COS221 Prac 5 Group 12

**Data types and explanation:**

Ids can be changed to varchar generation if the back end team believes this to be easier. Also please remember that everything has \_no\_ capital letter and no spaces, underscores are used.

ENTITIES:

wines :

1. Wine\_id : PK; Identifier for the wines; (INT())
2. image : wine image (url or base64) (VARCHAR(255))
3. Type: the type of wine (VARCHAR(50))
4. Price: the price of the specific wine (FLOAT())
5. User\_rating : derived attribute of the average of all the user ratings of this wine (FLOAT())
6. Quality: critics (using 100-point scale) rating of a wine (entry example : 92) (INT())
7. Alcohol: % alcohol of the wine (FLOAT())
8. Rating\_percentile: derived attribute, position this wine falls in compared to all other wines (as a %, for example 1% means top 1%, 100% means worst of the worst)(INT())
9. Grape\_type: type of grape used for the wine (for example Merlot)(VARCHAR(64))
10. Price\_percentile: derived attribute, position this wine falls in compared to all other wines’ price (as a %, for example 1% means top 1% most expensive, 100% means worst of the worst dirt cheap)(INT())

All\_users:

1. Last\_verified: the date that the account was last verified (user emails should be verified every 6 months)(DATE())
2. Cerified\_critic: is certified to be a critic or not (if true they get a nice font or something or a badge like on twitter when they make reviews)(TINYINT(1))
3. Name: composite attribute; name of user made from (first\_name(VARCHAR(32)) amd last\_name(VARCHAR(64))
4. Api\_key: uniquely generated primary key (VARCHAR(32))
5. Email: its their email address (VARCHAR(50))
6. Social media: multivalued composite attribute(made from : handle(VARCHAR(64)) and type(VARCHAR(32))
7. Password: hashed user password (VARCHAR(64))
8. Salt: salt for the password(VARCHAR(12))
9. Num\_reviews: number of reviews that user has made (INT())

Wineries:

1. Offers\_tours: do they offer tours? (TINYINT(1))
2. Operational: Are they operational right now? (TINYINT(1))
3. Certified: are they certified? (TINYINT(1))
4. Winery\_name: the name of the winery (VARCHAR(64))
5. Winery\_id: the uniquely generated id for the winery (INT())
6. Eco\_friendly: certified to be eco friendly (TINYINT(1))

Locations:

1. Country: VARCHAR(12)
2. Region1: VARCHAR(50)
3. Region2: VARCHAR(50)
4. Province: VARCHAR(30)
5. Location\_id: PK for each location uniquely generated (INT())

Reviews:

1. Review\_id: weak key for reviews, uniquely generated (INT())
2. Rating: rating from 0 to 5 stars that the user gave the wine (FLOAT())
3. Comment: comment reviewer gave about the specific wine. (VARCHAR(255))

RELATIONSHIPS:

Has\_owner:

1. Confirmed: is a confirmed owner (TINYINT(1))
2. Api\_key from the user that is the owner
3. Winery\_id from the winery that that user owns

(each user can only review a single wine once (meaning they can not review the same wine twice))

Made\_review:

1. Api\_key of user that made the review
2. Review\_id of the review that was made

Has\_reviews:

1. Wine\_id of the wine that was reviewed.
2. Review\_id of the review.

Is\_in\_location:

1. Winery\_id
2. Wine\_id (multivalued attribute of all the wines connected to said winery\_id)
3. Location\_id of the winery and its corresponding wines.

|  |  |  |
| --- | --- | --- |
| Location\_id | Winery\_id | Wine\_id (multivalued) |
| 123 | 123 | 123;234;345 |