Bono capítulo 15

Basics of Functional Dependencies and Normalization for Relational Databases

Consider the relation R, which has attributes that hold schedules of courses and sections at a university; R = {Course\_no, Sec\_no, Offering\_dept, Credit\_hours, Course\_level, Instructor\_ssn, Semester, Year, Days\_hours, Room\_no, No\_of\_students}. Suppose that the following functional dependencies hold on R:

{Course\_no} → {Offering\_dept, Credit\_hours, Course\_level}

{Course\_no, Sec\_no, Semester, Year} → {Days\_hours, Room\_no, No\_of\_students, Instructor\_ssn}

{Room\_no, Days\_hours, Semester, Year} → {Instructor\_ssn, Course\_no, Sec\_no}

Try to determine which sets of attributes form keys of R. How would you normalize this relation?

**Respuesta:**

1. Por definición de candidate key, podemos obtener los siguientes sets de llaves
   1. {Course\_no, Sec\_no, Semester, Year}
   2. {Room\_no, Days\_hours, Semester, Year}

Esto nos asegura que no tendremos más de una tupla asociada a un valor específico de algún set de candidate keys.

1. Llevamos la relación a la forma 2NF debido a que se presenta una dependencia funcional en el atributo Course\_no, el cual se encuentra participante en dos dependencias diferentes. Dicho esto, particionamos la relación en dos partes:
   1. R\_course: {Course\_no, Offering\_dept, Credit\_hours, Course\_level}
   2. R\_section: {Course\_no, Sec\_no, Semester, Year, Days\_hours, Room\_no, No\_of\_students, Instructor\_ssn}

Finalmente, debido a la ausencia de dependencias transitivas, R\_course y R\_section están en 3NF.