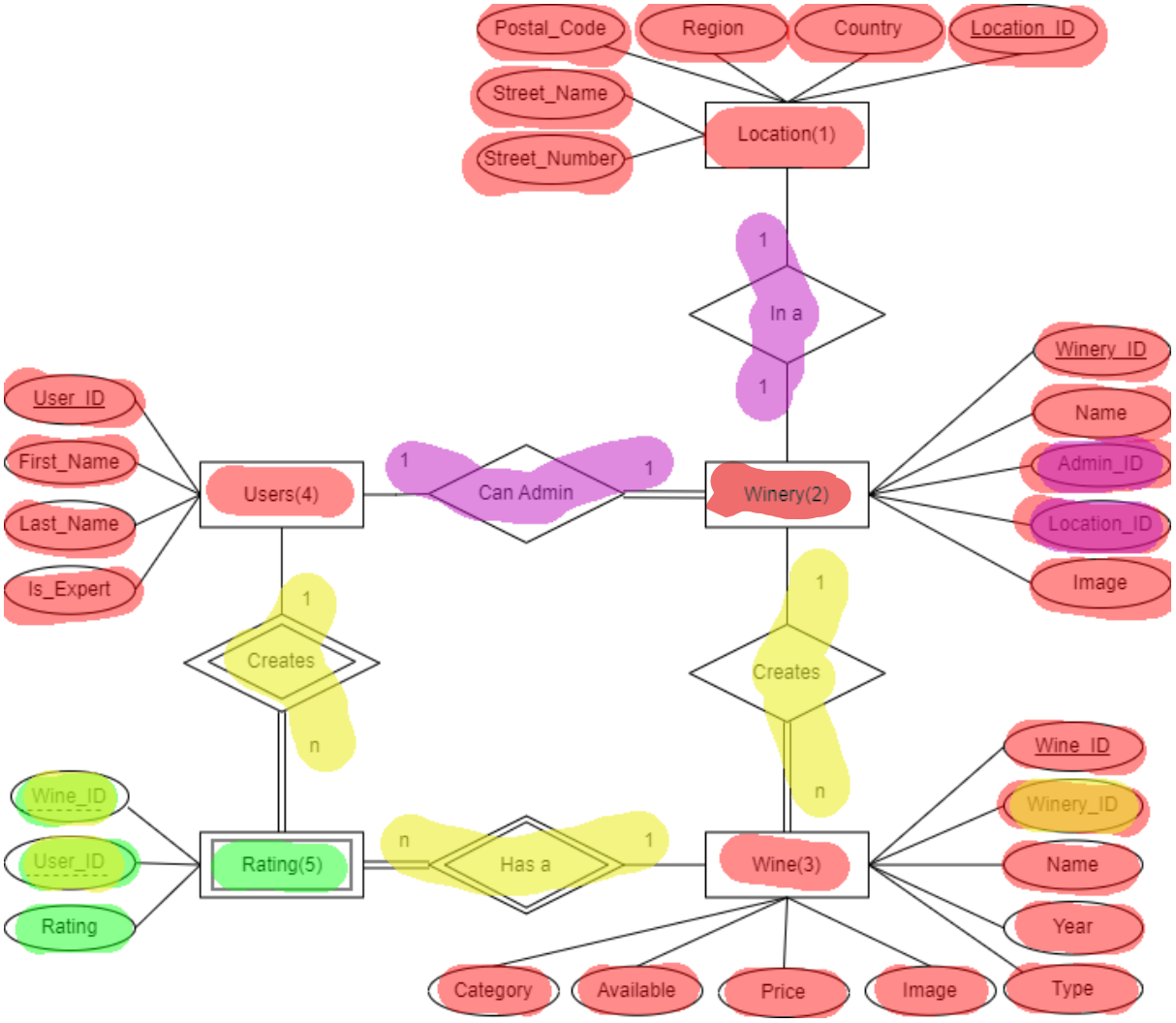
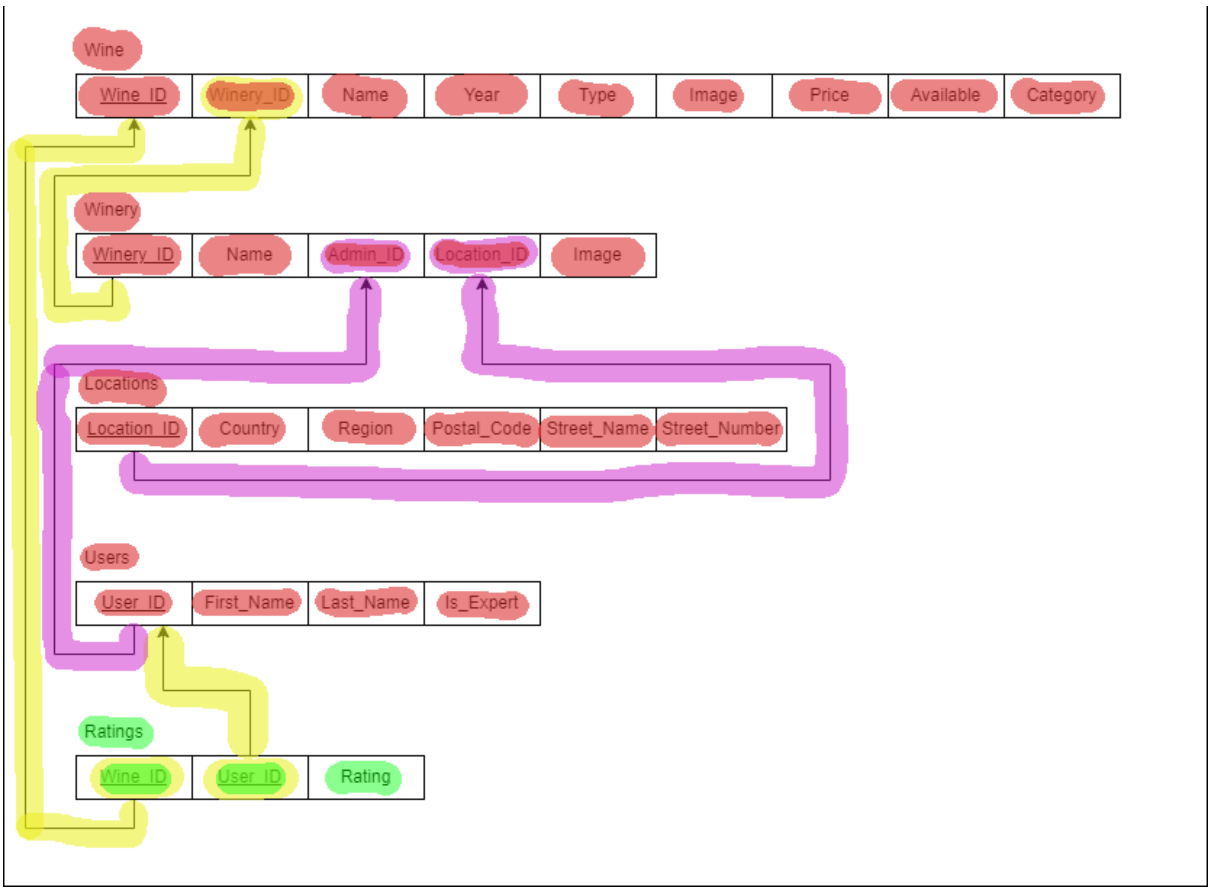
Task 1:

See included pdf in file titled “”Group3_P5_Research”

Task 2:





Step 1 (Indicated by red):
All entities highlighted are strong, and all of their attributes are simple

Step 2 (Indicated by green):
Ratings is a weak entity, inheriting its composite key attributes “Wine_ID” and “User_ID” from the Wine and Users entities respectively.

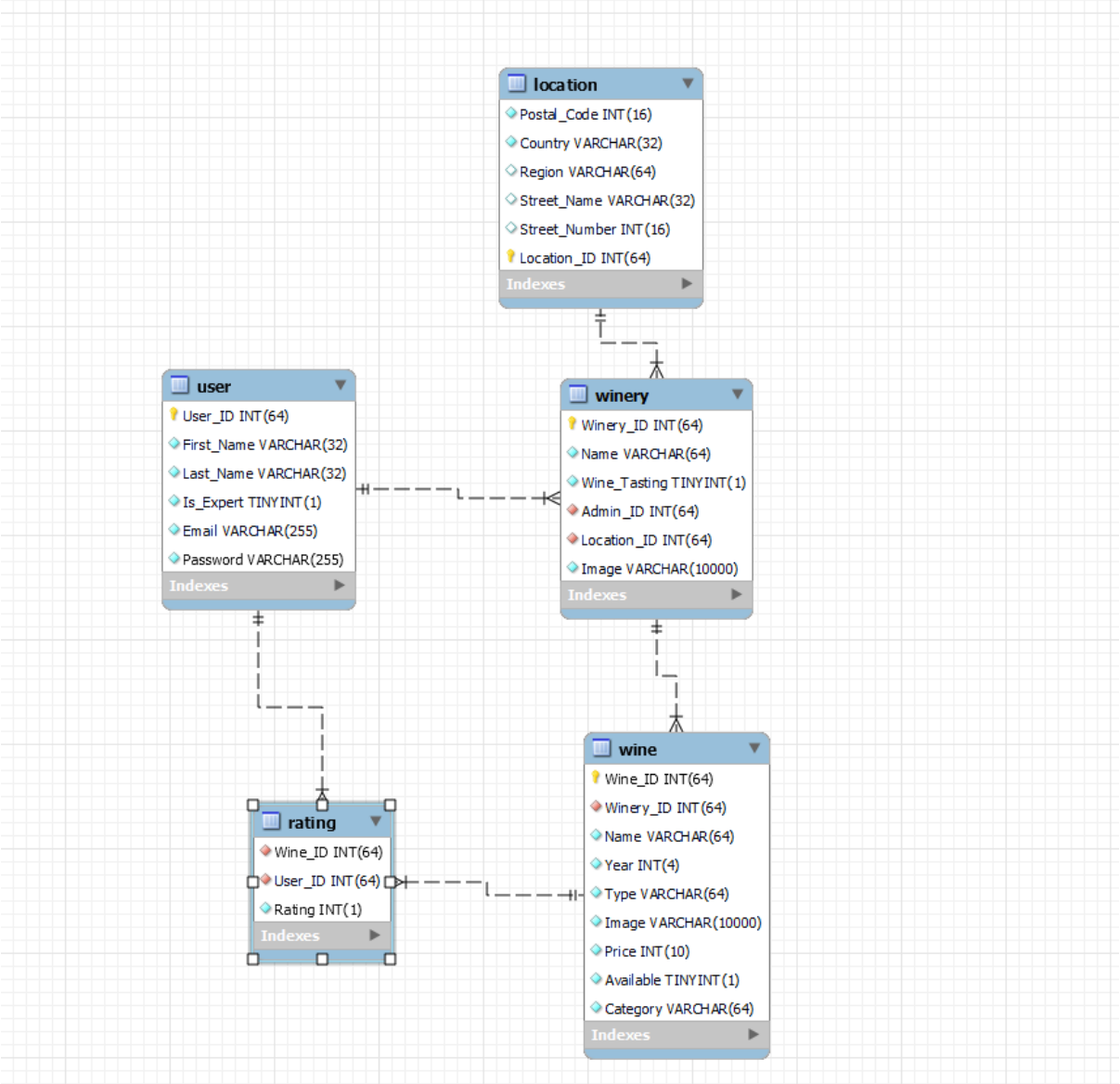
Step 3 (Indicated by purple):
The Admin_ID attribute comes from the relationship between Wine and Ratings, being a foreign key from the User_ID attribute from the Users entity. This relationship is 1:1 and, therefore, this attribute is highlighted accordingly. The same case is true for the relationship between Winery and Locations, they have a 1:1 relationship, with Winery having the foreign key Location_ID from the primary key of Locations by the same name.

Step 4 (indicated by yellow):
A user can rate multiple wines, thus Users and Ratings have a 1:N relationship. Wines have multiple ratings as well, giving Wine and Ratings a 1:N relationship too. Accordingly, the Ratings entity has both the User_ID and Wine_ID as foreign keys (the pair also doubling as the entity’s composite key). A winery can have multiple wines as well, making the WIne and Winery relationship 1:N, with Winery_ID being present in the Wine entity as a foreign key.

Steps 5 – 9:
N/A due to none of the properties being present in the Enhanced ER Diagram

Task 4:

Visual representation of the schema:



See database dump “DB_Dump/WinesDB.sql” for the SQL statements

Task 5:

See ReadMe file.

Task 6:

When populating the database, the provided wine APIs were used to keep information as real as possible. With them, only real wineries have been used as data. Wines made by these wineries were also gathered by this method. With this there was still some outstanding information that was not obtainable from the API, such as images of the wineries, their street names, numbers of wines and their prices.

I came up with the prices and the street names, they are not real. The images of the wineries are real images I gathered on the internet. With this we were able to keep information as real as possible without spending an unreasonable amount of time on gathering data.

When populating the Winery_Tasting information,Google searches were used. This is so that exclusive tasting isn't recommended to tourists.

Task 8:

- Timothy Whitaker (u22744968):Creation, maintenance, merging and conflict resolution of Github. Login and Signup validation (client and server). Worked on and fixed the API (mainly wine add/update/delete). PHP server setup and installation for local use. Logo creation. Added a Readme. General support and bug fixing.

- Scott Bebington (u201546216):
Worked on the API (getAllWines), created EER to relational mapping, co-design the website (Template), completed the design of the wines page and implemented the js, assisted with the wineries page, designed login and register page, fixing issues when merging branches. General support and bug fixing

- Charlize Hanekom (u22487222): Creation of wineries page (html and javascript), help with general design of website and the wineries part of the api. Also added some data to the database and did the research. General support and bug fixing

- Jayson Du Toit (u22571532): Put information in the database with real and mock data. Work on API. Made manage Wines pages as well as add Wines pages with their js. General support and bug fixing

- Stefan Jansen van Rensburg (u22550055):Creation of API, adjustments to database(adding tasting information). General support and bug fixing. API testing of all functions. Creation of API documentation.

- Willie van Rooyen (u22675699): Creation of API, Standardization, Styling of Login and Register Pages, Tech Support. Helped with the css Styling and general design. Assisted with integration between API and javascript. Conflict Resolution on Github. General support and bug fixing

- Ethan Groenendyk (u20743956): Creation of database, including setting up constraints such as foreign keys. Adjustments to final EER and relational model. Creation of this answer PDF. General support and idea supply. Dedicated Morning person

- All : Late night hee hee's.

Assumptions

We assumed that experts as well as admins are not made through the website but need to contact the company in order to be added.

We assumed that a winery only has one manager and that a manager can only manage one winery.

For recommending the best winery on location, the user can filter by location (Country) and sort by rating from high to low in order to get the recommendations.
We can make the assumption that the location (Country, region, street address etc.) won't change.