

numbers in a Social Security number—XXX-XX-XXXX). However, Social Security numbers are usually represented as just one attribute. The decision is based on how the database will be used. This exercise asks you to think about specific situations where dividing the SSN is useful.

- 5.19. Consider a STUDENT relation in a UNIVERSITY database with the following attributes (Name, Ssn, Local_phone, Address, Cell_phone, Age, Gpa). Note that the cell phone may be from a different city and state (or province) from the local phone. A possible tuple of the relation is shown below:

Name	Ssn	Local_phone	Address	Cell_phone	Age	Gpa
George Shaw	123-45-6789	555-1234	123 Main St., Anytown, CA 94539	555-4321	19	3.75
William Edwards						

- Identify the critical missing information from the Local_phone and Cell_phone attributes. (*Hint*: How do you call someone who lives in a different state or province?)
 - Would you store this additional information in the Local_phone and Cell_phone attributes or add new attributes to the schema for STUDENT?
 - Consider the Name attribute. What are the advantages and disadvantages of splitting this field from one attribute into three attributes (first name, middle name, and last name)?
 - What general guideline would you recommend for deciding when to store information in a single attribute and when to split the information?
 - Suppose the student can have between 0 and 5 phones. Suggest two different designs that allow this type of information.
- 5.20. Recent changes in privacy laws have disallowed organizations from using Social Security numbers to identify individuals unless certain restrictions are satisfied. As a result, most U.S. universities cannot use SSNs as primary keys (except for financial data). In practice, Student_id, a unique identifier assigned to every student, is likely to be used as the primary key rather than SSN since Student_id can be used throughout the system.
- Some database designers are reluctant to use generated keys (also known as *surrogate keys*) for primary keys (such as Student_id) because they are artificial. Can you propose any natural choices of keys that can be used to identify the student record in a UNIVERSITY database?
 - Suppose that you are able to guarantee uniqueness of a natural key that includes last name. Are you guaranteed that the last name will not change during the lifetime of the database? If last name can change, what solutions can you propose for creating a primary key that still includes last name but remains unique?
 - What are the advantages and disadvantages of using generated (surrogate) keys?