TUGAS PROJECT
DATA WAREHOUSE DAN INTELIGENSI BISNIS
PROJECT BASED MILESTONE 1

Oleh:

Marta Zuriadi	**24/548101/PPA/06919**
Muhammad Ashabul Kahfi	**24/537433/PPA/06796**
Silvanus Satno Nugraha	**24/548140/PPA/06921**

PROGRAM MAGISTER ILMU KOMPUTER
DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
2025

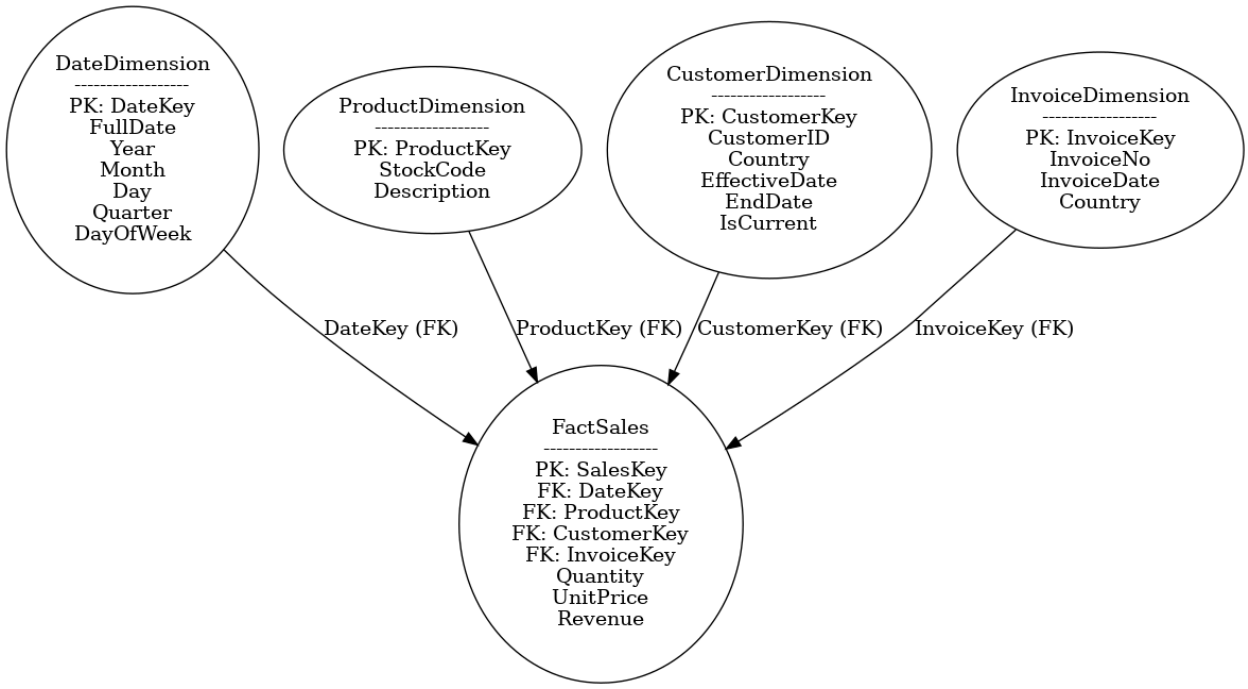
Link Github : https://github.com/satnodeus77/datawarehouse_project_duckdb.git
Dataset : Online Retail
Sumber : Kaggle Dataset
Link : https://www.kaggle.com/datasets/tunguz/online-retail
Domain : Retail

Dataset

Dataset ini merupakan kumpulan data transaksi yang mencakup range waktu terjadi antara 01/12/2010 dan 09/12/2011 untuk sebuah perusahaan ritel daring tanpa toko di UK. Perusahaan ini terutama menjual hadiah unik untuk berbagai acara dan kebanyakan customer merupakan pedagang grosir.

InvoiceNo	Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.
StockCode	Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product
Description	Product (item) name. Nominal.
Quantity	The quantities of each product (item) per transaction. Numeric.
InvoiceDate	Invice Date and time. Numeric, the day and time when each transaction was generated.
UnitPrice	nit price. Numeric, Product price per unit in sterling.
CustomerID	Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.
Country	Country name. Nominal, the name of the country where each customer resides.

ERD


DateDimension
PK: DateKey
FullDate
Year
Month
Day
Quarter
DayOfWeek
ProductDimension
PK: ProductKey
StockCode
Description
CustomerDimension
PK: CustomerKey
CustomerID
Country
EffectiveDate
EndDate
IsCurrent
InvoiceDimension
PK: InvoiceKey
InvoiceNo
InvoiceDate
Country
DateKey (FK)
ProductKey (FK)
CustomerKey (FK)
InvoiceKey (FK)
FactSales
PK: SalesKey
FK: DateKey
FK: ProductKey
FK: CustomerKey
FK: InvoiceKey
Quantity
UnitPrice
Revenue

Business Logic

1. Deskripsi Bisnis

Bisnis online retail ini bergerak di bidang penjualan produk secara daring (online) dengan fokus utama pada penjualan produk fisik. Operasional bisnis ini menghasilkan data transaksi yang komprehensif yang mencakup berbagai informasi penting seperti:

- Nomor Faktur (InvoiceNo): Identifikasi unik untuk setiap transaksi penjualan
- Kode Produk (StockCode): Kode unik untuk identifikasi produk dalam inventaris
- Deskripsi Produk (Description): Penjelasan mengenai produk yang dijual
- Jumlah Produk (Quantity): Jumlah unit produk yang dibeli dalam satu transaksi
- Tanggal Faktur (InvoiceDate): Tanggal dan waktu transaksi dilakukan
- Harga per Unit (UnitPrice): Harga satuan untuk setiap produk
- ID Pelanggan (CustomerID): Identifikasi unik untuk setiap pelanggan
- Negara (Country): Lokasi geografis asal pelanggan

Data transaksi ini memiliki peran strategis dalam bisnis karena digunakan untuk melakukan berbagai analisis penting, termasuk:

1. Analisis Kinerja Penjualan: Memahami pola penjualan, tren pendapatan, dan produk unggulan berdasarkan volume penjualan dan kontribusi pendapatan.
2. Analisis Perilaku Pelanggan: Mempelajari pola pembelian pelanggan, frekuensi transaksi, nilai rata-rata belanja, dan preferensi produk untuk mengembangkan strategi personalisasi dan retensi yang lebih efektif.
3. Evaluasi Strategi Pemasaran: Mengukur efektivitas kampanye pemasaran, promosi, dan strategi harga melalui analisis respons penjualan.
4. Optimalisasi Manajemen Persediaan: Mengidentifikasi produk dengan perputaran cepat dan lambat untuk meningkatkan efisiensi pengelolaan stok dan mengurangi biaya penyimpanan.
5. Analisis Geografis: Memahami distribusi pelanggan secara geografis untuk mengidentifikasi pasar potensial dan menyesuaikan strategi ekspansi.
6. Deteksi Pola Musiman: Mengidentifikasi variasi penjualan berdasarkan musim atau periode tertentu untuk perencanaan operasional dan pemasaran yang lebih baik.

Dengan memanfaatkan data transaksi secara maksimal, bisnis ini dapat mengambil keputusan berbasis data yang lebih akurat, meningkatkan efisiensi operasional, dan mengembangkan strategi yang lebih tepat sasaran untuk meningkatkan pertumbuhan dan profitabilitas jangka panjang.

2. Tujuan Analitis dan Pertanyaan Bisnis yang Harus Dijawab

Tujuan Analitis:

- Meningkatkan Pemahaman Kinerja Penjualan: Mengidentifikasi tren penjualan, pola musiman, dan faktor-faktor yang memengaruhi pendapatan.
- Optimasi Persediaan dan Kinerja Produk: Menentukan produk yang berkinerja tinggi dan rendah untuk meningkatkan efisiensi stok.
- Segmentasi dan Loyalitas Pelanggan: Mengelompokkan pelanggan berdasarkan perilaku pembelian dan nilai seumur hidup pelanggan.

- Evaluasi Strategi Harga dan Promosi: Menganalisis sensitivitas harga dan dampak promosi terhadap volume penjualan.
- Pelacakan Perubahan Historis Pelanggan: Memanfaatkan implementasi SCD Type 2 untuk memantau perubahan profil pelanggan seiring waktu.

Pertanyaan Bisnis:

- Berapa total dan rata-rata pendapatan yang diperoleh setiap hari, minggu, bulan, atau tahun?
- Pada periode apa saja terjadi peningkatan penjualan (pola musiman) dan bagaimana tren tersebut dapat dimanfaatkan untuk perencanaan inventaris?
- Produk apa saja yang menyumbang pendapatan tertinggi dan seberapa menguntungkannya produk tersebut?
- Bagaimana performa produk, dan produk mana yang memiliki perputaran stok rendah sehingga memerlukan diskon atau promosi khusus?
- Bagaimana perilaku pembelian ulang pelanggan, dan berapa nilai seumur hidup (Customer Lifetime Value) yang dapat dihasilkan?
- Negara atau wilayah mana yang memberikan kontribusi penjualan tertinggi?
- Seberapa tinggi tingkat pengembalian barang (return) dan apa dampaknya terhadap pendapatan bersih?
- Bagaimana perubahan data historis pelanggan (misalnya, perubahan negara asal) mempengaruhi penjualan dan distribusi pasar?

3. Sumber Data

Dataset utama yang tersedia untuk analisis adalah Online_Retail Dataset yang disimpan dalam format CSV. Dataset ini merupakan sumber data primer yang mencatat seluruh transaksi penjualan online dan terdiri dari beberapa kolom kunci:

- InvoiceNo: Nomor identifikasi unik untuk setiap transaksi penjualan. Nomor ini dapat digunakan untuk mengelompokkan item yang dibeli dalam satu transaksi yang sama.
- StockCode: Kode produk unik yang digunakan untuk mengidentifikasi setiap produk dalam inventaris perusahaan.
- Description: Deskripsi tekstual dari produk yang memberikan informasi detail tentang karakteristik dan spesifikasi produk.
- Quantity: Jumlah unit produk yang dibeli dalam transaksi tersebut. Nilai negatif pada kolom ini biasanya mengindikasikan transaksi pengembalian (return).
- InvoiceDate: Tanggal dan waktu terjadinya transaksi yang dapat digunakan untuk analisis tren waktu dan pola musiman.
- UnitPrice: Harga per unit produk yang dapat digunakan untuk menghitung nilai total transaksi.
- CustomerID: Identifikasi unik untuk setiap pelanggan yang memungkinkan analisis perilaku pembelian individual.
- Country: Negara asal pelanggan yang memungkinkan analisis geografis dan segmentasi berdasarkan wilayah.

Selain dataset utama, terdapat beberapa sumber data pendukung yang dapat diintegrasikan untuk analisis yang lebih komprehensif:

1. Data CRM (Customer Relationship Management):
    - Informasi demografi pelanggan seperti usia, jenis kelamin, dan preferensi
    - Riwayat interaksi pelanggan dengan layanan pelanggan
    - Data aktivitas dan respons kampanye pemasaran
2. Data Log Website:
    - Pola penelusuran produk oleh pelanggan
    - Waktu yang dihabiskan pada halaman produk tertentu
    - Jalur konversi dari penelusuran hingga pembelian
    - Tingkat abandonment cart (keranjang belanja yang ditinggalkan)
3. Laporan Pengiriman:
    - Waktu pengiriman dan penerimaan produk
    - Data pengembalian dan alasannya
    - Biaya pengiriman dan metode yang dipilih
    - Masalah dalam proses pengiriman

Integrasi data utama dengan data pelengkap ini akan memberikan perspektif yang lebih holistik tentang seluruh proses bisnis, memungkinkan analisis yang lebih mendalam tentang perilaku pelanggan, efektivitas operasional, dan faktor-faktor yang mempengaruhi kepuasan pelanggan serta loyalitas jangka panjang.

4. Indikator Utama (KPI) yang Perlu Dimonitor

Dalam pengelolaan bisnis retail online, pemantauan indikator kinerja utama (KPI) sangat penting untuk mengukur kesuksesan operasional dan mengidentifikasi area yang memerlukan perbaikan. Berikut adalah penjelasan detail tentang KPI utama yang perlu dimonitor:

1. **Total Pendapatan**

   Definisi: Keseluruhan nilai penjualan yang dihasilkan dari semua transaksi dalam periode waktu tertentu.

   Cara Perhitungan:

```
Unset
Total Pendapatan = Σ(Quantity × UnitPrice) untuk semua transaksi dengan Quantity > 0
```

   Signifikansi:
   - Mengukur pertumbuhan bisnis secara keseluruhan
   - Menjadi dasar untuk perhitungan profitabilitas
   - Membantu mengevaluasi efektivitas strategi pemasaran dan penjualan

   Visualisasi Ideal: Grafik garis waktu dengan kemampuan drill-down berdasarkan periode (hari, minggu, bulan, tahun)

2. **Pendapatan Rata-rata per Faktur**

   Definisi: Rata-rata nilai moneter yang diperoleh dari setiap transaksi penjualan.

   Cara Perhitungan:

```
Unset
Pendapatan Rata-rata per Faktur = Total Pendapatan ÷ Jumlah Faktur Unik
```

Signifikansi:

- Indikator nilai belanja rata-rata pelanggan
- Parameter untuk mengukur efektivitas strategi up-selling dan cross-selling
- Membantu mengidentifikasi tren perubahan pola belanja

Visualisasi Ideal: Grafik kombinasi dengan tren total pendapatan untuk mendeteksi korelasi

3. **Volume Penjualan Produk**

   Definisi: Jumlah total unit yang terjual dari setiap produk dalam katalog.

   Cara Perhitungan:

```
Unset
Volume Penjualan Produk = Σ(Quantity) untuk setiap StockCode dengan Quantity > 0
```

Signifikansi:

- Mengidentifikasi produk unggulan dan produk dengan performa rendah
- Dasar untuk keputusan manajemen persediaan
- Membantu perencanaan produksi dan pengadaan

Visualisasi Ideal: Diagram Pareto atau heat map produk berdasarkan kategori

4. **Tingkat Pengembalian Barang**

   Definisi: Persentase transaksi yang mengindikasikan pengembalian atau pembatalan produk.

   Cara Perhitungan:

```
Unset
Tingkat Pengembalian = (Jumlah Transaksi dengan Quantity < 0 ÷ Total Transaksi) × 100%
```

Signifikansi:

- Indikator kepuasan pelanggan dan kualitas produk
- Parameter untuk evaluasi deskripsi produk dan ekspektasi pelanggan
- Mengidentifikasi masalah dalam proses pengiriman atau kualitas produk

Visualisasi Ideal: Grafik tren waktu dan breakdown berdasarkan produk atau kategori produk.

5. **Nilai Seumur Hidup Pelanggan (CLV)**
   Definisi: Estimasi total pendapatan yang dihasilkan oleh seorang pelanggan selama hubungan bisnisnya dengan perusahaan.
   Cara Perhitungan:

```
Unset
```

- `CLV = Nilai Rata-rata Pembelian × Frekuensi Pembelian × Lama Hubungan Pelanggan`

   Signifikansi:
   - Dasar untuk strategi retensi pelanggan
   - Membantu mengalokasikan anggaran akuisisi pelanggan
   - Mengidentifikasi segmen pelanggan bernilai tinggi

   Visualisasi Ideal: Segmentasi pelanggan berdasarkan matriks RFM (Recency, Frequency, Monetary)
6. **Frekuensi Pembelian**
   Definisi: Jumlah rata-rata transaksi yang dilakukan oleh pelanggan dalam periode waktu tertentu.
   Cara Perhitungan:

```
Unset
Frekuensi Pembelian = Jumlah Faktur Unik per CustomerID ÷ Periode Waktu (dalam bulan)
```

   Signifikansi:
   - Indikator loyalitas dan retensi pelanggan
   - Parameter untuk mengukur efektivitas program loyalitas
   - Membantu mengidentifikasi pelanggan berpotensi churn

   Visualisasi Ideal: Distribusi frekuensi dan segmentasi pelanggan berdasarkan aktivitas
7. **Analisis Geografis**
   Definisi: Distribusi pendapatan berdasarkan negara atau wilayah geografis.
   Cara Perhitungan:

```
Unset
Pendapatan per Negara = Σ(Quantity × UnitPrice) dikelompokkan berdasarkan Country
```

   Signifikansi:
   - Mengidentifikasi pasar utama dan pasar berkembang
   - Dasar untuk strategi ekspansi geografis

- Membantu mengoptimalkan strategi logistik dan pengiriman

Visualisasi Ideal: Peta panas geografis dengan kemampuan drill-down ke tingkat regional

Idealnya, semua KPI ini ditampilkan dalam dashboard terintegrasi yang memungkinkan:

- Pemantauan real-time atau near real-time
- Perbandingan dengan periode sebelumnya
- Analisis tren jangka panjang
- Filter berdasarkan berbagai dimensi (waktu, produk, geografi)
- Penetapan target dan visualisasi pencapaian

Dengan memantau KPI ini secara konsisten, bisnis retail online dapat mengambil keputusan berbasis data yang lebih efektif, mengidentifikasi peluang pertumbuhan, dan mengatasi tantangan operasional dengan lebih cepat.

5. Jenis Laporan dan Analisis yang Dibutuhkan

Untuk mendukung pengambilan keputusan berbasis data yang efektif, beberapa jenis laporan dan analisis berikut perlu dikembangkan:

1. **Laporan Tren Penjualan**

   Deskripsi: Visualisasi komprehensif yang menampilkan pola penjualan dalam berbagai interval waktu.

   Komponen Utama:

   - Grafik garis interaktif yang menunjukkan pendapatan harian, mingguan, bulanan, dan tahunan
   - Diagram batang perbandingan pendapatan periode saat ini vs periode sebelumnya
   - Heat map kalender yang menunjukkan puncak aktivitas penjualan
   - Analisis pola musiman dan identifikasi anomali penjualan

   Manfaat Bisnis:

   - Membantu mengantisipasi fluktuasi permintaan untuk perencanaan inventaris
   - Mengidentifikasi tren jangka panjang untuk strategi pertumbuhan
   - Mendeteksi hari/periode puncak penjualan untuk optimasi staf dan operasional

   Contoh Implementasi: Dashboard interaktif dengan filter waktu dan kemampuan drill-down dari tahunan hingga harian.

2. **Analisis Kinerja Produk**

   Deskripsi: Evaluasi multidimensi performa produk dalam katalog.

   Komponen Utama:

   - Peringkat produk berdasarkan pendapatan dan volume penjualan

- Analisis ABC (80/20) untuk kategorisasi produk
- Matriks analisis produk berdasarkan margin dan volume
- Analisis afinitas produk (market basket analysis) untuk mengidentifikasi pola cross-selling
- Visualisasi tren produk berdasarkan musim atau periode waktu

Manfaat Bisnis:

- Optimalisasi katalog produk dan keputusan penghentian produk
- Panduan strategi promosi bundling produk
- Informasi untuk rekomendasi produk dan personalisasi

Contoh Implementasi: Dashboard dengan tabel peringkat produk, heat map afinitas produk, dan grafik tren performa produk dari waktu ke waktu.

3. **Segmentasi Pelanggan**

   Deskripsi: Kategorisasi pelanggan berdasarkan karakteristik dan perilaku pembelian.

   Komponen Utama:

   - Analisis RFM (Recency, Frequency, Monetary) untuk segmentasi pelanggan
   - Visualisasi distribusi pelanggan berdasarkan nilai seumur hidup (CLV)
   - Pemetaan geolokasi pelanggan dengan nilai tertinggi
   - Analisis perjalanan pelanggan (customer journey) dan titik kontak
   - Diagram Sankey yang menunjukkan alur konversi segmen pelanggan

   Manfaat Bisnis:

   - Strategi pemasaran yang lebih tepat sasaran
   - Peningkatan retensi melalui identifikasi pelanggan berisiko churn
   - Optimalisasi alokasi anggaran pemasaran berdasarkan nilai segmen

   Contoh Implementasi: Dashboard interaktif dengan matriks segmentasi, visualisasi geografis, dan detail perilaku per segmen.

4. **Analisis Pengembalian Barang**

   Deskripsi: Evaluasi komprehensif pola pengembalian produk dan dampaknya.

   Komponen Utama:

   - Tren tingkat pengembalian produk dari waktu ke waktu
   - Breakdown pengembalian berdasarkan kategori produk, harga, dan karakteristik
   - Analisis korelasi antara deskripsi produk dan tingkat pengembalian
   - Identifikasi pelanggan dengan pola pengembalian tinggi

   Manfaat Bisnis:

   - Peningkatan kualitas produk berdasarkan umpan balik pengembalian
   - Perbaikan akurasi deskripsi produk untuk mengurangi ketidaksesuaian ekspektasi
   - Optimalisasi kebijakan pengembalian barang

Contoh Implementasi: Dashboard dengan grafik tren pengembalian, diagram Pareto produk dengan tingkat pengembalian tertinggi, dan analisis alasan pengembalian.

5. **Laporan Harga dan Promosi**

   Deskripsi: Analisis hubungan antara strategi penetapan harga, promosi, dan respons penjualan.

   Komponen Utama:
   - Kurva elastisitas harga untuk kategori produk utama
   - Analisis dampak promosi terhadap volume penjualan dan margin
   - Perbandingan efektivitas berbagai jenis promosi
   - Analisis cannibalization effect antar produk selama promosi
   - Visualisasi harga optimal berdasarkan data historis

   Manfaat Bisnis:
   - Optimalisasi strategi harga untuk memaksimalkan margin
   - Peningkatan ROI kampanye promosi
   - Panduan penjadwalan dan penargetan promosi

   Contoh Implementasi: Dashboard dengan grafik elastisitas harga, perbandingan performa promosi, dan rekomendasi harga.

6. **Dashboard KPI Bisnis**

   Deskripsi: Tampilan konsolidasi metrik-metrik utama bisnis dalam satu antarmuka.

   Komponen Utama:
   - Ringkasan visual KPI utama seperti total pendapatan, margin rata-rata, dan CLV
   - Indikator performa dengan kode warna berdasarkan target
   - Tren historis KPI dengan proyeksi
   - Pemantauan jumlah pelanggan aktif dan tingkat akuisisi/churn

   Manfaat Bisnis:
   - Visibilitas instan terhadap kondisi bisnis secara keseluruhan
   - Deteksi dini area yang memerlukan perhatian
   - Pengukuran dampak inisiatif strategis terhadap KPI

   Contoh Implementasi: Dashboard eksekutif dengan kartu KPI, grafik tren, dan kemampuan drill-down untuk analisis lebih mendalam.

7. **Laporan Historis Pelanggan (SCD Type 2)**

   Deskripsi: Visualisasi perubahan atribut pelanggan dari waktu ke waktu menggunakan metodologi Slowly Changing Dimension Type 2.

   Komponen Utama:
   - Timeline perubahan atribut pelanggan utama (alamat, negara, preferensi)
   - Analisis dampak perubahan terhadap pola pembelian
   - Visualisasi migrasi pelanggan antar segmen atau wilayah geografis

- Identifikasi tren perubahan global dalam basis pelanggan

Manfaat Bisnis:

- Pemahaman lebih mendalam tentang evolusi basis pelanggan
- Deteksi tren relokasi pelanggan untuk perencanaan ekspansi
- Evaluasi dampak perubahan demografis terhadap strategi produk

Contoh Implementasi: Dashboard dengan timeline perubahan, peta migrasi pelanggan, dan analisis tren perubahan atribut.

Semua laporan dan analisis di atas sebaiknya dirancang dengan pendekatan interaktif yang memungkinkan pengguna untuk melakukan eksplorasi data mandiri, mengajukan pertanyaan lanjutan, dan mengkustomisasi visualisasi sesuai kebutuhan spesifik mereka.

ETL

Extract Data

```
    # Load CSV data
    df = pd.read_csv("Online_Retail.csv", encoding="latin1")
    logger.info("Data loaded successfully with shape: %s", df.shape)

    # Remove duplicate rows
    initial_count = len(df)
    df.drop_duplicates(inplace=True)
    logger.info("Duplicates removed: %d rows dropped", initial_count -
len(df))

    # Remove rows with missing critical values (InvoiceNo, StockCode,
InvoiceDate, Quantity, UnitPrice)
    required_cols = ['InvoiceNo', 'StockCode', 'InvoiceDate', 'Quantity',
'UnitPrice']
    df.dropna(subset=required_cols, inplace=True)
    logger.info("Rows with nulls in critical columns removed. Current shape:
%s", df.shape)

    # Convert InvoiceDate to datetime; drop rows where conversion fails
```

```
    df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'], errors='coerce')
    df = df[df['InvoiceDate'].notnull()]
    logger.info("InvoiceDate conversion completed. Current shape: %s",
df.shape)

    # Remove rows with negative UnitPrice if they are considered errors
    neg_unitprice_count = (df['UnitPrice'] < 0).sum()
    if neg_unitprice_count > 0:
        logger.warning("Found %d rows with negative UnitPrice. These rows
will be removed.", neg_unitprice_count)
        df = df[df['UnitPrice'] >= 0]
    logger.info("Final shape after data cleaning: %s", df.shape)

    # Compute Revenue column
    df['Revenue'] = df['Quantity'] * df['UnitPrice']
    logger.info("Revenue column calculated.")
```

Transform

```
    # Build DateDimension
    date_df = df[['InvoiceDate']].drop_duplicates().reset_index(drop=True)
    date_df['DateKey'] = date_df.index + 1
    date_df['FullDate'] = date_df['InvoiceDate'].dt.date
    date_df['Year'] = date_df['InvoiceDate'].dt.year
    date_df['Month'] = date_df['InvoiceDate'].dt.month
    date_df['Day'] = date_df['InvoiceDate'].dt.day
    date_df['Quarter'] = date_df['InvoiceDate'].dt.quarter
    date_df['DayOfWeek'] = date_df['InvoiceDate'].dt.dayofweek  # Monday=0,
Sunday=6
    date_dim =
date_df[['DateKey','FullDate','Year','Month','Day','Quarter','DayOfWeek']].c
opy()
    logger.info("Transformation: DateDimension built with %s rows",
len(date_dim))
```

```
# Build ProductDimension
    product_df =
df[['StockCode','Description']].drop_duplicates().reset_index(drop=True)
    product_df['ProductKey'] = product_df.index + 1
    product_dim =
product_df[['ProductKey','StockCode','Description']].copy()
    logger.info("Transformation: ProductDimension built with %s rows",
len(product_dim))

# Build CustomerDimension with SCD Type 2 columns
    customer_df =
df[['CustomerID','Country']].dropna(subset=['CustomerID']).drop_duplicates()
.copy()
    customer_df['CustomerID'] =
customer_df['CustomerID'].astype(int).astype(str)
    customer_df = customer_df.reset_index(drop=True)
    customer_df['CustomerKey'] = customer_df.index + 1
    today_date = date.today()  # plain Python date
    customer_dim =
customer_df[['CustomerKey','CustomerID','Country']].copy()
    customer_dim['EffectiveDate'] = today_date
    customer_dim['EndDate'] = None
    customer_dim['IsCurrent'] = True
    logger.info("Transformation: CustomerDimension built with %s rows",
len(customer_dim))

# Build InvoiceDimension (4th Dimension)
    invoice_df =
df[['InvoiceNo','InvoiceDate','Country']].drop_duplicates().reset_index(drop
=True)
    invoice_df['InvoiceKey'] = invoice_df.index + 1
    invoice_dim =
invoice_df[['InvoiceKey','InvoiceNo','InvoiceDate','Country']].copy()
    invoice_dim['InvoiceDate'] = invoice_dim['InvoiceDate'].dt.date
```

```
    logger.info("Transformation: InvoiceDimension built with %s rows",
len(invoice_dim))

# Build FactSales table
    df_for_fact = pd.merge(df, date_df[['InvoiceDate','DateKey']],
on='InvoiceDate', how='left')
    df_for_fact = pd.merge(
        df_for_fact,
        product_df[['StockCode','Description','ProductKey']],
        on=['StockCode','Description'],
        how='left'
    )
    df_for_fact['CustomerID'] =
df_for_fact['CustomerID'].astype('float').astype('Int64').astype(str)
    df_for_fact = pd.merge(
        df_for_fact,
        customer_df[['CustomerID','Country','CustomerKey']],
        left_on=['CustomerID','Country'],
        right_on=['CustomerID','Country'],
        how='left'
    )
    df_for_fact = pd.merge(
        df_for_fact,
        invoice_dim[['InvoiceNo','InvoiceKey']],
        on='InvoiceNo',
        how='left'
    )
    fact_sales = df_for_fact[[
        'InvoiceNo',
        'DateKey',
        'ProductKey',
        'CustomerKey',
        'InvoiceKey',
        'Quantity',
        'UnitPrice',
```

```
        'Revenue'
    ]].copy()
    fact_sales = fact_sales.reset_index(drop=True)
    fact_sales['SalesKey'] = fact_sales.index + 1
    fact_sales = fact_sales[[
        'SalesKey',
        'DateKey',
        'ProductKey',
        'CustomerKey',
        'InvoiceKey',
        'Quantity',
        'UnitPrice',
        'Revenue'
    ]]
    logger.info("Transformation: FactSales built with %s rows",
len(fact_sales))
except Exception as e:
    logger.error("Error building FactSales: %s", e)
    raise
```

Load

```
try:
    db_path = "online_retail.duckdb"
    con = duckdb.connect(db_path)
    logger.info("Connected to DuckDB at %s", db_path)
except Exception as e:
    logger.error("Error connecting to DuckDB: %s", e)
    raise

try:
    con.execute("CREATE SCHEMA IF NOT EXISTS retail")
    logger.info("Schema 'retail' is ready.")
```

```
con.execute("""
    CREATE OR REPLACE TABLE retail.DateDimension (
        DateKey INTEGER,
        FullDate DATE,
        Year INT,
        Month INT,
        Day INT,
        Quarter INT,
        DayOfWeek INT
    )
""")
logger.info("Table retail.DateDimension created.")

con.execute("""
    CREATE OR REPLACE TABLE retail.ProductDimension (
        ProductKey INTEGER,
        StockCode VARCHAR,
        Description VARCHAR
    )
""")
logger.info("Table retail.ProductDimension created.")

con.execute("""
    CREATE OR REPLACE TABLE retail.CustomerDimension (
        CustomerKey INTEGER,
        CustomerID VARCHAR,
        Country VARCHAR,
        EffectiveDate DATE,
        EndDate DATE,
        IsCurrent BOOLEAN
    )
""")
logger.info("Table retail.CustomerDimension created.")
```

```
    con.execute("""
        CREATE OR REPLACE TABLE retail.InvoiceDimension (
            InvoiceKey INTEGER,
            InvoiceNo VARCHAR,
            InvoiceDate DATE,
            Country VARCHAR
        )
    """)
    logger.info("Table retail.InvoiceDimension created.")

    con.execute("""
        CREATE OR REPLACE TABLE retail.FactSales (
            SalesKey INTEGER,
            DateKey INTEGER,
            ProductKey INTEGER,
            CustomerKey INTEGER,
            InvoiceKey INTEGER,
            Quantity INTEGER,
            UnitPrice DOUBLE,
            Revenue DOUBLE
        )
    """)
    logger.info("Table retail.FactSales created.")
except Exception as e:
    logger.error("Error creating tables: %s", e)
    raise
```