Submit via D2L. Late submission is not acceptable.

Assume we have a table

students (name, stnum, ssn, phone#, home_address, home_zip, home_state, course, grade, instructor_name, department, university, univ_address, univ_zip, univ_state);

Conduct normalizations until you get 3NF tables. You must submit all the intermediate results. For each step, you need to give a set of tables and their keys to completely express the relationships described in the original big table.

Step 1) **Determine the key** of this big table, ensuring every non-key column value can be decided when the key value is given.
   **Evaluation:** print out the entire table, but with each key column (i.e., attribute field of key, may have multiple columns) of that key underlined.

Step 2) For the key, without making any table split (creating sub table or new table), **explain why** the key set cannot be smaller.
   **Evaluation:** Show a counter-example for each smaller key to prove that missing any column in the current key may incur the loss of the dependence of a non-key column (i.e., who decides this non-key column) in count.

Step 3) **Explain why** the key set is sufficient enough (no need for any bigger) to support each non-key-column-value determination.
   **Self-evaluation (i.e., no need for any document work):** Verify each non-key column to have a dependence on one or more columns of that key (i.e., proving that this key can decide each of other columns).

Step 4) **Determine** which column may have **multi-value**. Be aware of the difference between multi-value and multi-record (for instance, phone can be multiple value, but course cannot. It must be multiple records).
   **Evaluation:** Indicate any repeating group.

Step 5) Determine if this 1NF table(s) (if no multi-value exists). Otherwise, **apply “remove and switch”** (remove any repeating group and copy its related key columns to a new table, **then switch** and make that repeating group attribute the key in new table).
   **Evaluation:** Print out 1NF tables (but not 2NF tables).

Step 6) **Apply the “remove”** (copy the partial relationship out to a new table, and delete that non-key column from original table. Keep the key column in original table and copy it to the new table). If you have one-column key, this 1NF is automatically 2NF.
   **Evaluation:** Print out 2NF tables (but not 3NF tables).

Step 7) **Apply the “copy” conversion** to get 3NF tables (copy any non-key relation, any dependency without the involvement of key(s), to the new table).
   **Evaluation:** Print out 3NF tables.