Courier Shipping Management System

Functional Dependencies :-

Branch

Branch -> City	BCNF
Branch -> Pin code	BCNF
Pin Code -> City	BCNF
Pin Code -> Branch	BCNF

Promotion

Promotion_id -> Price	BCNF
Promotion_id -> Tenure	BCNF
Promotion_id -> Company_name	BCNF

Vehicle

Vehicle_id -> distance	BCNF
Vehicle_id -> capacity	BCNF
Vehicle_id -> Cities	NOT In 1NF

Vehicle Promotion Mapping

{Promotion_id,Vehicle_id} -> Start Date	BCNF
{Promotion_id,Vehicle_id} -> End Date	BCNF

Employee

Employee_id -> name	BCNF
Employee_id -> Contact_number	BCNF
Employee_id -> Date of Joining	BCNF
Employee_id -> Role	BCNF
Employee_id -> Branch_id	BCNF
Contact_number -> Employee_id	BCNF
Contact_number -> name	BCNF
Contact_number -> Date of Joining	BCNF
Contact_number -> Role	BCNF
Contact_number -> Branch_id	BCNF

Driver

Id -> rating	BCNF
Id -> Vehicle_id	BCNF
Vehicle_id -> Id	BCNF
Vehicle_id -> rating	BCNF

Clerk

Id -> Accuracy	BCNF

IT_Support

Id -> Complaint_per_hour	BCNF

Customer

Customer_id -> name	BCNF
Customer_id -> contact_of_sender	BCNF
Customer_id -> contact_of_receiver	BCNF
Customer_id -> street_address	BCNF
Customer_id -> Branch_id	BCNF

Courier

Reference_id -> price	BCNF
Reference_id -> date	BCNF
Reference_id -> distance	BCNF
Reference_id -> weight	BCNF
Reference_id -> Expected_delivery_date	BCNF
Reference_id -> Status	BCNF
Reference_id -> Type	BCNF
Reference_id -> Customer_id	BCNF
Reference_id -> Vehicle_id	BCNF
Reference_id -> Branch_id	BCNF
Reference_id -> To_branch	BCNF

Payment

Payment_id -> date	BCNF
Payment_id -> status	BCNF

Payment_id -> method	BCNF
Payment_id -> amount	BCNF
Payment_id -> reference_id	BCNF
Reference_id -> Payment_id	BCNF
Reference_id -> date	BCNF
Reference_id -> status	BCNF
Reference_id -> method	BCNF
Reference_id -> amount	BCNF

Feedback

Feedback_id -> Date	BCNF
Feedback_id -> comment	BCNF
Feedback_id -> rating	BCNF
Feedback_id -> customer_id	BCNF

Delivery Partner

Registration_number -> Name	2NF
Registration_number -> Price_per_courier	2NF
Registration_number -> Start_Date	2NF
Registration_number -> End_Date	2NF
Reference_id -> Regestration_number	BCNF
Reference_id -> Name	BCNF
Reference_id -> Price_per_courier	BCNF
Reference_id -> Start_Date	BCNF
Reference_id -> End_Date	BCNF

Insurance

Insurance_id -> Type	BCNF
Insurance_id -> Coverage_Amount	BCNF
Insurance_id -> Premium	BCNF
Insurance_id -> Claim_amount	BCNF
Insurance_id -> Reference_id	BCNF
Reference_id -> Insurance_id	BCNF
Reference_id -> Type	BCNF
Reference_id -> Coverage_Amount	BCNF
Reference_id -> Premium	BCNF
Reference_id -> Claim_amount	BCNF

Works_on

{Employee_id,Reference_id} -> Assigned_Date	BCNF

Path we Followed to achieved at BCNF Design:-

1) In the Vehicle Table our cities attribute was a multivalued attribute.So in one tuple we were storing more than one cities it caused our functional dependency (Vehicle_id -> Cities) to not to be in 1NF. So we converted it into BCNF using the Decomposition Algorithm as below:

   Here,

   X = Vehicle_id
   Y = Cities
   S = { Vehicle_id,distance, capacity, Cities }

   X^+ = S1 = {Vehicle_id, Cities}

   S2 = {(S - X^+) U X}

S2 = { Vehicle_id,distance, capacity }

- Now, S1 and S2 both are in BCNF.
- Here, S1 has Composite Primary Key of Vehicle_id & Cities and S2 has Vehicle_id as Primary Key.

2) In the Delivery_Partner Table our functional dependency

(Registration_number -> Name) was in 2NF.

Now, we will apply the Decomposition Algorithm on Delivery_Partner Table as below:

Here,

X = Registration_number

Y = Name

S = {Registration_number,Name,Price_per_courier,Start_Date,End_Date,

Reference_id}

X^+ = S1 = {Registration_number,Name,Price_per_courier,Start_Date,End_Date}

S2 = {(S - X^+) U X}

S2 = {Registration_number , Reference_id}

- Now, in table S1 (Registration_number -> Name) became BCNF.
- And the other three (Registration_number -> Price_per_courier, Registration_number -> Start_Date,Registration_number -> End_Date) also became BCNF.
- Now, S1 and S2 both are in BCNF.
- Here, S1 has Primary Key of Registration_number and S2 has Reference_id as Primary Key.