A → B
      - A → C
      - A → F
  - These functional dependencies show that B, C, and F are dependent on A, which is a candidate key by itself, so there is no partial dependency with respect to A.
  - However, for the combined candidate key AG:
    - AG → D
    - AG → E
    - Since A is a subset of AG and:
      - A → B
      - A → C
      - A → F

- These show that B, C, and F are partially dependent on A (a proper subset of AG), thus indicating a partial dependency, which violates 2NF.
- Therefore, the given relation does not satisfy 2NF due to partial dependencies of non-prime attributes on a part of a candidate key.



- **3NF (Third Normal Form):**

-Condition: A relation is in 3NF if it is in 2NF and for every functional dependency X → Y, either X is a super key or Y is a prime attribute.

- Explanation:
- The relation is already in 2NF.
- For A->B,C,F
- There are no transitive dependencies (dependencies where a non-prime attribute depends on another non-prime attribute).
- Conclusion: The relation is in 3NF because it is in 2NF and for every functional dependency, the left-hand side is a super key.

-For AG->DE:

- There are no transitive dependencies (dependencies where a non-prime attribute depends on another non-prime attribute).
- Conclusion: The relation is in 3NF because it is in 2NF and for every functional dependency, the left-hand side is a super key.

- **BCNF (Boyce-Codd Normal Form):**

  - Condition:A relation is in BCNF if for every non-trivial functional dependency X → Y, X is a superkey.

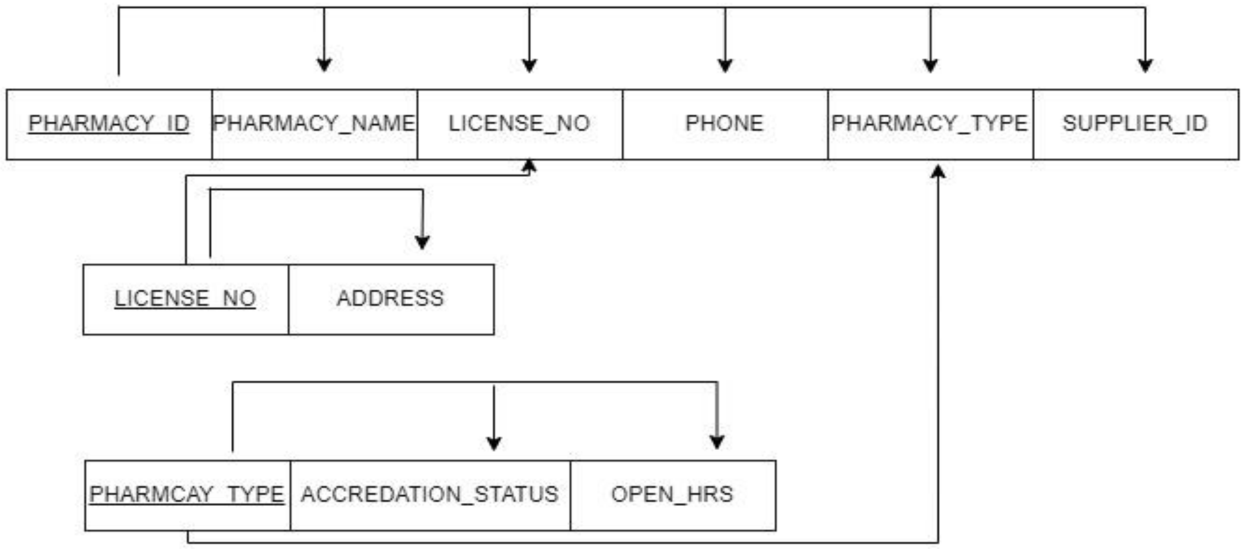
  - Explanation:

  - The relation is already in 3NF.

  - The given functional dependency A → B, C F and AG ->DE shows that the primary key A and AG determines all other attributes of the same table.

FINAL SCHEMA:

PHARMACY

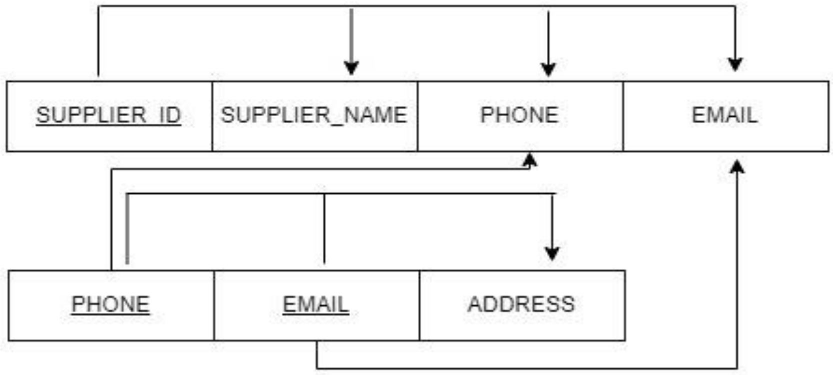


PHARMACY_ID	PHARMACY_NAME	LICENSE_NO	PHONE	PHARMACY_TYPE	SUPPLIER_ID

LICENSE_NO	ADDRESS

PHARMCAY_TYPE	ACCREDATION_STATUS	OPEN_HRS

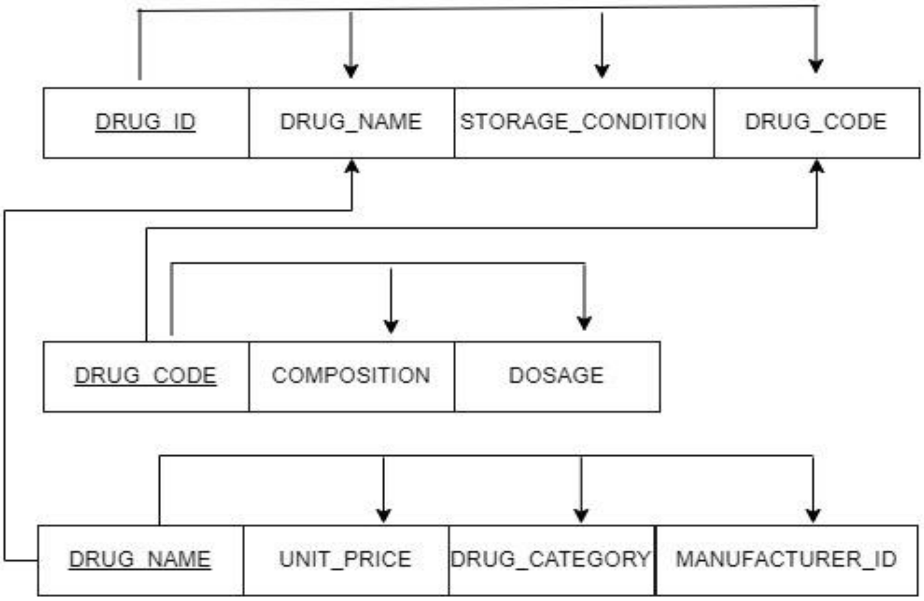
SUPPLIER



SUPPLIER_ID	SUPPLIER_NAME	PHONE	EMAIL

PHONE	EMAIL	ADDRESS

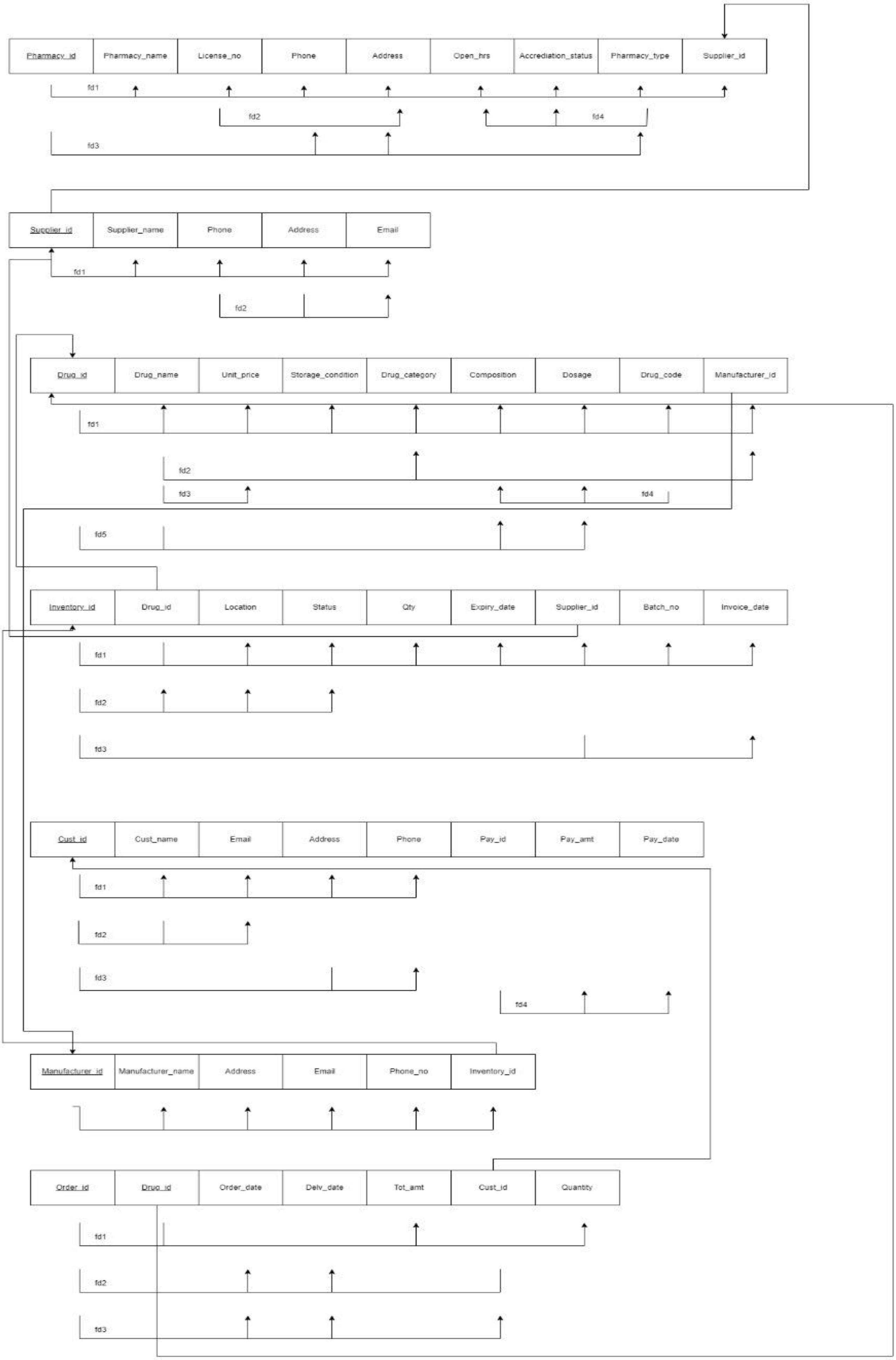
DRUG



DRUG_ID	DRUG_NAME	STORAGE_CONDITION	DRUG_CODE

DRUG_CODE	COMPOSITION	DOSAGE

DRUG_NAME	UNIT_PRICE	DRUG_CATEGORY	MANUFACTURER_ID


Pharmacy_id
Pharmacy_name
License_no
Phone
Address
Open_hrs
Accrediation_status
Pharmacy_type
Supplier_id
fd1
fd2
fd4
fd3
Supplier_id
Supplier_name
Phone
Address
Email
fd1
fd2
Drug_id
Drug_name
Unit_price
Storage_condition
Drug_category
Composition
Dosage
Drug_code
Manufacturer_id
fd1
fd2
fd3
fd4
fd5
Inventory_id
Drug_id
Location
Status
Qty
Expiry_date
Supplier_id
Batch_no
Invoice_date
fd1
fd2
fd3
Cust_id
Cust_name
Email
Address
Phone
Pay_id
Pay_amt
Pay_date
fd1
fd2
fd3
fd4
Manufacturer_id
Manufacturer_name
Address
Email
Phone_no
Inventory_id
Order_id
Drug_id
Order_date
Delv_date
Tot_amt
Cust_id
Quantity
fd1
fd2
fd3

SQL CODE:

```
DROP TABLE ORDER_LIST;
DROP TABLE ORDERS;
DROP TABLE PHARMACY;
DROP TABLE MANUFACTURER;
DROP TABLE SUPPLIER;
DROP TABLE INVENTORY;
DROP TABLE DRUGS;
DROP TABLE CUSTOMER;


CREATE TABLE CUSTOMER(
    USER_NAME VARCHAR2(100) CONSTRAINT C_ID_PK PRIMARY KEY,
    PASSWORD VARCHAR2(100),
    CUST_NAME VARCHAR2(100),
    EMAIL VARCHAR2(100),
    DOOR_NO INT,
    CITY VARCHAR2(100),
    PINCODE NUMERIC(6)
);

CREATE TABLE DRUGS(
    DRUG_ID CHAR(4) CONSTRAINT D_ID_PK PRIMARY KEY,
    DRUG_NAME VARCHAR2(100),
    STORAGE_CONDITION VARCHAR2(100),
    DRUG_CODE INT,
    UNIT_PRICE INT,
    CONSTRAINT D_ID_CH CHECK(DRUG_ID LIKE 'D%')
);

CREATE TABLE INVENTORY(
    INVENTORY_ID CHAR(4) CONSTRAINT IN_PK PRIMARY KEY,
    DRUG_ID CHAR(4) CONSTRAINT I_FK_D REFERENCES DRUGS(DRUG_ID),
    EXPIRY_DATE DATE,
    QTY INT,
    CONSTRAINT I_ID_CH CHECK(INVENTORY_ID LIKE 'I%')
);

CREATE TABLE SUPPLIER(
    SUPPLIER_ID CHAR(4) CONSTRAINT SUPPLIER_PK PRIMARY KEY,
    INVENTORY_ID CHAR(4) CONSTRAINT I_FK REFERENCES INVENTORY(INVENTORY_ID),
    DRUG_ID CHAR(4) CONSTRAINT D_FK REFERENCES DRUGS(DRUG_ID),
    SUPPLIER_NAME VARCHAR2(100),
    PHONE_NO VARCHAR2(10) CONSTRAINT S_PH_U UNIQUE,
    EMAIL VARCHAR2(100),
    CONSTRAINT S_ID_CH CHECK(SUPPLIER_ID LIKE 'S%')
);
```

```
CREATE TABLE MANUFACTURER(
    MANUFACTURER_ID CHAR(4) CONSTRAINT MF_PK PRIMARY KEY,
    MANUFACTURER_NAME VARCHAR2(100),
    DRUG_ID CONSTRAINT D1_FK REFERENCES DRUGS(DRUG_ID),
    ADDRESS VARCHAR2(100),
    PHONE_NO NUMERIC CONSTRAINT M_PH_U UNIQUE,
    EMAIL VARCHAR(100),
    INVENTORY_ID CHAR(4) CONSTRAINT M_FK_I REFERENCES INVENTORY(INVENTORY_ID),
    CONSTRAINT M_ID_CH CHECK(MANUFACTURER_ID LIKE 'M%')
);
CREATE TABLE PHARMACY(
    DRUG_ID CHAR(4) CONSTRAINT D_FK_P REFERENCES DRUGS(DRUG_ID),
    SUPPLIER_ID CHAR(4) CONSTRAINT S_FK_P REFERENCES SUPPLIER(SUPPLIER_ID),
    SUPPLIED_QTY INT,
    PHARMACY_NAME VARCHAR2(100),
    PHONE_NO NUMERIC(10) CONSTRAINT P_PH_U UNIQUE,
    CONSTRAINT P_ID_PK PRIMARY KEY(SUPPLIER_ID,DRUG_ID)
);

CREATE TABLE ORDERS(
    ORDER_ID CHAR(4) CONSTRAINT O_PK PRIMARY KEY,
    USER_NAME VARCHAR2(100) CONSTRAINT O_FK_C REFERENCES CUSTOMER(USER_NAME),
    ORDER_DATE DATE,
    TOTAL_AMT INT,
    CONSTRAINT O_ID_CH CHECK(ORDER_ID LIKE 'O%')
);

CREATE TABLE ORDER_LIST(
    ORDER_ID CHAR(4) CONSTRAINT O_FK_OL REFERENCES ORDERS(ORDER_ID),
    DRUG_ID CHAR(4) CONSTRAINT D_FK_O REFERENCES DRUGS(DRUG_ID),
    QTY INT,
    CONSTRAINT CMP_KEY_OL PRIMARY KEY(ORDER_ID,DRUG_ID)
);
```

```
INSERT INTO CUSTOMER (USER_NAME, PASSWORD, CUST_NAME, EMAIL, DOOR_NO, CITY,
PINCODE) VALUES
('john_doe', 'password123', 'John Doe', 'john.doe@example.com', 101, 'New York',
10001);

INSERT INTO CUSTOMER (USER_NAME, PASSWORD, CUST_NAME, EMAIL, DOOR_NO, CITY,
PINCODE) VALUES
('jane_smith', 'pass456', 'Jane Smith', 'jane.smith@example.com', 202, 'Los Angeles',
90001);
```

```
INSERT INTO CUSTOMER (USER_NAME, PASSWORD, CUST_NAME, EMAIL, DOOR_NO, CITY,
PINCODE) VALUES
('alice_w', 'alice789', 'Alice Wonderland', 'alice.w@example.com', 303, 'Chicago',
60007);



INSERT INTO DRUGS (DRUG_ID, DRUG_NAME, STORAGE_CONDITION, DRUG_CODE, UNIT_PRICE)
VALUES
('D001', 'Aspirin', 'Room Temperature', 1001, 10);

INSERT INTO DRUGS (DRUG_ID, DRUG_NAME, STORAGE_CONDITION, DRUG_CODE, UNIT_PRICE)
VALUES
('D002', 'Paracetamol', 'Cool and Dry', 1002, 15);

INSERT INTO DRUGS (DRUG_ID, DRUG_NAME, STORAGE_CONDITION, DRUG_CODE, UNIT_PRICE)
VALUES
('D003', 'Ibuprofen', 'Room Temperature', 1003, 12);



INSERT INTO INVENTORY (INVENTORY_ID, DRUG_ID, EXPIRY_DATE, QTY) VALUES
('I001', 'D001','31-DEC-2023', 100);

INSERT INTO INVENTORY (INVENTORY_ID, DRUG_ID, EXPIRY_DATE, QTY) VALUES
('I002', 'D002','30-NOV-2023', 200);

INSERT INTO INVENTORY (INVENTORY_ID, DRUG_ID, EXPIRY_DATE, QTY) VALUES
('I003', 'D003', '31-OCT-2023', 150);



INSERT INTO SUPPLIER (SUPPLIER_ID, INVENTORY_ID, DRUG_ID, SUPPLIER_NAME, PHONE_NO,
EMAIL) VALUES
('S001', 'I001', 'D001', 'Supplier One', '1234567890', 'supplier1@example.com');

INSERT INTO SUPPLIER (SUPPLIER_ID, INVENTORY_ID, DRUG_ID, SUPPLIER_NAME, PHONE_NO,
EMAIL) VALUES
('S002', 'I002', 'D002', 'Supplier Two', '0987654321', 'supplier2@example.com');

INSERT INTO SUPPLIER (SUPPLIER_ID, INVENTORY_ID, DRUG_ID, SUPPLIER_NAME, PHONE_NO,
EMAIL) VALUES
('S003', 'I003', 'D003', 'Supplier Three', '1122334455', 'supplier3@example.com');


INSERT INTO MANUFACTURER (MANUFACTURER_ID, MANUFACTURER_NAME, DRUG_ID, ADDRESS,
PHONE_NO, EMAIL, INVENTORY_ID) VALUES
('M001', 'Manufacturer One', 'D001', '123 Street, City', 1112223333, 'manu1@example.com', 'I001');
```

```sql
INSERT INTO MANUFACTURER (MANUFACTURER_ID, MANUFACTURER_NAME, DRUG_ID, ADDRESS,
PHONE_NO, EMAIL, INVENTORY_ID) VALUES
('M002', 'Manufacturer Two', 'D002', '456 Avenue, City', 4445556666, 'manu2@example.com', 'I002');

INSERT INTO MANUFACTURER (MANUFACTURER_ID, MANUFACTURER_NAME, DRUG_ID, ADDRESS,
PHONE_NO, EMAIL, INVENTORY_ID) VALUES
('M003', 'Manufacturer Three', 'D003', '789 Boulevard, City', 7778889999,
'manu3@example.com', 'I003');



INSERT INTO PHARMACY (DRUG_ID, SUPPLIER_ID, SUPPLIED_QTY, PHARMACY_NAME, PHONE_NO)
VALUES
('D001', 'S001', 50, 'Pharmacy One', 1231231231);

INSERT INTO PHARMACY (DRUG_ID, SUPPLIER_ID, SUPPLIED_QTY, PHARMACY_NAME, PHONE_NO)
VALUES
('D002', 'S002', 75, 'Pharmacy Two', 4564564564);

INSERT INTO PHARMACY (DRUG_ID, SUPPLIER_ID, SUPPLIED_QTY, PHARMACY_NAME, PHONE_NO)
VALUES
('D003', 'S003', 100, 'Pharmacy Three', 7897897897);



INSERT INTO ORDERS (ORDER_ID, USER_NAME, ORDER_DATE, TOTAL_AMT) VALUES
('O001', 'john_doe','15-JUN-2023', 150);

INSERT INTO ORDERS (ORDER_ID, USER_NAME, ORDER_DATE, TOTAL_AMT) VALUES
('O002', 'jane_smith', '16-JUN-2023', 200);

INSERT INTO ORDERS (ORDER_ID, USER_NAME, ORDER_DATE, TOTAL_AMT) VALUES
('O003', 'alice_w','17-JUN-2023', 180);




INSERT INTO ORDER_LIST (ORDER_ID, DRUG_ID, QTY) VALUES
('O001', 'D001', 10);

INSERT INTO ORDER_LIST (ORDER_ID, DRUG_ID, QTY) VALUES
('O001', 'D002', 5);

INSERT INTO ORDER_LIST (ORDER_ID, DRUG_ID, QTY) VALUES
('O002', 'D002', 10);

INSERT INTO ORDER_LIST (ORDER_ID, DRUG_ID, QTY) VALUES
('O002', 'D003', 8);
```

```
INSERT INTO ORDER_LIST (ORDER_ID, DRUG_ID, QTY) VALUES
('O003', 'D001', 6);

INSERT INTO ORDER_LIST (ORDER_ID, DRUG_ID, QTY) VALUES
('O003', 'D003', 12);
```

Trigger used:

Trigger is used to make sure that order date is prior to expiry date.

```
CREATE OR REPLACE TRIGGER trg_check_order_date
BEFORE INSERT ON ORDER_LIST
FOR EACH ROW
DECLARE
    v_order_date ORDERS.ORDER_DATE%TYPE;
    v_expiry_date INVENTORY.EXPIRY_DATE%TYPE;
BEGIN
    -- Retrieve the order date from the ORDERS table
    SELECT ORDER_DATE INTO v_order_date
    FROM ORDERS
    WHERE ORDER_ID = :NEW.ORDER_ID;

    -- Retrieve the expiry date from the INVENTORY table
    SELECT EXPIRY_DATE INTO v_expiry_date
    FROM INVENTORY
    WHERE DRUG_ID = :NEW.DRUG_ID;

    -- Check if the order date is before the expiry date
    IF v_order_date >= v_expiry_date THEN
        RAISE_APPLICATION_ERROR(-20001, 'Order date must be before the expiry date of the drug.');
    END IF;
END;
/
```

Procedure used:

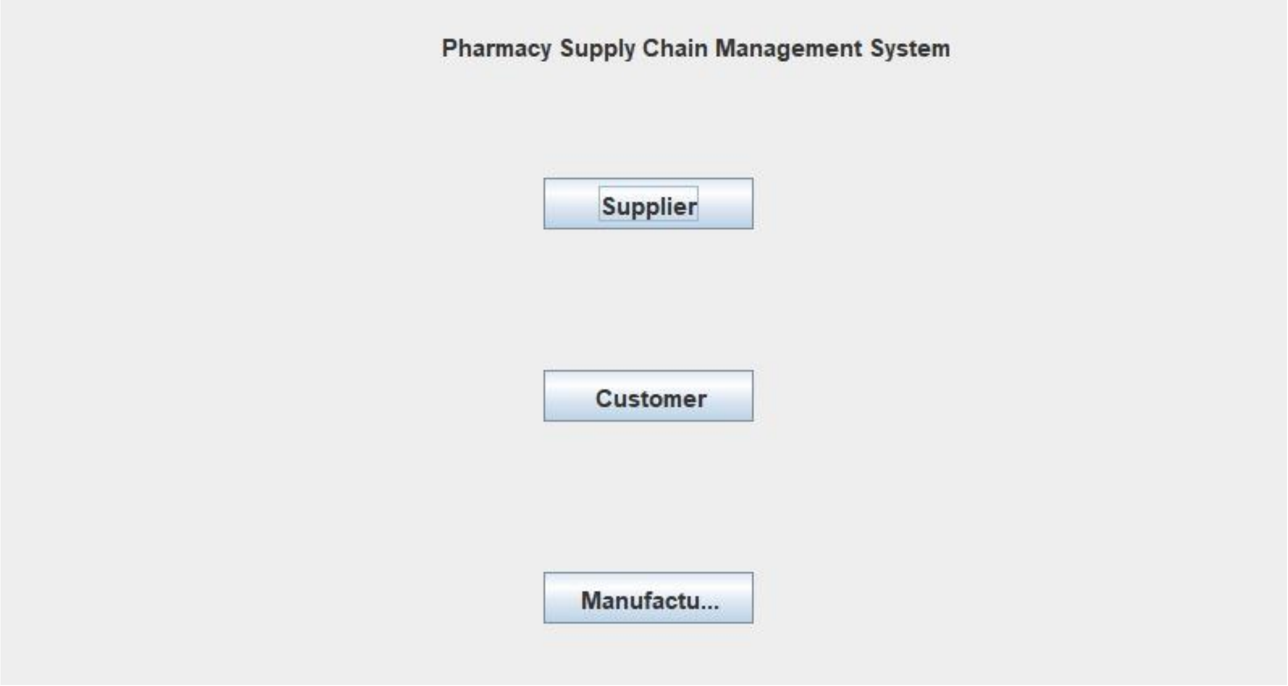
Procedure is used to calculate the bill amount for the order.

```
CREATE OR REPLACE PROCEDURE CALCULATE_TOTAL_AMOUNT (
    p_order_id IN CHAR
) AS
    v_total_amt INT := 0;
BEGIN
    -- Calculate the total amount for the given order ID
    SELECT SUM(ol.QTY * d.UNIT_PRICE)
    INTO v_total_amt
    FROM ORDER_LIST ol
    JOIN DRUGS d ON ol.DRUG_ID = d.DRUG_ID
    WHERE ol.ORDER_ID = p_order_id;

    -- Update the total amount in the ORDERS table
    UPDATE ORDERS
    SET TOTAL_AMT = v_total_amt
    WHERE ORDER_ID = p_order_id;

    COMMIT;
END;
/
```

OUTPUT SCREENSHOTS:

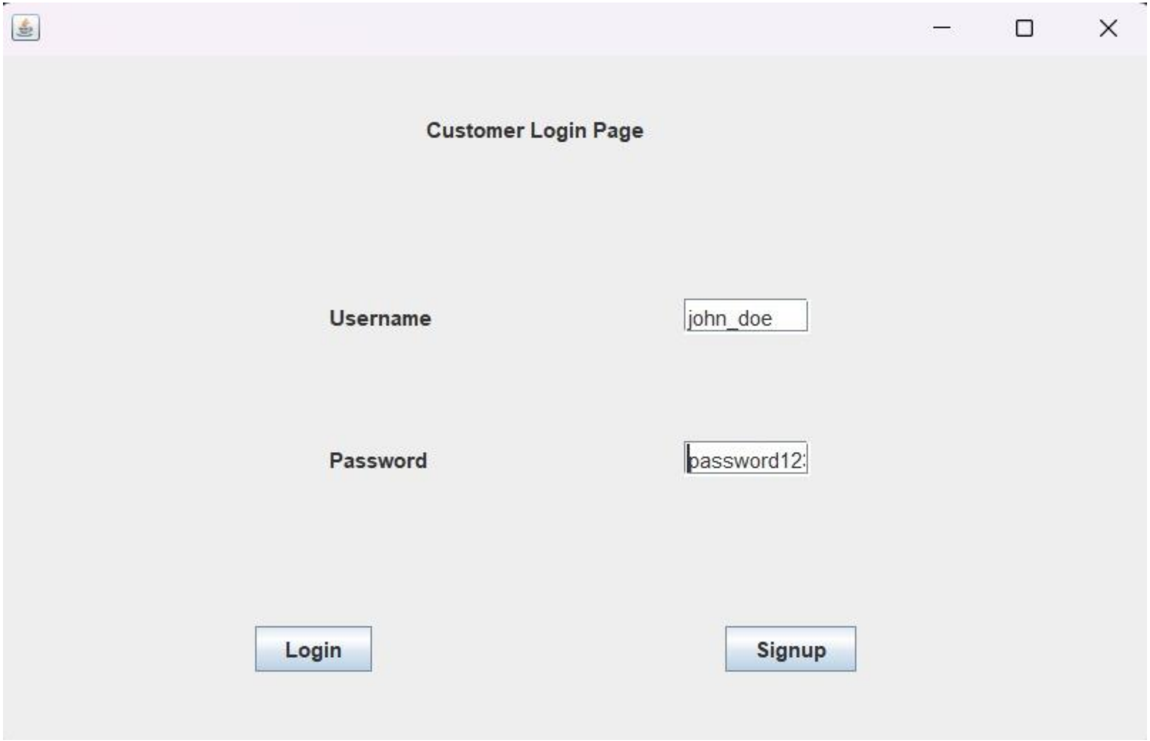
Login Page:

This is the basic login page that would be displayed to the user where the user has to enter their login credentials to access the page further.



Pharmacy Supply Chain Management System

Customer Login:



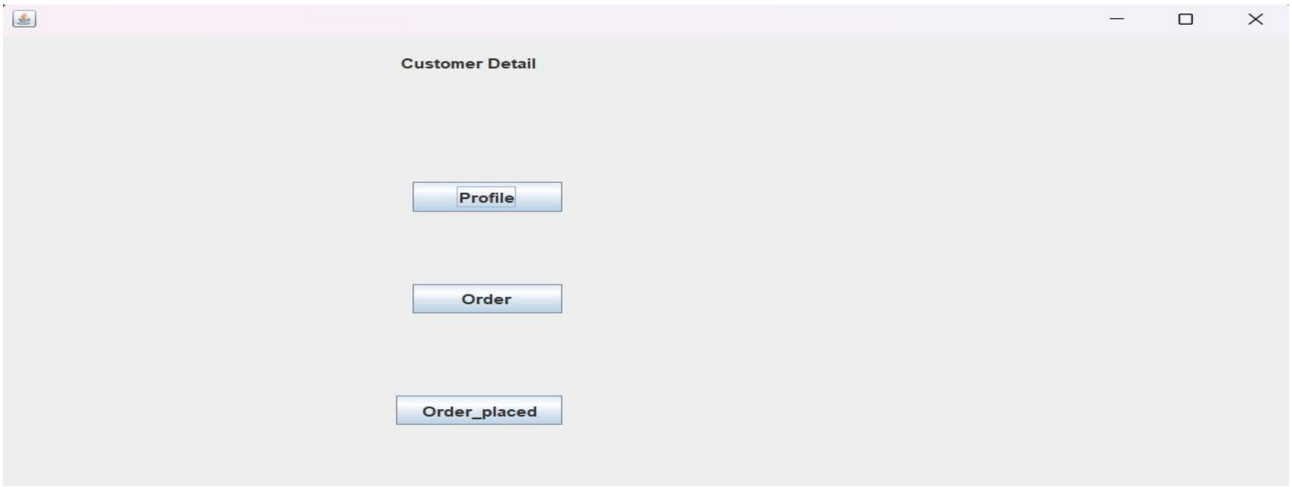
Customer Login Page

Username: john_doe

Password: password12:

Menu Page:

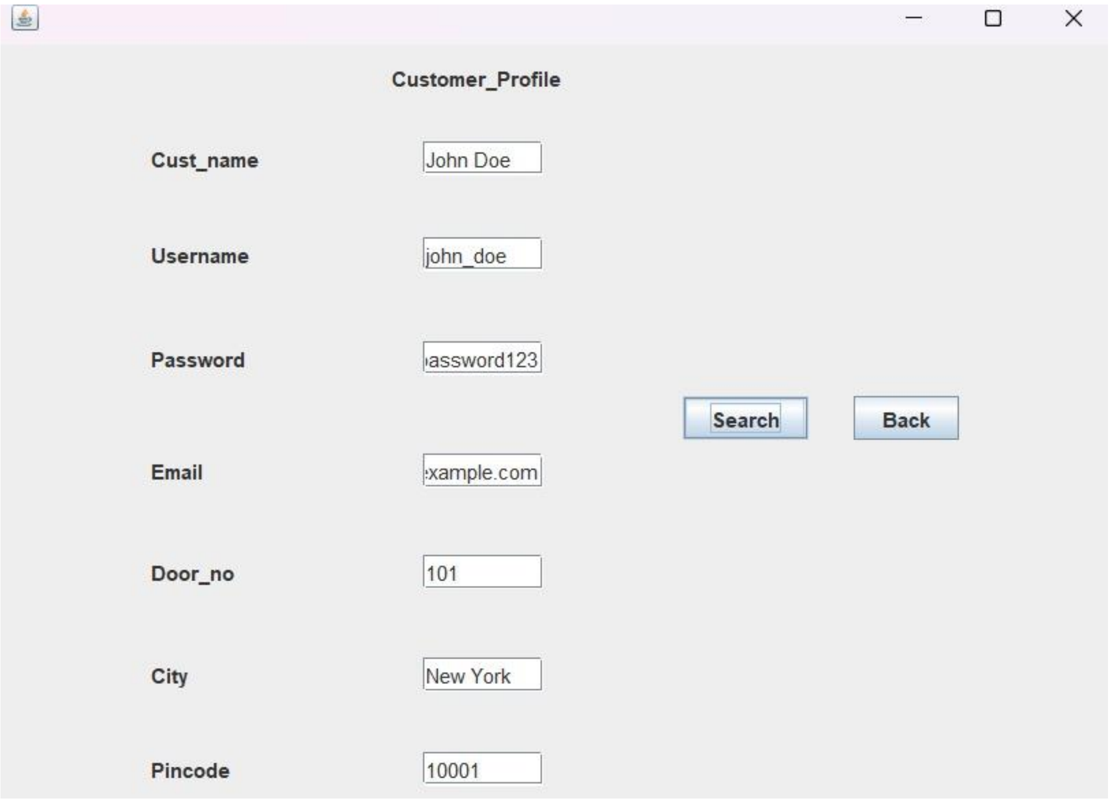
Menu Page will display the various options that are available to the user. The options include gateway to view the profile ,to visit the orders placed, to place new order.



Profile Page:

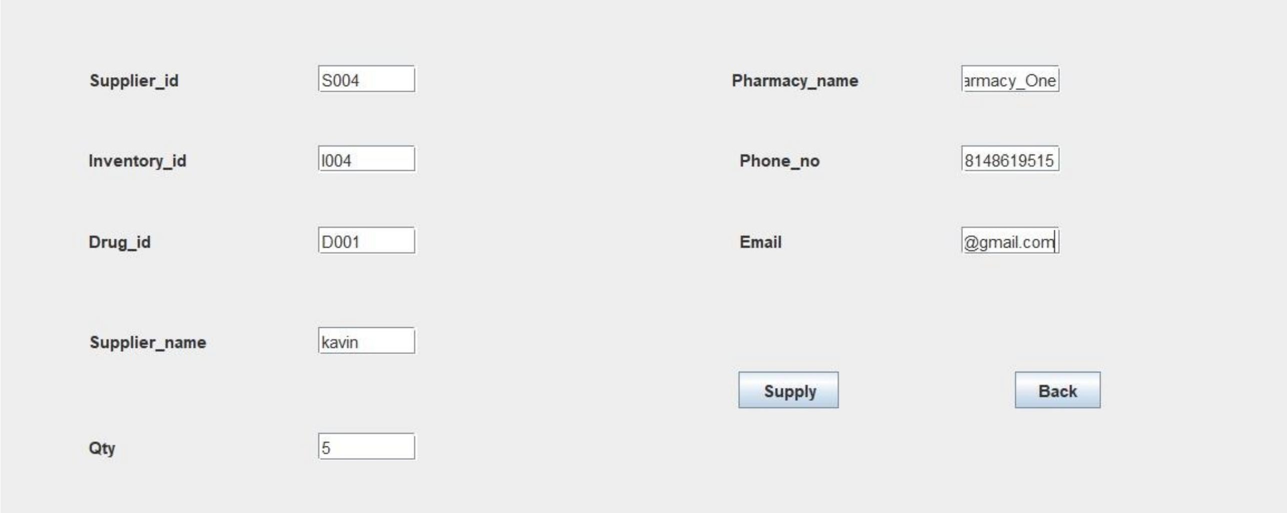
Profile page will display the details of the customer that includes their customer name, user name,password,mail_id,address details,etc.

Customer Profile:



Cust_name	John Doe
Username	john_doe
Password	ıassword123
Email	ꜟxample.com
Door_no	101
City	New York
Pincode	10001

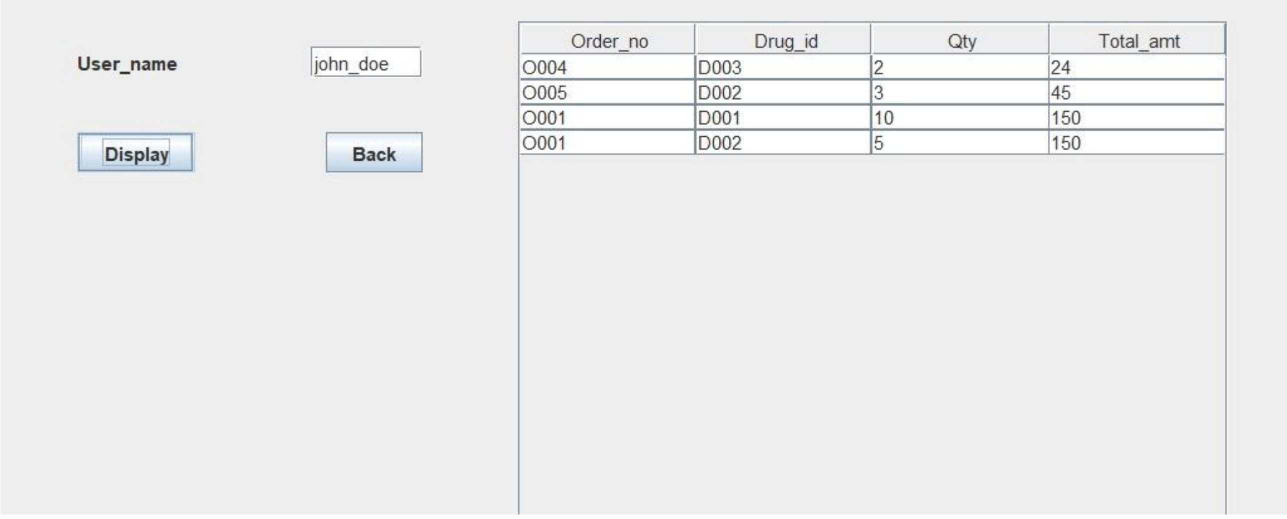
Supplier profile:



Supplier_id	S004	Pharmacy_name	armacy_One
Inventory_id	I004	Phone_no	8148619515
Drug_id	D001	Email	@gmail.com
Supplier_name	kavin		
Qty	5		

Orders Placed:

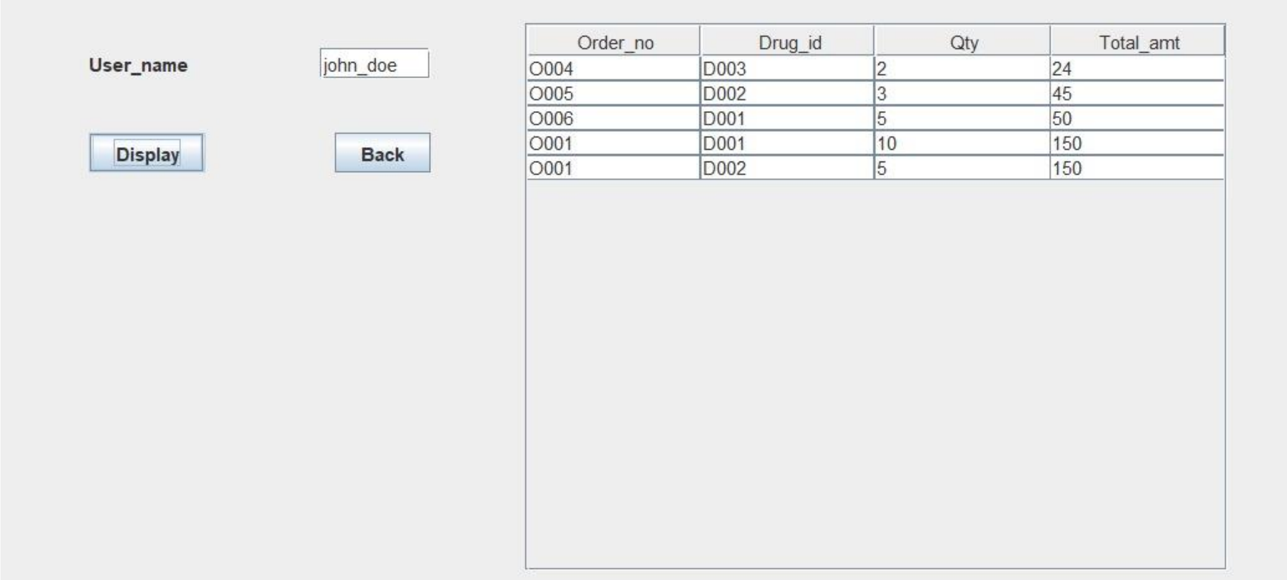
Orders Placed will contain in detail contents of the orders placed by the customer. It provides the details of order_id,drug id of the drug ordered, quantity orders,total amount.



User_name john_doe

Order_no	Drug_id	Qty	Total_amt
O004	D003	2	24
O005	D002	3	45
O001	D001	10	150
O001	D002	5	150

Orders List After Placing New order:

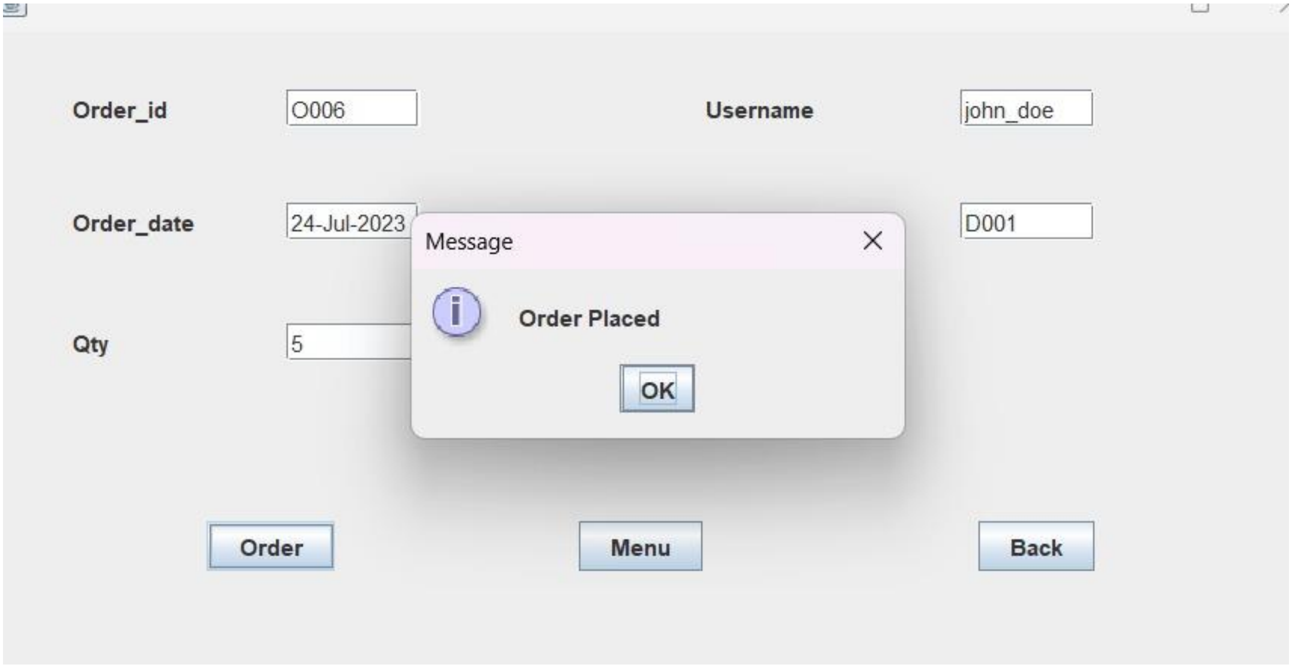


User_name john_doe

Order_no	Drug_id	Qty	Total_amt
O004	D003	2	24
O005	D002	3	45
O006	D001	5	50
O001	D001	10	150
O001	D002	5	150

Placing New order:

Orders page will open the gateway to order new drug where user must give inputs like drug id,quantity to be ordered.



Order_id O006

Username john_doe

Order_date 24-Jul-2023

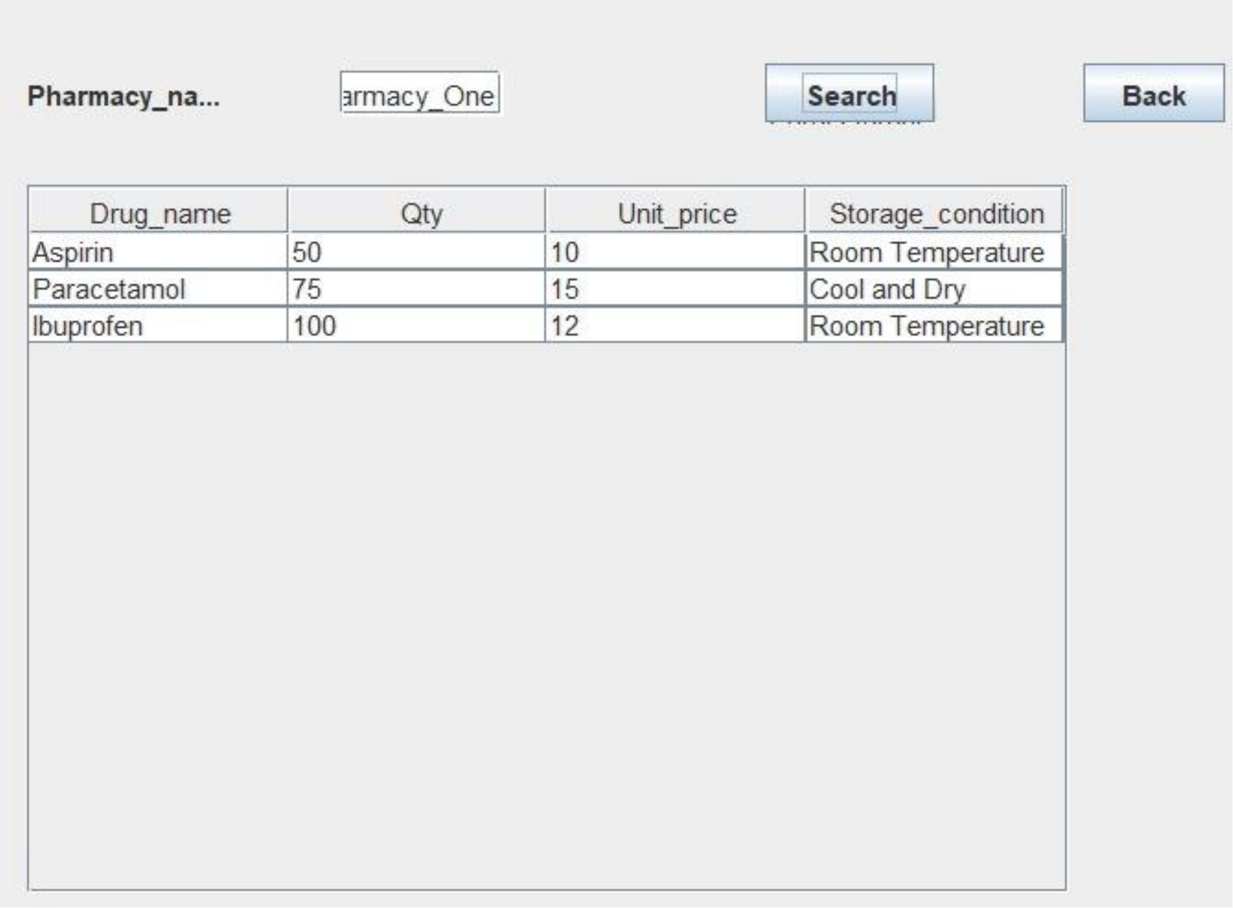
D001

Qty 5

Message

Order Placed

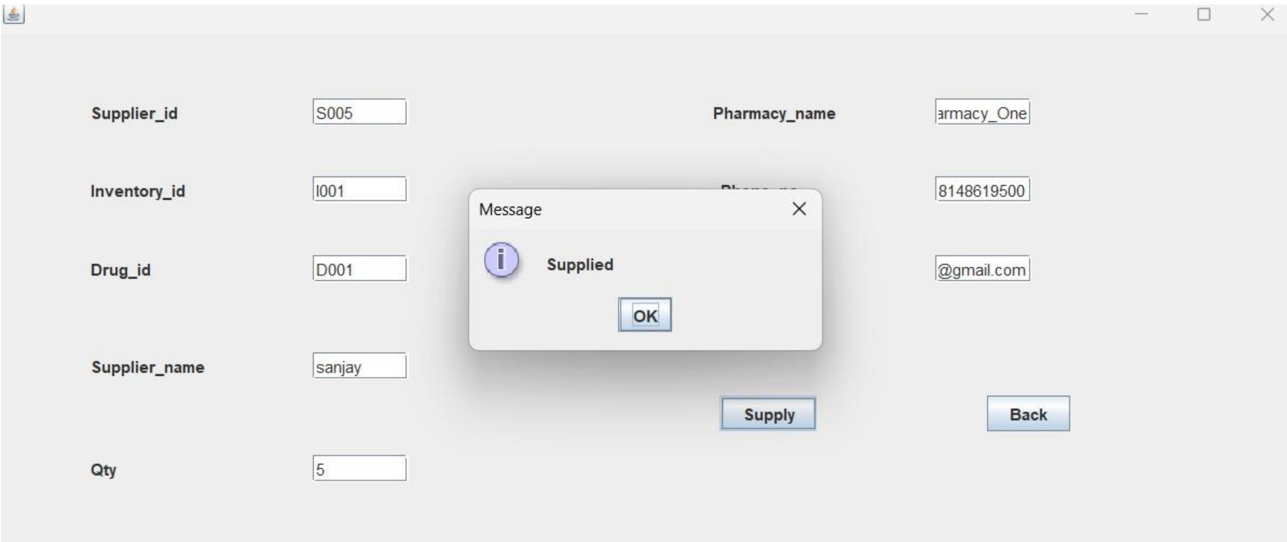
Available Stock:



Pharmacy_na... armacy_One

Drug_name	Qty	Unit_price	Storage_condition
Aspirin	50	10	Room Temperature
Paracetamol	75	15	Cool and Dry
Ibuprofen	100	12	Room Temperature

Supplier Adding New Stock:

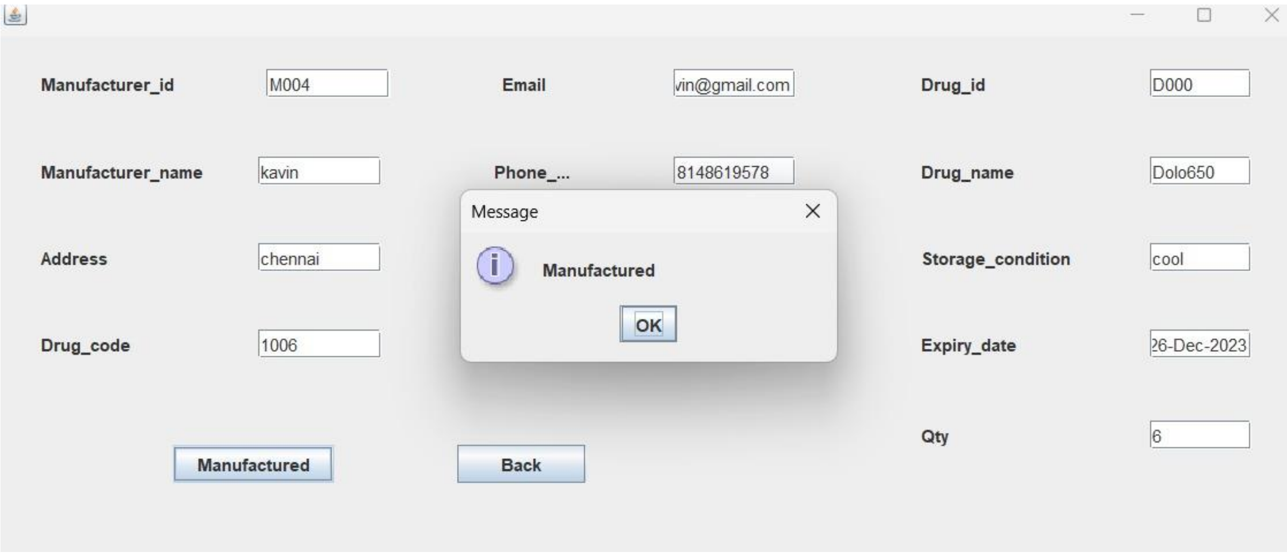


Supplier_id	S005	Pharmacy_name	armacy_One
Inventory_id	I001		8148619500
Drug_id	D001		@gmail.com
Supplier_name	sanjay		
Qty	5		

Message

Supplied

Manufacturer Adding New Drug:



Manufacturer_id	M004	Email	vin@gmail.com	Drug_id	D000
Manufacturer_name	kavin	Phone_...	8148619578	Drug_name	Dolo650
Address	chennai			Storage_condition	cool
Drug_code	1006			Expiry_date	26-Dec-2023
				Qty	6

Message

Manufactured

NOVELTY:

• Deploy the pharmacy management system on a cloud platform to leverage scalability, reliability, and flexibility.

• Use AI to predict future trends, helping the pharmacy to manage inventory more effectively and anticipate demand.

• Analyze feedback and reviews to continually improve services and user satisfaction

• Implement advanced security features to protect user data and ensure secure transactions.

• Integrate the pharmacy management system with external services and platforms for enhanced functionality.