Scenario

Last semester you were introduced to a hypothetical College Enrollment System. Let's start from the beginning with the concept of a *Course Offering*.

We can capture the ***offering*** requirement for the system as follows:

> ***Rqmt #00000****: An Offering is a college course scheduled at various times in a semester, taught by an instructor at one or more locations at a campus of a college.*

Intuitively and naïvely, we can collect the following fields as being pertinent to an offering. These could have been discovered from documentation, user interviews, or examining prior or competing systems.

Offering Table	
Course Name	like "Database II"
Course Number	like "CP1555"
Course Description	like "A course in…"
Course Program	like "SD"
College Name	like "CNA"
College Campus	like "PPD"
Course Capacity	like 25
College Address	like "1 Prince Philip, St. John's"
Semester	like "Winter"
Weekly time 1	like "Mon, 8:30, 2hrs"
Weekly time 2	like "Tue, 8:30, 2hrs"
Weekly time 3	like "Wed, 8:30, 2hrs"
Weekly location 1	like "K205"
Weekly location 2	like "K205"
Weekly location 3	like "K205"
Instructor Name	like "Paul Drover"
Instructor Dept	like "IT/BUS"
Instructor Email	like "paul.drover@cna.nl.ca"
Instructor Office	like "K212d"
Course Enrollment	like 22
Course Wait-list	like 0

It's always a good idea to list any assumptions. These can also be called *business rules*.

1. The college is always CNA.
2. There are 3 sessions of a course per week.
3. Course numbers can contain letters.
4. Semester is one of "Fall", "Winter", "Spring", "Summer"
5. Waitlist is only > 0 if enrollment = capacity

Your task

The above table is *unnormalized*. Using whatever tool you are comfortable with (Workbench is not required. Indeed, this can be done with pen and paper. Excel works too, as does Word.):

1. Bring this table into **1NF**. **Show** the resulting table (screen shot is fine).
2. **Identify** primary key components. The key must uniquely identify every record in the 1NF table. **State** what the primary key should be.
3. For every non-key field **identify**:
    a. **State** all parts of the key that the field is **dependant** on.
    b. **State** if there is a partial-key violation on the field that keeps the table from 2NF.
4. Bring 1NF into **2NF**. **Show** the resulting table(s).
5. For every non-key field in every table **identify**:
    a. if there are any data that is not dependant on the entire primary key
    b. if there are any data that are used by more than one table.
6. Bring 2NF into **3NF**. **Show** the resulting table(s).
7. Would you consider any **denormalization**? Why or why not?

Submission

Create a word file that contains all responses to items 1-7 above. Insert screenshots (photos, if you used pen and paper) of any results from another application (like Excel) as appropriate.

Save your responses and upload the file to D2L Dropbox and submit electronically.