Import libraries and modules

```
import pandas as pd
from sqlalchemy import create_engine
```

Connect to postgresql

```
%load_ext sql
```

```
username = 'postgres'
password = 'BostonMA2022'
hostname = 'localhost'
dbname = 'Target_AirFryer_Project'
```

```
connection_string = f'postgresql://{username}:{password}@{hostname}/{dbname}'
```

```
%sql $connection_string
```

Import the 1st table into the database

```
review_rating = pd.read_csv(r"C:\Users\Admin\Downloads\review_rating.csv")
```

```
review_rating.to_sql("review_rating", connection_string )
```

Start cleaning the data

```
%%sql
SELECT *
FROM review_rating
```

Check if there are any duplicate values

```
%%sql
SELECT *
FROM review_rating
WHERE review IN (
    SELECT review
    FROM review_rating
    GROUP BY review
    HAVING COUNT(review) > 1
);
```

Update null values with 0

```
%%sql
UPDATE review_rating
SET rating = 0
WHERE rating IS NULL
```

```
%%sql
UPDATE review_rating
SET design = 0
WHERE design IS NULL
```

```
%%sql
UPDATE review_rating
SET quality = 0
WHERE quality IS NULL
```

```
%%sql
UPDATE review_rating
SET easy_of_use = 0
WHERE easy_of_use IS NULL
```

```
%%sql
UPDATE review_rating
SET value_rating= 0
WHERE value_rating IS NULL
```

```
%%sql
UPDATE review_rating
SET easy_to_clean= 0
WHERE easy_to_clean IS NULL
```

```
%%sql
SELECT *
FROM review_rating
WHERE rating IS NULL OR quality IS NULL OR easy_of_use IS NULL OR value_rating IS NULL OR easy_to_clean IS NULL OR design IS NULL
```

```
%%sql
SELECT *
FROM review_rating
WHERE review IS NULL
```

Drop the index column since we don't need it

```
%%sql
ALTER TABLE review_rating
DROP COLUMN index ;
```

Add a column to classify promotion and non-promotion reviews.

1: yes, it is a promotion review

0: no, it is not a promotion review

```
%%sql
SELECT *,
CASE
    WHEN review LIKE '[This review was collected as part of a promotion.]%' THEN 1
    ELSE 0
END AS promotion_review
FROM review_rating
```

turn it into a dataframe and save as a csv file

```
output = %sql SELECT *, CASE WHEN review LIKE '[This review was collected as part of a promotion.]%' THEN 1 ELSE 0 END AS promotion_
```

```
classified_review = pd.DataFrame(output)
```

```
classified_review.to_csv(r'C:\Users\Admin\Downloads\classified_review.csv', index=False)
```

Import the 2nd table into the database

```
product_infor = pd.read_csv(r"C:\Users\Admin\Downloads\product_infor.csv")
```

```
product_infor.to_sql("product_infor", connection_string )
```

Start cleaning the data

```
%%sql
SELECT *
FROM product_infor
```

Check if there are any duplicate rows in the Item_ID column

```
%%sql
SELECT *
FROM product_infor
WHERE item_id IN (
    SELECT item_id
    FROM product_infor
    GROUP BY item_id
    HAVING COUNT(item_id) > 1
);
```

Replace null value with 0

```
%%sql
UPDATE product_infor
SET avg_rating = 0
WHERE avg_rating IS NULL
```

```
%%sql
UPDATE product_infor
SET rating_count = 0
WHERE rating_count IS NULL
```

```
%%sql
SELECT *
FROM product_infor
WHERE avg_rating IS NULL OR rating_count IS NULL
```

turn it into a dataframe and save as a csv file

```
cleaned_productInfor = %sql SELECT * FROM product_infor
```

```
output_product = pd.DataFrame(cleaned_productInfor)
```

```
output_product.to_csv(r'C:\Users\Admin\Downloads\output_product.csv', index=False)
```