Inciso B

15.19 Suppose that we have the following requirements for a university database that is used to keep track of students’ transcripts:

a. The university keeps track of each student’s name (Sname), student number (Snum), Social Security number (Ssn), current address (Sc\_addr) andphone (Sc\_phone), permanent address (Sp\_addr) and phone (Sp\_phone), birth date (Bdate), sex (Sex), class (Class) (‘freshman’, ‘sophomore’, ... , ‘graduate’), major department (Major\_code), minor department (Minor\_code) (if any), and degree program (Prog) (‘b.a.’, ‘b.s.’, ... , ‘ph.d.’). Both Ssn and student number have unique values for each student.

b. Each department is described by a name (Dname), department code (Dcode), office number (Doffice), office phone (Dphone), and college (Dcollege). Both name and code have unique values for each department.

c. Each course has a course name (Cname), description (Cdesc), course number (Cnum), number of semester hours (Credit), level (Level), and offering department (Cdept). The course number is unique for each course.

d. Each section has an instructor (Iname), semester (Semester), year (Year), course (Sec\_course), and section number (Sec\_num). The section number distinguishes different sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ..., up to the total number of sections taught during each semester.

e. A grade record refers to a student (Ssn), a particular section, and a grade (Grade).

Design a relational database schema for this database application. First show all the functional dependencies that should hold among the attributes. Then design relation schemas for the database that are each in 3NF or BCNF. Specify the key attributes of each relation. Note any unspecified requirements and make appropriate assumptions to render the specification complete.

* **Esquemas iniciales (0NF).**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Student |  |  |  |  |  |  |  |  |  |  |  |  |
| Sname | Snum | Ssn | Sc\_addr | Sc\_phone | Sp\_addr | Sp\_phone | Bdate | Sex | Class | Major\_code | Minor\_code | Prog |

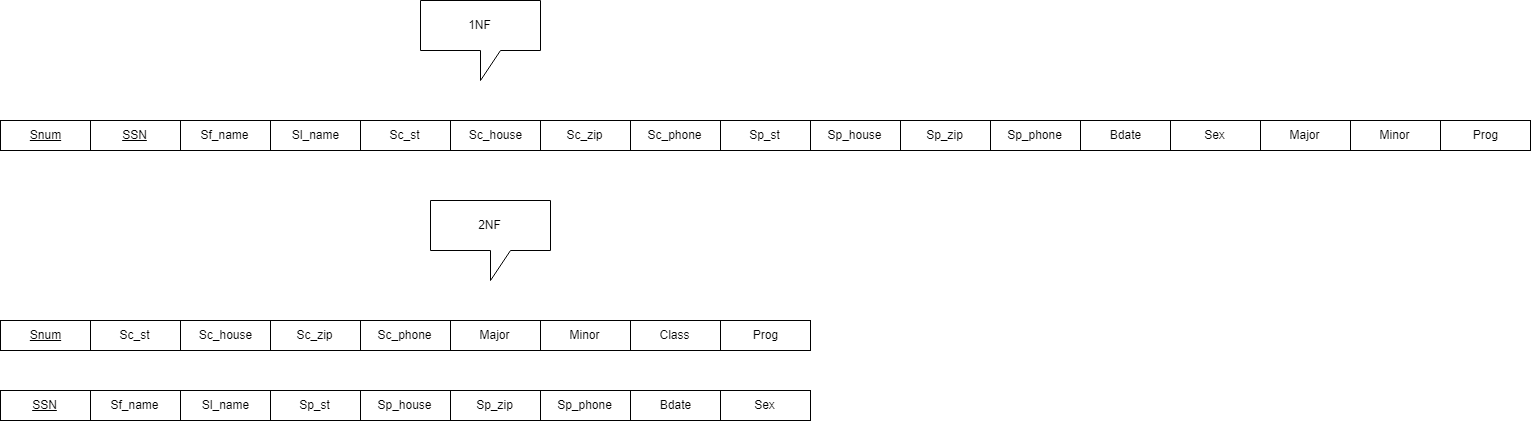
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Department | | | | |  |  | | |  | |  | |
| Dname | | Dcode | | | Doffice | Dphone | | | Dcollege | |  | |
|  | |  | | |  |  | | |  | |  | |
| Course | |  | | |  |  | | |  | |  | |
| Cname | | Cdesc | | | Cnum | Credit | | | Level | | Cdept | |
|  | |  | | |  |  | | |  | |  | |
| Section | |  | | |  |  | | |  | |  | |
| Iname | | Semester | | | Year | Cnum | | | Sec\_num | | Id\_sec | |
|  | |  | | |  |  | | |  | |  | |
| Grade | |  | | |  |  | | |  | |  | |
| Ssn | | Grade | | | Id\_sec |  | | |  | |  | |
| Instructor |  | |  | |  |  | | |  |  | |  | |  | |  |  |  |  |
| Iname | Inum | | Ssn | | Ic\_addr | Ic\_phone | | | Ip\_addr | Ip\_phone | | Bdate | | Sex | |
|  | |  | | |  |  | | |  | |  | |

**Dependencias funcionales**:

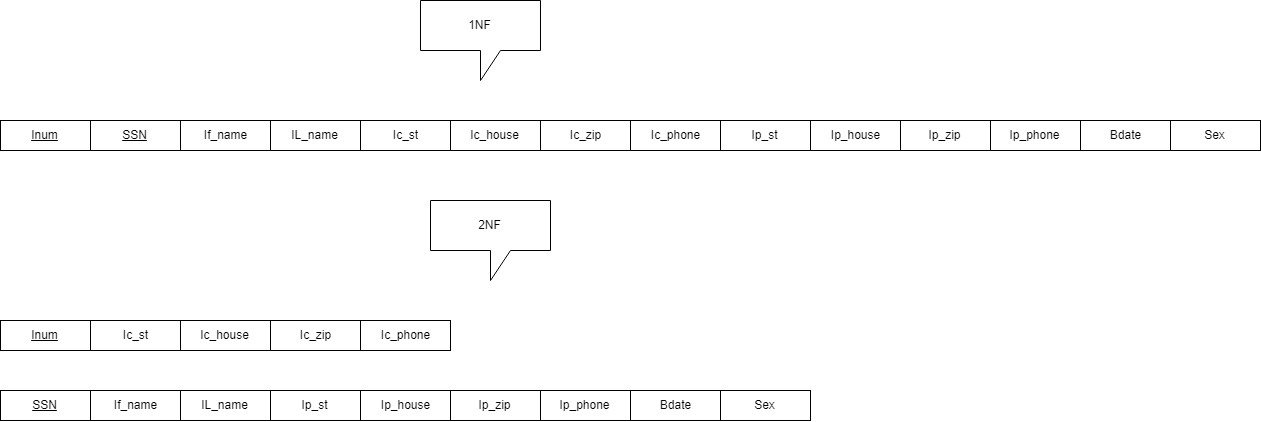
* Student
* FD1: {Ssn} -> {Sname, Sp\_addr, Sp\_phone, Bdate, Sex}
* FD2: {Snum} -> {Sc\_phone, Sc\_addr, Class, Major, Minor, Prog}
* Department
* FD3: {Dcode} -> {Dname, Doffice, Dphone, Dcollege}
* Curso
* FD4: {Cnum} -> {Cname, Cdesc, Credit, Level, Cdept}
* Section
* FD5: {SSN, Id\_sec} -> {Iname}
* Instructor
* FD6: {Ssn} -> {Iname, Ip\_add, Ip\_phone, Sex, Bdate}
* FD7: {Inum} -> {Ic\_phone, Ic\_addr}

**Normarlización**

* Student: La relación Student es llevada a 1NF debido a sus atributos compuestos de Sname, Sc\_addr y Sp\_addr. Luego, es llevada a la 2NF para separar los atributos que dependen de las dos llaves diferentes. Finalmente, la relación ya queda en 3NF debido a que no se forma ninguna dependencia transitiva.



* Instructor: En un caso muy similar a la relación Student, encontramos que Instructor debe sufrir transformaciones a 1NF y a 2NF. La forma normal 3 se consigue automáticamente después de transformar a la forma normal 2 debido a la inexistencia de dependencias transitivas.



* Section: finalmente, la relación Section debe ser llevada a 1NF por el atributo compuesto Iname. Luego de esto, también queda en estado 2NF ya que no existen dependencias parciales y por eso no se toma en cuenta. Para terminar, fue llevada a la forma 3NF por la dependencia transitiva Inum -> Iname. En este caso, la relación resultante de la transformación es obviada porque ya existe una relación Instructor, por lo que solamente se señala el Foriegn key de Inum.

