**Assignment 3 – Solution**

**Question 1:**

1. The student should provide examples of insertion, deletion and update anomalies using the data shown in the table. An example of an update anomaly is if we wish to change the name of the employee called ‘Smith J’, we may only change the entry in the first row and not the last with the result that the database becomes inconsistent.

The student should state any assumptions made about the data shown in the table. For example, we may assume that a hotel may be associated with one or more contracts.

NIN

contractNo

hours

eName

hNo

hLoc

NIN

contractNo

hours

NIN

eName

NIN

contractNo

hours

contractNo

hNo

hNo

hLoc

PK

PK

FK

PK

PK

fd1

fd2

fd3

fd4

fd2 and fd3 violates 2NF

#### 2NF

#### 3NF

fd1

fd3

fd4

#### 1NF

contractNo

hNo

hLoc

fd4 violates 3NF

FK

FK

NIN

eName

PK

fd2

Question 2:

patientNo fullName

wardNo wardName

wardName wardNo

drugNo  name, description, dosage, methodOfAdmin

patientNo, drugNo, startDate  unitsPerDay, finishDate

The functional dependencies for bedNo are unclear. If bedNo was a unique number for the entire hospital, then could say that bedNo wardNo. However, from further examination of the requirements specification, we can observe that bedNo is to do with the allocation of patients on the waiting list to beds.

**First Normal Form**

patientNo, drugNo, startDate, fullName, wardNo, wardName, bedNo, name, description, dosage, methodOfAdmin, unitsPerDay, finishDate

**Second Normal Form**

patientNo, drugNo, startDate, wardNo, wardName, bedNo, unitsPerDay, finish Date

drugNo, name, description, dosage, methodOfAdmin

patientNo, fullName

**Third Normal Form**

patientNo, drugNo, startDate, wardNo, bedNo, unitsPerDay, finish Date

drugNo, name, description, dosage, methodOfAdmin

patientNo, fullName

wardNo, wardName

patientNo (FK), drugNo(FK), startDate, wardNo(FK), bedNo, unitsPerDay, finish Date

drugNo, name, description, dosage, methodOfAdmin

patientNo, fullName

wardNo, wardName (AK)

(Primary keys underlined.)

Question 3.

***Answer:***

From the above description, we can presume that the following functional dependencies

hold on the attributes:

FD1: {SSSN} -> {SNAME, SNUM, SCADDR, SCPHONE, SPADDR, SPPHONE, BDATE, SEX, CLASS,

MAJOR, MINOR, PROG}

FD2: {SNUM} -> {SNAME, SSSN, SCADDR, SCPHONE, SPADDR, SPPHONE, BDATE, SEX, CLASS,

MAJOR, MINOR, PROG}

FD3: {DEPTNAME} -> {DEPTCODE, DEPTOFFICE, DEPTPHONE, DEPTCOLLEGE}

FD4: {DEPTCODE} -> {DEPTNAME, DEPTOFFICE, DEPTPHONE, DEPTCOLLEGE}

FD5: {CNUM} -> {CNAME, CDESC, CREDIT, LEVEL, CDEPT}

FD6: {SECCOURSE, SEMESTER, YEAR, SECNUM} -> {INSTRUCTORNAME}

FD7: {SECCOURSE, SEMESTER, YEAR, SECNUM, SSSN} -> {GRADE}

These are the basic FDs that we can define from the given requirements;

we can deduce many others. FD1 and FD2 refer to student attributes;

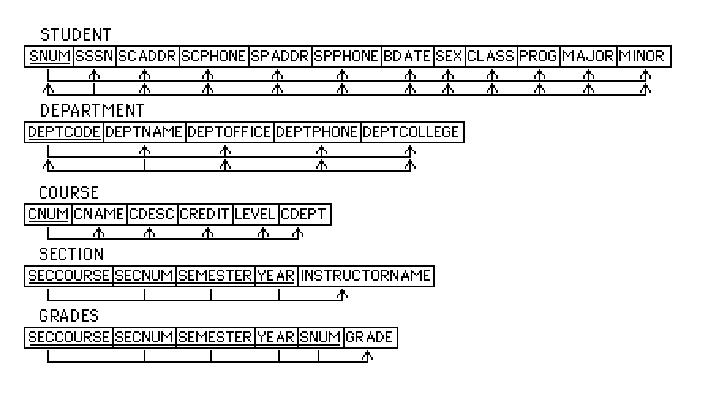
we can define a relation STUDENT and choose either SSSN or SNUM as its primary key.

Similarly, FD3 and FD4 refer to department attributes, with either DEPTNAME or

DEPTCODE as primary key. FD5 defines COURSE attributes, and FD6 SECTION attributes.

Finally, FD7 defines GRADES attributes. We can create one relation for each of STUDENT,

DEPARTMENT, COURSE, SECTION, and GRADES as shown below, where the primary keys are underlined. The COURSE, SECTION, and GRADES relations are in 3NF and BCNF if no other dependencies exist. The STUDENT and DEPARTMENT relations are in 3NF and BCNF according to the general definition



The foreign keys will be as follows:

STUDENT.MAJOR -> DEPARTMENT.DEPTCODE

STUDENT.MINOR -> DEPARTMENT.DEPTCODE

COURSE.CDEPT -> DEPARTMENT.DEPTCODE

SECTION.SECCOURSE -> COURSE.CNUM

GRADES.(SECCOURSE, SEMESTER, YEAR, SECNUM) ->

SECTION.(SECCOURSE, SEMESTER, YEAR, SECNUM)

GRADES.SNUM -> STUDENT.SNUM

Note: We arbitrarily chose SNUM over SSSN for primary key of STUDENT, and

DEPTCODE over DEPTNAME for primary key of DEPARTMENT.