PHASE 1 Goal: Research-grade transaction timeline

```
# STEP 1.1 – Dataset Loading

import pandas as pd

# Load both sheets
df_1 = pd.read_excel("/content/online_retail_II.xlsx", sheet_name=0)
df_2 = pd.read_excel("/content/online_retail_II.xlsx", sheet_name=1)

# Combine
df = pd.concat([df_1, df_2], ignore_index=True)

print(df.shape)
df.head()
```

```
(1067371, 8)
```

{"type":"dataframe","variable_name":"df"}

```
# STEP 1.2 – Column Sanity Check

df.columns
```

```
Index(['Invoice', 'StockCode', 'Description', 'Quantity',
'InvoiceDate',
       'Price', 'Customer ID', 'Country'],
      dtype='object')
```

```
# STEP 1.3 – Hard Cleaning Rules (CLV-Safe)

# Rule 1: Drop missing customers
df = df.dropna(subset=["Customer ID"])

# Rule 2: Convert date properly
df["InvoiceDate"] = pd.to_datetime(df["InvoiceDate"])

# Rule 3: Handle cancellations carefully
df["is_cancelled"] = df["Invoice"].astype(str).str.startswith("C")

# STEP 1.4 – Monetary Value Construction

df["revenue"] = df["Quantity"] * df["Price"]

# STEP 1.5 – Temporal Ordering (MOST IMPORTANT)

df = df.sort_values(
    by=["Customer ID", "InvoiceDate", "Invoice"]
).reset_index(drop=True)

# STEP 1.6 – Build Customer Event Index
```

```python
df["event_index"] = (
    df.groupby("Customer ID")
      .cumcount()
)
```

```python
# STEP 1.7 — Phase 1 Validation Checks
```

```python
# Check monotonic time per customer
check = (
    df.groupby("Customer ID")["InvoiceDate"]
      .apply(lambda x: x.is_monotonic_increasing)
)
```

```python
print("All customers sorted correctly:", check.all())
```

```
All customers sorted correctly: True
```

```python
# Check customers with at least 2 transactions
(df.groupby("Customer ID").size() >= 2).mean()
```

```
np.float64(0.9754291484348704)
```

```python
# STEP 1.8 — Save Phase 1 Artifact
```

```python
df["Invoice"] = df["Invoice"].astype(str)
df["StockCode"] = df["StockCode"].astype(str)
df.to_parquet("phase1_clean_transactions.parquet", index=False)
```