

Task-8: Normalising database using functional
Date: 30/9/25. Dependencies upto BCNF

Aim: In Perform normalization upto BCNF Based on given dependencies.

Banking Database:

1. Identify Banking attributes:- customer, Account, Branch, Banker Info, Loan, credit-card.
2. Relational schema:- Banking (customer Account, Branch, Banker Info, Loan, credit-card).
3. Functional dependencies (FD's between Attributes):
 - customer ID \rightarrow Name, Address, Ph-no.
 - Account-number \rightarrow Account-name, category.
 - Branch-ID \rightarrow Branch Name, Location, ifsc-code.
 - Banker-ID \rightarrow Banker-name, ph-no.
 - customer-ID \rightarrow Account-number.
 - Loan-ID \rightarrow Loan-Amount
 - customer-ID \rightarrow Loan-ID.

Step-2: Convert to 1NF:

- * No repeating groups or Arrays.
- * All attributes are atomic

The schema is in 1NF.

Step-3: Convert to 2NF

- * All primary Keys are single-column keys, so no partial dependencies exist.
- * However, we ensure foreign key attributes are managed correctly.

output: The schema is already in 2NF.

Step-4: Convert it to 3NF.

Eliminate Transitive dependencies:

- * customer-ID \rightarrow Account-number \rightarrow Loan-ID.
 \rightarrow move Loan-ID to a separate loan table.
- * customer-ID \rightarrow name, Address, ph-no.
 \rightarrow Already in separate users table

* Account-number \rightarrow customer-ID \rightarrow Branch ID.
 \rightarrow No redundancy.

All transitive dependencies removed.

Step-5: Convert to BCNF.

check if every determinant is a Candidate Key.

* customer-ID, Account-number, Branch-ID, loan-ID are all unique keys for their respective tables.

* Foreign Keys like customer-ID, do not violate BCNF Rules.

All FD's comply with BCNF - no further decomposition needed.

Using Griffith Tool:

1. Input relational schema and functional dependencies
2. Griffith tool generates a dependency graph.
3. Analyze the graph to identify normalization issues.
4. Apply normalization to transform schema.
5. Verify the resulting schema meets BCNF criteria.

Griffith tool steps:

1. create a new Project in Griffith.
2. Define the relational schema and FD's.
3. Run the "dependency graph" tool.
4. Analyze the graph for normalization issues.
5. Analyze the graph for normalization issues.
6. Apply Transformations using the 'Normalize' tool.
7. Verify BCNF Compliance using "BCNF check" tool.

Normalization schema:

customer (customer-ID, Name, ph-no).

Account (Account-number, Account-name, category).

Branch (Branch-ID, Branch-name, Location,
IFSC-code).

loan (loan-ID, customer-ID, Account).

credit-card (credit-card number, customer-ID, limit).

Step 1: Backup Database

BACKUP DATABASE [Banking] TO DISK = 'backup.mbak'
WITH NOFORMAT, NOINIT, NAME = 'Full Database Backup',
SKIP, REWIND, NOONLINE, STATS = 0

Step 2: Create Incremental Backup

BACKUP DATABASE [Banking] TO DISK = 'Incremental
Backup.bak' WITH DIFFERENTIAL, NOFORMAT, NOINIT,
NAME = 'Incremental Database Backup', SKIP, NOONLINE,
STATS = 0

Step 3: Simulate Data loss

Intentionally delete data

Step 4: Restore Database

RESTORE DATABASE [Banking]

WITH REPLACE

Step 5: Verify Database Backup

VEL TECH	
EX No.	8
PERFORMANCE (5)	5
RESULT AND ANALYSIS (3)	5
VIVA VOCE (3)	4
RECORD (4)	14
TOTAL (5)	20
SIGNATURE DATE	15/10

Result: Thus, the implementation of normalizing the database upto BCNF Based on given dependency was successfully executed.