

```
In [1]: import sqlite3
import pandas as pd

connection = sqlite3.connect('Amazon.db')

cursor = connection.cursor()
```

```
In [2]: df = pd.read_csv(r"C:\Users\Owner\Documents\FDA files\Amazon Sales data.csv")

df.head()
```

Out[2]:

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	5/28/2010	669165933	6/27/2010	9925	255.28	159.42	2533654.00	1582243.50	951410.50
1	Central America and the Caribbean	Grenada	Cereal	Online	C	8/22/2012	963881480	9/15/2012	2804	205.70	117.11	576782.80	328376.44	248406.36
2	Europe	Russia	Office Supplies	Offline	L	05-02-2014	341417157	05-08-2014	1779	651.21	524.96	1158502.59	933903.84	224598.75
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	6/20/2014	514321792	07-05-2014	8102	9.33	6.92	75591.66	56065.84	19525.82
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	02-01-2013	115456712	02-06-2013	5062	651.21	524.96	3296425.02	2657347.52	639077.50

```
In [3]: df.to_sql('Amaz', connection, if_exists='append', index=False)

# Commit the changes
connection.commit()
```

```
In [4]: connection = sqlite3.connect('Amazon.db')
cursor = connection.cursor()

cursor.execute("SELECT * FROM Amaz")
rows = cursor.fetchall()

columns = [description[0] for description in cursor.description]

df = pd.DataFrame(rows, columns=columns)

display(df)

connection.commit()
```

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	5/28/2010	669165933	6/27/2010	9925	255.28	159.42	2533654.00	1582243.50	951410.50
1	Central America and the Caribbean	Grenada	Cereal	Online	C	8/22/2012	963881480	9/15/2012	2804	205.70	117.11	576782.80	328376.44	248406.36
2	Europe	Russia	Office Supplies	Offline	L	05-02-2014	341417157	05-08-2014	1779	651.21	524.96	1158502.59	933903.84	224598.75
3	Sub-Saharan Africa	Sao Tome and Principe	Fruits	Online	C	6/20/2014	514321792	07-05-2014	8102	9.33	6.92	75591.66	56065.84	19525.82
4	Sub-Saharan Africa	Rwanda	Office Supplies	Offline	L	02-01-2013	115456712	02-06-2013	5062	651.21	524.96	3296425.02	2657347.52	639077.50
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	Sub-Saharan Africa	Mali	Clothes	Online	M	7/26/2011	512878119	09-03-2011	888	109.28	35.84	97040.64	31825.92	65214.72
996	Asia	Malaysia	Fruits	Offline	L	11-11-2011	810711038	12/28/2011	6267	9.33	6.92	58471.11	43367.64	15103.47
997	Sub-Saharan Africa	Sierra Leone	Vegetables	Offline	C	06-01-2016	728815257	6/29/2016	1485	154.06	90.93	228779.10	135031.05	93748.05
998	North America	Mexico	Personal Care	Offline	M	7/30/2015	559427106	08-08-2015	5767	81.73	56.67	471336.91	326815.89	144521.02
999	Sub-Saharan Africa	Mozambique	Household	Offline	L	02-10-2012	665095412	2/15/2012	5367	668.27	502.54	3586605.09	2697132.18	889472.91

1000 rows × 14 columns

```
In [5]: connection = sqlite3.connect('Amazon.db')
cursor = connection.cursor()

# Execute PRAGMA table_info to get column info
cursor.execute("PRAGMA table_info(Amaz);")

# Fetch all column information
columns_info = cursor.fetchall()

# Print column names and their data types
for column in columns_info:
    print(f"Column: {column[1]}, Data Type: {column[2]}")

connection.close()
```

Column: Region, Data Type: TEXT  
Column: Country, Data Type: TEXT  
Column: Item Type, Data Type: TEXT  
Column: Sales Channel, Data Type: TEXT  
Column: Order Priority, Data Type: TEXT  
Column: Order Date, Data Type: TEXT  
Column: Order ID, Data Type: INTEGER  
Column: Ship Date, Data Type: TEXT  
Column: Units Sold, Data Type: INTEGER  
Column: Unit Price, Data Type: REAL  
Column: Unit Cost, Data Type: REAL  
Column: Total Revenue, Data Type: REAL  
Column: Total Cost, Data Type: REAL  
Column: Total Profit, Data Type: REAL

```
In [6]: connection = sqlite3.connect('Amazon.db')

cursor = connection.cursor()
cursor.execute("SELECT DISTINCT Region FROM Amaz")

rows = cursor.fetchall()
```

```
columns = [description[0] for description in cursor.description]
df = pd.DataFrame(rows, columns=columns)

display(df)

connection.commit()
connection.close()
```

Region	
0	Australia and Oceania
1	Central America and the Caribbean
2	Europe
3	Sub-Saharan Africa
4	Asia
5	Middle East and North Africa
6	North America

```
In [7]: connection = sqlite3.connect('Amazon.db')
cursor = connection.cursor()
cursor.execute("SELECT DISTINCT Country FROM Amaz")

rows = cursor.fetchall()

columns = [description[0] for description in cursor.description]
df = pd.DataFrame(rows, columns=columns)

display(df)

connection.commit()
connection.close()
```

Country	
0	Tuvalu
1	Grenada
2	Russia
3	Sao Tome and Principe
4	Rwanda
...	...
71	Slovenia
72	Romania
73	Nicaragua
74	Malaysia
75	Mozambique

76 rows × 1 columns

```
In [8]: connection = sqlite3.connect('Amazon.db')
cursor = connection.cursor()
cursor.execute('SELECT DISTINCT "Item Type" FROM Amaz')

rows = cursor.fetchall()

columns = [description[0] for description in cursor.description]
df = pd.DataFrame(rows, columns=columns)

display(df)

connection.commit()
connection.close()
```

Item Type	
0	Baby Food
1	Cereal
2	Office Supplies
3	Fruits
4	Household
5	Vegetables
6	Personal Care
7	Clothes
8	Cosmetics
9	Beverages
10	Meat
11	Snacks

```
In [9]: connection = sqlite3.connect('Amazon.db')
cursor = connection.cursor()

cursor.execute("""
SELECT "Country", SUM("Total Revenue") AS "Total Revenue"
FROM Amaz
GROUP BY "Country"
ORDER BY "Total Revenue" DESC
""")

rows = cursor.fetchall()

columns = [description[0] for description in cursor.description]
```

```
df = pd.DataFrame(rows, columns=columns)

display(df)

connection.commit()
connection.close()
```

	Country	Total Revenue
0	Honduras	63365454.8
1	Myanmar	61612579.0
2	Djibouti	60528908.6
3	Turkmenistan	58220362.0
4	Mexico	56433565.5
...	...	...
71	Syria	353047.2
72	Slovakia	263442.6
73	New Zealand	204047.1
74	Kyrgyzstan	191034.4
75	Kuwait	48702.6

76 rows × 2 columns

```
In [10]: connection = sqlite3.connect('Amazon.db')
cursor = connection.cursor()

cursor.execute("""
SELECT "Country", SUM("Total Profit") AS "Total Profit"
FROM Amaz
GROUP BY "Country"
ORDER BY "Total Profit" DESC
""")

rows = cursor.fetchall()

# Get column names (use cursor.description for column names)
columns = [description[0] for description in cursor.description]

df = pd.DataFrame(rows, columns=columns)

display(df)

connection.commit()
connection.close()
```

	Country	Total Profit
0	Djibouti	24253178.7
1	Myanmar	18027717.0
2	Pakistan	17199220.4
3	Samoa	16785409.8
4	Honduras	16099475.2
...	...	...
71	Slovakia	107952.3
72	Syria	91194.4
73	Kyrgyzstan	78281.2
74	New Zealand	52706.7
75	Kuwait	12580.2

76 rows × 2 columns

```
In [11]: connection = sqlite3.connect('Amazon.db')
cursor = connection.cursor()

cursor.execute("""
SELECT "Region", SUM("Total Revenue") AS "Total Revenue"
FROM Amaz
GROUP BY "Region"
ORDER BY "Total Revenue" DESC
""")

rows = cursor.fetchall()

columns = [description[0] for description in cursor.description]

df = pd.DataFrame(rows, columns=columns)

display(df)

pd.set_option('display.float_format', '{:,.0f}'.format)

connection.commit()
connection.close()
```

	Region	Total Revenue
0	Sub-Saharan Africa	396720314.3
1	Europe	333689321.1
2	Asia	213470910.2
3	Australia and Oceania	140942651.3
4	Middle East and North Africa	140527065.8
5	Central America and the Caribbean	91703854.9
6	North America	56433565.5

```
In [12]: connection = sqlite3.connect('Amazon.db')
cursor = connection.cursor()

cursor.execute("""
SELECT "Region", SUM("Total Profit") AS "Total Profit"
FROM Amaz
GROUP BY "Region"
ORDER BY "Total Profit" DESC
""")

rows = cursor.fetchall()

columns = [description[0] for description in cursor.description]

df = pd.DataFrame(rows, columns=columns)

display(df)

connection.commit()
connection.close()
```

	Region	Total Profit
0	Sub-Saharan Africa	121,832,114
1	Europe	110,829,386
2	Asia	61,138,459
3	Middle East and North Africa	57,611,919
4	Australia and Oceania	47,221,600
5	Central America and the Caribbean	28,469,078
6	North America	14,579,428

```
In [13]: connection = sqlite3.connect('Amazon.db')
cursor = connection.cursor()

cursor.execute("""
SELECT "Item Type", SUM("Total Revenue") AS "Total Revenue"
FROM Amaz
GROUP BY "Item Type"
ORDER BY "Total Revenue" DESC
""")

rows = cursor.fetchall()

columns = [description[0] for description in cursor.description]

df = pd.DataFrame(rows, columns=columns)

display(df)

pd.set_option('display.float_format', '{:,.0f}'.format)

connection.commit()
connection.close()
```

	Item Type	Total Revenue
0	Cosmetics	366,015,096
1	Office Supplies	305,853,801
2	Household	298,897,123
3	Baby Food	103,503,276
4	Clothes	77,872,928
5	Cereal	53,228,989
6	Meat	45,036,758
7	Personal Care	39,809,048
8	Vegetables	30,890,571
9	Beverages	26,907,946
10	Snacks	20,807,335
11	Fruits	4,664,813

```
In [14]: connection = sqlite3.connect('Amazon.db')
cursor = connection.cursor()

cursor.execute("""
SELECT "Item Type", SUM("Total Profit") AS "Total Profit"
FROM Amaz
GROUP BY "Item Type"
ORDER BY "Total Profit" DESC
""")

rows = cursor.fetchall()

columns = [description[0] for description in cursor.description]

df = pd.DataFrame(rows, columns=columns)

display(df)

pd.set_option('display.float_format', '{:,.0f}'.format)

connection.commit()
connection.close()
```

	Item Type	Total Profit
0	Cosmetics	145,560,487
1	Household	74,126,057
2	Office Supplies	59,295,838
3	Clothes	52,333,344
4	Baby Food	38,866,437
5	Cereal	22,924,434
6	Vegetables	12,658,196
7	Personal Care	12,206,225
8	Beverages	8,880,473
9	Snacks	7,519,442
10	Meat	6,106,100
11	Fruits	1,204,952

```
In [15]: connection = sqlite3.connect('Amazon.db')
        cursor = connection.cursor()

        cursor.execute("""
        SELECT "Item Type", SUM("Units Sold") AS "Units Sold"
        FROM Amaz
        GROUP BY "Item Type"
        ORDER BY "Units Sold" DESC
        """)

        rows = cursor.fetchall()

        columns = [description[0] for description in cursor.description]

        df = pd.DataFrame(rows, columns=columns)

        display(df)

        connection.commit()
        connection.close()
```

	Item Type	Units Sold
0	Cosmetics	837180
1	Clothes	712600
2	Beverages	567080
3	Fruits	499980
4	Personal Care	487080
5	Office Supplies	469670
6	Household	447270
7	Baby Food	405450
8	Cereal	258770
9	Vegetables	200510
10	Snacks	136370
11	Meat	106750

```
In [16]: connection = sqlite3.connect('Amazon.db')
        cursor = connection.cursor()

        cursor.execute("""
        SELECT "Item Type", SUM("Total Profit") AS "Total Profit", SUM("Total Revenue") AS "Total Revenue"
        FROM Amaz
        GROUP BY "Item Type"
        ORDER BY "Total Profit" DESC
        """)

        rows = cursor.fetchall()
        columns = [description[0] for description in cursor.description]

        df = pd.DataFrame(rows, columns=columns)

        df["Percentage Profit"] = (df["Total Profit"] / df["Total Revenue"]) * 100

        df["Percentage Profit"] = df["Percentage Profit"].apply(lambda x: "{:.2f}%".format(x))

        display(df)

        connection.commit()
        connection.close()
```

	Item Type	Total Profit	Total Revenue	Percentage Profit
0	Cosmetics	145,560,487	366,015,096	39.77%
1	Household	74,126,057	298,897,123	24.80%
2	Office Supplies	59,295,838	305,853,801	19.39%
3	Clothes	52,333,344	77,872,928	67.20%
4	Baby Food	38,866,437	103,503,276	37.55%
5	Cereal	22,924,434	53,228,989	43.07%
6	Vegetables	12,658,196	30,890,571	40.98%
7	Personal Care	12,206,225	39,809,048	30.66%
8	Beverages	8,880,473	26,907,946	33.00%
9	Snacks	7,519,442	20,807,335	36.14%
10	Meat	6,106,100	45,036,758	13.56%
11	Fruits	1,204,952	4,664,813	25.83%

```
In [17]: connection = sqlite3.connect('Amazon.db')
cursor = connection.cursor()

cursor.execute("""
SELECT "Country", "Item Type", SUM("Units Sold") AS "Units Sold"
FROM Amaz
GROUP BY "Country", "Item Type"
ORDER BY "Country", "Item Type", "Units Sold" DESC
""")

rows = cursor.fetchall()

columns = [description[0] for description in cursor.description]

df = pd.DataFrame(rows, columns=columns)

display(df)

connection.commit()
connection.close()
```

	Country	Item Type	Units Sold
0	Albania	Clothes	22690
1	Angola	Household	41870
2	Australia	Beverages	93890
3	Australia	Cereal	6820
4	Australia	Office Supplies	29240
...	...	...	...
92	Turkmenistan	Household	38300
93	Turkmenistan	Office Supplies	50100
94	Tuvalu	Baby Food	99250
95	United Kingdom	Household	2820
96	Zambia	Snacks	40850

97 rows × 3 columns

```
In [18]: connection = sqlite3.connect('Amazon.db')
cursor = connection.cursor()

cursor.execute("""
SELECT "Country", "Item Type", SUM("Total Profit") AS "Total Profit", SUM("Total Revenue") AS "Total Revenue"
FROM Amaz
GROUP BY "Country", "Item Type"
ORDER BY "Country", "Total Profit" DESC
""")

rows = cursor.fetchall()
columns = [description[0] for description in cursor.description]

df = pd.DataFrame(rows, columns=columns)

df["Percentage Profit"] = (df["Total Profit"] / df["Total Revenue"]) * 100

df["Percentage Profit"] = df["Percentage Profit"].apply(lambda x: "{:.2f}%".format(x))

display(df)

connection.commit()
connection.close()
```

	Country	Item Type	Total Profit	Total Revenue	Percentage Profit
0	Albania	Clothes	1,666,354	2,479,563	67.20%
1	Angola	Household	6,939,115	27,980,465	24.80%
2	Australia	Office Supplies	3,691,550	19,041,380	19.39%
3	Australia	Beverages	1,470,317	4,455,080	33.00%
4	Australia	Cereal	604,184	1,402,874	43.07%
...	...	...	...	...	...
92	Turkmenistan	Household	6,347,459	25,594,741	24.80%
93	Turkmenistan	Office Supplies	6,325,125	32,625,621	19.39%
94	Tuvalu	Baby Food	9,514,105	25,336,540	37.55%
95	United Kingdom	Household	467,359	1,884,521	24.80%
96	Zambia	Snacks	2,252,469	6,232,893	36.14%

97 rows × 5 columns

```
In [19]: connection = sqlite3.connect('Amazon.db')
        cursor = connection.cursor()

        cursor.execute("""
        SELECT "Region", "Item Type", SUM("Units Sold") AS "Units Sold"
        FROM Amaz
        GROUP BY "Region", "Item Type"
        ORDER BY "Region", "Item Type", "Units Sold" DESC
        """)

        rows = cursor.fetchall()

        columns = [description[0] for description in cursor.description]

        df = pd.DataFrame(rows, columns=columns)

        display(df)

        connection.commit()
        connection.close()
```

	Region	Item Type	Units Sold
0	Asia	Clothes	141930
1	Asia	Cosmetics	69520
2	Asia	Fruits	62670
3	Asia	Household	120800
4	Asia	Office Supplies	117180
5	Asia	Personal Care	49010
6	Asia	Vegetables	38560
7	Australia and Oceania	Baby Food	128990
8	Australia and Oceania	Beverages	187680
9	Australia and Oceania	Cereal	6820
10	Australia and Oceania	Clothes	99050
11	Australia and Oceania	Cosmetics	96540
12	Australia and Oceania	Fruits	75850
13	Australia and Oceania	Meat	59080
14	Australia and Oceania	Office Supplies	29240
15	Central America and the Caribbean	Beverages	81560
16	Central America and the Caribbean	Cereal	28040
17	Central America and the Caribbean	Clothes	54980
18	Central America and the Caribbean	Cosmetics	17050
19	Central America and the Caribbean	Household	89740
20	Central America and the Caribbean	Personal Care	64090
21	Central America and the Caribbean	Snacks	22250
22	Europe	Baby Food	220870
23	Europe	Beverages	97840
24	Europe	Clothes	117840
25	Europe	Cosmetics	301000
26	Europe	Household	47950
27	Europe	Office Supplies	140530
28	Europe	Personal Care	53430
29	Europe	Vegetables	1710
30	Middle East and North Africa	Cereal	40630
31	Middle East and North Africa	Clothes	140000
32	Middle East and North Africa	Cosmetics	236150
33	Middle East and North Africa	Fruits	49790
34	Middle East and North Africa	Office Supplies	20210
35	North America	Household	69540
36	North America	Personal Care	121890
37	Sub-Saharan Africa	Baby Food	55590
38	Sub-Saharan Africa	Beverages	200000
39	Sub-Saharan Africa	Cereal	183280
40	Sub-Saharan Africa	Clothes	158800
41	Sub-Saharan Africa	Cosmetics	116920
42	Sub-Saharan Africa	Fruits	311670
43	Sub-Saharan Africa	Household	119240
44	Sub-Saharan Africa	Meat	47670
45	Sub-Saharan Africa	Office Supplies	162510
46	Sub-Saharan Africa	Personal Care	198660
47	Sub-Saharan Africa	Snacks	114120
48	Sub-Saharan Africa	Vegetables	160240

```
In [20]: connection = sqlite3.connect('Amazon.db')
        cursor = connection.cursor()

        cursor.execute("""
        SELECT "Region", "Item Type", SUM("Total Profit") AS "Total Profit", SUM("Total Revenue") AS "Total Revenue"
        FROM Amaz
        GROUP BY "Region", "Item Type"
        ORDER BY "Region", "Total Profit" DESC
        """)

        rows = cursor.fetchall()
        columns = [description[0] for description in cursor.description]

        df = pd.DataFrame(rows, columns=columns)

        df["Percentage Profit"] = (df["Total Profit"] / df["Total Revenue"]) * 100

        df["Percentage Profit"] = df["Percentage Profit"].apply(lambda x: "{:.2f}%".format(x))

        display(df)

        connection.commit()
        connection.close()
```



	Region	Item Type	Total Profit	Total Revenue	Percentage Profit
0	Asia	Household	20,020,184	80,727,016	24.80%
1	Asia	Office Supplies	14,793,975	76,308,788	19.39%
2	Asia	Cosmetics	12,087,442	30,394,144	39.77%
3	Asia	Clothes	10,423,339	15,510,110	67.20%
4	Asia	Vegetables	2,434,293	5,940,554	40.98%
5	Asia	Personal Care	1,228,191	4,005,587	30.66%
6	Asia	Fruits	151,035	584,711	25.83%
7	Australia and Oceania	Cosmetics	16,785,410	42,207,288	39.77%
8	Australia and Oceania	Baby Food	12,364,981	32,928,567	37.55%
9	Australia and Oceania	Clothes	7,274,232	10,824,184	67.20%
10	Australia and Oceania	Office Supplies	3,691,550	19,041,380	19.39%
11	Australia and Oceania	Meat	3,379,376	24,925,261	13.56%
12	Australia and Oceania	Beverages	2,939,069	8,905,416	33.00%
13	Australia and Oceania	Cereal	604,184	1,402,874	43.07%
14	Australia and Oceania	Fruits	182,798	707,680	25.83%
15	Central America and the Caribbean	Household	14,872,610	59,970,550	24.80%
16	Central America and the Caribbean	Clothes	4,037,731	6,008,214	67.20%
17	Central America and the Caribbean	Cosmetics	2,964,484	7,454,260	39.77%
18	Central America and the Caribbean	Cereal	2,484,064	5,767,828	43.07%
19	Central America and the Caribbean	Personal Care	1,606,095	5,238,076	30.66%
20	Central America and the Caribbean	Beverages	1,277,230	3,870,022	33.00%
21	Central America and the Caribbean	Snacks	1,226,865	3,394,905	36.14%
22	Europe	Cosmetics	52,334,870	131,597,200	39.77%
23	Europe	Baby Food	21,172,598	56,383,694	37.55%
24	Europe	Office Supplies	17,741,912	91,514,541	19.39%
25	Europe	Clothes	8,654,170	12,877,555	67.20%
26	Europe	Household	7,946,754	32,043,546	24.80%
27	Europe	Beverages	1,532,174	4,642,508	33.00%
28	Europe	Personal Care	1,338,956	4,366,834	30.66%
29	Europe	Vegetables	107,952	263,443	40.98%
30	Middle East and North Africa	Cosmetics	41,059,400	103,244,780	39.77%
31	Middle East and North Africa	Clothes	10,281,600	15,299,200	67.20%
32	Middle East and North Africa	Cereal	3,599,412	8,357,591	43.07%
33	Middle East and North Africa	Office Supplies	2,551,512	13,160,954	19.39%
34	Middle East and North Africa	Fruits	119,994	464,541	25.83%
35	North America	Household	11,524,864	46,471,496	24.80%
36	North America	Personal Care	3,054,563	9,962,070	30.66%
37	Sub-Saharan Africa	Office Supplies	20,516,888	105,828,137	19.39%
38	Sub-Saharan Africa	Cosmetics	20,328,880	51,117,424	39.77%
39	Sub-Saharan Africa	Household	19,761,645	79,684,515	24.80%
40	Sub-Saharan Africa	Cereal	16,236,775	37,700,696	43.07%
41	Sub-Saharan Africa	Clothes	11,662,272	17,353,664	67.20%
42	Sub-Saharan Africa	Vegetables	10,115,951	24,686,574	40.98%
43	Sub-Saharan Africa	Snacks	6,292,577	17,412,430	36.14%
44	Sub-Saharan Africa	Baby Food	5,328,857	14,191,015	37.55%
45	Sub-Saharan Africa	Personal Care	4,978,420	16,236,482	30.66%
46	Sub-Saharan Africa	Beverages	3,132,000	9,490,000	33.00%
47	Sub-Saharan Africa	Meat	2,726,724	20,111,496	13.56%
48	Sub-Saharan Africa	Fruits	751,125	2,907,881	25.83%

```
In [21]: connection = sqlite3.connect('Amazon.db')
        cursor = connection.cursor()

        cursor.execute("""
        SELECT "Sales Channel", COUNT(*) AS "Total Count"
        FROM Amaz
        GROUP BY "Sales Channel"
        ORDER BY "Total Count" DESC

        """)
        results = cursor.fetchall()

        df = pd.DataFrame(results, columns=["Sales Channel", "Total Count"])

        display(df)

        connection.commit()
        connection.close()
```

	Sales Channel	Total Count
0	Online	500
1	Offline	500

```
In [22]: connection = sqlite3.connect('Amazon.db')
        cursor = connection.cursor()

        cursor.execute("""
        SELECT "Order Priority", COUNT(*) AS "Total Count"
        FROM Amaz
        GROUP BY "Order Priority"
```

```
ORDER BY "Total Count" DESC

"""
results = cursor.fetchall()

df = pd.DataFrame(results, columns=["Order Priority", "Total Count"])

display(df)

connection.commit()
connection.close()
```

	Order Priority	Total Count
0	H	300
1	L	270
2	C	220
3	M	210

```
In [23]: connection = sqlite3.connect('Amazon.db')
cursor = connection.cursor()

cursor.execute("""
SELECT "Country", "Sales Channel", COUNT(*) AS "Total Count"
FROM Amaz
GROUP BY "Country", "Sales Channel"
ORDER BY "Country", "Total Count" DESC

""")
results = cursor.fetchall()

df = pd.DataFrame(results, columns=["Country", "Sales Channel", "Total Count"])

display(df)

connection.commit()
connection.close()
```

	Country	Sales Channel	Total Count
0	Albania	Online	10
1	Angola	Offline	10
2	Australia	Offline	20
3	Australia	Online	10
4	Austria	Offline	10
...	...	...	...
82	Turkmenistan	Online	10
83	Turkmenistan	Offline	10
84	Tuvalu	Offline	10
85	United Kingdom	Online	10
86	Zambia	Online	10

87 rows × 3 columns

```
In [24]: connection = sqlite3.connect('Amazon.db')
cursor = connection.cursor()

cursor.execute("""
SELECT "Region", "Sales Channel", COUNT(*) AS "Total Count"
FROM Amaz
GROUP BY "Region", "Sales Channel"
ORDER BY "Region", "Total Count" DESC

""")
results = cursor.fetchall()

df = pd.DataFrame(results, columns=["Region", "Sales Channel", "Total Count"])

display(df)

connection.commit()
connection.close()
```

	Region	Sales Channel	Total Count
0	Asia	Offline	60
1	Asia	Online	50
2	Australia and Oceania	Online	70
3	Australia and Oceania	Offline	40
4	Central America and the Caribbean	Offline	50
5	Central America and the Caribbean	Online	20
6	Europe	Online	130
7	Europe	Offline	90
8	Middle East and North Africa	Online	80
9	Middle East and North Africa	Offline	20
10	North America	Offline	30
11	Sub-Saharan Africa	Offline	210
12	Sub-Saharan Africa	Online	150

```
In [25]: connection = sqlite3.connect('Amazon.db')
cursor = connection.cursor()

cursor.execute("""
SELECT "Item Type", "Sales Channel", COUNT(*) AS "Total Count"
FROM Amaz
GROUP BY "Item Type", "Sales Channel"
ORDER BY "Item Type", "Total Count" DESC

""")
results = cursor.fetchall()
```

```
df = pd.DataFrame(results, columns=["Item Type", "Sales Channel", "Total Count"])

display(df)

connection.commit()
connection.close()
```

	Item Type	Sales Channel	Total Count
0	Baby Food	Online	40
1	Baby Food	Offline	30
2	Beverages	Offline	60
3	Beverages	Online	20
4	Cereal	Online	40
5	Cereal	Offline	30
6	Clothes	Online	70
7	Clothes	Offline	60
8	Cosmetics	Offline	70
9	Cosmetics	Online	60
10	Fruits	Online	80
11	Fruits	Offline	20
12	Household	Offline	80
13	Household	Online	10
14	Meat	Online	20
15	Office Supplies	Online	60
16	Office Supplies	Offline	60
17	Personal Care	Offline	70
18	Personal Care	Online	30
19	Snacks	Online	30
20	Vegetables	Online	40
21	Vegetables	Offline	20

```
In [26]: connection = sqlite3.connect('Amazon.db')
cursor = connection.cursor()

cursor.execute("""
SELECT "Item Type", "Order Priority", COUNT(*) AS "Total Count"
FROM Amaz
GROUP BY "Item Type", "Order Priority"
ORDER BY "Item Type", "Total Count" DESC

""")
results = cursor.fetchall()

df = pd.DataFrame(results, columns=["Item Type", "Order Priority", "Total Count"])

display(df)

connection.commit()
connection.close()
```

	Item Type	Order Priority	Total Count
0	Baby Food	H	30
1	Baby Food	L	20
2	Baby Food	M	10
3	Baby Food	C	10
4	Beverages	C	70
5	Beverages	H	10
6	Cereal	H	50
7	Cereal	M	10
8	Cereal	C	10
9	Clothes	C	40
10	Clothes	M	30
11	Clothes	L	30
12	Clothes	H	30
13	Cosmetics	H	80
14	Cosmetics	M	30
15	Cosmetics	L	10
16	Cosmetics	C	10
17	Fruits	L	50
18	Fruits	M	20
19	Fruits	H	20
20	Fruits	C	10
21	Household	L	50
22	Household	H	20
23	Household	M	10
24	Household	C	10
25	Meat	M	10
26	Meat	L	10
27	Office Supplies	M	50
28	Office Supplies	L	30
29	Office Supplies	H	20
30	Office Supplies	C	20
31	Personal Care	L	40
32	Personal Care	M	30
33	Personal Care	C	20
34	Personal Care	H	10
35	Snacks	L	20
36	Snacks	M	10
37	Vegetables	H	30
38	Vegetables	C	20
39	Vegetables	L	10

```
In [27]: connection = sqlite3.connect('Amazon.db')
        cursor = connection.cursor()

        cursor.execute("""
        SELECT "Region", "Order Priority", COUNT(*) AS "Total Count"
        FROM Amaz
        GROUP BY "Region", "Order Priority"
        ORDER BY "Region", "Total Count" DESC

        """)
        results = cursor.fetchall()

        df = pd.DataFrame(results, columns=["Region", "Order Priority", "Total Count"])

        display(df)

        connection.commit()
        connection.close()
```

	Region	Order Priority	Total Count
0	Asia	L	40
1	Asia	H	30
2	Asia	M	20
3	Asia	C	20
4	Australia and Oceania	H	50
5	Australia and Oceania	C	40
6	Australia and Oceania	M	10
7	Australia and Oceania	L	10
8	Central America and the Caribbean	L	20
9	Central America and the Caribbean	H	20
10	Central America and the Caribbean	C	20
11	Central America and the Caribbean	M	10
12	Europe	H	70
13	Europe	L	60
14	Europe	C	50
15	Europe	M	40
16	Middle East and North Africa	M	40
17	Middle East and North Africa	L	40
18	Middle East and North Africa	H	20
19	North America	M	10
20	North America	L	10
21	North America	C	10
22	Sub-Saharan Africa	H	110
23	Sub-Saharan Africa	L	90
24	Sub-Saharan Africa	M	80
25	Sub-Saharan Africa	C	80

```
In [28]: connection = sqlite3.connect('Amazon.db')
cursor = connection.cursor()

cursor.execute("""
SELECT "Country", "Order Priority", COUNT(*) AS "Total Count"
FROM Amaz
GROUP BY "Country", "Order Priority"
ORDER BY "Country", "Total Count" DESC
""")
results = cursor.fetchall()

df = pd.DataFrame(results, columns=["Country", "Order Priority", "Total Count"])

display(df)

connection.commit()
connection.close()
```

	Country	Order Priority	Total Count
0	Albania	C	10
1	Angola	M	10
2	Australia	H	20
3	Australia	C	10
4	Austria	H	10
...	...	...	...
87	Turkmenistan	M	10
88	Turkmenistan	L	10
89	Tuvalu	H	10
90	United Kingdom	L	10
91	Zambia	L	10

92 rows × 3 columns

```
In [29]: connection = sqlite3.connect('Amazon.db')
cursor = connection.cursor()

cursor.execute("""
SELECT "Item Type",
       "Country",
       "Ship Date" - "Order Date" AS "Delivery Time (Days)"
FROM Amaz
ORDER BY "Item Type", "Country"
""")

rows = cursor.fetchall()

columns = [description[0] for description in cursor.description]

df = pd.DataFrame(rows, columns=columns)

display(df)

connection.commit()
connection.close()
```

	Item Type	Country	Delivery Time (Days)
0	Baby Food	Monaco	1
1	Baby Food	Monaco	1
2	Baby Food	Monaco	1
3	Baby Food	Monaco	1
4	Baby Food	Monaco	1
...	...	...	...
995	Vegetables	Slovakia	1
996	Vegetables	Slovakia	1
997	Vegetables	Slovakia	1
998	Vegetables	Slovakia	1
999	Vegetables	Slovakia	1

1000 rows × 3 columns

In [ ]:

In [ ]: