Wiwino market analysis

- Repository: `wine-market-analysis`
- Type: `Consolidation`
- Duration: `1 week`
- Deadline: `29/08/2024 3:00 PM (analysis)`
- Team: `group`

Mission Objectives

Consolidate your knowledge in SQL, specifically in:

- SELECT operations
- JOIN operations
- GROUP BY operations
- AGGREGATIONS operations *(average, sum, ...)*
- LIMIT operations
- ... and surely many others 😁

Learning Objectives

- Teamwork through ticketing and Trello
- To be able to read and understand a SQL database diagram
- To be able to query a SQL database
- To be able to write efficient SQL queries
- To be able to create visuals from aggregated insights coming from queries
- To be able to present a market analysis with clear numbers and graphs

The Mission

> We are the company *Wiwino*, proudly active in the wine industry. We have been gathering data about wines from our users for years. We want to have a better understanding of the wine market by analyzing this data.

Do the analysis, summarize your output, and present the results.

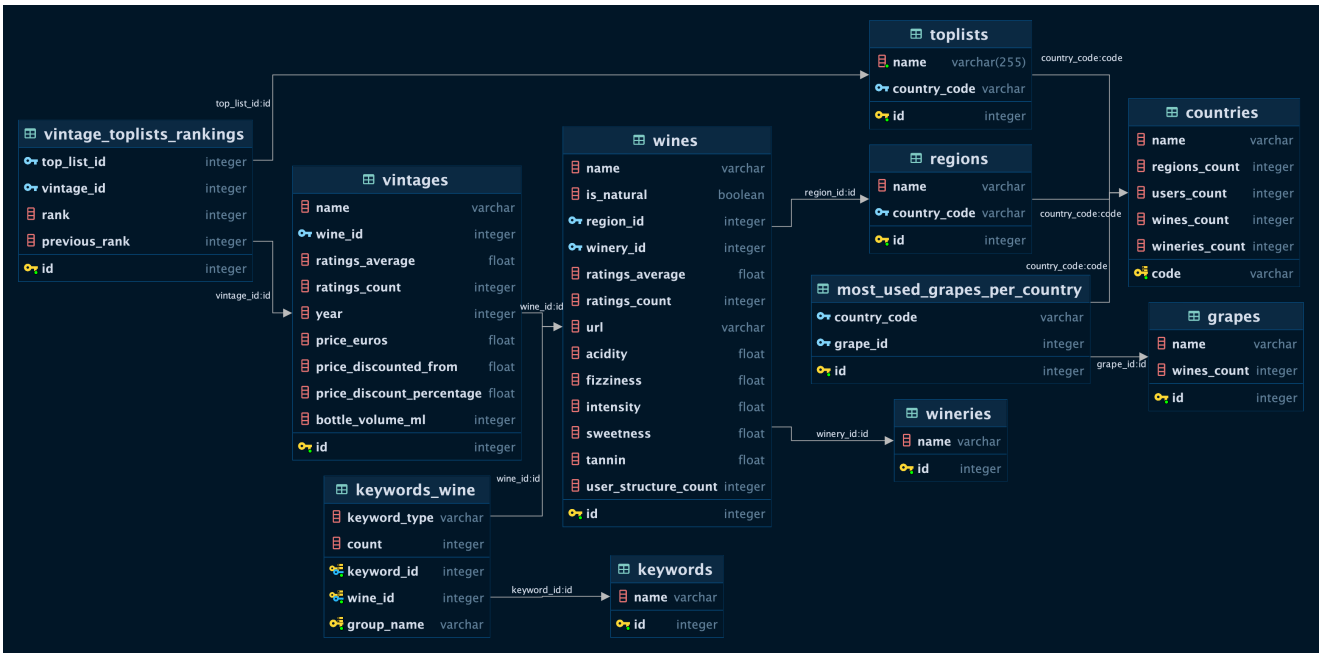
The first two days will be dedicated to individual exploration while we will next week split groups in two. Data engineer minded people will focus on gathering the data in CSV files which will be requested by the Data Analysts through tickets on a Trello.

Data

Wiwino was so kind to gather their data into a database. You can find the SQLite database in the `db/` folder or download it from here.

Below is the database diagram. The `yellow keys` symbol represents `PRIMARY KEYS` while the `blue keys` symbol represents `FOREIGN KEYS`. Each column is typed. You can see that the types are not exactly the

same as Python's types. Here is a list of SQL types.



Must-have features

Crunch the numbers to arrive at a complete market analysis.

You should at least answers these questions from the perspective of Wiwino:

- We want to highlight 10 wines to increase our sales. Which ones should we choose and why?
- We have a limited marketing budget for this year. Which country should we prioritise and why?
- We would like to give awards to the best wineries. Come up with 3 relevant ones. Which wineries should we choose and why?
- We detected that a big cluster of customers likes a specific combination of tastes. We identified a few keywords that match these tastes: *coffee*, *toast*, *green apple*, *cream*, and *citrus* (note that these keywords are case sensitive ⚠). We would like you to find all the wines that are related to these keywords. Check that **at least 10 users confirm those keywords**, to ensure the accuracy of the selection. Additionally, identify an appropriate group name for this cluster.
- We would like to select wines that are easy to find all over the world. **Find the top 3 most common `grapes` all over the world** and **for each grape, give us the the 5 best rated wines**.
- We would like to create a country leaderboard. Come up with a visual that shows the **average wine rating for each `country`**. Do the same for the `vintages`.
- One of our VIP clients likes *Cabernet Sauvignon* and would like our top 5 recommendations. Which wines would you recommend to him?

Give us any other useful insights you found in the data. **Be creative!** 😉

If a certain question is hard or not possible to answer with the data you have been given, document what is missing. This can always happen...

Known issues

- The `wineries` table doesn't link properly to the `wines` table. Match only gives four items so not useful.
- The wineries name is (probably) the wine name.
- The `wines_count` variable is instead in the `most_used_grapes_per_country` table and it doesn't make sense because it is the same for all (so probably a worldwide figure and not per country).
- The `regions_count` variable from the `countries` table doesn't correspond to the actual number of regions found in the database (e.g. way less regions in France than the count variable indicates).
- Not clear what the `user_structure_count` variable means.
- It is not possible to find the exact grape for an individual wine. With string matching you can try to circumvent this, but the matching rate is not very high (due to many wines not referencing the name of their grape).

Nice-to-have features

- Optimise your queries to obtain the results as fast as possible.
- How would you improve the data (quality), the database schema, or the typing?
- Implement visualization best practices
    - Data storytelling, nice design, relevancy to the questions asked, ...

Constraints

- You are not allowed to use pandas or similar tools for the data analysis, you should use SQL and SQL only
    - For instance, use SQL `JOIN`s to cross-reference tables, not `pd.merge()`
    - But you can of course use a Python ORM library if you like
- Write your queries in dedicated `.sql` files or a `queries.py` file with the queries as strings
- For visualizing your insights, use either Python or Excel

Deliverables

1. Publish your source code in a GitHub repository
2. Pimp the README file
    - Include the main insights in it
3. Show us your results in a nice presentation
    - Can be with PowerPoint, a smooth Jupyter notebook & Markdown, a printout, ...

Steps

1. Create the repository
2. Study the project brief (Who? Why? What?)
3. Identify the technical challenges (How?)
4. Start exploring the data
5. Answer the questions with clear queries
6. Create a presentation with your findings
7. Clean your code and finish up your repository

> Again: be creative in both the analysis and delivery!!! Try to impress us.

Evaluation

Criterion	Indicator	Yes/No
1. Is complete	You have an answer for each must-have question	
	You push your changes to GitHub at least once a day	
	There is a visualization available when it makes sense	
2. Is great	You SQL queries are optimized	
	Your code is commented/typed	
	You presentation is clear and well designed	

Final note