TASK-8 Normalizing database using functional dependencies upto BCNF

Aim: To determine Possible functional dependencies (fds)

cust-ID -> cust-Name cust-Phone No, cust-city

cust-Arrray-Pad

Bill-ID -> Price, cust-ID

Phone-ID -> model-Name

Admin-ID -> Password

convert to INF

Already in INF

All Attributes have atomic sigle values; there are no repeating groups

convert to 2NF

check for Partial dependencies - they occur only if there is is composite key

Since all table have single attributes Primary keys cust-ID, Bill-ID, Phone-ID Admin-ID),

convert to 3ND :-

1. Ensure there are no transitive dependencies
2. Move non-key attributes to seperate tables if they depend on another key

cust-ID -> cust-Name, cust-Phone No

cust-city, cust-Amount Paid

convert to BCNF

1. Ensure every determinant is a candidate key
2. check for overlapping candidate keys
3. Decompose relations to eliminate redundancy

No further decomposition needed. using Griffith tool.

1. Input relational schema and functional dependecies
2. Griffith the graph to Identify normalization issues
3. Analyze the graph to Identify normalization the schema
4. Apply normalization rules to transform the schma
5. Verify the resulting schema meets BCNF criteria.

Griffith Tool steps:

1. create a new Project in griffith.
2. Define the relational schema and FDS
3. Run the dependency Graph Tool.
4. Analyze the graph for normalization issues
5. Apply information using the "Normalize" tool
6. Verify BCNF compliance using "BCNF check tools

Normalized schema:

Customer (cust-ID PK, cust-Name, cust-Phone.No. cust-city, cust-Amount Paid)

Bill (Bill-ID AK, Price, cust-ID FK -> customer cust-ID

MOBILE (Phone-ID PK, model-Name, modelPrice)

LOGIN (Admin-ID PK, Password)

VEL TECH - CSE	
EX NO.	7
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	
TOTAL (20)	15
SIGN WITH DATE	

30/9/16

Result: Thus the implemention of normalizing database dependencies has been successfully Executed