Task 2

Question 1

Functional dependencies: Input\_id --> labeldem, Input\_id --> labelgop, Input\_id --> labeldjt

Question 2

No, the *comments* relation does not look normalized because there are redundancies in the data. For example, the relation contains the *subreddit*, *subreddit\_id*, and *subreddit\_type* attributes. The *subreddit\_id* attribute functionally determines *subreddit* and *subreddit\_type*, and the comment’s *id* attribute functionally determines *subreddit\_id*:

id --> subreddit\_id

subreddit\_id --> subreddit

subreddit\_id --> subreddit\_type

Therefore, *subreddit* and *subreddit\_type* are redundant because we can use transitivity to determine these attributes using the comment *id*: *id* --> {*subreddit, subreddit\_type*}. There are other redundant attributes as well, such as *can\_gild*, *author\_flair\_text, author\_flair\_css\_class*, and *author\_cakeday,* which can each be functionally determined by *author* or {*author, subreddit\_id*}.

We can decompose the comments relation by taking a nontrivial functional dependency, such as *id --> subreddit\_id*, and splitting the relation into one relation containing {id, subreddit\_id} and a second relation containing *id* and every other attribute except *subreddit\_id*. We can repeat this process with other nontrivial functional dependencies until all relations are in BCNF or some other desired normal form.

We believe that the collector of the *comments* data stored the data this way for several reasons. First, having all data associated with a comment in one relation means that we do not need to compute joins across several tables, which are expensive operations. Also, the collector likely assumed that we will not have data integrity issues because we will not be adding more comment data to the relation in this project. Further, organizing all comment data into one table makes it easy for students to see what information is attached to each comment; it would be difficult for us to begin navigating this project if data were spread across many different tables.