Завдання 2. Очистіть дані, видаляючи будь-які рядки з пропущеними значеннями.

- Кількість записів у таблиці products

```
df_products = products.agg(
    count("category").alias("count_category"),
    count("price").alias("count_price"),
    count("product_name").alias("count_product_name"),
    count("product_id").alias("product_id_cnt")
)

df_products.show()

#no Nulls
```

✓ 0.6s

```
+--------------+-----------+------------------+--------------+
|count_category|count_price|count_product_name|product_id_cnt|
+--------------+-----------+------------------+--------------+
|            49|         49|                49|            49|
+--------------+-----------+------------------+--------------+
```

- Кількість записів у таблиці users до та після видалення пропусків.

```
def users_count(table = users):
    """Функція для підрахунку кількості записів у таблиці users"""
    df_users = table.agg(
    count("name").alias("count_name"),
    count("age").alias("count_age"),
    count("email").alias("count_email"),
    count("user_id").alias("user_id_count")
    )

    df_users.show()

users_count()
```

✓ 0.1s

```
+----------+---------+-----------+-------------+
|count_name|count_age|count_email|user_id_count|
+----------+---------+-----------+-------------+
|        98|       98|         99|          100|
+----------+---------+-----------+-------------+
```

```
users_cleaned = users.dropna()
users_count(users_cleaned)
```

✓ 0.2s

```
+----------+---------+-----------+-------------+
|count_name|count_age|count_email|user_id_count|
+----------+---------+-----------+-------------+
|        95|       95|         95|           95|
+----------+---------+-----------+-------------+
```

- Кількість записів у таблиці purchases до та після видалення пропусків.

```
def orders_count(table=orders):
    """Функція для підрахунку кількості записів у таблиці purchases (переіменувала на orders)"""
    df_orders = table.agg(
    count("user_id").alias("count_user_id"),
    count("product_id").alias("count_product_id"),
    count("date").alias("count_date"),
    count("quantity").alias("count_quantity"),
    count("purchase_id").alias("user_id_count")
    )

    df_orders.show()

orders_count()
```

[10] ✓ 0.3s

```
+-------------+----------------+----------+--------------+-------------+
|count_user_id|count_product_id|count_date|count_quantity|user_id_count|
+-------------+----------------+----------+--------------+-------------+
|          198|             199|       199|           199|          200|
+-------------+----------------+----------+--------------+-------------+
```

```
orders_cleaned = orders.dropna()

orders_count(orders_cleaned)
```

[11] ✓ 0.3s

```
+-------------+----------------+----------+--------------+-------------+
|count_user_id|count_product_id|count_date|count_quantity|user_id_count|
+-------------+----------------+----------+--------------+-------------+
|          195|             195|       195|           195|          195|
+-------------+----------------+----------+--------------+-------------+
```

Завдання 3. Визначте загальну суму покупок за кожною категорією продуктів.

```
orders_tbl = orders_cleaned.select("product_id", "date", "quantity")

combined_table = orders_tbl.join(products, orders_tbl.product_id==products.product_id, "left")\
                .drop(products.product_id)\
                .dropna(subset=["category"])\
                .withColumn("price", col("price").cast("float"))\
                .withColumn("quantity", col("quantity").cast("int"))\
                .withColumn("total_price", round(col("price")*(col("quantity")), 2))\
                .select("category", "total_price")\
                .groupBy("category")\
                .agg(round(sum("total_price"), 2).alias("revenue_by_category"))

combined_table.show()
```

[12] ✓ 1.2s

```
+-----------+------------------+
|   category|revenue_by_category|
+-----------+------------------+
|       Home|            1552.2|
|     Sports|            1802.5|
|Electronics|            1174.8|
|   Clothing|             790.3|
|     Beauty|             459.9|
+-----------+------------------+
```

4. Визначте суму покупок за кожною категорією продуктів для вікової категорії від 18 до 25 включно.

```
users_1825 = users_cleaned.select("*")\
                          .withColumn("age", col("age").cast("int"))\
                          .where((col("age")>=18) & (col("age")<=25))
usr_orders = orders_cleaned.join(
        users_1825, orders_cleaned.user_id==users_1825.user_id, "inner")\
        .join(products, orders_cleaned.product_id==products.product_id, "left")\
        .withColumn("price", col("price").cast("float"))\
        .withColumn("quantity", col("quantity").cast("int"))\
        .withColumn("total_price", round(col("price")*(col("quantity")), 2))\
        .groupBy("age", "category")\
        .agg(round(sum("total_price"), 2).alias("sum_total_price")).dropna()

usr_orders.orderBy(col("age").asc(), col("sum_total_price").desc()).show()
```

[48] ✓ 0.3s

```
+---+-----------+---------------+
|age|   category|sum_total_price|
+---+-----------+---------------+
| 18|       Home|          186.4|
| 18|Electronics|           73.6|
| 20|       Home|           84.6|
| 20|     Sports|           76.8|
| 20|Electronics|           46.0|
| 20|   Clothing|           36.0|
| 20|     Beauty|           33.2|
| 21|   Clothing|           21.0|
| 21|       Home|            8.8|
| 21|     Sports|            8.4|
| 23|Electronics|           84.6|
| 23|     Sports|           67.5|
| 23|       Home|           28.7|
| 24|       Home|           81.3|
```

5. Визначте частку покупок за кожною категорією товарів від сумарних витрат для вікової категорії від 18 до 25 років.

```
total_spent = usr_orders.agg(sum("sum_total_price").alias("total_spent"))\
                        .collect()[0]["total_spent"]

users_1825_stats = usr_orders.groupBy("category") \
            .agg(round(sum("sum_total_price"), 2).alias("spent_by_price")) \
            .withColumn("total_spent", round(lit(total_spent), 2))\
            .withColumn("spent_ratio",
            round(col("spent_by_price").cast("float") / col("total_spent").cast("float"), 2))

users_1825_stats.show()
```

[51] ✓ 0.9s

```
+-----------+--------------+-----------+-----------+
|   category|spent_by_price|total_spent|spent_ratio|
+-----------+--------------+-----------+-----------+
|       Home|         389.8|     1236.3|       0.32|
|     Sports|         310.5|     1236.3|       0.25|
|Electronics|         249.6|     1236.3|        0.2|
|   Clothing|         245.0|     1236.3|        0.2|
|     Beauty|          41.4|     1236.3|       0.03|
+-----------+--------------+-----------+-----------+
```

6. Виберіть 3 категорії продуктів з найвищим відсотком витрат споживачами віком від 18 до 25 років.

```
users_1825_stats.select("category", "spent_by_price", "total_spent", "spent_ratio")\
                .orderBy(col("spent_ratio").desc()).limit(3).show()
```

[55] ✓ 0.4s

```
+-----------+-------------+-----------+-----------+
|   category|spent_by_price|total_spent|spent_ratio|
+-----------+-------------+-----------+-----------+
|       Home|         389.8|     1236.3|       0.32|
|     Sports|         310.5|     1236.3|       0.25|
|Electronics|         249.6|     1236.3|        0.2|
+-----------+-------------+-----------+-----------+
```