# Goal

The purpose of this assignment is to showcase your skills in analyzing business needs and in utilizing Python and SQL to provide solutions.

# Assignment

Scenario:

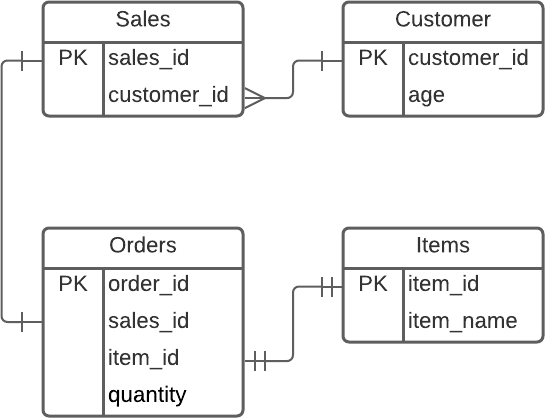
Company XYZ held a promo sale for their signature items named: x,y,z. Sales are at an

all-time high, but they want to create a marketing strategy to target age groups of people by looking at total quantities purchased.

They then created a database with these business rules:

* A sales receipt can have multiple items in an order.
* For every order, the clerk records all quantities for all items, including items not bought (which they denote with quantity=NULL).
* Each customer can do multiple sales transactions, and has his/her age stored in a database.

Refer to the image below for the table structures and relationships.



Objectives

Create a Python script that can:

1. connect to the SQLite3 database provided
2. extract the total quantities of each item bought per customer aged 18-35.
   * For each customer, get the sum of each item
   * Items with no purchase (total quantity=0) should be omitted from the final list
   * No decimal points allowed (The company doesn’t sell half of an item ;) ) Challenge: Provide 2 solutions, one using purely SQL, the other using Pandas
3. store the query to a CSV file, delimiter should be the semicolon character (';')

Test case:

Customer 1 bought Item X on multiple occasions, totaling 10 for Item X only Customer 2 bought one of each item only once, totaling 1 each Item Customer 3 bought Item Z on two occasions, totaling 2 for Item Z only

Then the output file should look like the example below:

Customer;Age;Item;Quantity 1;21;x;10

2;23;x;1

2;23;y;1

2;23;z;1

3;35;z;2

(Note: Actual values will vary)

# Delivery

Provide a link to a public repository where we can download the code containing all files.

# Answer for above Assignment:

We need to write a Python script that:

1. Connects to the SQLite database.

2. Extracts total quantities of each item bought by customers aged 18–35.

Sum each item’s quantity per customer.

Ignore items with total quantity = 0 or NULL.

Output should be integers only.

3. Provide two solutions:

One using pure SQL.

One using Pandas after fetching data.

4. Export results to CSV with ; as delimiter.

#SQL Query:

SELECT

c.customer\_id AS Customer,

c.age AS Age,

i.item\_name AS Item,

SUM(o.quantity) AS Quantity

FROM Customer c

JOIN Sales s ON c.customer\_id = s.customer\_id

JOIN Orders o ON s.sales\_id = o.sales\_id

JOIN Items i ON o.item\_id = i.item\_id

WHERE c.age BETWEEN 18 AND 35

AND o.quantity IS NOT NULL

GROUP BY c.customer\_id, c.age, i.item\_name

HAVING SUM(o.quantity) > 0

ORDER BY c.customer\_id, i.item\_name;

# We can solve this Python Script with Two Solutions:

import sqlite3

import pandas as pd

conn = sqlite3.connect("sales.db")

cursor = conn.cursor()

# Solution 1: By using SQL

query = “””

SELECT

c.customer\_id AS Customer,

c.age AS Age,

i.item\_name AS Item,

SUM(o.quantity) AS Quantity

FROM Customer c

JOIN Sales s ON c.customer\_id = s.customer\_id

JOIN Orders o ON s.sales\_id = o.sales\_id

JOIN Items i ON o.item\_id = i.item\_id

WHERE c.age BETWEEN 18 AND 35

AND o.quantity IS NOT NULL

GROUP BY c.customer\_id, c.age, i.item\_name

HAVING SUM(o.quantity) > 0

ORDER BY c.customer\_id, i.item\_name;

”””

df\_sql = pd.read\_sql\_query(query, conn)

df\_sql.to\_csv("output\_sql.csv", sep=";", index=False)

# Solution 2: Using Pandas

customers = pd.read\_sql\_query("SELECT \* FROM Customer", conn)

sales = pd.read\_sql\_query("SELECT \* FROM Sales", conn)

orders = pd.read\_sql\_query("SELECT \* FROM Orders", conn)

items = pd.read\_sql\_query("SELECT \* FROM Items", conn)

merged = (

orders

.merge(sales, on="sales\_id")

.merge(customers, on="customer\_id")

.merge(items, on="item\_id")

)

merged = merged[(merged["age"] >= 18) & (merged["age"] <= 35)]

merged["quantity"] = merged["quantity"].fillna(0)

result = (

merged.groupby(["customer\_id", "age", "item\_name"], as\_index=False)["quantity"]

.sum()

)

result = result[result["quantity"] > 0]

result.rename(

columns={

"customer\_id": "Customer",

"age": "Age",

"item\_name": "Item",

"quantity": "Quantity"

},

inplace=True

)

result.to\_csv("output\_pandas.csv", sep=";", index=False)

conn.close()

print("CSV files generated: output\_sql.csv, output\_pandas.csv")

# Final Output (from test case given)

output\_sql.csv and output\_pandas.csv should look like:

Customer;Age;Item;Quantity

1;21;x;10

2;23;x;1

2;23;y;1

2;23;z;1

3;35;z;2