CALIFORNIA STATE UNIVERSITY, SAN BERNARDINO
SCHOOL OF COMPUTER SCIENCE & ENGINEEERING

CSE 572-F2013 : PART 2B PROJECT SPECIFICATIONS – NORMALIZATION
Due: November 4 (Monday, by 6 PM)

Normalization is a formal technique for analyzing relations based on their primary key (or candidate keys) and functional dependencies. The technique involves a series of rules that test individual relations so that a database can be normalized to any degree. For this project, we are to normalize up to Third Normal Form (3NF) only. As normalization progresses, the relations become more well-structured and therefore less vulnerable to anomalies.

This document explains what needs to be done for Part 2B – Normalization.

WHAT TO DO

You will need to use both corrected Relational Model from Part 2A and the Revised Conceptual Model from Part 1 and the Business Rules from the project option description/specifications.

1. Using the corrected Relational Model and the constraints described in the description for the project option that you have chosen,
   a. list all the functional dependencies
   b. Cite either the constraints and or ERD/EERD part that you used to arrive at the functional dependencies.
   c. Make sure that each relation or table is in 3NF.
   If relation/table is already in 3NF justify. If not, identify the highest normal form of the relation/table and convert to 3NF. Describe the details for normalization done for each table.

Example1: EMPLOYEE Relation/Table

Fname	Minit	Lname	Sex	SSN	Address	Salary	Bdate	***Dno***	***Superssn***

Primary Key: SSN
Prime Attributes: SSN
Non-prime Attributes: Fname Minit Lname Sex Address Salary Bdate ***Dno Superssn***
FDs:

SSN → Fname, Minit, Lname, Sex, Address, Salary, Bdate
From the project description: We store each employee's name, social security number, address, salary, sex and birthdate; SSN is the primary key attribute that uniquely identifies an employee.

SSN → Dno
From the business rule: An employee is assigned to one department and the 1:N WORKS_FOR relationship between EMPLOYEE and DEPARTMENT

SSN → Superssn
From the project description: We also keep track of the direct supervisor of each employee and from 1:N SUPERVISES relationship between EMPLOYEE(Supervisor) and EMPLOYEE(Supervisee)

Normalization Explanation:

The table EMPLOYEE has only one CK (SSN) which becomes the PK.
The composite attribute Name of EMPLOYEE has already been converted to RM as three separate attributes Fname, Minit, Lname. There are no multivalued attributes in EMPLOYEE. **Thus EMPLOYEE is in 1NF.**

Since the only CK of EMPLOYEE has only one attribute, then it is already in **2NF**. Partial functional dependency cannot exist!

There is no transitive dependency that involves the non-prime attributes of EMPLOYEE; also in all FDs, the LHS of each FD is a full key. Thus EMPLOYEE is in **3NF**.

Example2: EmpProj Relation/Table

Ssn	Pnumber	Hours	Ename	Pname	Plocation

Primary Key: composite (SSn,Pnumber)
Prime Attributes: Ssn, Pnumber
Non-prime Attributes: Hours Ename Pname Plocation
FDs:

FD1: Ssn,Pnumber → Hours, Ename, Pname,Plocation
From the primary key constraint

FD2: Ssn,Pnumber → Hours
From business rule, 'We keep track of the ***number of hours per week than an employee works on each project' and from the N:M WORKS_ON relationship between EMPLOYEE and PROJECT.***

FD3: Ssn → Ename
From business rule, 'We store each ***employee's name, social security number***, address, salary, sex, and birthdate.

FD4: Pnumber → Pname,Plocation
From business rule, 'A **project *has a unique name*, a unique number, and *a single location'.***

Normalization Explanation:

*EmpProj is in **1NF** since there are no multivalued nor composite attributes in EmpProj.*
*EmpProj Relation is **not in 2NF**!*
The primary key of EmpProj is the composite attribute Ssn,Pnumber.
We need to show that there are partial FDs in the relation EmpProj.

FD2 is a full FD since the non-prime attribute Hours is functionally dependent on the full key.

However, FD3 and FD4 are not full FDs but partial FDs since
FD3: nonprime attribute Ename is functionally dependent on only part of the key,[Ssn, Pnumber] of EmpProj relation; it is dependent only on SSn and not on both SSn and Pnumber.
FD4: nonprime attributes Pname and Plocation are functionally dependent on only part of the key,[SSn, Pnumber] of EmpProj relation; it is dependent only on Pnumber and not on both Ssn and Pnumber.

Since EmpProj is not in 2NF we normalize as follows.
FD3 violates 2NF. Remove nonprime attribute Ename from EmpProj. Create a new table EmpProj2 with attributes found in FD3.

EmpProj2

Ssn	Ename

FD: Ssn → Ename // no mv nor composite attributes therefore it is 1NF

Primary Key: Ssn // one attribute in PK therefore it is in 2NF
Prime Attribute: Ssn
Non-prime attribute: Ename // has only one nonprime attribute thus it is in 3NF

FD4 violates 2NF. Remove nonprime attribute Pname, Plocation from EmpProj.
Create a new table EmpProj3 with attributes found in FD4.

EmpProj3

Pnumber	Pname	Plocation

FDs: Pnumber → Pname, Plocation
Pname → Plocation
// no mv nor composite attributes therefore 1NF
Candidate keys: Pnumber, Pname // one attribute in PK therefore 2NF
Primary key: Pnumber Unique Key: Pname
Prime Attributes: Pnumber, Pname
Non-prime attribute: Plocation // has only one nonprime attribute thus 3NF

Therefore EmpProj has now been normalized to three 3NF relations, EmpProj1, EmpProj2 and EmpProj3

EmpProj1(WORKS_ON) | Ssn | Pnumber |Hours | PK: Ssn
FD: Ssn,Pnumber → Hours

EmpProj2(EMPLOYEE) | Ssn | Ename | PK: Ename
FD: Ssn → Ename

EmpProj3 (PROJECT) | Pnumber |Pname | Plocation | PK: Pnumber
FD: Pnumber → Pname, Plocation

Example3: EmpDeptRelation/Table

| Ssn | Ename | Bdate | Address | Dnumber | Dname | DMgrSsn |

Primary Key: Ssn
Prime Attributes: Ssn
Non-prime Attributes: Ename Bdate Address Dnumber Dname DMgrSsn

FDs:

FD1a-f: Ssn → Ename, Bdate, Address, Dnumber, Dname, DmgrSsn
From business rule, 'We store each ***employee's name, social security number***, address, salary, sex, and birthdate.

FD2a-b: Dnumber → Dname, DmgrSsn
From the business rule, 'Each department has a unique name, a unique number, and a particular employee who manages the department' and from the 1-1 Manages between EMPLOYEE and DEPARTMENT

Normalization Explanation:
EmpDept Relation is in 1NF! There are no multivalued nor composite attributes in EmpDept!
EmpDept Relation is in 2NF! The PK contains only one attribute therefore partial FD cannot be violated!
EmpDept Relation is not in 3NF!
FD1f: Ssn → DmgrSsn is transitively dependent through Dnumber in EmpDept.
That is, Ssn → Dnumber (FD1d);
Dnumber → DmgrSSn (FD2b) therefore
Ssn → Dmgrssn (FD1f) by transitivity.

Or in short because non-prime attribute (Dnumber) → non-prime attribute, Dname or DmgrSsn
Therefore EmpDept is not in 3NF.

Since EmpProj is not in 3NF we normalize as follows.
FD2a-b violates 3NF. Remove nonprime attribute Dname and DmgrSsn from Dept Create a new table EmpDept2 with attributes found in FD2a-b.

EmpDeptj2 | Dnumber | Dname | DmgrSsn |
FDs
Dnumber → Ename, DmgrSsn
// no mv nor composite attributes thus 1NF
Candidate key/Primary Key: Dnumber // one attribute in PK therefore 2NF
Prime Attribute: Dnumber
Non-prime attributes: Dname,DmgrSsn // has only one nonprime attribute 3NF
Therefore EmpDept has now been normalized to two 3NF relations, EmpDep1, EmpDept2

EmpDept1 | Ssn | Ename | Bdate | Address | Dnumber |

PK: Ssn
FD: Ssn → Ename,Bdate, Address, Dnumber

EmpDept2 | Dnumber | Dname |DmgrSsn |

PK: Dnumber; CK: Dname
FD: Dnumber → Dname, DmgrSsn

2. Submit in your 3-ring binder the sheets in the following order (immediately after PART 2A Section)

a. DIVIDER PAGE with PART 2A : REVISED tab label
b. Listing of the revisions made and why each change was made.
c. Revised Pages from PART 2A
d. DIVIDER PAGE with PART 2B: Normalization tab label
e. COVER PAGE. See format on page 6. This cover page should not have a footer.
f. Revised Relational Model: If no revisions were made, provide a copy of the relational model you submitted for PART 2A.
g. Description of normalizing each table in the relational model. This must be done for each relation in your relational database schema.
Refer to Step 2 above.
h. **Comments on Project – Normalization Phase**

A. Difficulties you faced in doing this implementation phase and how they were resolved (to be summarized by the Leader based on each member's comments)

B. Likes and dislikes about this part of the project (to be done by each team member as well as by Team Leader; present this section arranged alphabetically by team member's name)

C. What was the most challenging aspect of this part of the design? (to be done by each team member as well as by Team Leader; present this section arranged alphabetically by team member's name)

D. Suggestions on how to improve this part of project (to be done by each team member as well as by Team Leader; present this section arranged alphabetically by team member's name)

PAGINATE EACH PAGE by including a footer in the document as follows (BOLD, font size 8). Start page numbering with 1.

CSE572F13: TEAM NAME PROJECT: PART II-B PAGE 5 of 30

4. **Team Leader** will evaluate each member of the team based on the evaluation criteria agreed upon. The team leader will submit via email his/her evaluation using the following format.

Format
SUBJECT:
CSE 572: PART 2B - TEAM <name>: RATING BY LEADER <name>
BODY:
MEMBER1: <name> -
Criteria item rating & justification
MEMBER2: <name> -
Criteria item rating & justification
MEMBER3: <name> - (if any)
Criteria item rating & justification

5. Each **TEAM MEMBER** will evaluate the team leader and other team members and must submit the rating to me via email using the format below. Note the team member submitting the evaluation must exclude himself/herself from the list of members below. So for a team of three members, there should be 1 other member that should be evaluated; for a team of four members, there should be two other members that need to be evaluated.

Format

SUBJECT: CSE 572: PART 2B- TEAM <name>: RATING BY MEMBER <name>

BODY:

LEADER: <name>
Criteria item 1 rating & justification
….
MEMBER1: <name> -
Criteria item rating & justification
….
MEMBER3: <name> - (if any)
Criteria item rating & justification

This part of the project will be graded according to the following criteria:

Presentation ……………………………………………………………………………. 5 pts
(no spelling errors, no grammar errors, neatness, followed directions, correct sequence of pages, legible and readable)

Content (Correct Normalization)…………………………………………………….. 35 pts

Group/Individual Evaluation…………………………………………………… 10 pts

Total Points…………………………………………………………………………… 50 pts

CALIFORNIA STATE UNIVERSITY, SAN BERNARDINO
SCHOOL OF COMPUTER SCIENCE & ENGINEERING

CSE 572 F13: Database Design Project
PART 2 B – Normalization

Title of Project Option
PROJECT OPTION

Name of the Team
TEAM NAME

Lastname, Firstname
TEAM LEADER

TEAM MEMBERS
Adams, Mark
Smith, Clark
Zeta, Pam