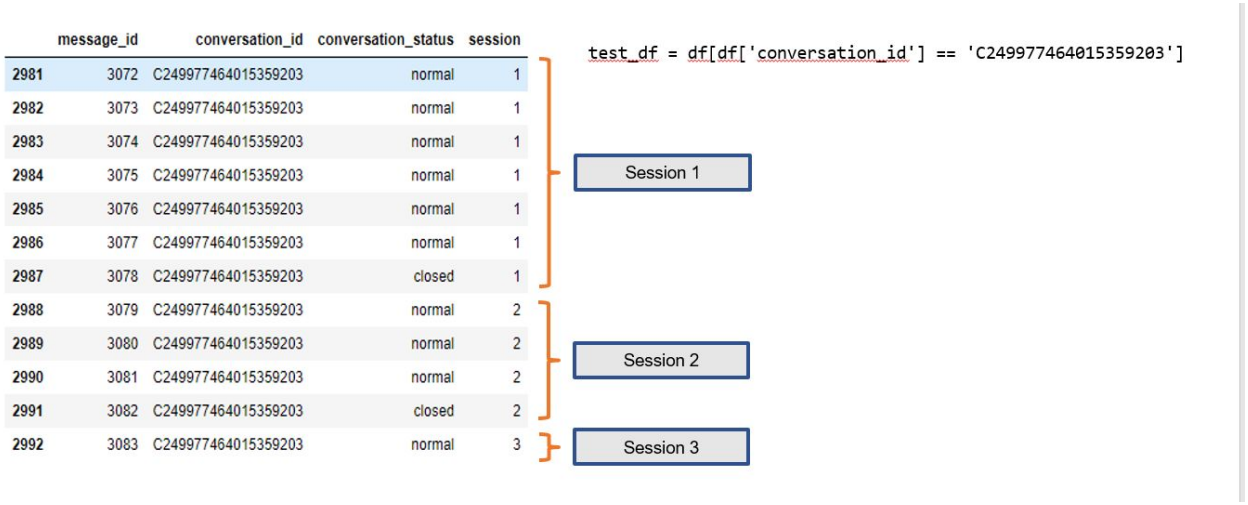
Introduction Questions

1) Aggregating
    a) Pivot Table
        i) How many messages are attributed to each Shopee shop?
        ii) How many messages each buyer sent in total?
        iii) Remove from the dataset, messages from the top 3 buyers

2) Merging
    a) Merge them with the CSAT table (csat.csv) to obtain the CSAT ID
    b) Obtain the CSAT ID for chats that have a Survey Reply

3) Groupby
    a) For each conversation id, split each conversation into different sessions
        i) If a message has the conversation_status “closed”, it means that it is the end of the session
        ii) Assign a new column “session“



	message_id	conversation_id	conversation_status	session
2981	3072	C249977464015359203	normal	1
2982	3073	C249977464015359203	normal	1
2983	3074	C249977464015359203	normal	1
2984	3075	C249977464015359203	normal	1
2985	3076	C249977464015359203	normal	1
2986	3077	C249977464015359203	normal	1
2987	3078	C249977464015359203	closed	1
2988	3079	C249977464015359203	normal	2
2989	3080	C249977464015359203	normal	2
2990	3081	C249977464015359203	normal	2
2991	3082	C249977464015359203	closed	2
2992	3083	C249977464015359203	normal	3

4) Dealing with Datetime
    a) For each conversation get the time difference between the order time and the earliest message.

Workshop Questions

1) **Get the product id for each conversation from the product_url column**
    a) Product id is the numbers after “pid=”

2) **Flag all conversations that have no agents replying with the column missed_chat = True and remove all these chats where missed_chat =True.**

   (Hint: Groupby sessions, check if each session has any is_seller = True, you can check if the dataframe has any True values by using `any(df['is_seller'])`

3) **Assign a new column -> Chat Duration, the time difference between the first chat and the last chat for each session.**

   One session will all have the same value of ‘chat_duration’

   (Hint: Group by each session, find the max and minimum time and then find the difference. Assign it back into the dataframe)
   df[col].min() and df[col].max() gives you the min and max time respectively

   Remember to convert to datetime if you have done so.
   `df['timestamp']  = pd.to_datetime(df['timestamp'])`

4) **Assign a new column -> first reply time in seconds.**

   The reply time for each chat is the time taken from the first buyer message and the next seller message after the first buyer message.

   One session will all have the same value of ‘first_reply_time’

   Example:

is_buyer	is_seller	timestamp
False	True	2020-04-01T00:00:01.000
True	False	2020-04-01T00:01:01.000
False	True	2020-04-01T00:02:03.000

The first buyer message is at **2020-04-01T00:01:01.000**
The first seller message after the first buyer message is **2020-04-01T00:02:03.000**
First reply time is 62 seconds.