***Assignment 1 – Instructions:***

Top of Form

This assignment tests your familiarity with normalization techniques and basic SQL commands.

You are expected to do the following:

* 1. Analyze the [attached table](https://adauniversity-my.sharepoint.com/:x:/g/personal/nsadili_ada_edu_az/EX_LHa70bIFJhMxY_qlZocQBzKXjgV8GJf6yvBsEgtv3LQ?e=ZVhsAO) and identify the potential problems associated with the design of the given relation: StudentsCourses.
  2. Design a strategy to convert it to 1st, 2nd, and 3rd normal forms. Clearly state which normalization rule is violated and how you are addressing it.
  3. Represent each normalized relation in a table format, clearly specifying the attributes and sample tuples.
  4. Implement the mentioned plan using the PostgreSQL environment:
  5. Upload the given data to Postgres (you may want to convert it to CSV first and then upload it).
  6. start building the necessary tables using SQL commands representing the normalized relations.
  7. populate the new tables with appropriate data from the StudentsCourses relation.
  8. Add a few more records to your new tables.

**Hints for Normalization**

*Violations of 1NF:*

* 1. Examine for atomicity. Do all attributes have a single value?
  2. Consider the level of granularity for each attribute. Is there any repetition in the groups or arrays?

*Violations of 2NF:*

* 1. Determine the primary key of the relation. Does it consist of multiple attributes?
  2. Is it possible to have non-key properties that are only dependent on a subset of the primary key? This would indicate a partial dependency.

*Violations of 3NF:*

* 1. Check if non-key attributes are dependent on other non-key attributes. This is termed transitive dependency.
  2. For example, if you know the value of attribute A (say, it is not a primary key), and it gives you the value of attribute B, and B is not a primary key either, you might have a transitive dependency.