ECE 356 Winter 2019: Lab2 Part1

(1) decompose it, per BCNF

Employee (empID, empName, job, deptID, salary)

The empName is the full name of each employee, and consists of a First and Last Name, and optionally a Middle Name. This violates the INF. So the Employee table should have three extra columes (firstName, lastName and middleName) to replace the empName.

The decomposed Employee that under BCNF will be:

EmployeeBCNF (empID, firstName, lastName, middleName, job, salary)

deptID shouldn't be in the Employee table or else it will be redundent because there is only one candidate key (empID) and an employee can be in more than one department. We can't just discard the deptID, because it will result data loss. So we need a new table, EmployeeDept, in order to look up deptID for employees.

New EmployeeDept table:

EmployeeDept (empID, deptID)

Project (projID, title, budget, funds)

The Project table seems fine and under BCNF with the candidate key, projID. Title, budget and funds only depends on projID. So no decomposition is needed.

Project (projID, title, budget, funds)

Assigned (empID, projID, role)

The Assigned table seems fine and under BCNF with the candidate key, {empID, projID, role}. Although it seems that the candidate key is a trivial key (consists of all columns in that table), the candidate key will only be unique when all three columns are included. So no decomposition is needed.

Assigned (empID, projID, role)

Department (deptID, deptName, location)

location shouldn't be in the Department table or else BCNF will be violated because there is only one candidate key (deptID) and a department may have multiple locations. So location column should be removed and form a separated table.

The decomposed Department that under BCNF will be:

DepartmentBCNF (deptID, deptName)

We know that a department location is a full address, comprising the street number, street name, city name, province, and postal code. However, postal code is partially dependent on city name and province, they cannot be in the same table (violates BCNF) and need to be decomposed into a separate table other than Location.

New Location table:

Location (locationID, streetName, streetNumber, postalCode)

New PostalCode table:

PostalCode (postalCode, cityName, province)

We can't just discard the location from Department table, because it will result data loss. So we need a new table, DepartmentLocation, in order to link deptID and location together.
New DepartmentLocation table:
DepartmentLocation (deptID, locationID)

After decomposition:

EmployeeBCNF

empID INT(11),
firstName VARCHAR(100),
lastName VARCHAR(100),
middleName VARCHAR(100),
job VARCHAR(100),
salary INT(11)

EmployeeDept

empID INT(11),
deptID INT(11)

Project

projID INT(11),
title VARCHAR(100),
budget INT(11),
funds INT(11)

Assigned

empID INT(11),
projID INT(11),
role VARCHAR(100)

DepartmentBCNF

deptID INT(11),
deptName VARCHAR(100)

Location

locationID INT(11),
streetName VARCHAR(100),
streetNumber VARCHAR(100),
postalCode VARCHAR(100)

PostalCode

postalCode VARCHAR(100),
cityName VARCHAR(100),
province VARCHAR(100)

DepartmentLocation

deptID INT(11),
locationID INT(11)

(2) determine primary keys

primary keys:
EmployeeBCNF = {empID}
EmployeeDept ={ empID, deptID }
Project = { projID }
Assigned = { empID, projID, role}
DepartmentBCNF = { deptID }
Location = { postalCode }
PostalCode = { postalCode }
DepartmentLocation = { deptID, locationID }

(3) determine foreign keys

foreign keys:
EmployeeDept.empID -> EmployeeBCNF.empID
EmployeeDept.deptID -> DepartmentBCNF.deptID
Assigned.empID -> EmployeeBCNF.empID
Assigned.projID -> Project.projID
Location.postalCode -> PostalCode.postalCode
DepartmentLocation. deptID -> DepartmentBCNF. deptID
DepartmentLocation.locationID -> Location.locationID