Apply the Functional Dependency and Normalize to 1NF.

Step1: 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, payment_status.

customer(cust_ID, cust_Name, cust_contact, cust_email.

*FD2: 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 1NF (First Normal Form)

* ensure that each column contains only atomic 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 dependencies UPTO BCNF

objective :-

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

2. Normalize the relations using FD+ and d+

* 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, payment_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_status.

* FD2: cust_ID → cust_Name, cust_contact, cust_email, cust_address.

* FD3: Item_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 :-

* order table (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→y), x is a super key.

* Identify and remove transitive dependencies.

Normalization to BCNF :-

* OrderTable (orderID, cust-ID, order-Date, order-Table, payment-status).

*customer (cust-ID, cost-Name, cust-contact, cust-email, cust-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).

* Menu-Item (Item-ID, Item-Name, price, category, Rest-ID)

VELTECH	
EX No.	3
PERFORMANCE (5)	5
RESULT AND ANALYSIS (3)	3
VIVA VOCE (3)	3
RECORD (4)	4
TOTAL (15)	15
WITH DATE	

Result :-

Thus the study of normalizing database using functional dependencies upto BCNF is successfully verified.