Database Design and Normalization

Normalization based on functional dependencies (FDs) make it easier to update the relations (by eliminating some problems that can occur – problems that we call update anomalies). But if we normalize a database, then some queries may run slower than they would have if we hadn't normalized.

Consider the following DB designs to keep track of classes (offered in a particular quarter at a university) and the instructors who teach them.

Design A:

Class (CRN, Dept, Number, Bldg, Room, Time-days, Instructor-id, Instructor-name) where CRN is the only key for this table.

Design B:

Class (CRN, Dept, Number, Bldg, Room, Time-days, Instructor-id, Instructor-name) where CRN and Instructor-id together serve as the only key for this table.

Design C:

Class (CRN, Dept, Number, Bldg, Room, Time-days) where CRN is the only key
Instructor (Instructor-id, Instructor-name) where Instructor-id is the only key
Teaches (CRN, Instructor-id) where CRN and Instructor-id, together, are the only key

For each design, answer the following questions. Assume that the DBMS enforces all keys. No other constraint checking or constraint enforcement is done.

1. How many instructors can have class have? If one class can have more than one instructor, provide sample data for the tables in each design that demonstrates this point.
2. How many classes can one instructor teach? If an instructor is allowed to teach more than one class, provide sample data for the tables in each design that demonstrates this point.
3. Is it possible for one Instructor-id to appear with two different names? If yes, provide sample data that demonstrates this point.
4. Suppose you want to print a list of classes with the following attributes. If a class has more than one instructor, then the class will appear multiple times in this report, once for each instructor.

   CRN Dept Number Bldg, Room Time-days Instructor-id Instructor-name

   Write the query that will produce this query answer, for each design. Which design will likely be the fastest to run the query? Which design will likely be the slowest?