Note: The keys of every table are in bold and underlined.

Delivery Details:

ProductID (FK)	**VendorID** (FK)	**StoreID** (FK)	Cost_per_kg	Quantity_in_kg

Cost_per_packet	Quantity_in_packets	Cost_per_carton	Number_of_cartons	Arrival_date

Departure_date	Payment_status

Not All IN BCNF:

- { ProductID } --> cost_per_kg (1NF)
- { ProductID } --> cost_per_packet (1NF)
- { ProductID } --> cost_per_carton (1NF)
- { ProductID, VendorID, StoreID } --> quantity_in_kg (BCNF)
- { ProductID, VendorID, StoreID } --> quantity_in_packets (BCNF)
- { ProductID, VendorID, StoreID } --> number_of_cartons (BCNF)
- { ProductID, VendorID, StoreID } --> arrival_date (BCNF)
- { ProductID, VendorID, StoreID } --> departure_date (BCNF)
- { ProductID, VendorID, StoreID } --> payment_status (BCNF)

Cost_per_kg, cost_per_packet, cost_per_carton are not dependent on the key therefore they are not in BCNF.

Those dependencies also not have a prime attribute at right side so its not in 3NF.

Cost_per_kg, cost_per_packet, cost_per_carton are all partially dependent on key and not completely so they are not in 2NF also.

The values in them are atomic so they are in 1NF.

Anomalies:

Insertion anomaly – when we want to enter a new record for the same product we need to enter the costs every time. This can also result to redundancy.

Modification anomaly – On modifying cost of one product it may result in having inconsisitent data where same product have different costs.

Normalization:

Using BCNF decomposition algorithm.

We take dependent { productID } --> cost_per_kg, which violates BCNF.

We generate closure for productID

ProductID$^+$ = { productID, cost_per_kg, cost_per_packet, cost_per_carton }

We choose this closure as one relation.

The projected dependencies will be:

- { ProductID } --> cost_per_kg
- { ProductID } --> cost_per_packet
- { ProductID } --> cost_per_carton

All are in BCNF.

The other relation will be

{ VendorID, StoreID, quantity_in_kg, quantity_in_packets, number_of_cartons, arrival_date, departure_date, ProductID }

Projected dependencies will be:

- { ProductID, VendorID, StoreID } --> quantity_in_kg
- { ProductID, VendorID, StoreID } --> quantity_in_packets
- { ProductID, VendorID, StoreID } --> number_of_cartons
- { ProductID, VendorID, StoreID } --> arrival_date
- { ProductID, VendorID, StoreID } --> departure_date
- { ProductID, VendorID, StoreID } --> payment_status

All are in BCNF.

Customer:

CustomerID	Customer_ name	Plot_number	location	city	PIN	Contact_number	Special_ discount

All In BCNF :

- { CustomerID } --> Customer_name
- { CustomerID } --> Plot_Number
- { CustomerID } --> location
- { CustomerID } --> city
- { CustomerID } --> PIN
- { CustomerID } --> contact_number
- { CustomerID } --> special_discount

All the attributes are directly dependent on the key, therefore the relation is in BCNF.

As it is in BCNF it also confirms 3NF, 2NF and 1NF.

Product:

ProductID	Product_ name	brand	Discount_details	Cost_per _kg	MRP	Cost_per _packet	CategoryID (FK)

All In BCNF:

- { ProductID } --> product_name
- { ProductID } --> brand
- { ProductID } --> discount_details
- { ProductID } --> cost_per_kg
- { ProductID } --> MRP
- { ProductID } --> cost_per_packet
- { ProductID } --> CategoryID

All the attributes are directly dependent on the key, therefore the relation is in BCNF.

As it is in BCNF it also confirms 3NF, 2NF and 1NF.

Category:

CategoryID	Category_name

In BCNF:

- { CategoryID } --> category_name

All the attributes are directly dependent on the key, therefore the relation is in BCNF.

As it is in BCNF it also confirms 3NF, 2NF and 1NF.

Store:

StoreID	location	city	PIN	street	PlotNo.

All In BCNF:

- { StoreID } --> location
- { StoreID } --> city
- { StoreID } --> PIN
- { StoreID } --> street
- { StoreID } --> plotNo.

All the attributes are directly dependent on the key, therefore the relation is in BCNF.

As it is in BCNF it also confirms 3NF, 2NF and 1NF.

Employee:

EmployeeID	Full_name	gender	DOB	designation	Shift_start_time

Date_of_hiring	Adhaar_number	Salary_status	supervisorID	isManager

StoreID (FK)	CategoryID (FK)

All In BCNF:

- { EmployeeID } --> full_name
- { EmployeeID } --> gender
- { EmployeeID } --> DOB
- { EmployeeID } --> designation
- { EmployeeID } --> shift_start_time
- { EmployeeID } --> date_of_hiring
- { EmployeeID } --> adhaar_number
- { EmployeeID } --> salary_status
- { EmployeeID } --> supervisorID
- { EmployeeID } --> isManager
- { EmployeeID } --> StoreID
- { EmployeeID } --> categoryID

All the attributes are directly dependent on the key, therefore the relation is in BCNF.

As it is in BCNF it also confirms 3NF, 2NF and 1NF.

Contains:

StoreID (FK)	**ProductID** (FK)	Expiry_date	stock

IN BCNF:

- { StoreID, ProductID } --> expiry_date
- { StoreID, ProductID } --> stock

All the attributes are directly dependent on the key, therefore the relation is in BCNF.

As it is in BCNF it also confirms 3NF, 2NF and 1NF.

Made_on:

TransactionID (FK)	**ProductID** (FK)	quantity

In BCNF:

- { TransactionID, ProductID } --> quantity

All the attributes are directly dependent on the key, therefore the relation is in BCNF.

As it is in BCNF it also confirms 3NF, 2NF and 1NF.

Fills:

EmployeeID (FK)	**Work_date** (FK)	status

In BCNF:

- { EmployeeID, work_date } --> status

All the attributes are directly dependent on the key, therefore the relation is in BCNF.

As it is in BCNF it also confirms 3NF, 2NF and 1NF.

Vendor_contact_number:

VendorID (FK)	**Contact_number**

In BCNF:

- { vendorID , Contact_number } --> vendorID
- { vendorID , Contact_number } --> contact_number

All the attributes are directly dependent on the key, therefore the relation is in BCNF.

As it is in BCNF it also confirms 3NF, 2NF and 1NF.

Employee_contact_number:

EmployeeID (FK)	**Contact_number**

In BCNF:

- { employeeID, contact_number } --> emloyeeID
- { employeeID, contact_number } --> contact_number

All the attributes are directly dependent on the key, therefore the relation is in BCNF.

As it is in BCNF it also confirms 3NF, 2NF and 1NF.