UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

# COS 221

Practical Assignment 5


Group 5

Error 404: Group Name Not Found

| | |
|---|---|
| Jacobus Smit | u21489476 |
| Iwan de Jong | u22498037 |
| Rebecca Oosthuizen | u20512008 |
| Ashley Tullis | u22486985 |
| Dawie Reyneke | u21438112 |
| Tiaan Pouwels | u21675229 |


7 June 2023

# Task 1: Research

As of May 2022, the tourism industry in South Africa accounts for 3,7% of the country's gross domestic product. This industry is an important driver of South Africa's economy and contributes to job creation and cultural exchange. In the fallout of the Covid-19 pandemic, the tourism industry was among the most negatively affected sectors, experiencing a global decline and resulting in a loss of countless jobs and businesses.

Wine is an alcoholic beverage made of fermented grape juice, while a winery is a licensed property that produces wine. There are a number of wine and winery varieties and characteristics that are of interest to a wine tourist when deciding where to visit and what wines to try.

The National Department of Tourism wants to make South Africa a wine tourism hotspot, given its position as a global leader in wine production. We, as a team of second year Computer Science students, have been contracted to design and implement a wine tourism application. Using our research on wine and wineries, these are the specifications of the project we will build:

**Functional Requirements:**

- The user should be able to login to the web-based application.
- The user should be able to view the various wines in the system, along with their year, type, producer, country of origin, alcohol percentage, price, rating, and the winery that holds them. The wines will be able to be easily searched and filtered through.
- The user should be able to view the various wineries in the system, along with their locations and verification statuses. The wineries will be able to be easily searched and filtered through.
- Users can be of type 'general user', 'connoisseur' or 'manager'. General users will be able to leave general reviews of wines, while connoisseurs will be able to leave critic reviews of wines. Managers will each manage one winery.
- Verified wineries should be able to add new wines to their catalog.
- The application should be able to manage users, wines and wineries.
- The application should be able to update the database.
- The application should be able to suggest the best destinations based on a user's location.

**GitHub Repository:**

https://github.com/PA5G5/PA5.git

**References:**


[1] https://www.gov.za/speeches/deputy-minister-lindiwe-sisulu-tourism-dept-budget-vote-202223-19-may-2022-0000

[2] https://www.unwto.org/tourism-data/international-tourism-and-covid-19

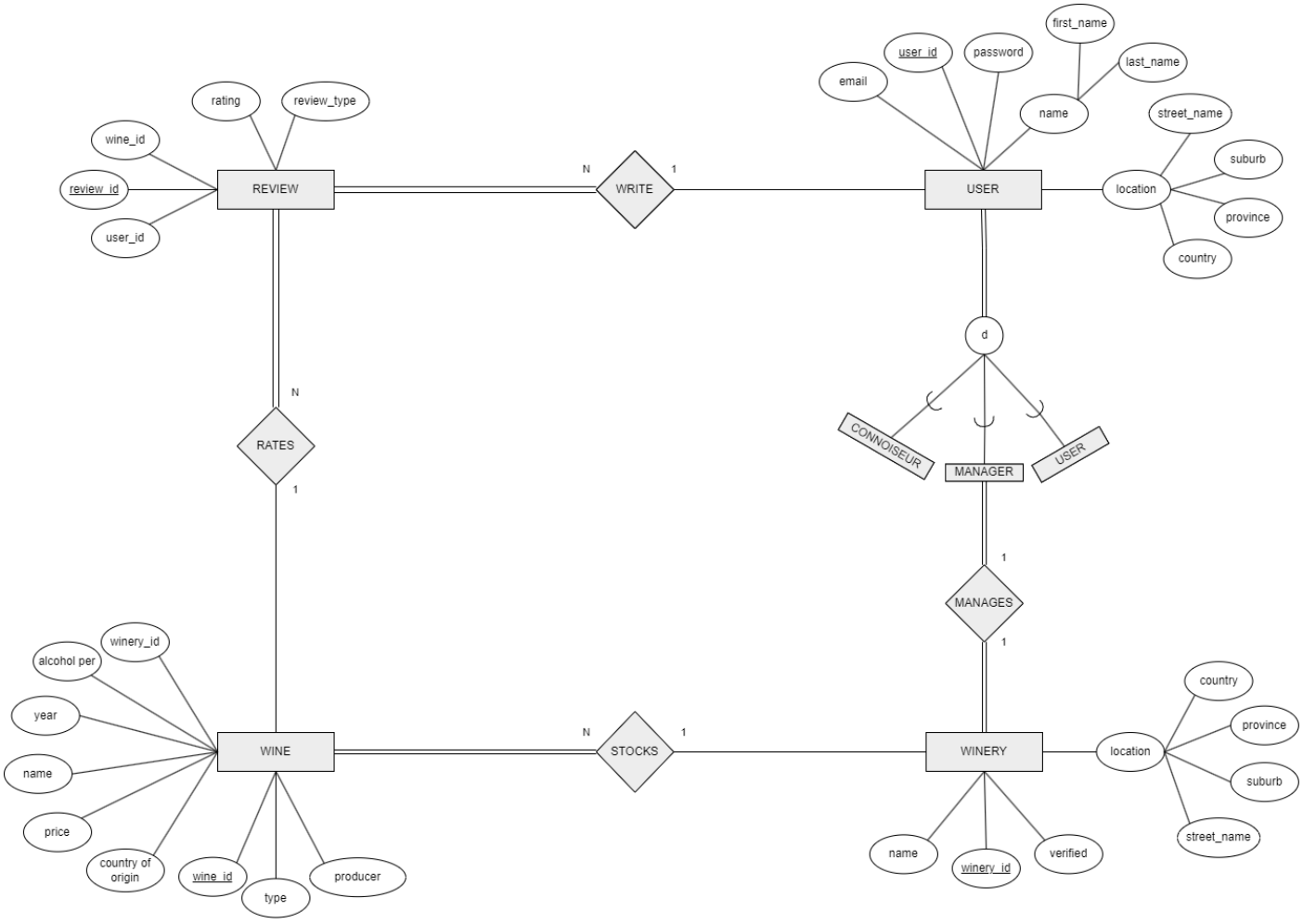
[3] https://winefolly.com/deep-dive/what-is-wine/

[4] https://foodandroad.com/what-is-wine-tourism/

[5] https://cgtwines.com/the-difference-between-a-winery-and-a-vineyard/

[6] https://www.vivino.com/US/en/

# Task 2: (E)ER-Diagram




**Assumptions and other Information:**

- Each wine can only be stored at one winery at a time.

# Task 3: (E)ER-Diagram to Relational Mapping

### Step 1: Mapping of Regular Entity Types

**Wine**

| Wine_id | Type | Producer | AlcoholPer | Year | Name | Price | Country_of_Origin |
|---|---|---|---|---|---|---|---|

**Winery**

| Winery_id | Name | Street_name | Suburb | Province | Country | Verified |
|---|---|---|---|---|---|---|

**User**

| User_id | Email | Password | First_name | Last_name | Street_name | Suburb | Province | Country |
|---|---|---|---|---|---|---|---|---|

**Review**

| Review_id | Rating | Review_type |
|---|---|---|

### Step 2: Mapping of Weak Entity Types

Not applicable

### Step 3: Mapping of Binary 1:1 Relationships

Not applicable

### Step 4: Mapping of Binary 1:N Relationships

**Wine**

| Wine_id | Type | Producer | AlcoholPer | Year | Name | Price | Country_of_Origin | Winery_id |
|---|---|---|---|---|---|---|---|---|

**Winery**

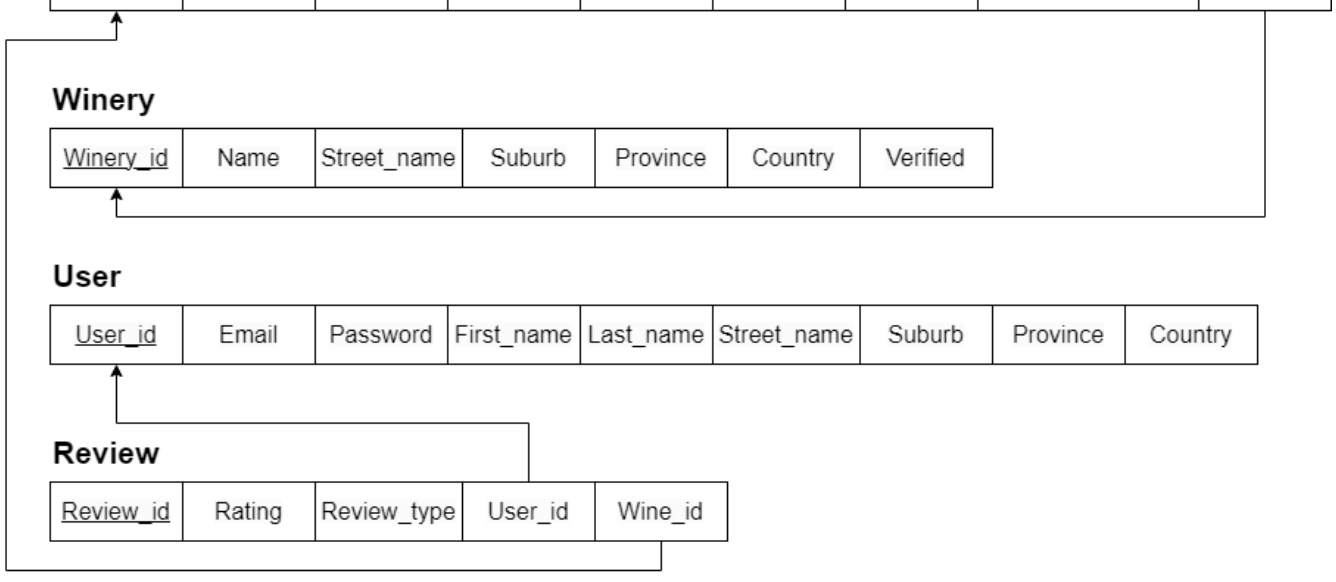
| Winery_id | Name | Street_name | Suburb | Province | Country | Verified |
|---|---|---|---|---|---|---|

**User**

| User_id | Email | Password | First_name | Last_name | Street_name | Suburb | Province | Country |
|---|---|---|---|---|---|---|---|---|

**Review**

| Review_id | Rating | Review_type | User_id | Wine_id |
|---|---|---|---|---|



**Step 5: Mapping of Binary M:N Relationships**

Not applicable

**Step 6: Mapping of Multivalued Attributes**

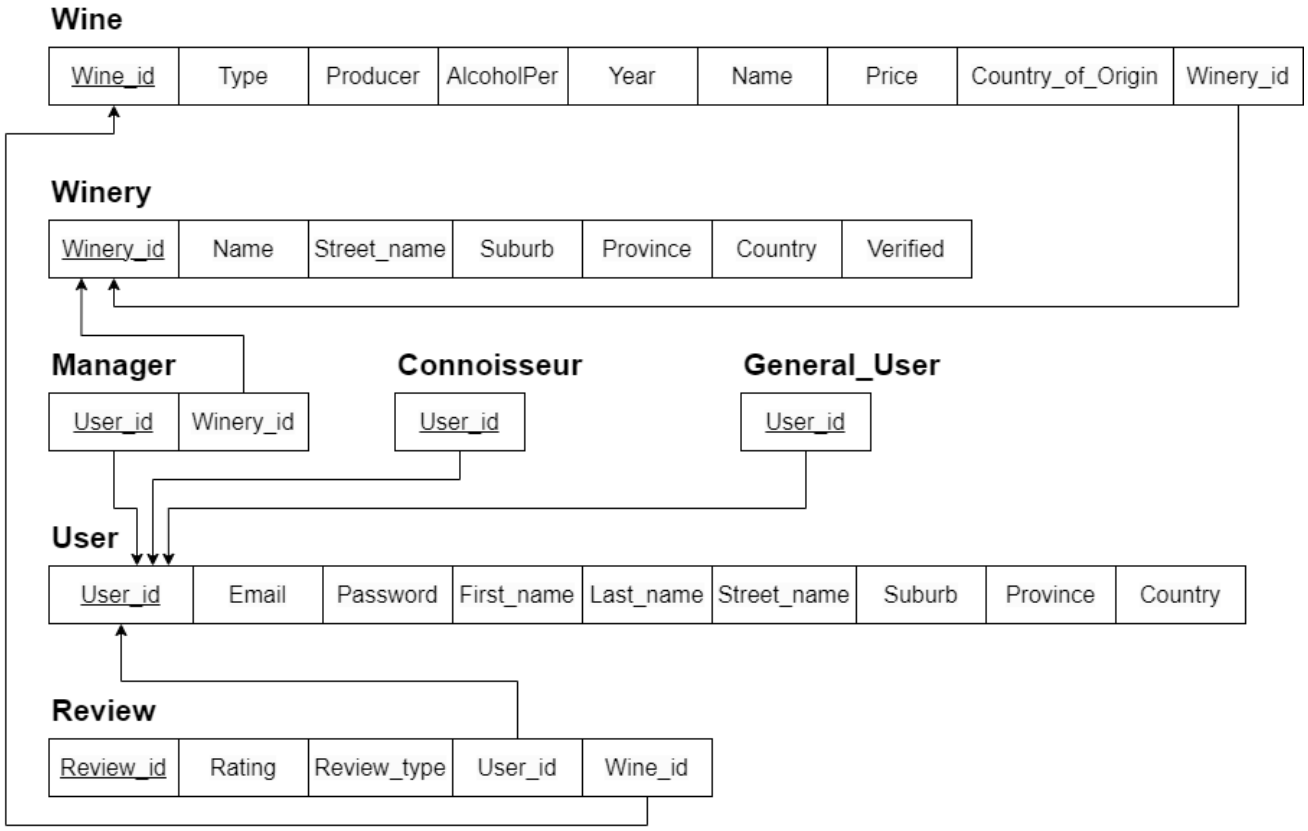
Not applicable

**Step 7: Mapping of N-ary Relationships**

Not applicable

**Step 8: Mapping Specialisation and Generalisation**

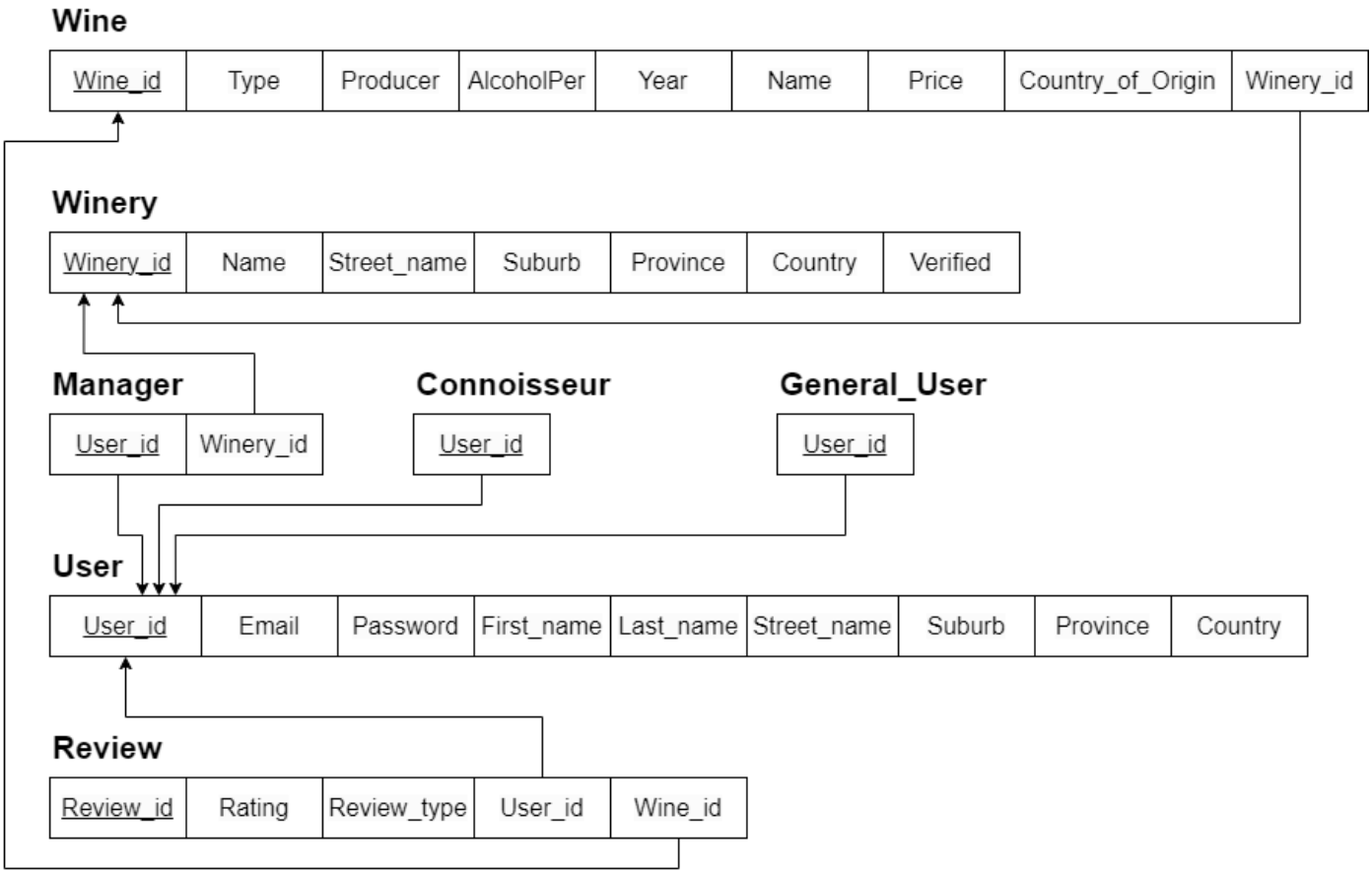
Option 8B: Multiple relations - subclass relations only




**Step 9: Mapping Unions**

Not applicable

## Final Relational Mapping



# Task 4: Relational Schema

## Visual Diagram

### Wine

| Wine_id | Type | Producer | AlcoholPer | Name | Price | Country_of_Origin | Year | Winery_id |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

### Winery

| Winery_id | Name | Street_name | Suburb | Province | Country | Verified |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

### User

| User_id | Email | Password | First_name | Last_name | Street_name | Suburb | Province | Country |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

### Manager

| User_id | Email |
|---|---|
| | |
| | |
| | |
| | |

### General_User

| User_id |
|---|
| |
| |
| |
| |

### Connoisseur

| User_id |
|---|
| |
| |
| |
| |

### Review

| Review_id | User_id | Wine_id | Rating | Review_type |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |

**SQL Statements**

```
CREATE DATABASE Practical5;

USE Practical5;

CREATE TABLE User (
     User_id INT PRIMARY KEY,
     Email VARCHAR(50),
     Password VARCHAR(50),
     First_name VARCHAR(50),
     Last_name VARCHAR(50),
     Street_name VARCHAR(50),
     Suburb VARCHAR(50),
     Province VARCHAR(50),
     Country VARCHAR(50)
);

CREATE TABLE Winery(
     Winery_id INT PRIMARY KEY,
     Name VARCHAR(50),
     Street_name VARCHAR(50),
     Suburb VARCHAR(50),
     Province VARCHAR(50),
     Country VARCHAR(50),
     Verified INT
);

CREATE TABLE Wine(
     Wine_id INT PRIMARY KEY,
     Type VARCHAR(50),
     Producer VARCHAR(50),
     Alcohol_per DECIMAL(5,2),
     Year INT,
     Name VARCHAR(50),
     Price INT,
     Country_of_Origin VARCHAR(50),
     Winery_id INT,
     FOREIGN KEY (Winery_id) REFERENCES Winery(Winery_id)
);

CREATE TABLE Review(
     Write_id INT AUTO_INCREMENT PRIMARY KEY,
     User_id INT,
     Wine_id INT,
     Rating VARCHAR(500),
     Review_type VARCHAR(50)
     FOREIGN KEY (User_id) REFERENCES User(User_id),
     FOREIGN KEY (Wine_id) REFERENCES Wine(Wine_id)
);

CREATE TABLE Manager(
     Winery_id INT,
     User_id INT PRIMARY KEY,
     FOREIGN KEY (User_id) REFERENCES User(User_id),
     FOREIGN KEY (Wine_id) REFERENCES Wine(Wine_id)
);
```

```
CREATE TABLE Connoisseur(
     User_id INT PRIMARY KEY,
     FOREIGN KEY (User_id) REFERENCES User(User_id),
);

CREATE TABLE General_user(
     User_id INT PRIMARY KEY,
     FOREIGN KEY (User_id) REFERENCES User(User_id)
);
```

## Task 6: Data

The database was populated manually as the addresses of the wineries were not available in the API. Addresses were therefore researched and inserted into the database. Only a few wineries and their wines were taken from the API. Further, other wineries and wines were generated, and the users were generated randomly from the countries of the wineries that are in the database

## Task 8: Contributions

Jacobus Smit
- Database schema and visual diagram
- Implemented table management

Iwan de Jong
- Created and managed Google Cloud and GitHub repository
- Created the web-based application with Bootstrap/PHP
- Implemented location suggestions
- Created a logo with DALL-E 2 ;)

Rebecca Oosthuizen
- Managed PDF
- Implemented filtering and searching of tables

Ashley Tullis
- EER to Relational Mapping
- Implemented login/logout functionality

Dawie Reyneke
- Created, populated and managed database
- Implemented database operations functionality

Tiaan Pouwels
- EER Diagram
- Implemented Reviews
- Implemented connoisseur login functionality