The list I have here contains data on the cpu, performance, and sold. The first thing I notice that is wrong with this list is that it need keys so that if one of the data is deleted or changed than it would not coorelate with the with one of the column. Like if cpu name is changed than it would not correlate with the other column because that change name might not correlate with the date , socket type, series name. To fix this I placed primary keys to the set of data that uniquely identify a column. I placed a CPU\_ID column to determine the cpu name, socket type, and series name. A date\_id will determine the date release, number sold on release and price on release. I placed a performance\_id which determine the 3d model score 3d rendering speed and physics score. Then I look at the relation each entity have with each other and I determine that CPU and Can have many performance but the performance it has correlate with one cpu which is a one to many relationship. Another relationship I notice is that cpu can have many performance and each performance is correlated to one CPU. And that is also a one to many relationship. After I established the relationship, I placed each entity into separate tables. Since they are both one to many relationship, the key from the one side goes to the many sideThe table is still not well formed so I used normalization to fix that. I already determine that functional tendency when I placed the key to uniquely select a table and made a separate table for each functional tendency I made. The original table need to be rid of everything except the determinant so in the end we are left with Only the primary key in the original table