Project 4 Not So Minority

UML Diagram:

Insert UML Diagram (Kia)

Relational Model:

Refer to the NotSoMinority\_Project4\_UML.sql file.

Step by Step BCNF:

Functional Dependencies for Course:

CNUM-> CDESC, CDEPT,

(CNUM, CNAME)->CREDIT, LEVEL,

CNUM is a primary key and (CNUM, CNAME) is a super key

Tables:

Course1(CNUM, CDESC, CDEPT)

Course2(CNUM, CNAM, CREDIT, LEVEL)

With the assumption that you can only get one CDESC and CDEPT per CNUM and only one CREDIT and LEVEL per (CNUM,CNAME).

Functional Dependencies for Section:

(SECCOURSE, SECNUM) ->INSTRUCTORNAME, SEMESTER, YEAR

(SECCOURSE,SECNUM) is a composite primary key.

Table:

Section(SECCOURSE, SECNUM, INSTRUCTORNAME, SEMESTER, YEAR)

With the assumption that you can only get one Instructor, one Semester, one Year per (SECCOURSE, SECNUM).

Functional Dependencies for Grade:

Grade will not have any functional dependencies

(SSN, SECTION,GRADE) is the composite primary key.

Table:

Grade(SSN, SECTION, GRADE)

Functional Dependencies for Department:

DEPTNAME->DEPTCODE, DEPTOFFICE, DEPTPHONE, DEPTCOLLEGE

DEPTNAME is the primary key.

Table:

Department(DEPTNAME, DEPTCODE, DEPTOFFICE, DEPTPHONE, DEPTCOLLEGE)

With the assumption that you get only one DEPTCODE, one DEPTOFFICE, one DEPTPHONE, and one DEPTCOLLEGE per DEPTNAME

Functional Dependencies for Student:

SSSN -> SNAME, SNUM, SPADDR, SPPHONE, BDATE, SEX

SNUM->SCADDR, SCPHONE, PROG, CLASS

PROG->MAJORDEPTCODE,MINORDEPTCODE

SSSN is the primary key, SNUM is the primary key, and PROG is the primary key.

Table:

Student1(SSSN, SNAME, SNUM, SPADDR, SPPHONE, BDATE, SEX)

Student2(SNUM, SCADDR, SCPHONE, PROG, CLASS)

Student3(PROG, MAJORDEPTCODE, MINORDEPTCODE)

With the assumption that you get only one SNAME, SNUM, SPADDR, SPPHONE, BDATE, and SEX per SSSN; you get only one SCADDR, SCPHONE, PROG, CLASS per SNUM; and only one MAJORDEPTCODE, MINORDEPTCODE per PROG

Step by Step 3NF:

Functional dependencies:

Student Table:

snum -> sssn, sname, scaddr, scphone, spaddr, spphone, bdate, sex, class, prog

prog -> majordeptcode, minordeptcode

snum key for student table. decompose on prog. assume that prog has relevant info  
to determine the department since program of study is linked to department in the college.

department table. Prog becomes a primary key.

Student(snum, sssn, sname, scaddr, scphone, spaddr, spphone, bdate, sex, class, prog)

StudentProgram(prog, majordeptcode, minordeptcode)

Department Table:

deptcode -> deptname  
deptname -> deptoffice, deptphone

deptcode, deptname -> deptcollege

deptcode,deptname key for department table. no decomposition since all left hand side are keys.

Department(deptname, deptcode, deptoffice, deptphone, deptcollege)

Course Table:

cnum -> cname,cdept cnum,cname,level -> cdesc,credit

decompose on level because level is not a key in the course table   
Course(cnum,cname cdept,level)

CourseInfo(cnum,cname,level,cdesc,credit)

Section Table:  
secnum,seccourse -> instructor, semester, year  
instructor -> semester, year -> we can link instructor to when semester and year course section was taught

decomposition on instructor:

Section(secnum,seccourse,instructor)

SectionTaught(instructor,semester,year)

Grade Table:

grade table no changes

|  |  |  |  |
| --- | --- | --- | --- |
|  | ER / UML | BCNF | 3NF |
| No. of tables | 5 | 8 | 8 |
| Lossless | no | yes | no |
| Functional dependencies preserved | yes | no | yes |