

11-9-25

dependencies upto BCNF

Task-8: Normalization databases using functional dependencies upon relational tables created in Task-2, Perform normalization upto BCNF based on given dependencies as following:

Employee Database:

- ① Identify employee attributes: Employee-ID, Name, Department, Job-Title, Manager-ID, Hire-Date, Salary
- ② Define relational schema: Employee
- ③ Determine functional dependencies (FDs) between attributes
  - Employee-ID  $\rightarrow$  Name, Department, Job-Title, Manager-ID, Hire-Date, Salary
  - Department  $\rightarrow$  Manager-ID
  - Manager-ID  $\rightarrow$  Name

Step 2: Convert to 1NF

- ① Eliminate repeating groups or arrays
- ② Create separate tables for each repeating group

Step 3: Convert to 2NF

- ① Ensure each non-key attribute depends on the entire primary key
- ② Move non-key attributes to separate tables if they depend on only part of the primary key
  - Create Department table: Department(Department-ID, Manager-ID, Name)
  - Create Employee table: Employee(Employee-ID, Name, Dept-ID, Job-Title, Salary, Hire-Date)

Step 4: Convert to 3NF

- ① Ensure there are no transitive dependencies
- ② Move non-key attributes to separate tables if they depend on another non-key attribute
  - Create Manager table: Manager(Manager-ID, Name)
  - Update Department table: Department(Department-ID, Manager-ID)

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
- 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 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 and FDs
- ③ Run the "Dependency Graph" tool.
- ④ Analyze the graph for normalization issues.
- ⑤ Apply transformations using the "Normalize" tool.
- ⑥ Verify BCNF compliance using the "BCNF check" tool.

11-2-25

## Task - 8

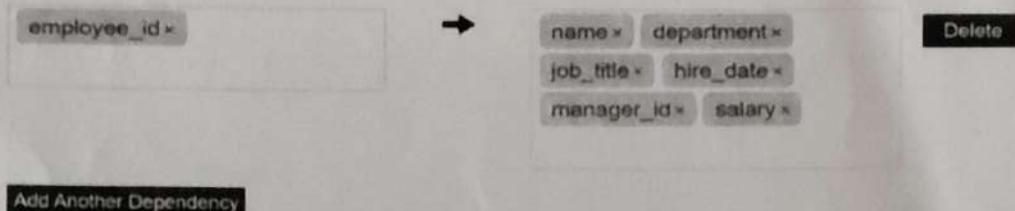
### FUNCTIONAL DEPENDENCY :

#### Attributes in Table

① Separate attributes using a comma ( , )

employee\_id, name, department, job\_title, manager\_id, hire\_date, salary

#### Functional Dependencies



### 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 candidate 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  $\rightarrow$  name,department,job\_title,hire\_date,manager\_id,salary

##### 3NF

find all candidate keys. The candidate 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  $\rightarrow$  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  $\rightarrow$  name  
employee\_id  $\rightarrow$  department  
employee\_id  $\rightarrow$  job\_title  
employee\_id  $\rightarrow$  hire\_date  
employee\_id  $\rightarrow$  manager\_id  
employee\_id  $\rightarrow$  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



False primary key for BCNF relation test

VEL TECH	
PERFORMANCE (5)	6
PROJECT AND EVALUATION (5)	5
PHYSIQUE (6)	5
RECORD (5)	20
PAI (29)	20
WITH DATE	

Dinesh  
18/4/2016

18/4/2016

Normalized Schema.

- ① Employee (Employee\_ID, Name, Department\_ID, Job\_Title,  
Hire\_Date, Salary)
- ② Department (Department\_ID, Manager\_ID)
- ③ Manages (Manages\_ID, Name,

VISUALS	DATA
3 NO	✓
PERFORMANCE (P)	✓
RESULT AND ANALYSIS (R)	✓
DATA VOICE (D)	✓
RECORD (R)	✓
TOTAL (T)	✓
DATA WITH DATES	✓

Result: Thus the Normalization database using  
Functional dependencies upto BCNF has been  
executed successfully.