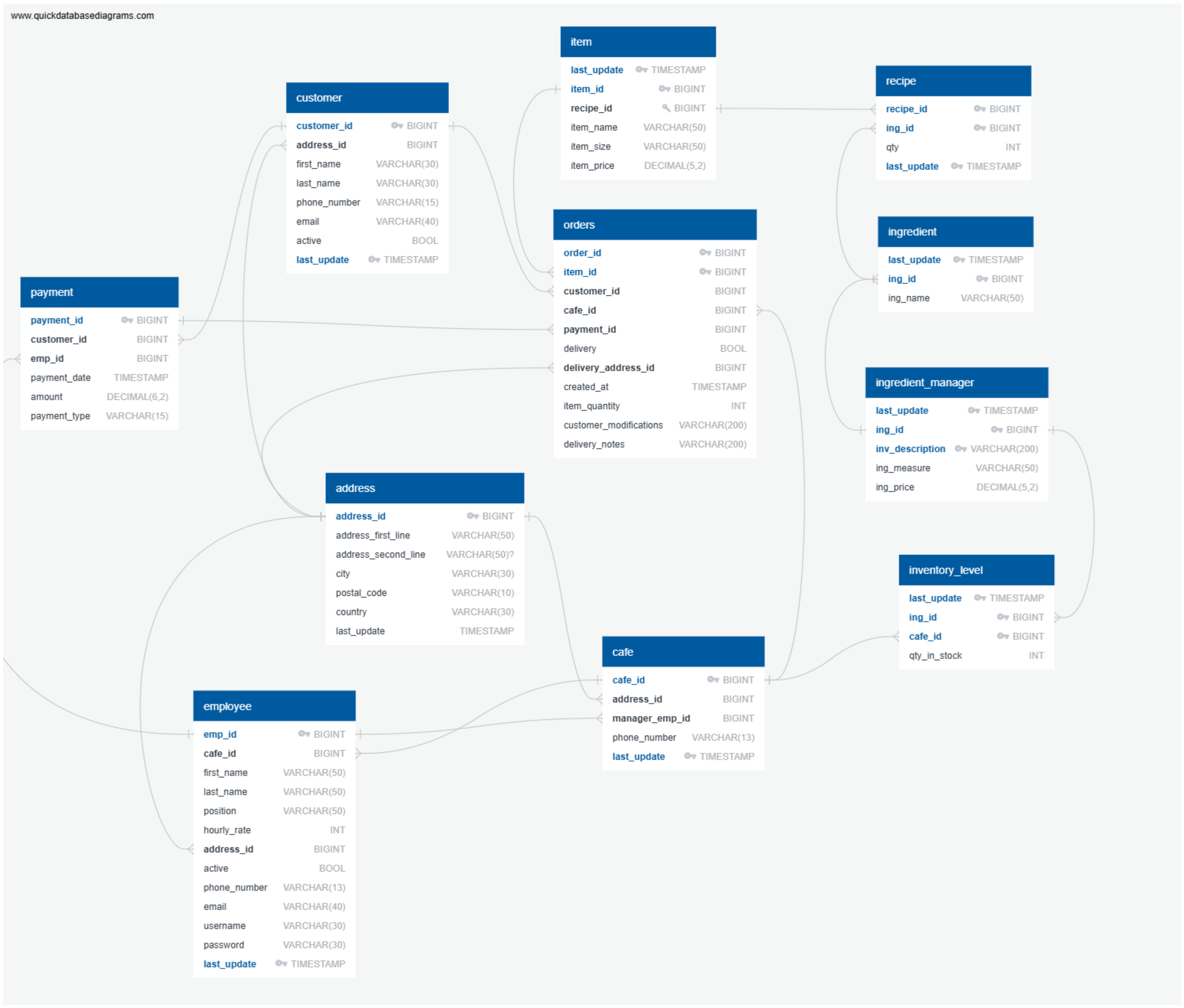
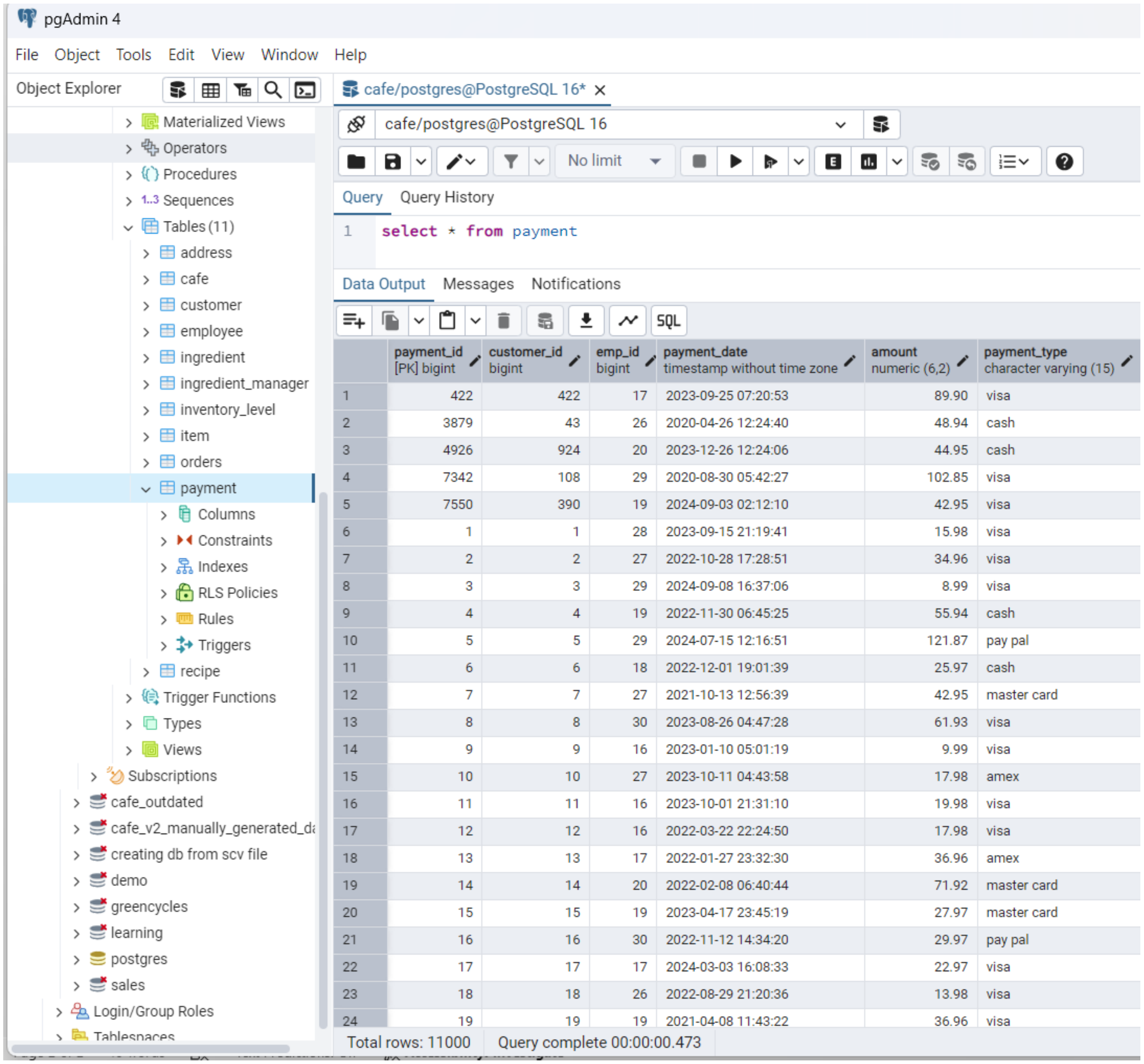
Part 1: Building SQL data base

This SQL database consists of 11 tables and contains information about transactions, orders, delivery addresses, customers, employees, etc

Data base diagram:



Example showing content of the data base. Snippet of “payment” table



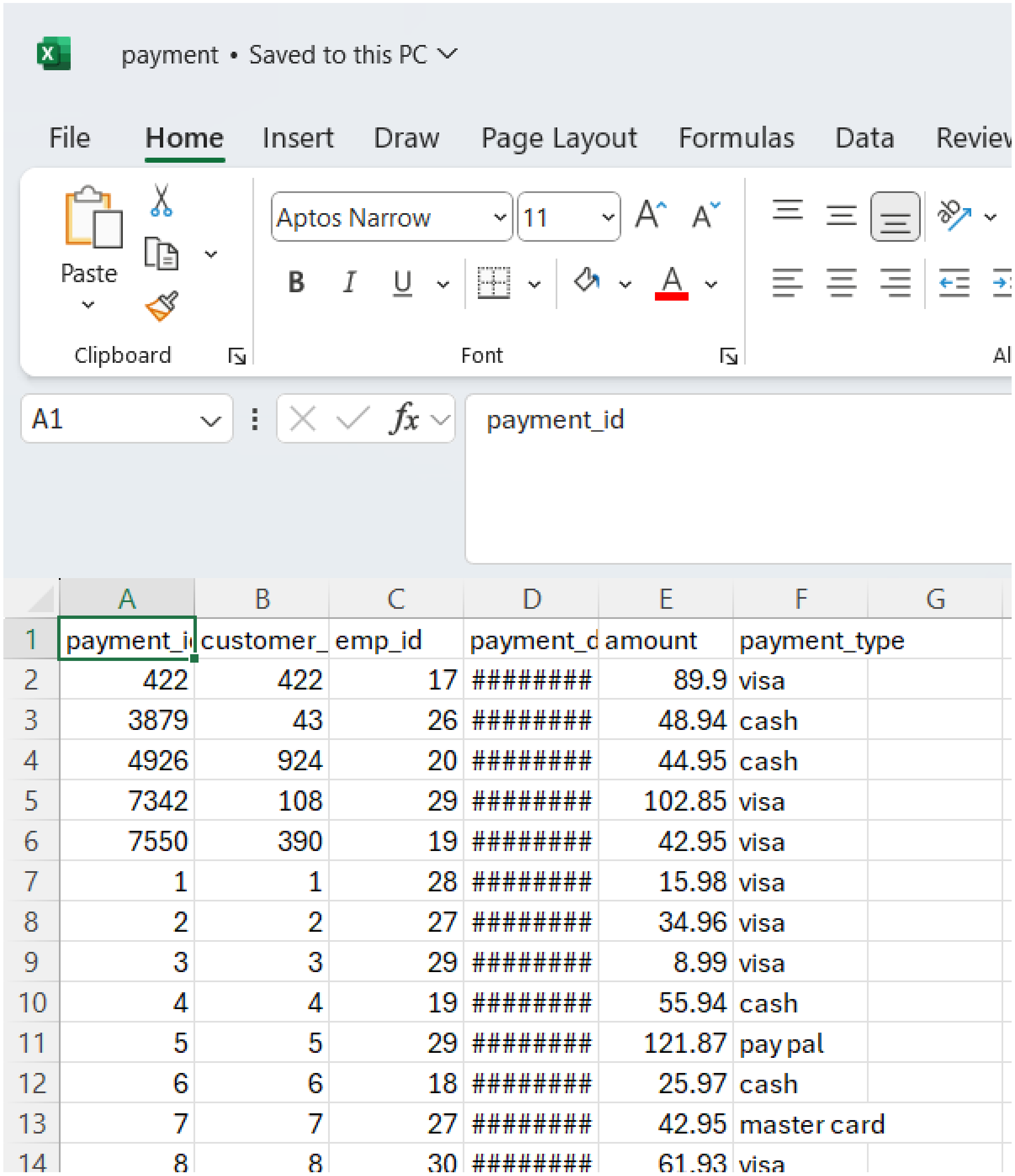
```
select * from payment
```

| | payment_id [PK] bigint | customer_id bigint | emp_id bigint | payment_date timestamp without time zone | amount numeric (6,2) | payment_type character varying (15) |
|---|---|---|---|---|---|---|
| 1 | 422 | 422 | 17 | 2023-09-25 07:20:53 | 89.90 | visa |
| 2 | 3879 | 43 | 26 | 2020-04-26 12:24:40 | 48.94 | cash |
| 3 | 4926 | 924 | 20 | 2023-12-26 12:24:06 | 44.95 | cash |
| 4 | 7342 | 108 | 29 | 2020-08-30 05:42:27 | 102.85 | visa |
| 5 | 7550 | 390 | 19 | 2024-09-03 02:12:10 | 42.95 | visa |
| 6 | 1 | 1 | 28 | 2023-09-15 21:19:41 | 15.98 | visa |
| 7 | 2 | 2 | 27 | 2022-10-28 17:28:51 | 34.96 | visa |
| 8 | 3 | 3 | 29 | 2024-09-08 16:37:06 | 8.99 | visa |
| 9 | 4 | 4 | 19 | 2022-11-30 06:45:25 | 55.94 | cash |
| 10 | 5 | 5 | 29 | 2024-07-15 12:16:51 | 121.87 | pay pal |
| 11 | 6 | 6 | 18 | 2022-12-01 19:01:39 | 25.97 | cash |
| 12 | 7 | 7 | 27 | 2021-10-13 12:56:39 | 42.95 | master card |
| 13 | 8 | 8 | 30 | 2023-08-26 04:47:28 | 61.93 | visa |
| 14 | 9 | 9 | 16 | 2023-01-10 05:01:19 | 9.99 | visa |
| 15 | 10 | 10 | 27 | 2023-10-11 04:43:58 | 17.98 | amex |
| 16 | 11 | 11 | 16 | 2023-10-01 21:31:10 | 19.98 | visa |
| 17 | 12 | 12 | 16 | 2022-03-22 22:24:50 | 17.98 | visa |
| 18 | 13 | 13 | 17 | 2022-01-27 23:32:30 | 36.96 | amex |
| 19 | 14 | 14 | 20 | 2022-02-08 06:40:44 | 71.92 | master card |
| 20 | 15 | 15 | 19 | 2023-04-17 23:45:19 | 27.97 | master card |
| 21 | 16 | 16 | 30 | 2022-11-12 14:34:20 | 29.97 | pay pal |
| 22 | 17 | 17 | 17 | 2024-03-03 16:08:33 | 22.97 | visa |
| 23 | 18 | 18 | 26 | 2022-08-29 21:20:36 | 13.98 | visa |
| 24 | 19 | 19 | 19 | 2021-04-08 11:43:22 | 36.96 | visa |

Total rows: 11000 Query complete 00:00:00.473

Extracting sales data from database in .csv format

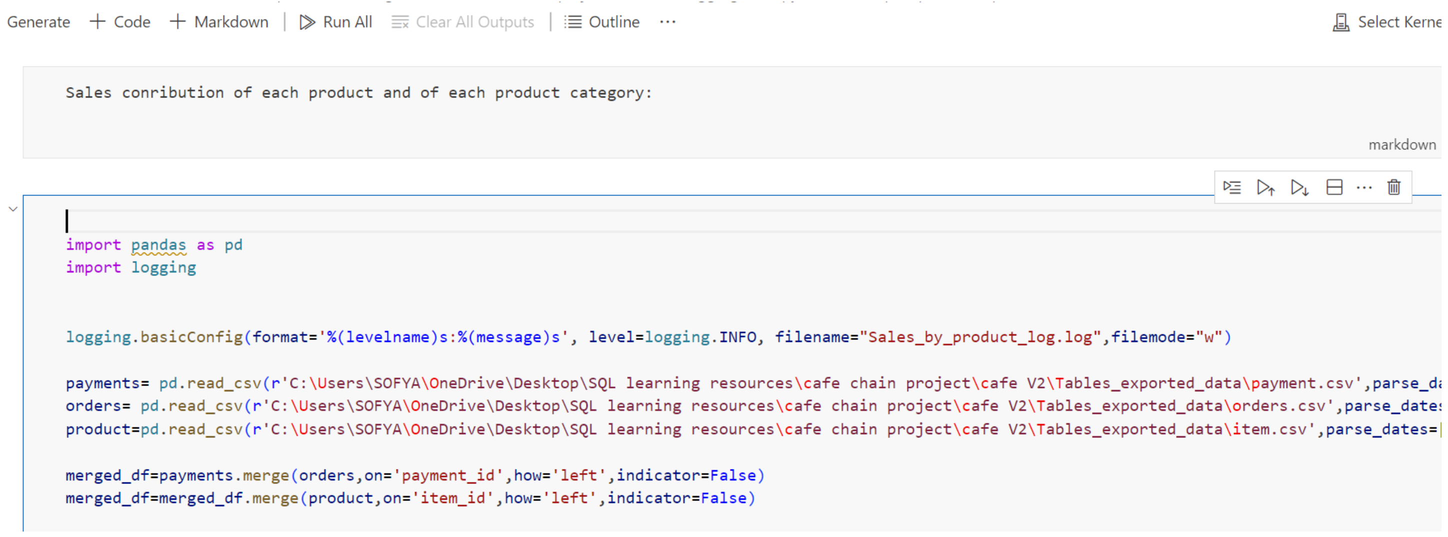
Screenshot of one of the extracted tables:



| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | payment_i | customer_ | emp_id | payment_d | amount | payment_type | |
| 2 | 422 | 422 | 17 | ######## | 89.9 | visa | |
| 3 | 3879 | 43 | 26 | ######## | 48.94 | cash | |
| 4 | 4926 | 924 | 20 | ######## | 44.95 | cash | |
| 5 | 7342 | 108 | 29 | ######## | 102.85 | visa | |
| 6 | 7550 | 390 | 19 | ######## | 42.95 | visa | |
| 7 | 1 | 1 | 28 | ######## | 15.98 | visa | |
| 8 | 2 | 2 | 27 | ######## | 34.96 | visa | |
| 9 | 3 | 3 | 29 | ######## | 8.99 | visa | |
| 10 | 4 | 4 | 19 | ######## | 55.94 | cash | |
| 11 | 5 | 5 | 29 | ######## | 121.87 | pay pal | |
| 12 | 6 | 6 | 18 | ######## | 25.97 | cash | |
| 13 | 7 | 7 | 27 | ######## | 42.95 | master card | |
| 14 | 8 | 8 | 30 | ######## | 61.93 | visa | |

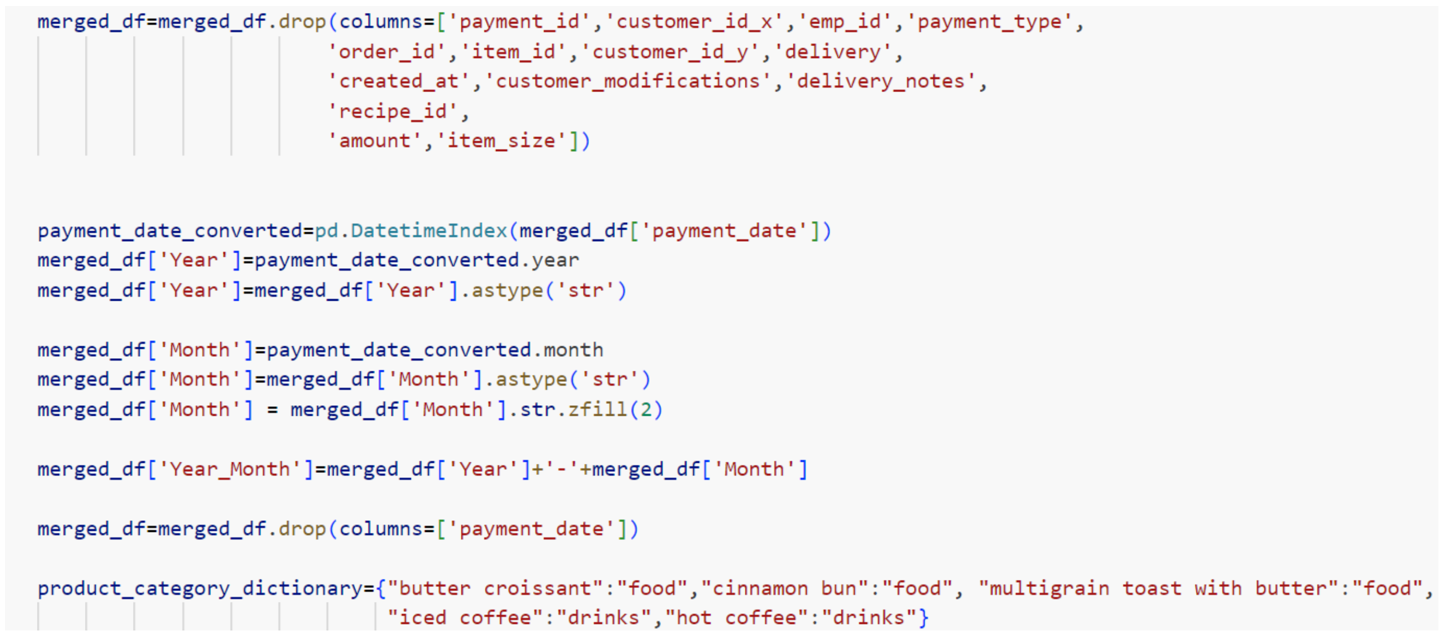
Part 2: Transforming and aggregating sales data using Python Pandas library

Snippet of the code that aggregates data:



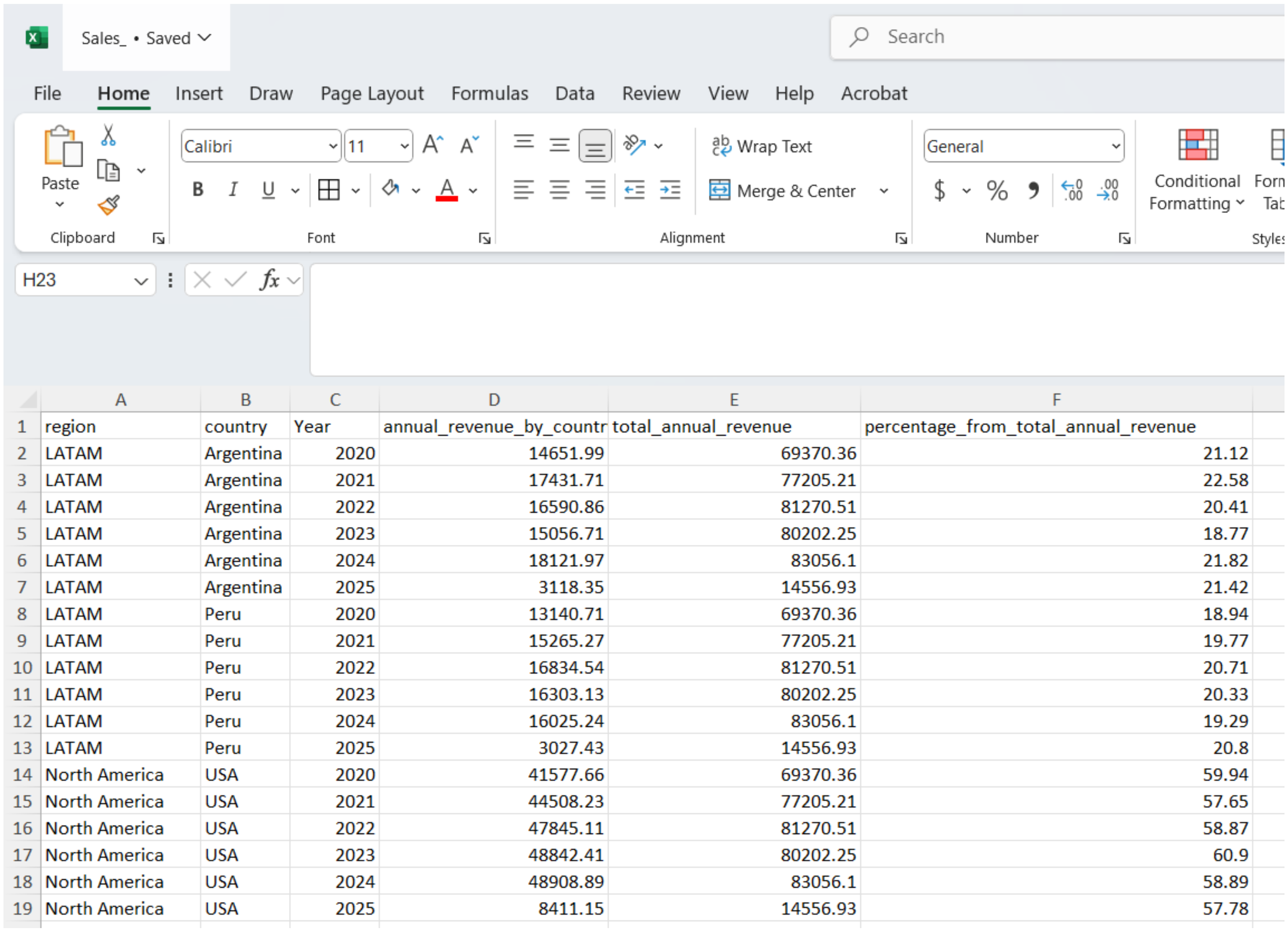
```
Sales conribution of each product and of each product category:
```

```
import pandas as pd
import logging


logging.basicConfig(format='%(levelname)s:%(message)s', level=logging.INFO, filename="Sales_by_product_log.log",filemode="w")

payments= pd.read_csv(r'C:\Users\SOFYA\OneDrive\Desktop\SQL learning resources\cafe chain project\cafe V2\Tables_exported_data\payment.csv',parse_da
orders= pd.read_csv(r'C:\Users\SOFYA\OneDrive\Desktop\SQL learning resources\cafe chain project\cafe V2\Tables_exported_data\orders.csv',parse_dates
product=pd.read_csv(r'C:\Users\SOFYA\OneDrive\Desktop\SQL learning resources\cafe chain project\cafe V2\Tables_exported_data\item.csv',parse_dates=

merged_df=payments.merge(orders,on='payment_id',how='left',indicator=False)
merged_df=merged_df.merge(product,on='item_id',how='left',indicator=False)
```



```
merged_df=merged_df.drop(columns=['payment_id','customer_id_x','emp_id','payment_type',
                                  'order_id','item_id','customer_id_y','delivery',
                                  'created_at','customer_modifications','delivery_notes',
                                  'recipe_id',
                                  'amount','item_size'])


payment_date_converted=pd.DatetimeIndex(merged_df['payment_date'])
merged_df['Year']=payment_date_converted.year
merged_df['Year']=merged_df['Year'].astype('str')

merged_df['Month']=payment_date_converted.month
merged_df['Month']=merged_df['Month'].astype('str')
merged_df['Month'] = merged_df['Month'].str.zfill(2)

merged_df['Year_Month']=merged_df['Year']+'-'+merged_df['Month']

merged_df=merged_df.drop(columns=['payment_date'])

product_category_dictionary={"butter croissant":"food","cinnamon bun":"food", "multigrain toast with butter":"food",
                                    "iced coffee":"drinks","hot coffee":"drinks"}
```

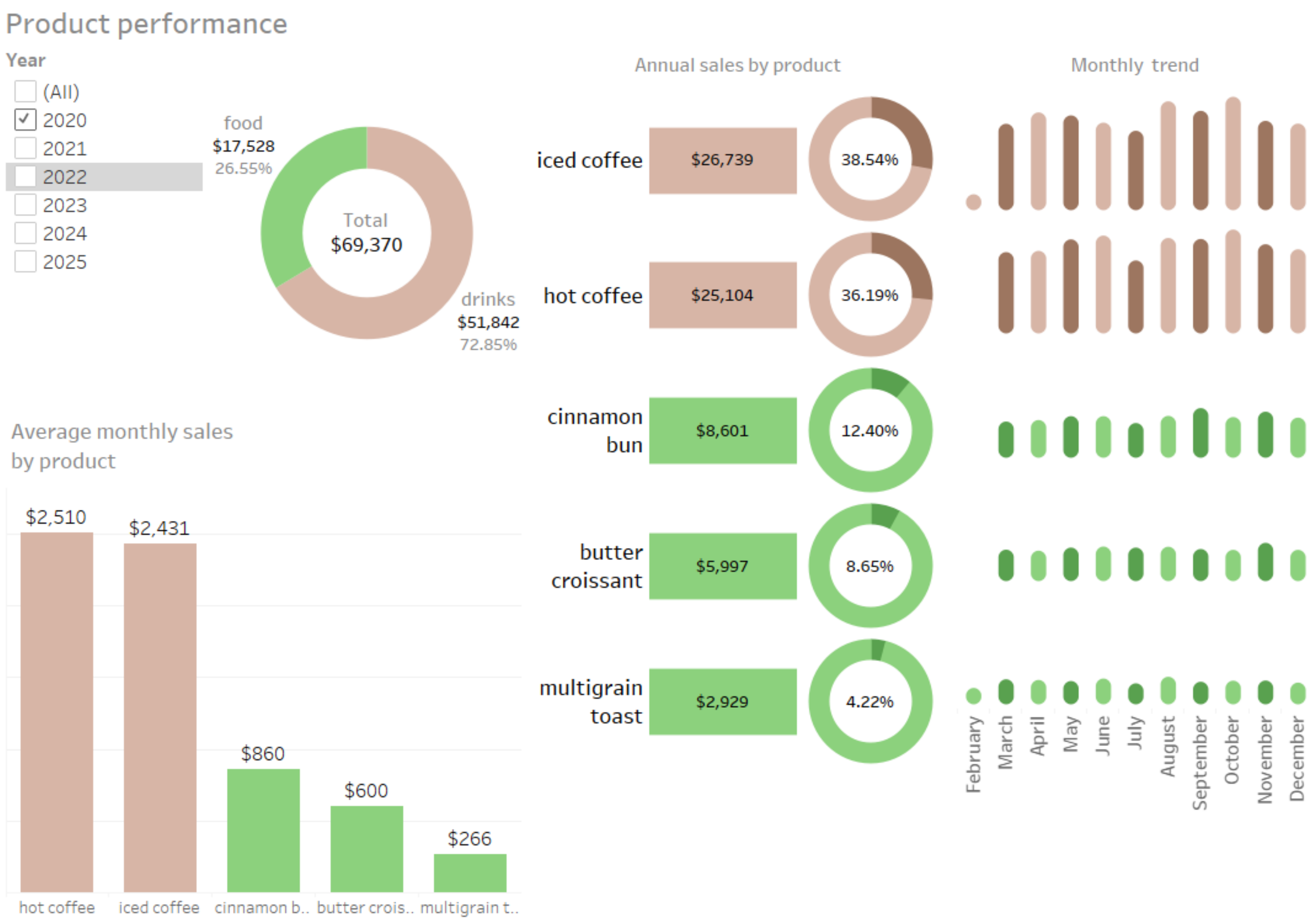
Screenshot of the output of the code (.csv file with aggregated data):



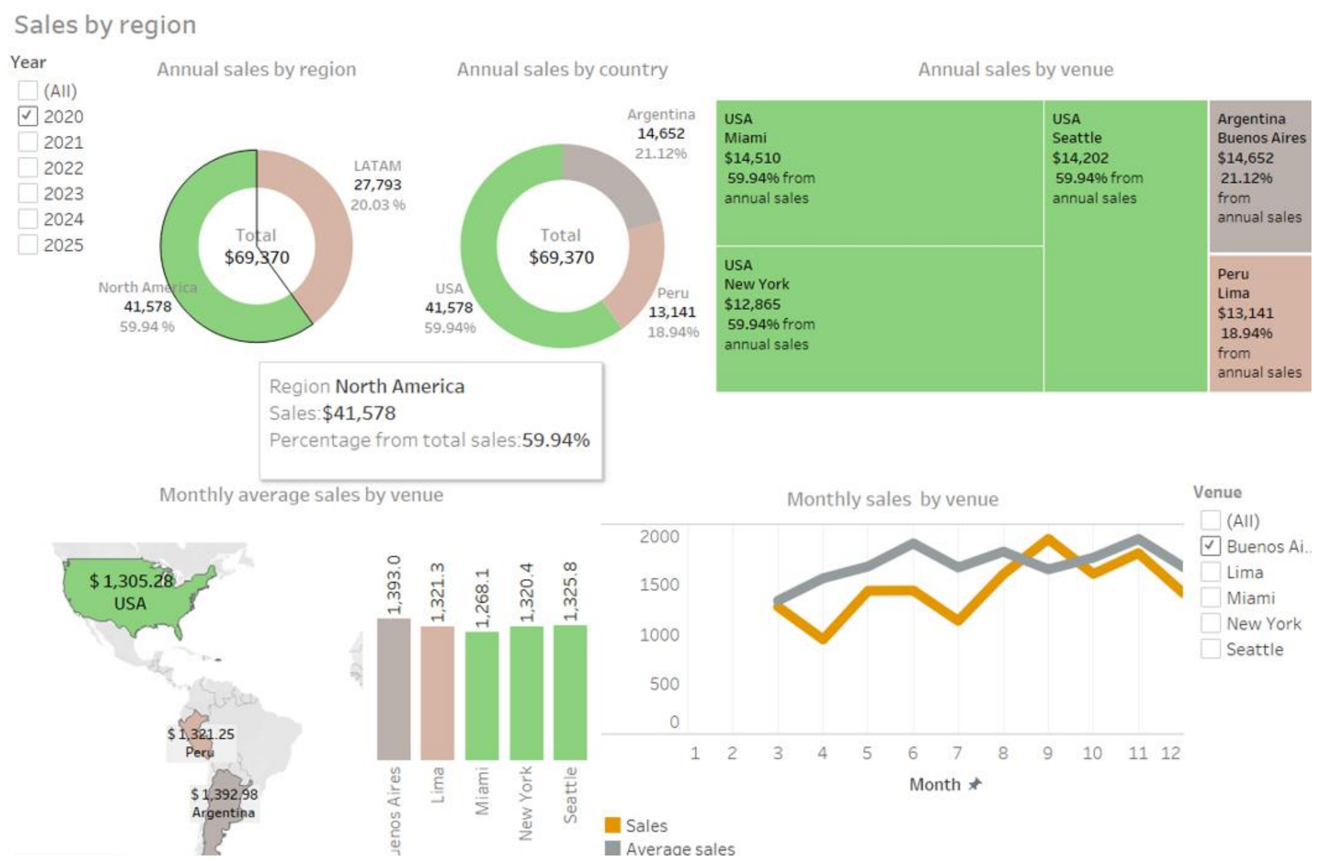
| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | region | country | Year | annual_revenue_by_countr | total_annual_revenue | percentage_from_total_annual_revenue |
| 2 | LATAM | Argentina | 2020 | 14651.99 | 69370.36 | 21.12 |
| 3 | LATAM | Argentina | 2021 | 17431.71 | 77205.21 | 22.58 |
| 4 | LATAM | Argentina | 2022 | 16590.86 | 81270.51 | 20.41 |
| 5 | LATAM | Argentina | 2023 | 15056.71 | 80202.25 | 18.77 |
| 6 | LATAM | Argentina | 2024 | 18121.97 | 83056.1 | 21.82 |
| 7 | LATAM | Argentina | 2025 | 3118.35 | 14556.93 | 21.42 |
| 8 | LATAM | Peru | 2020 | 13140.71 | 69370.36 | 18.94 |
| 9 | LATAM | Peru | 2021 | 15265.27 | 77205.21 | 19.77 |
| 10 | LATAM | Peru | 2022 | 16834.54 | 81270.51 | 20.71 |
| 11 | LATAM | Peru | 2023 | 16303.13 | 80202.25 | 20.33 |
| 12 | LATAM | Peru | 2024 | 16025.24 | 83056.1 | 19.29 |
| 13 | LATAM | Peru | 2025 | 3027.43 | 14556.93 | 20.8 |
| 14 | North America | USA | 2020 | 41577.66 | 69370.36 | 59.94 |
| 15 | North America | USA | 2021 | 44508.23 | 77205.21 | 57.65 |
| 16 | North America | USA | 2022 | 47845.11 | 81270.51 | 58.87 |
| 17 | North America | USA | 2023 | 48842.41 | 80202.25 | 60.9 |
| 18 | North America | USA | 2024 | 48908.89 | 83056.1 | 58.89 |
| 19 | North America | USA | 2025 | 8411.15 | 14556.93 | 57.78 |

Part 3:Visualisations.Creating two dashboards in Tableau: first dashboard shows sales overview across regions and second offers sales comparison between products and between product categories.

The use of filters makes dashboards dynamic (user can utilize filters to display information for selected years):



Dashboard with sales by geographic region shows comparison of annual sales between regions, between countries and between individual coffee shops as well as monthly sales in each country and chart that compares monthly sales of a particular coffee shop for the selected year with average monthly sales of this coffee shop.



: