1. What went well in the project GitHub Events table?
    a. We had an early start on the ERD. This gave us an understanding about the relationship between different entities and gave us a bird's eye view of how we will execute our project.
    b. Functions: We invested a good amount of time to create a logical renaming of columns once we un nest the dataset from payload. We had a function that un nests our nested Object type data into flattened table. We also had a function that automatically drops the columns that were 100 percent null. This saved us a lot of time in otherwise manual process and made our work error free.
    c. Preserving the dataset: We tried to preserve every cell in our data set. This gave us an opportunity to pivot our dataset in every possible column and analyze our dataset in many forms.
    d. Ability to change: During the second last day, we realized that join operations were computationally expensive. We found an alternate logic on the fly that virtually eliminated all the joins and kept our existing code logic as much as possible. This showed agility of our team to be flexible as per the situations demand.

2. What did not go well on our table?
    a. Not understanding the project requirements: Our understanding of the project was to preserve each and every cell of the table. It was later clarified that this wasn’t necessary.
    b. Creating clustered index for every column: We did not need to do that for any table. From Jessies presentation, we learned that we could have converted the User attributes from each sub table to an RDD, infer a common schema and joined them via Union operation. This would have been a narrow transformation, and we could have dropped duplicates on one final table.
    c. Communication: Team members had a tunnel vision for how they wanted to process their dataset. We could spend more time listening to every team member perspective and coming up with a conclusion that was approved than Project Manager (aka Jordan and Lee). We could have been honest about how approachable each of the team members were in the beginning. We realized that we should have more confidence in our ideas and should not be afraid to express them in a team environment. We should have agreed upon the standards of how we were going to name the variables, how we were going to create the functions and should have created better markdowns to make our code more reader friendly.
    d. Join operations: Join operations are computationally expensive and should be avoided at all times.
3. What hindered our progress in the project?
    a. Lack of central communication: We could have created a list of the things that needed to be done and highlighted each team member by unique color. There could have been a column which indicated our progress on each single task as follows: initiated, running into problems, done.

b. Creation of agreed upon rules: How we would define variables, how we would write functions, and how we would use existing indices to justify relationships between each table.

c. Communication with our superiors: We should have had more frequent feedback from Jordan and Lee. This would have helped us catch early on our errors.

4. Things you learned from this project?
   a. Over engineering: This is a real phenomenon and should be caught on early in the project.
   b. Occom's razor: The simplest solution is usually the right one.