Task 1 - Data Analysis & Understanding

Import Python Libraries Connect to DB

In [1]:
```
#pip install pandas
import sqlite3
import pandas as pd

# DB File Path
db_path = "/Users/deepthi.matta/dbt_test_projects/customer_invoices.db"
# Connect to customer_invoices database
conn = sqlite3.connect(db_path)
```

Query the tables

Abrechnung_Kunden

In [2]:
```
query = "SELECT * FROM Abrechnung_Kunden;"
dataframe = pd.read_sql_query(query, conn)
# Display the output
dataframe
```

Out[2]:

	id	Kdnr	Verlagsname	Region
0	5	20172	1. FC Nürnberg	Nürnberg
1	19	20137	Allgäuer Zeitung / Allgäuer Zeitungsverlag GmbH	Bodensee
2	27	20115	Augsburger Allgemeine	München
3	69	10113	Brainpool TV Productions	Rheinland
4	72	10154	RFW / Redaktionsbüro Wipperfürth	Hamburg
...	...	...	...	...
596	14014	81391	PPF	NULL
597	14021	81398	VLAAMSE RADIO- EN TELEVISIEOMROEP (VRT)	NULL
598	14025	81402	Umweltinstitut München	NULL
599	14031	81408	Everprod - Groupe Elephant	NULL
600	14034	81411	Lennart Homeyer	NULL

601 rows × 4 columns

Abrechnung_Rechnungen

In [3]:

```
query = "SELECT * FROM Abrechnung_Rechnungen;"
dataframe = pd.read_sql_query(query, conn)
# Display the output
dataframe
```

Out[3]:

	ReNummer	SummeNetto	MwStSatz	ZahlungsbetragBrutto	KdNr	Summenebenkosten	ReDatum	Zahlungsdatum
0	101602	145.0	0	145	81044	0	2024-11-04 00:00:00.000	2024-11-08 00:00:00.000
1	101603	375.0	7	401.25	20843	0	2024-11-04 00:00:00.000	2024-11-28 00:00:00.000
2	101604	94.5	7	101.12	20843	0	2024-11-04 00:00:00.000	2024-11-18 00:00:00.000
3	101605	3450.0	7	3691.5	20020	0	2024-11-04 00:00:00.000	2024-11-12 00:00:00.000
4	101606	37550.0	0	37842.17	78962	-292.17	2024-11-04 00:00:00.000	2024-11-07 00:00:00.000
...	...	...	...	...	...	...	...	...
1995	103597	1260.0	7	0	20213	0	2025-04-03 00:00:00.000	NULL
1996	103598	225.0	7	0	10383	0	2025-04-01 00:00:00.000	NULL
1997	103599	160.0	7	0	30145	0	2025-04-04 00:00:00.000	NULL
1998	103600	379.0	7	0	79666	0	2025-04-03 00:00:00.000	NULL
1999	103601	11786.7	0	0	78911	0	2025-04-04 00:00:00.000	NULL

2000 rows × 8 columns

Abrechnung_Positionen

In [4]:

```
query = "SELECT * FROM Abrechnung_Positionen;"
dataframe = pd.read_sql_query(query, conn)
# Display the output
dataframe
```

Out[4]:

	id	ReId	KdNr	Nettobetrag	Bildnummer	VerDatum
0	4069567	103172	30035	45.0	92104298	2021-03-15 00:00:00.000
1	4069568	103172	30035	15.0	76396227	2021-03-15 00:00:00.000
2	4069569	103172	30035	140.0	88872289	2021-04-15 00:00:00.000
3	4069570	103172	30035	30.0	78670291	2021-05-15 00:00:00.000
4	4069571	103172	30035	45.0	51407649	2021-05-15 00:00:00.000
...	...	...	...	...	...	...
129087	5726954	103389	20115	20.0	1058373276	2025-02-05 00:00:00.000
129088	5726955	103389	20115	20.0	105481711	2025-02-07 00:00:00.000
129089	5726956	103389	20115	20.0	100000000	2025-02-08 00:00:00.000
129090	5726957	103389	20115	20.0	1058638486	2025-02-10 00:00:00.000
129091	5726958	103389	20115	20.0	1058677007	2025-02-11 00:00:00.000

129092 rows × 6 columns

Source Files Data Checks

In [5]:

```
# Run SQL query -  Check for NULL value in the fields.
query_null_checks = '''select count(1) as null_count, 'id' as col_nm , 'Abrechnung_Kunden' AS table_name
from Abrechnung_Kunden where id ='NULL'
        union
        select count(1) , 'Kdnr' as col_nm , 'Abrechnung_Kunden' AS table_name
        from Abrechnung_Kunden where Kdnr ='NULL'
        union
        select count(1) , 'Verlagsname' as col_nm , 'Abrechnung_Kunden' AS table_name
        from Abrechnung_Kunden where Verlagsname ='NULL'
        union
        select count(1) , 'Region' as col_nm , 'Abrechnung_Kunden' AS table_name
        from Abrechnung_Kunden where Region ='NULL'
        union
        select count(1) , 'id' as col_nm , 'Abrechnung_Positionen' AS table_name
        from Abrechnung_Positionen where id ='NULL'
        union
```

```
        select count(1) , 'Reid' as col_nm , 'Abrechnung_Positionen' AS table_name
        from Abrechnung_Positionen where Reid ='NULL'
        union
        select count(1) , 'KdNr' as col_nm , 'Abrechnung_Positionen' AS table_name
        from Abrechnung_Positionen where KdNr ='NULL'
        union
        select count(1) , 'Nettobetrag' as col_nm, 'Abrechnung_Positionen' AS table_name
        from Abrechnung_Positionen where Nettobetrag ='NULL'
        union
        select count(1) , 'Bildnummer' as col_nm, 'Abrechnung_Positionen' AS table_name
        from Abrechnung_Positionen where Bildnummer ='NULL'
        union
        select count(1) , 'VerDatum' as col_nm, 'Abrechnung_Positionen' AS table_name
        from Abrechnung_Positionen where VerDatum ='NULL'

        union
        select count(1) , 'ReNummer' as col_nm , 'Abrechnung_Rechnungen' AS table_name
        from Abrechnung_Rechnungen where ReNummer ='NULL'
        union
        select count(1) , 'SummeNetto' as col_nm , 'Abrechnung_Rechnungen' AS table_name
        from Abrechnung_Rechnungen where SummeNetto ='NULL'
        union
        select count(1) , 'MwStSatz' as col_nm , 'Abrechnung_Rechnungen' AS table_name
        from Abrechnung_Rechnungen where MwStSatz ='NULL'
        union
        select count(1) , 'ZahlungsbetragBrutto' as col_nm , 'Abrechnung_Rechnungen' AS table_name
        from Abrechnung_Rechnungen where ZahlungsbetragBrutto ='NULL'
        union
        select count(1) , 'KdNr' as col_nm , 'Abrechnung_Rechnungen' AS table_name
        from Abrechnung_Rechnungen where KdNr ='NULL'
        union
        select count(1) , 'Summenebenkosten' as col_nm, 'Abrechnung_Rechnungen' AS table_name
        from Abrechnung_Rechnungen where Summenebenkosten ='NULL'
        union
        select count(1) , 'ReDatum' as col_nm, 'Abrechnung_Rechnungen' AS table_name
        from Abrechnung_Rechnungen where ReDatum ='NULL'
        union
        select count(1) , 'Zahlungsdatum' as col_nm, 'Abrechnung_Rechnungen' AS table_name
        from Abrechnung_Rechnungen where Zahlungsdatum ='NULL'
        order by table_name, col_nm

        ;
       '''

dataframe_null_checks = pd.read_sql_query(query_null_checks, conn)
```

```
# Display the output
dataframe_null_checks
```

Out[5]:

	null_count	col_nm	table_name
0	0	Kdnr	Abrechnung_Kunden
1	321	Region	Abrechnung_Kunden
2	0	Verlagsname	Abrechnung_Kunden
3	0	id	Abrechnung_Kunden
4	1	Bildnummer	Abrechnung_Positionen
5	1	KdNr	Abrechnung_Positionen
6	1	Nettobetrag	Abrechnung_Positionen
7	0	Reid	Abrechnung_Positionen
8	4	VerDatum	Abrechnung_Positionen
9	0	id	Abrechnung_Positionen
10	0	KdNr	Abrechnung_Rechnungen
11	0	MwStSatz	Abrechnung_Rechnungen
12	0	ReDatum	Abrechnung_Rechnungen
13	0	ReNummer	Abrechnung_Rechnungen
14	0	SummeNetto	Abrechnung_Rechnungen
15	2	Summenebenkosten	Abrechnung_Rechnungen
16	1	ZahlungsbetragBrutto	Abrechnung_Rechnungen
17	399	Zahlungsdatum	Abrechnung_Rechnungen

Q1.How many positions are linked to invoices that are missing payment info

Using Join

In [6]:
```
# Run SQL query -  query_positions_missing_payment_info
query_positions_missing_payment_info = '''
       SELECT COUNT(id) AS positions_missing_payment_info
       FROM Abrechnung_Positionen p
       LEFT JOIN Abrechnung_Rechnungen r
       ON p.ReId = r.ReNummer
       WHERE r.Zahlungsdatum IS NULL OR r.Zahlungsdatum = 'NULL'
        ;
       '''
dataframe_positions_missing_payment_info = pd.read_sql_query(query_positions_missing_payment_info, conn)
dataframe_positions_missing_payment_info
```

Out[6]:

	positions_missing_payment_info
0	18011

Using Exists

In [7]:
```
# Run SQL query -  query_positions_missing_payment_info
query_positions_missing_payment_info = '''
       SELECT COUNT(p.id) AS positions_missing_payment_info
       FROM Abrechnung_Positionen p WHERE EXISTS
       ( SELECT 1 FROM Abrechnung_Rechnungen r
       WHERE p.ReId = r.ReNummer
       AND (r.Zahlungsdatum IS NULL OR r.Zahlungsdatum = 'NULL'));'''
dataframe_positions_missing_payment_info = pd.read_sql_query(query_positions_missing_payment_info, conn)
# Display the output
dataframe_positions_missing_payment_info
```

Out[7]:

	positions_missing_payment_info
0	18011

Q2.How much revenue is attributed to placeholder media ID '100000000'

In [8]:
```
# Run SQL query -  query_revenue_placeholder_media
query_revenue_placeholder_media = '''
       SELECT Bildnummer , SUM(Nettobetrag) AS placeholder_media_revenue
       FROM Abrechnung_Positionen
       WHERE Bildnummer = 100000000
       GROUP BY 1;'''

dataframe_revenue_placeholder_media = pd.read_sql_query(query_revenue_placeholder_media, conn)

# Display the output
dataframe_revenue_placeholder_media
```

Out[8]:

	Bildnummer	placeholder_media_revenue
0	100000000	1319897.91

Q3.How many invoices have no positions attached

Using Join

In [9]:
```
query_invoices_without_any_positions = '''
       SELECT COUNT(1) AS invoices_without_any_positions
        FROM Abrechnung_Rechnungen r
        LEFT JOIN Abrechnung_Positionen p ON r.ReNummer = p.ReId
        WHERE p.ReId IS NULL
        ;
       '''

dataframe_invoices_without_any_positions = pd.read_sql_query(query_invoices_without_any_positions, conn)
dataframe_invoices_without_any_positions
```

Out[9]:

	invoices_without_any_positions
0	2

Using Exists

In [10]:
```
# Run SQL query -  query_invoices_without_any_positions
query_invoices_without_any_positions = '''
       SELECT COUNT(1) AS invoices_without_any_positions
        FROM Abrechnung_Rechnungen r WHERE NOT EXISTS (
        SELECT 1 FROM Abrechnung_Positionen p
        WHERE r.ReNummer = p.ReId
        );'''
dataframe_invoices_without_any_positions = pd.read_sql_query(query_invoices_without_any_positions, conn)
dataframe_invoices_without_any_positions
```

Out[10]:

	invoices_without_any_positions
0	2