Apply the Functional Dependency and Normalize to INF

Step 1:- Identify Functional Dependencies (FDs).

Consider the following relations and FDs:

OrderTable (Order_ID, Cust_ID, order_Date, Order_Total, Payment_Status)

• FD1: order_ID → Cust_ID, Order_Date, Order_Total, Payme-nt_Status.

Customer (Cust_ID, Cust_Name, Cust_Contact, Cust_Email

• FD2: ~~Order~~ Cust_ID → Cust_Name, Cust_Contact, Cust_Email, Cust_Address.

Menu_Item (Item_ID, Item_Name, Price_Category, Rest_ID)

• FD3: Item_ID → Item_Name, Price, Category, Rest_ID.

Normalization to INF (First Normal Form)

• Ensure that each column contains only atomic ~~val~~ values.

• Remove any repeating groups.

• Example:-

Order_ID	Cust_ID	order_Date	Order_Total	Payment_Status
1	1	2025-01-20	800	Paid
2	2	2025-01-21	500	unpaid.

TASK-8 - Normalizing Database using Functional Depend-cies up to BCNF

Objective:-

To normalize the database created in TASK-2 using functional dependencies (FDs) and apply normalization.

2. Normalize the Relations using FD+ and α+

• Compute FD+ (Closure of FDs) using Armstrong's Axioms.

• Identify minimal keys and remove redundant FDs.

Closure for orderTable:-

• FD+: {order_ID → Cust_ID, order_Date, order_Total, Payme-nt_Status}.

Closure for customer:-

• FD: {Cust_ID → cust_Name, Cust_Contact, Cust_Email, Cust-Address}

Closure for Menu_Item:-

• FD+ {Item_ID → Item_Name, Price, Category, Rest_ID}

3. Find the minimal Cover and Canonical Cover:-

Minimal Cover:

• FD1: Order_ID → Cust_ID, order_Date, order_Total, Payment_sta

• FD2: Cust_ID → Cust_Name, Cust_Contact, Cust_Email, Cust_Addre

• FD3: Item_ID → Name, Price, Category, Rest_ID.

Canonical Cover:-

• No redundancy detected.

4. Normalize to 2NF:

- A relation is in 2NF if it is in 1NF and has no Partial dependencies.
- Remove partial dependencies by creating separate relations.

Normalization to 2NF:-

- OrderTable (Order_ID, Order_Date, Order_Total, Payment_Status)
- Customer (Cust_ID, Cust_Name, Cust_Contact, cust_Email, Cost_Address)
- Menu_Item (Item_ID, Item_Name, Price, Category, Rest_ID).

5. Normalize to BCNF:

- A relation is in BCNF if, for every functional dependency $(X \rightarrow Y)$, X is a Super Key.
- Identify and remove transitive dependencies.

Normalization to BCNF:-

- OrderTable (order_ID, cost_ID, order_Date, Order_Table, Payment_Status).
- Customer (Cust_ID, Cost_Name, Cost_Contact, Cost_Email, cost_Address)
- Menu_Item (Item_ID, Item_Name, Price, Category, Rest ID).

6. Normalize to 3NF:-

- A relation is in 3NF if it is in 2NF and has no transitive dependencies.
- Ensure non-Prime attributes depend only on Primary Keys.

Normalization to 3NF:

- Restaurant (Rest ID, Rest_Name, Rest_location, Rest_Contact).

VEL TECH	
EX No.	8
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	3
VIVA VOCE (5)	3
RECORD (5)	4
TOTAL (25)	15
SIGN WITH DATE	

Result:- Hence, Normalizing Database using functional dependences up to BCNF done successfully.