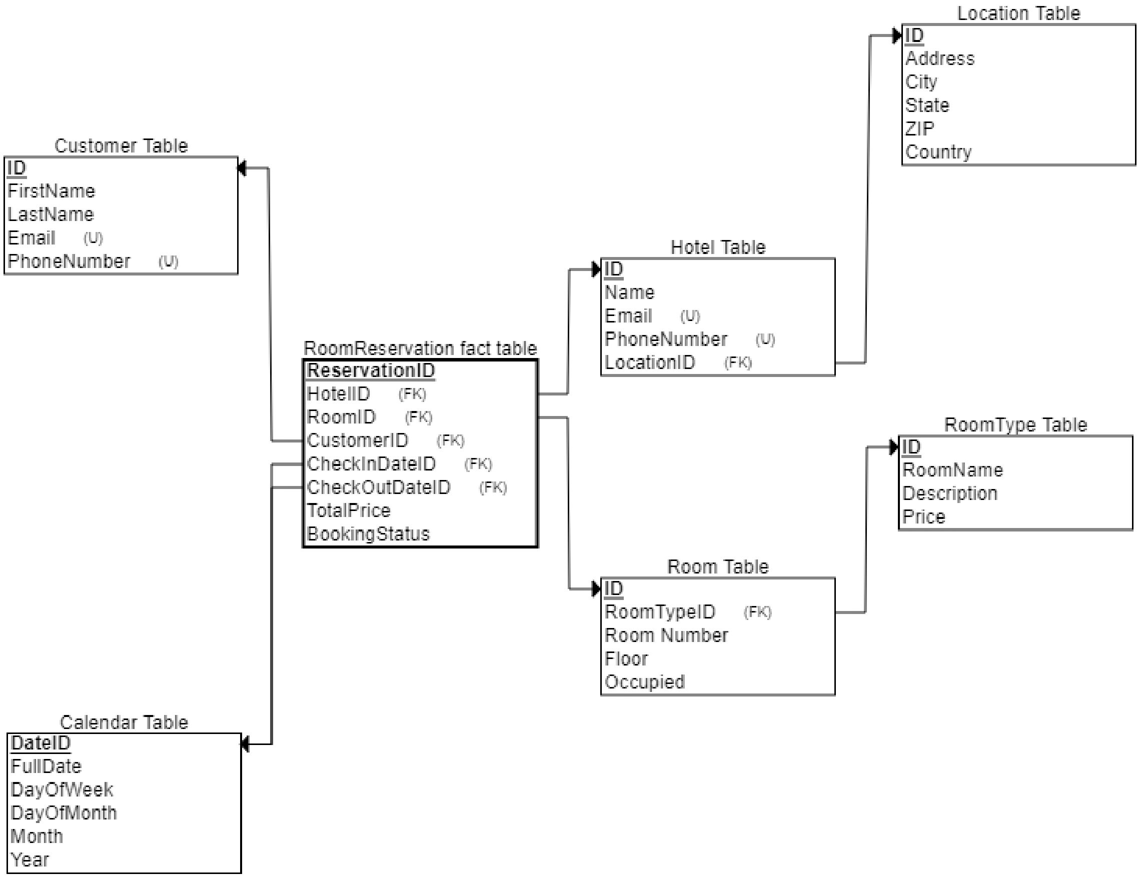
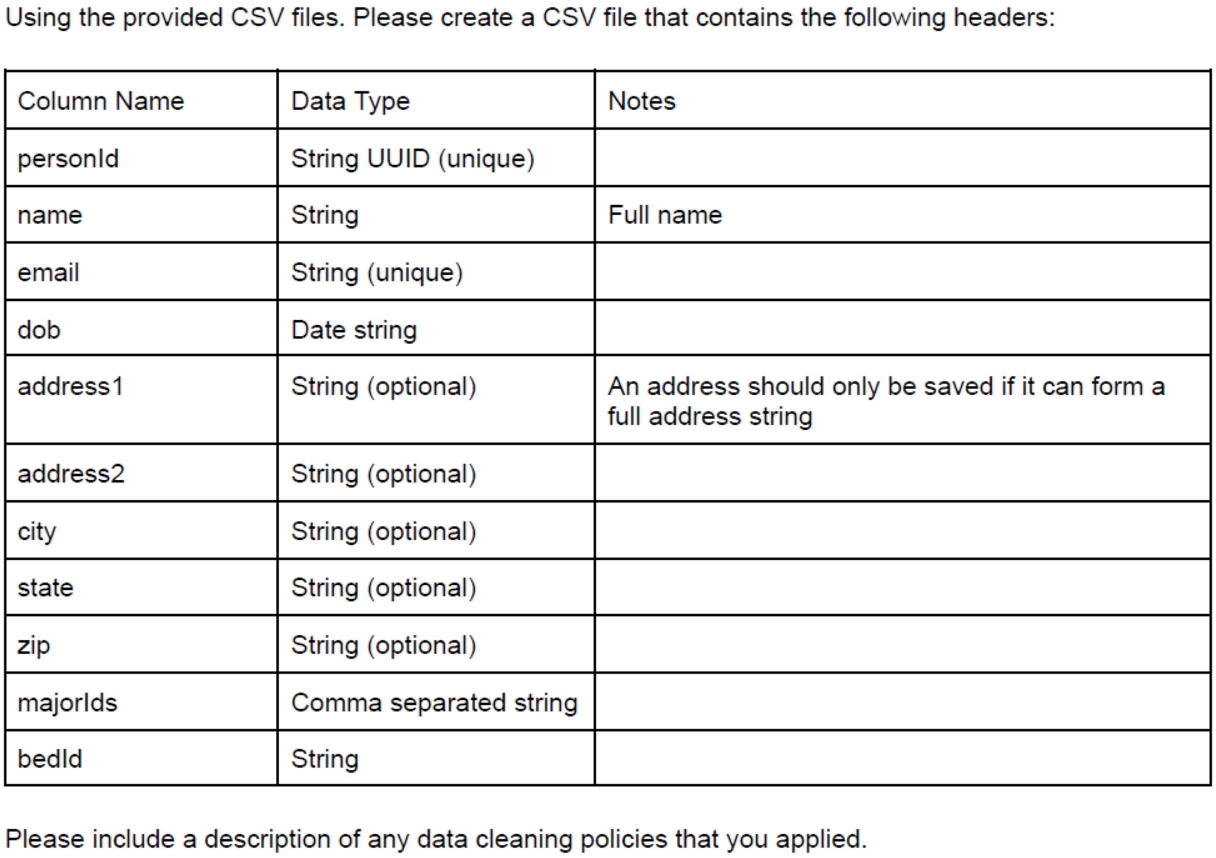
Question 1)



I created a Snowflake Schema implementation for a Room Reservation System for a Global Hotel Chain. I also implemented the same schema using MS SQL.

We have a central Room Reservation fact table which contains keys to reference other dimension tables. We can get information about the specific hotel, the room, the customer , the check in and check out dates, along with the total price and the booking status of the reservation.

Question 2)



Using the provided CSV files. Please create a CSV file that contains the following headers:

Column Name	Data Type	Notes
personId	String UUID (unique)	
name	String	Full name
email	String (unique)	
dob	Date string	
address1	String (optional)	An address should only be saved if it can form a full address string
address2	String (optional)	
city	String (optional)	
state	String (optional)	
zip	String (optional)	
majorIds	Comma separated string	
bedId	String	

Please include a description of any data cleaning policies that you applied.

I first loaded all the csv files as tables into an Excel Workbook. **persons_data.csv, occupancy_data.csv, inventory_data.csv, majors_data.csv**

Step1 ->
Loaded the **occupany_data.csv** and **inventory_data.csv.**
I then transformed the data and used First Row as Headers and trimmed roomName and bedName columns in both tables to remove whitespaces. roomName and bedName form a composite candidate key in inventory_data.csv. I performed a merge query using an inner join. Now we have personIds and linked bedIds in the occupancy table file.

Step2 ->
In the majors_data.csv file, I noticed that multiple major names are associated with the same major id. I first duplicated the entire majors_csv file. Then I removed duplicate major names using Remove Duplicates from

the original file. To get the flagged rows, I performed a merge query using a left anti join on majorids column using the original and duplicate files. This will filter out all the flagged rows in the duplicates csv file.

Step 3 ->
Similarly, for person_data.csv. I dropped all duplicate emails and kept the first occurrence of the duplicate emails. This will filter out 4900 rows of flagged data. I put this data in majors_flagged_data file using the same left anti-join technique as above. We are left with 100 rows of data.

Step4 ->

I first merged the firstName and lastName columns using a space delimiter. Then I split the Address column by delimiter – “,”. I then trimmed the columns to remove whitespace.

I noticed there are also multiple major names present in the majors_data.csv file. We need to map all the major names in every cell to majorIds.

There are multiple ways to do this.
We can use Text to Columns feature to split majors column to multiple columns and then perform XLOOKUP with cell values in each of the newly split columns to create separate columns corresponding to every majorId. Then we can merge these columns into a single column containing majorIds.

I created a separate column named majorIds. I wrote a function ->

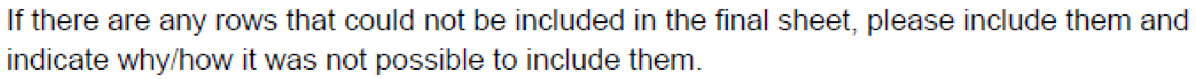
```
=TEXTJOIN(", ", TRUE, IFERROR(XLOOKUP(TEXTSPLIT(J2, ", "),
majors_data[name], majors_data[id], ""), ""))
```

This splits major names in every cell of column J (containing major names) based on delimiter “, “; and then performs XLOOKUP on each of the split major names from the cell. We lookup against the major names in the majors_data.csv column and return the corresponding majorID from the majors_data.csv file for every major name in the cell. Finally the TEXTJOIN function returns the combined string of majorIds. I then deleted the major names column.

Step5 ->

I then performed XLOOKUP to get the bedIds into the majors_data file by looking up personId in the the occupancy_table which I had merged with the inventory table containing bedIds. I saved the result in the **final_data.csv file**. I also flagged data from the other CSV files using FILTER functions.

To cross check my results, I also wrote a Python script to perform the same task and generated a separate **final_data_python.csv** file. Both the resulting CSV files have the same data after processing.



If there are any rows that could not be included in the final sheet, please include them and indicate why/how it was not possible to include them.

I flagged a lot of data which couldn’t be included. EmailIDs are supposed to be unique, but there were tons of duplicate emailIDs in the person_data file. So, I removed all the duplicates and put them in a separate file. Similarly, lots of rows were removed in the majors_data file because of duplicate major names.
There was lot of data in the occupany and inventory files not being used because they did not match the personIds in the final_data file.