Bernsteins's Algorithm on Employees Table

We will be using Bernsteins' algorithm to build/check the schema for Employees. We will start with only a list of attributes that belong to the table and a list of functional dependencies in the table.

Step 1:

List of attributes:

- UserId
- BranchID
- EmployeeMonthlySalary
- EmployeeSINNumber
- EmployeePermissions

F = {

UserID -> {BranchID, EmployeeMonthlySalary,EmployeeSINNumber EmployeePermissions},

EmployeeSINNumber -> UserID,

}

Step 2a:

F = {

UserID -> BranchID,
UserID -> EmployeeMonthlySalary,
UserID -> EmployeeSINNumber
UserID -> EmployeePermissions,
EmployeeSINNumber -> UserID,

}

Now we perform closure on the determinant of each functional dependency to remove redundant FD's

1. We determine that **UserID ->** BranchID is not a redundant FD. If we modify **F** by removing **UserID ->** BranchID (**MF**) and then subsequently calculate the closure of UserID using **MF**, BranchID is not in that closure.
2. We determine that **UserID ->** EmployeeMonthlySalary is not a redundant FD. If we modify **F** by removing **UserID ->** EmployeeMonthlySalary (**MF**) and then subsequently calculate the

closure of UserID using **MF**, EmployeeMonthlySalary is not in that closure.

3. We determine **UserID ->** EmployeeSINNumber is not a redundant FD. If we modify **F** by removing **UserID ->** EmployeeSInNumber (**MF**) and then subsequently calculate the closure of UserID using **MF**, EmployeeSINNumber is not in that closure.

4. We determine **UserID ->** EmployeePermissions is not a redundant FD. If we modify **F** by removing **UserID ->** EmployeePermissions (**MF**) and then subsequently calculate the closure of UserID using **MF**, EmployeePermissions is not in that closure.

Step 2b:

In step 2b we will check if each FD itself is redundant.

Here is our current F:

F = {

UserID -> BranchID,
UserID -> EmployeeMonthlySalary,
UserID -> EmployeeSINNumber
UserID -> EmployeePermissions,
EmployeeSINNumber -> UserID,

}

Checking each FD involves the following steps:

1. Eliminate an attribute A on the LHS of one of the FDs
2. Look at the remainder Q of attributes on the LHS for that FD
3. Find Q+ in the original set of FDs
4. If Q+ contains the RHS of the FD in question then the attribute A is redundant

However, this process is unnecessary because each determinant in our current F is only 1 attribute. Therefore, no FD's are redundant and step 2b is completed implicitly

Step 3:

Now we must determine the keys. An attribute is a key if its closure is every attribute initially listed in Step 1.

UserID + = {UserID, BranchID, EmployeeMonthlySalary,EmployeeSINNumber EmployeePermissions}

EmployeeSINNumber + = {UserID, BranchID, EmployeeMonthlySalary,EmployeeSINNumber EmployeePermissions}

Therefore, UserID and EmployeeSINNumber are the keys. We don't need to check if the subsets of {UserID} and {EmployeeSINNumber} are keys.

Step 4:

We combine the FD's with the same left hand side:

F = {

UserID -> BranchID,
UserID -> EmployeeMonthlySalary,
UserID -> EmployeeSINNumber
UserID -> EmployeePermissions,
EmployeeSINNumber -> UserID,

}

Below if the final F after combining:

F = {

UserID -> {BranchID, EmployeeMonthlySalary,EmployeeSINNumber EmployeePermissions},

EmployeeSINNumber -> UserID,

}