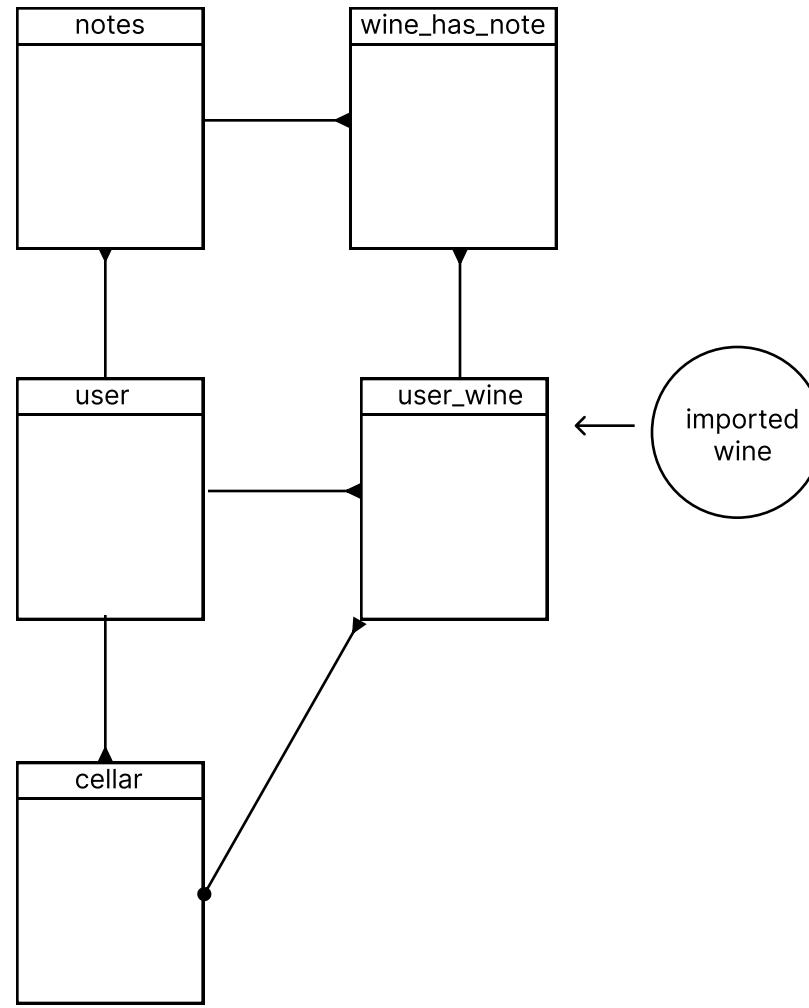


Schémas possibles

Ces systèmes permettent tous de gérer le projet avec chacun leur avantages et désavantages.
Il est aussi possible de combiner un ou plusieurs systèmes

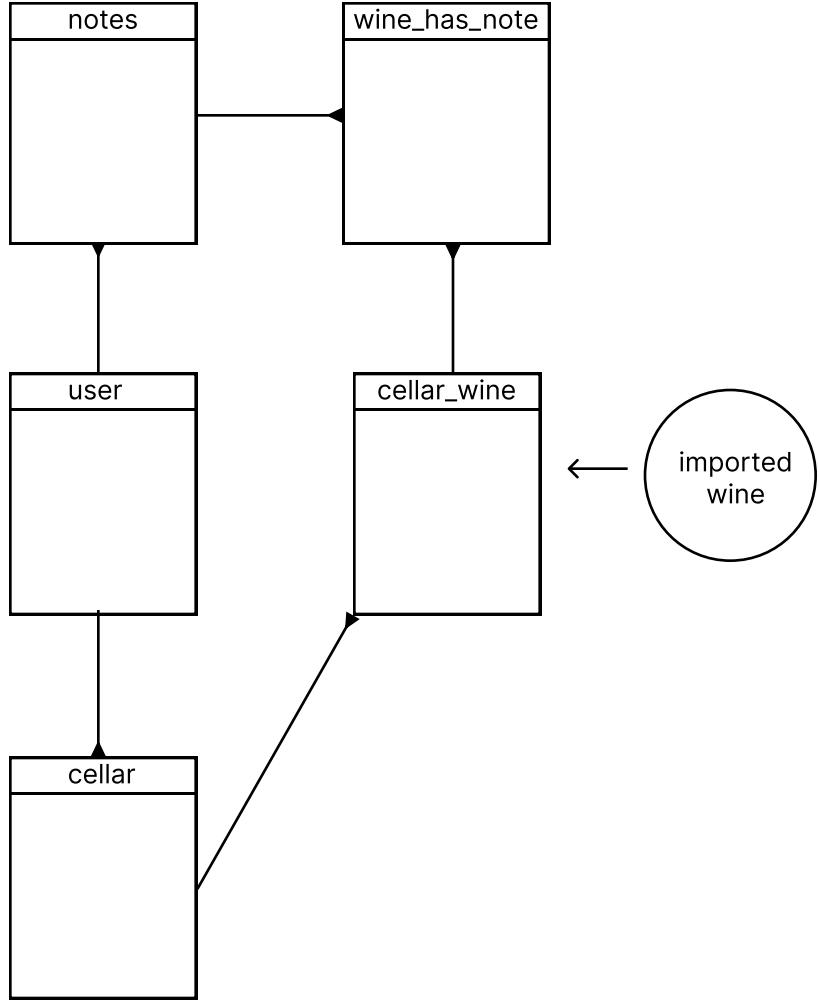


RELATION USER → WINE + USER → CELLIER

selon moi le meilleur choix

PRO: permet plus de flexibilité. Un user peut ou non avoir un cellier, peut avoir des vins même si pas de place dans cellier. peut avoir des notes non associées.

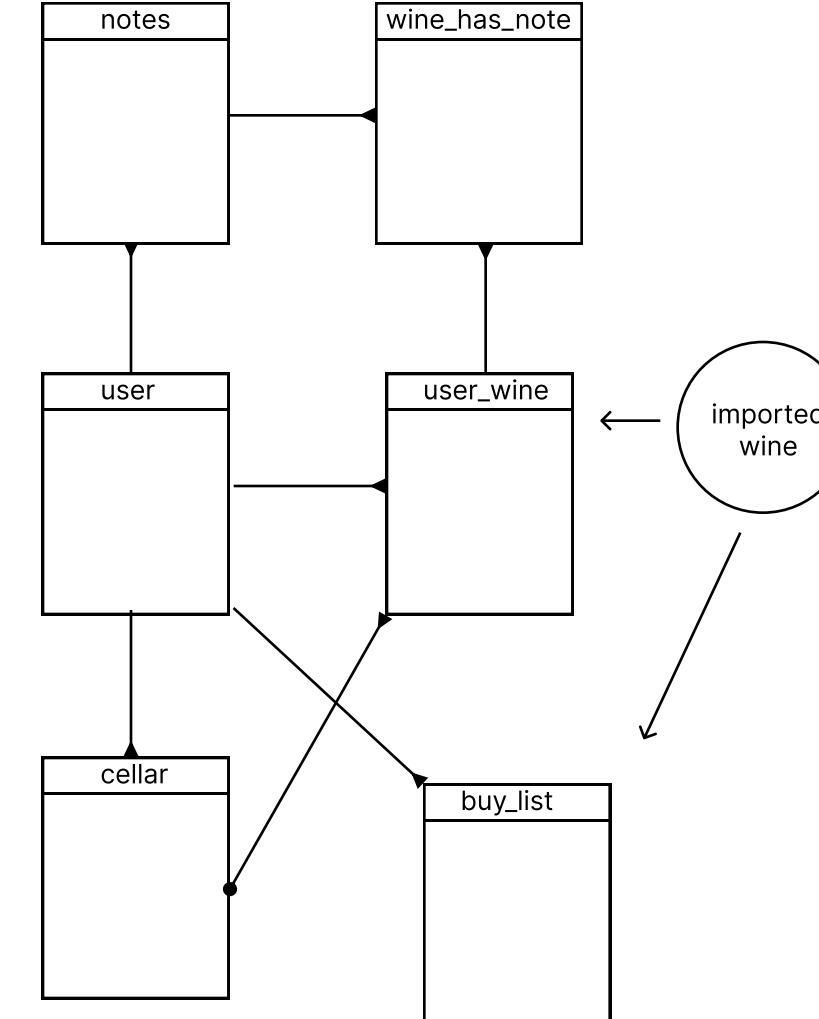
CON: demande des relations plus complexes et abstraites



RELATION USER → CELLIER → WINE

PRO: permet de bien gérer tout le système et relation plus simple

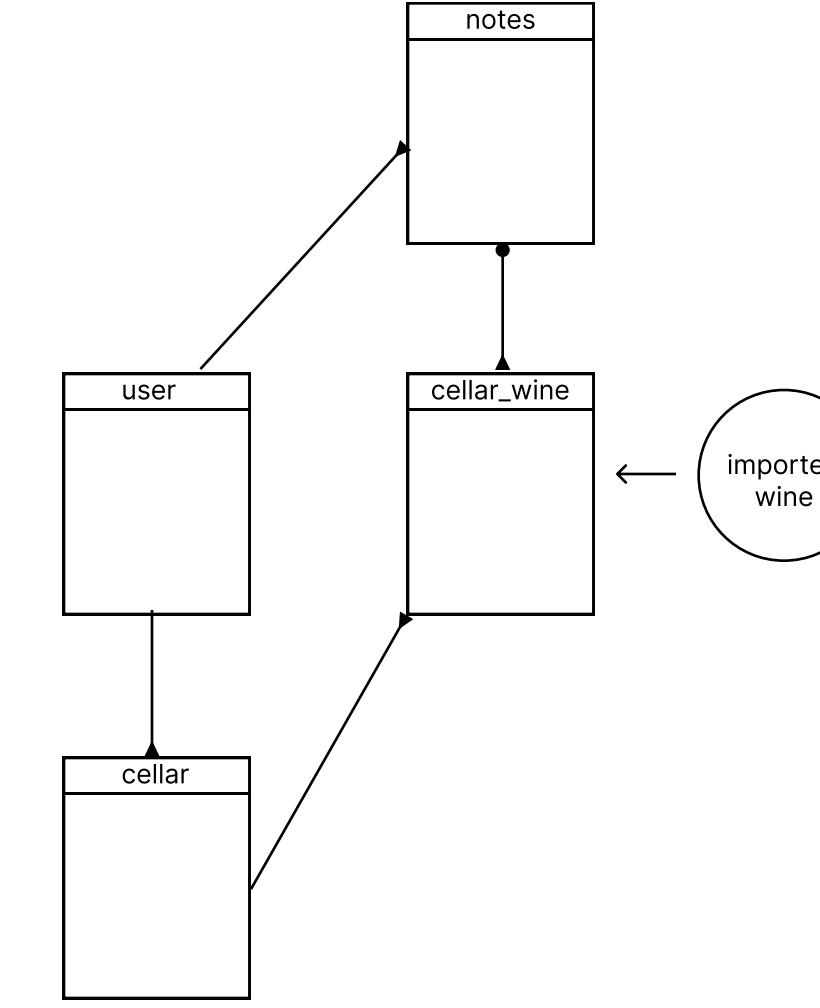
CON: limite le système à la capacité du cellier.



LISTE ACHAT INDÉPENDANTE

PRO: simplifie les requêtes

CON: permet moins de facilité à instaurer un bon UX



VIN A UNE SEULE NOTE ASSOCIÉ

PRO: plus logique qu'un vin n'ai qu'une note. Peut tout de même être copié selon le code_saq.

CON: dans ce cas et les autres, une notes doit être associé à un vin logiquement. Pour avoir des notes indépendantes, il faudrait, encore une fois faire une copie de user_wine ou autre table de donnée

User_wine

- Chaque item est traité comme un objet physique.
- Chaque item doit avoir un statut: (cellier, bu, to_buy,unassigned)
- Chaque item à une date associé à l'ajout et à la consommation
- Un item peut être ajouté manuellement ou être copié de imported_wine ou de user_wine

Note

- Une note appartient à un user et peut être associé à un vin, possibilité de copier à partir du code_saq

Stats et associations virtuelles

- Il est possible de mesurer la table user_wine unitairement ou en utilisant le code_saq ou code_custom
- En utilisant le code_saq/ code_custom, il est possible d'associer les notes et autres infos aux user_wine similaires

Code_saq + code_custom

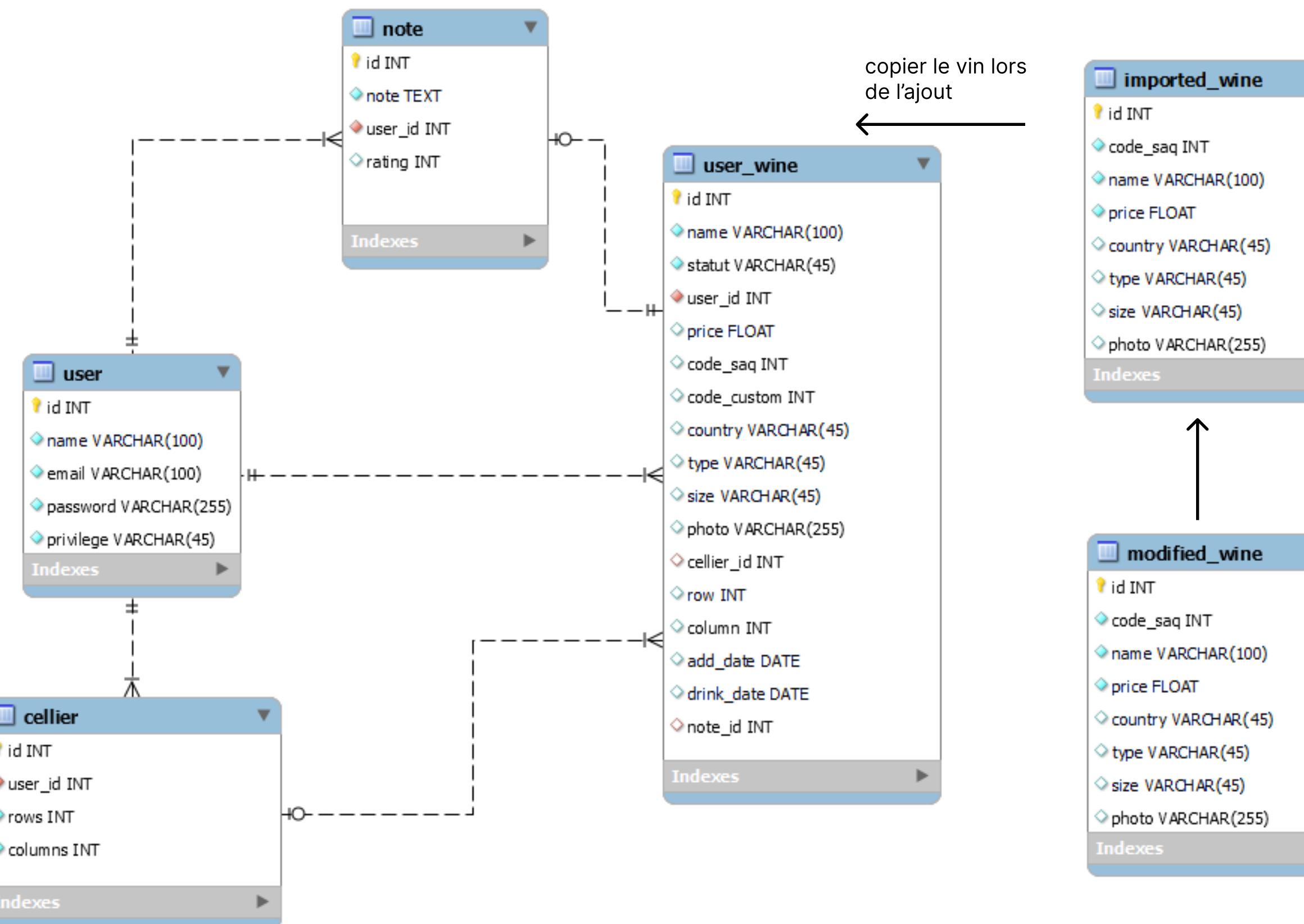
- Chaque item peut avoir un code_saq si importé(unique)
- Chaque item peut avoir un code_custom si ajouté manuellement(auto-increment+unique)

Cellier

- Les champs rows et columns du cellier et row et column du user_wine sont seulement applicables si interface graphique
- calcul total de bouteille == rows*columns

La liste d'achat

la liste d'achat est créée selon le statut des user_wine, to_buy.



Exemple de queries:

premiere fois que vous avez bu ce vin → SELECT * FROM user_wine WHERE user_id == "" AND code_saq == "" ORDER BY drink_date FIRST

associer une note lors de l'ajout d'un vin → SELECT * FROM user_wine_has_note INNER JOIN [...] WHERE USER_ID == "" AND code_saq == "" → COUNT!= 0 déjà bu

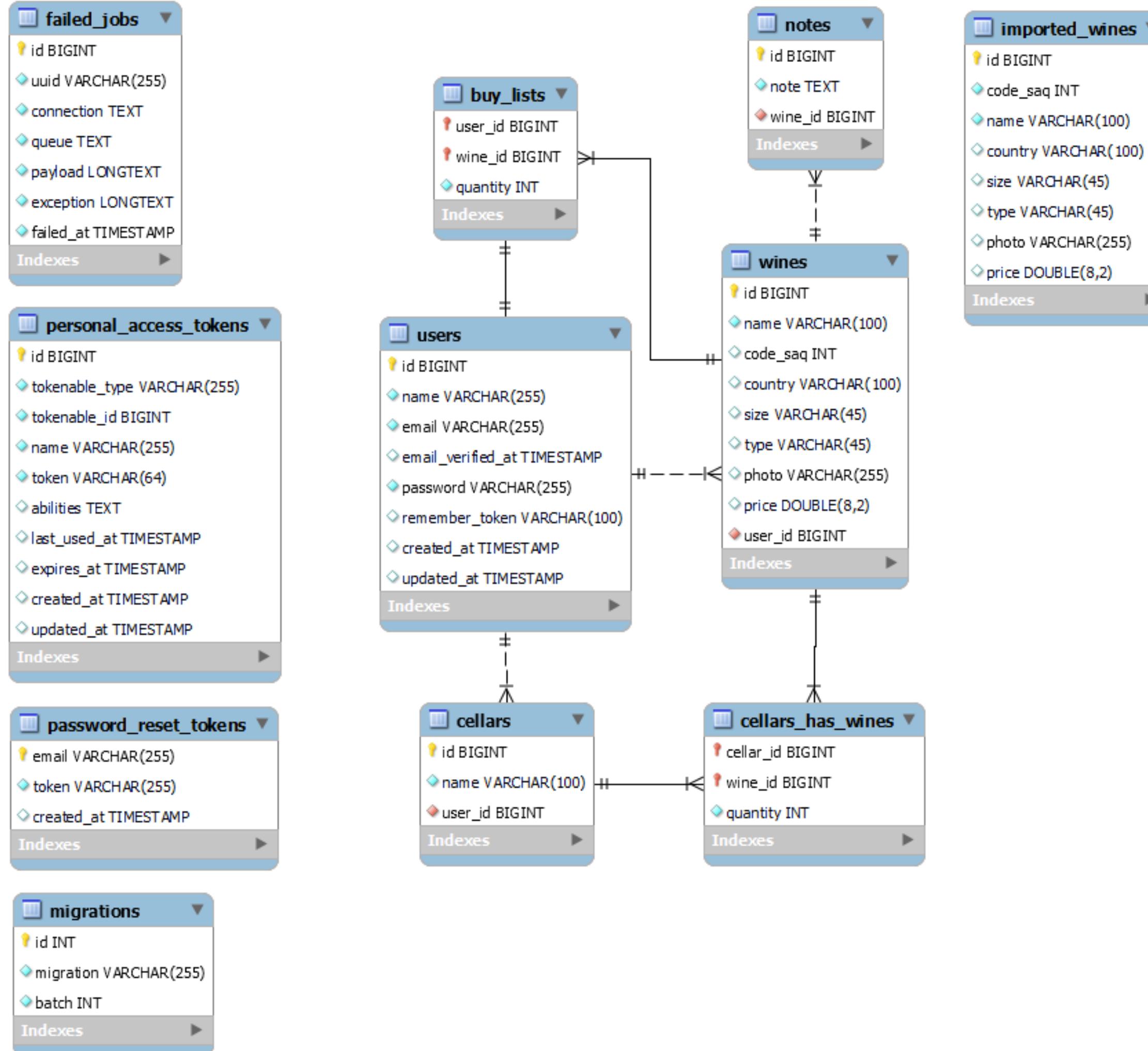
afficher la liste d'achat → SELECT * FROM user_wine WHERE user_id == "" AND statut == 'to_buy'

afficher les vin customs → SELECT * FROM user_wine WHERE user_id == "" AND code_custom == NOT NULL

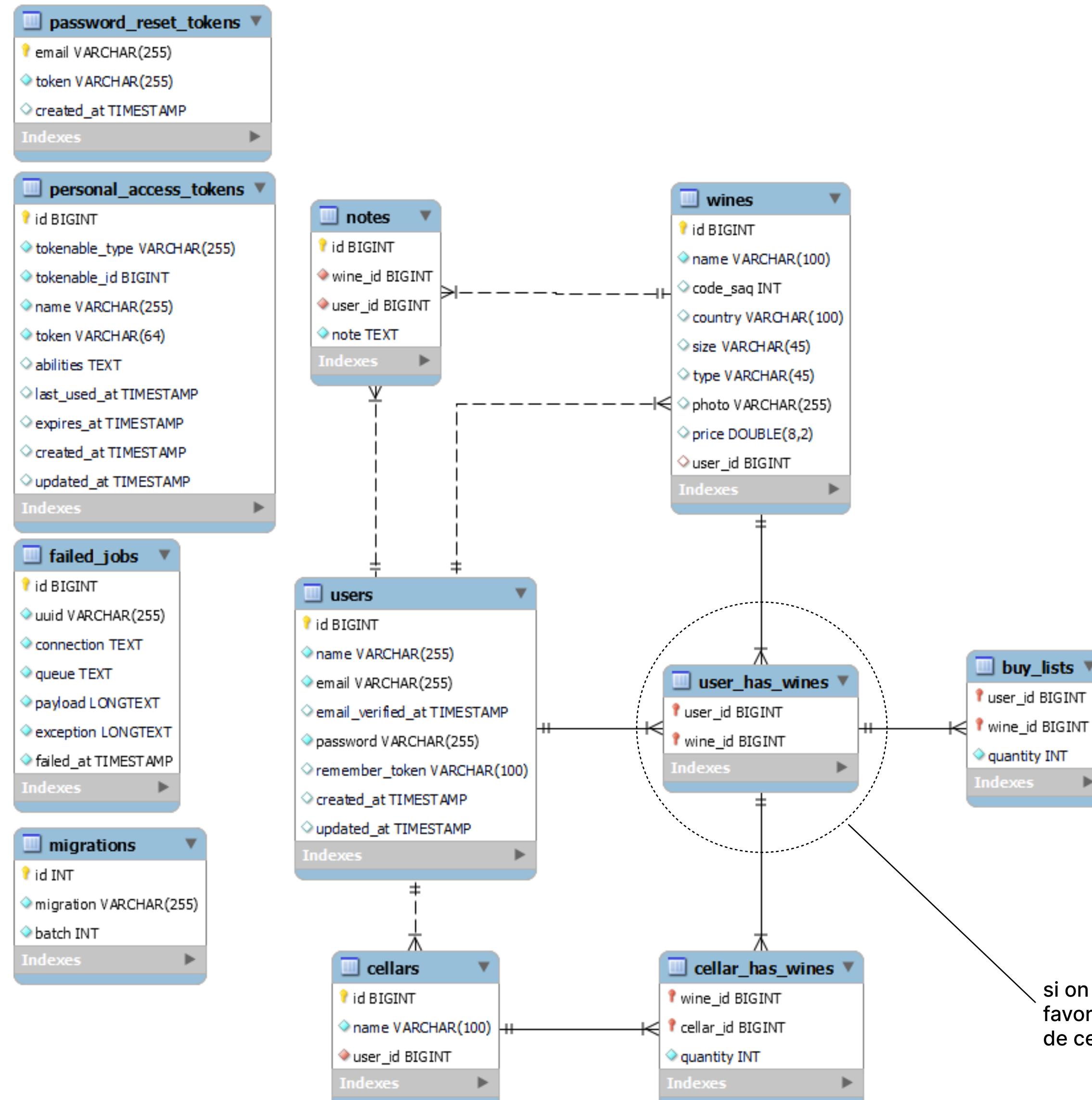
exemple de cas d'usage usant du code_saq ou code_custom

- l'utilisateur rajoute un vin qu'il a déjà noté, on lui propose de copier la note à ce vin
- l'utilisateur rajoute un vin custom avec un nom qu'il a déjà rajouté, on lui propose de copier ce vin
- l'utilisateur rajoute un vin qu'il a modifié par le passé, on lui propose de cloner la version modifiée ou l'original

Modèle comme discuté ce matin/ bon modèle , facilite les requêtes mais crée énormément de données



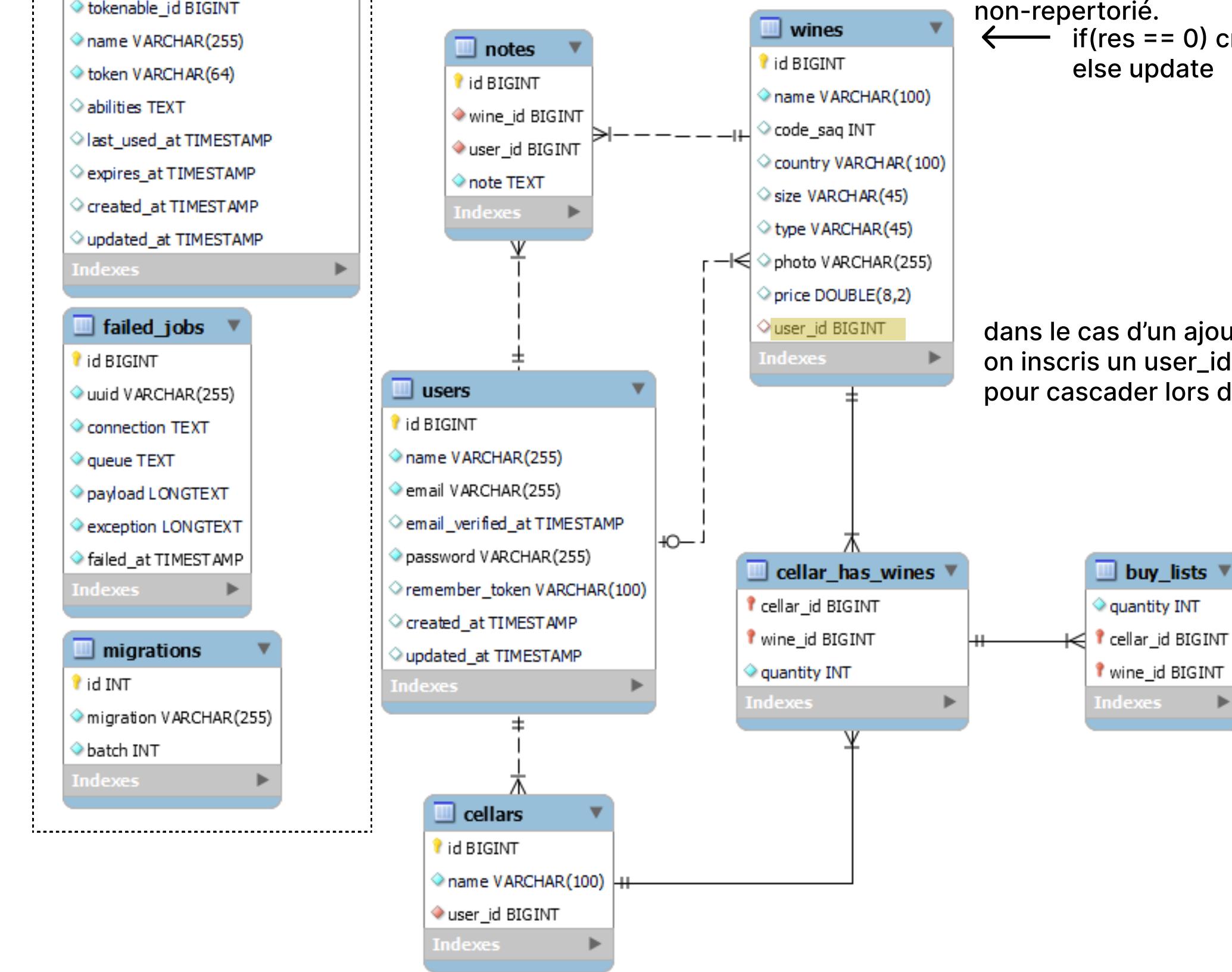
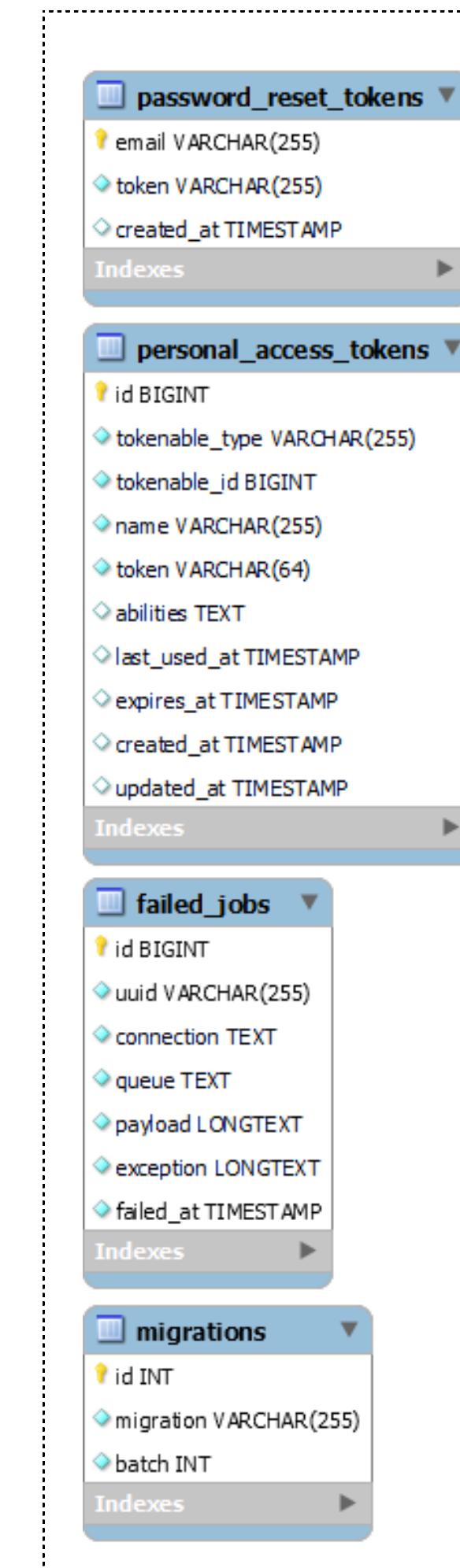
Modèle 3A



si on utilise cette table, il pourrait être plus facile de repérer des vins favoris ou d'autres stats, c'est je pense l'argument le plus concret en faveur de cette structure

- la table notes pourrait avoir un champ 'public' si l'utilisateur veut rendre son commentaire public.
- certaines tables pourront aussi avoir un timestamps, si on veut rajouter des fonctionnalités liés au temps

tables systèmes

Script d'importation vers cette table

Update les données à partir du code_saq ou ajoute donnée non-repertorié.
 ← if(res == 0) create
 else update

dans le cas d'un ajout de bouteille custom,
 on inscrit un user_id pour qu'il soit le seul à pouvoir la voir et
 pour cascader lors de la suppression d'un user

OK, je te laisse regarder et si on veut ajuster demain, on en parleras

Je pense que c'est beaucoup mieux maintenant. :)



