Data Engineering Questions:

Question 1, Which city is best for each traveler type? For each traveler type, recommend the best city based on the given reviews:

First, we merge the three tables to manipulate them easily.

```
df_q1 = df_reviews[['hotel_id', 'user_id', 'score_overall']].merge(
    df_hotels[['hotel_id', 'city']],
    on='hotel_id',
    how='left'
).merge(
    df_users[['user_id', 'traveller_type']],
    on='user_id',
    how='left'
)
```

This step groups the table by the ‘traveller_type’ and ‘city’, and for each [‘traveller_type’, ‘city’] pair, it gets the average ‘score_overall’ of the corresponding reviews and their count

```
city_traveler_ratings = df_q1.groupby(['traveller_type',
'city'])['score_overall'].agg(['mean', 'count']).reset_index()
```

We disregard pairs with fewer than 10 reviews

```
city_traveler_ratings =
city_traveler_ratings[city_traveler_ratings['count'] >= 10]
```

This is the final step, it gets the maximum average ‘score_overall’ for each ‘traveller_type’

```
best_cities =
city_traveler_ratings.loc[city_traveler_ratings.groupby('traveller_type')['
mean'].idxmax()]
```

Question 2, What are the top 3 countries with the best value-for-money score per traveler's age group?:

First, we merge the three tables to manipulate them easily.

```
df_q2 = df_reviews[['hotel_id', 'user_id', 'score_value_for_money']].merge(
    df_hotels[['hotel_id', 'country']],
    on='hotel_id',
    how='left'
).merge(
    df_users[['user_id', 'age_group']],
    on='user_id',
    how='left'
)
```

This step groups the table by the 'age_group' and 'country', and for each ['age_group', 'country'] pair, it gets the average 'score_value_for_money' of the corresponding reviews and their count

```
vfm_by_country_age = df_q2.groupby(['age_group',
'country'])['score_value_for_money'].agg(['mean', 'count']).reset_index()
```

We disregard pairs with fewer than 10 reviews

```
vfm_by_country_age = vfm_by_country_age[vfm_by_country_age['count'] >= 10]
```

Finally, we get the top 3 countries with the highest 'score_value_for_money' for each 'age_group'

```
top_3_countries = vfm_by_country_age.sort_values(['age_group', 'mean'],
ascending=[True, False]).groupby('age_group').head(3)
```

The Reason Behind the chosen features

Score features: These features represent the quality of the hotel according to the users' reviews; they can be used to determine the overall quality of hotels in each country group

Base features: These features can be used along with the score features to find the discrepancy between them; this way, the model can see what country groups overpromise

User gender: see trends in which a certain gender frequents specific country groups, for example, females may visit countries more commonly visit countries that are considered safer.

User age group: Relate between the age and country groups, lower ages may frequent less expensive and more adventurous places, while older ages may go to the opposite characteristics.

Traveler Type: Some countries are frequented more in business trips, so a Business type may be more likely to visit them.