11/9/24

Task 8: Normalizing database using functional dependencies upto BCNF

[Tool: GV / Table Normalization Tool, ALM: Jigsaw) cos, k3.

upon relational tables created in task 2.2 perform based on given Dependence as following for the assumed relations specified below.

Employee Database:

1. Identify employee attributes: Employee_ID, Name, Department, Job-Tittle, manager_ID, Hire-Date, Salary.

2. Define relation schema: Employee (ID, Name, Dept, Job_tittle, mager, Salary).

3. Determine functional dependencies:

- Employee_ID -> Name, Department, Job-tittle, manager_ID, Hire-Date, Salary.
- Department -> manager_ID
- manager_ID -> Name.

Step 2 convert to 1NF

1. Elimination repeating groups of arrays
2. Create Seperate tables for each repeating group

Step 3: Convert to 2NF

1. Ensure each Non-key attribute dependent on the
2. move non-key attribute if they depend on only part

Step 4: Convert to 3NF.

1. Ensure there are no transitive dependencies
2. move non-key attribute to seperate table if they depend on another non-key attribute.

Step 5: Convert to BCNF

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

using Griffin tool

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

Griffith Tool Steps

1. Create a new project in Griffith.
2. Define the relational schema & FDs.
3. Run the "Dependency Graph" tool.
4. Analyze the graph for normalization
5. Apply transform using the normalization tool

Normalized Schema:

1. Employee (Emp-ID, Name, Dept-ID, Job-title, Salary).
2. Department (Department-ID, manager-ID).
3. manager (manager-ID, Name).

Result:- Thus, the program has been executed successfully. 18/9/25

sk 8:- Normalizing database using functional dependencies upto BCNF

FUNCTIONAL DEPENDENCY :

Attributes in Table

Separate attributes using a comma (,)

employee_id, name, department, job_title, manager_id, hire_date, salary

Functional Dependencies

employee_id × → name × department × job_title × hire_date × manager_id × salary ×

Delete

Add Another Dependency

NORMAL FORM :

Check Normal Form

2NF
The table is in 2NF

3NF
The table is in 3NF

BCNF
The table is in BCNF

Show Steps

2NF

find all candidate keys. The candiates keys are { employee_id}, The set of key attributes are: { employee_id }
for each non-trivial FD, check whether the LHS is a proper subset of some candidate key or the RHS are not all key attributes
checking FD: employee_id --> name,department,job_title,hire_date,manager_id,salary

3NF

find all cadnidate keys. The candiates keys are { employee_id}, The set of key attributes are: { employee_id }
for each FD, check whether the LHS is superkey or the RHS are all key attributes
checking functional dependency employee_id --> name,department,job_title,hire_date,manager_id,salary

BCNF

A table is in BCNF if and only if for every non-trivial FD, the LHS is a superkey.

CONVERT 2NF :

Normalize to 2NF

Attributes

employee_id | name | department | job_title | manager_id | hire_date | salary

Functional Dependencies

employee_id → name department job_title hire_date manager_id salary

Show Steps

First, find the minimal cover of the FDs, which includes the FDs
employee_id --> name
employee_id --> department
employee_id --> job_title
employee_id --> hire_date
employee_id --> manager_id
employee_id --> salary

Initially rel[1] is the original table:

Round1: checking table rel[1]

***** The table is in 2NF already, send it to output *****

CONVERT 3NF :

1NF to 3NF

Attributes

employee_id | name | department | job_title | manager_id | hire_date | salary

Functional Dependencies

employee_id → name
employee_id → department
employee_id → job_title
employee_id → hire_date
employee_id → manager_id
employee_id → salary

Show Steps

Table already in 3NF

CONVERT BCNF :

Normalize to BCNF

Attributes

employee_id | name | department | job_title | manager_id | hire_date | salary

Functional Dependencies

employee_id → name department job_title hire_date manager_id salary

Show Steps

Table already in BCNF, return itself.

Result:- Thus, the program has been executed successfully.

VEL TECH	
EX No.	8
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	20
SIGN WITH DATE	

18/4/25