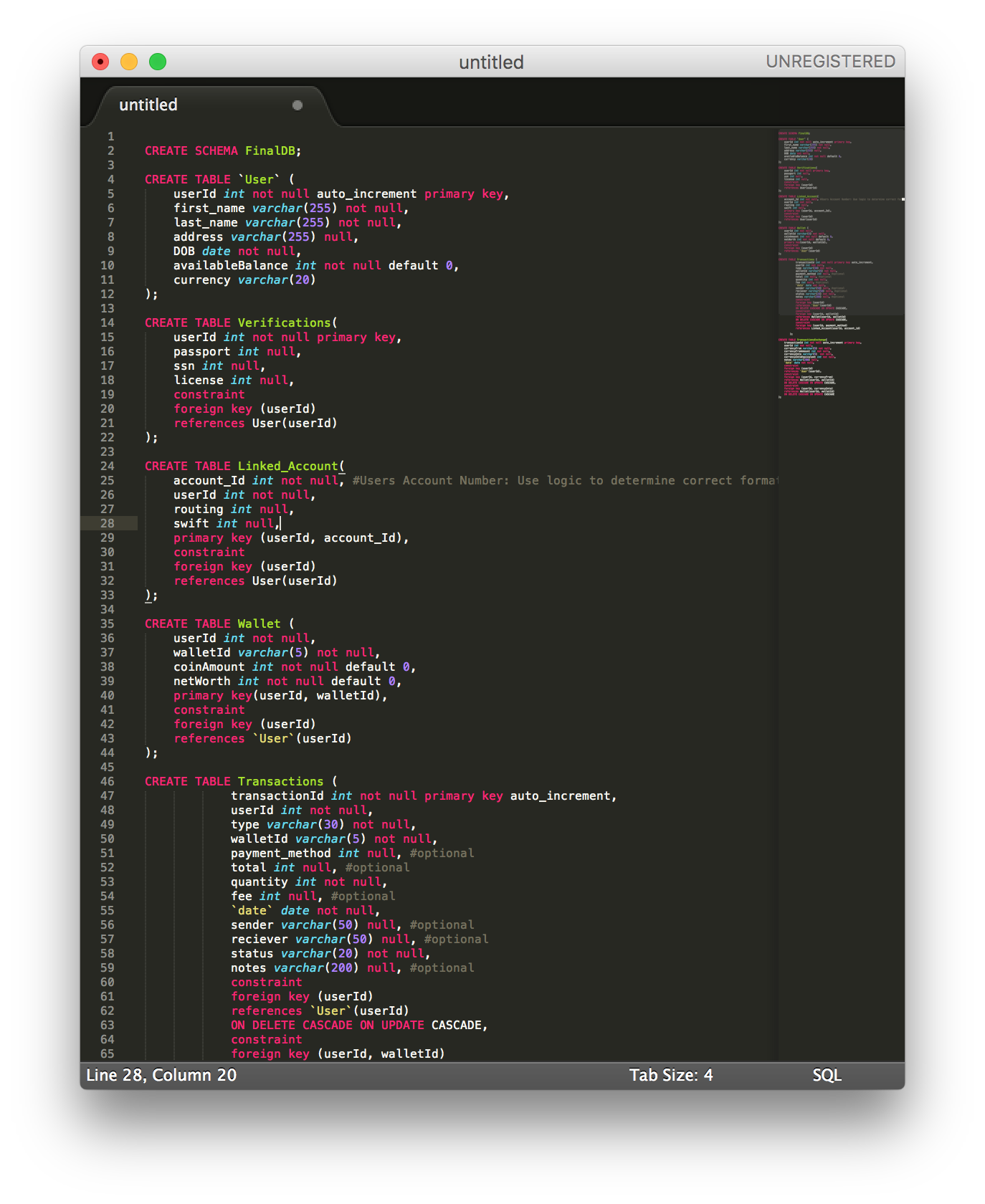
Ian Hoyos

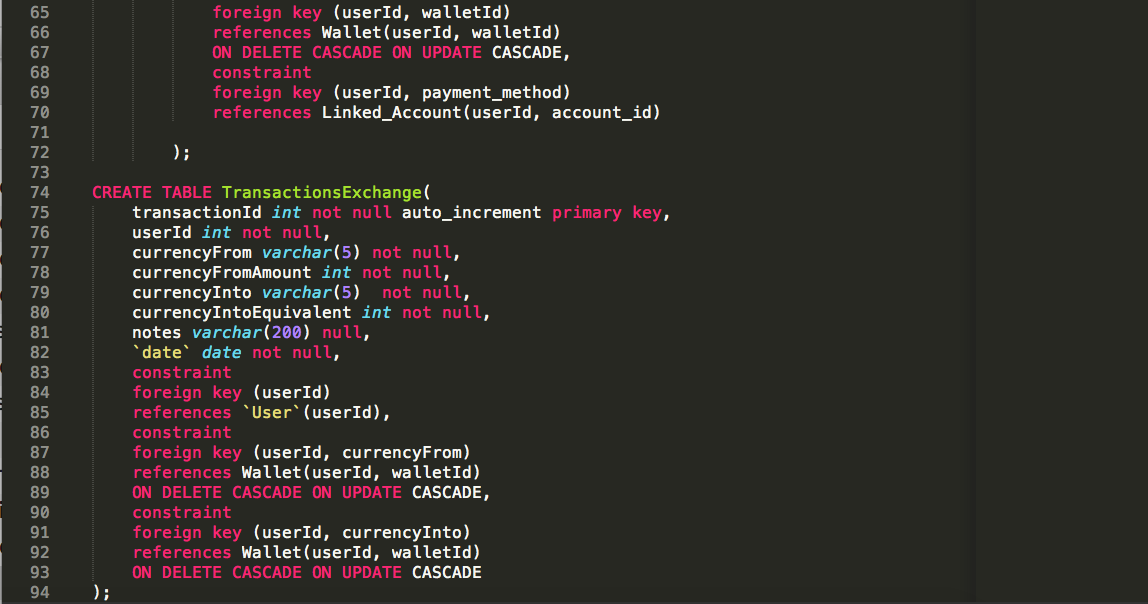
CSC-423

Prof. Sayed

FINAL EXAM

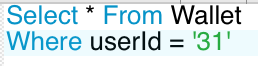


­­­



For the creation of the database I assume a centralized role of the user, in which the userId is the primary field for identification of almost every resource pertinent to the user. Every user, can have up to three wallets, one for each currency supported by my hypothetical platform. The three currencies are BTC (BitCoin), LTC (LiteCoin), and ETH (Ethereum).

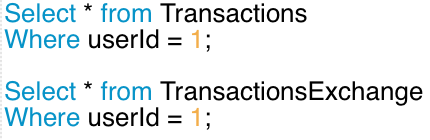
I decided to give userId a centralized role as the primary key of many tables because it simplifies the query process, when requesting data such as all wallets held by user,



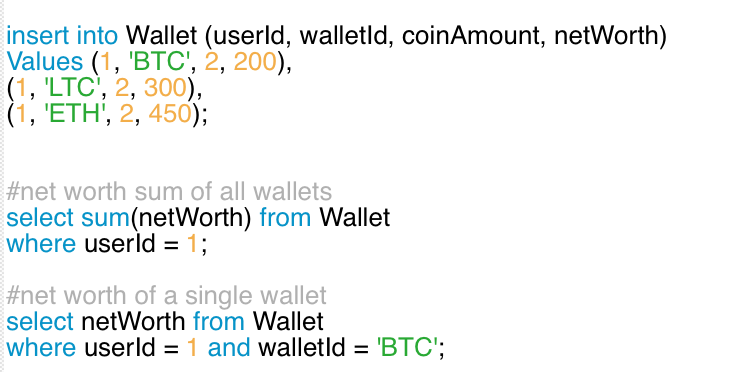
The Wallets table is in charge of wallet definition, by applying a walletId and userId as a composite, (walletId input would be limited to ‘BTC’, ‘LTC’, and ‘ETH’), currency specific wallets can be made uniquely for each user. A lot of the functionality for this platform I would implement on the backend. Things like wallet limit would be handled in a simple if condition statement checking for previous instantiation of a currency wallet, this would handle the limiting one wallet per currency per user.

I attempted to integrate several different transaction schemas, but ultimately centralizing all transactions into one table deemed to be the most efficient for both querying and storage capacity/efficiency.

From this table design all queries per user can be made simply and easily. This same design decision simplifies the process of creating the IRS documents. All transactions can be compiled with a simple,

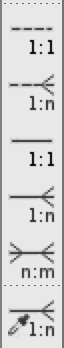
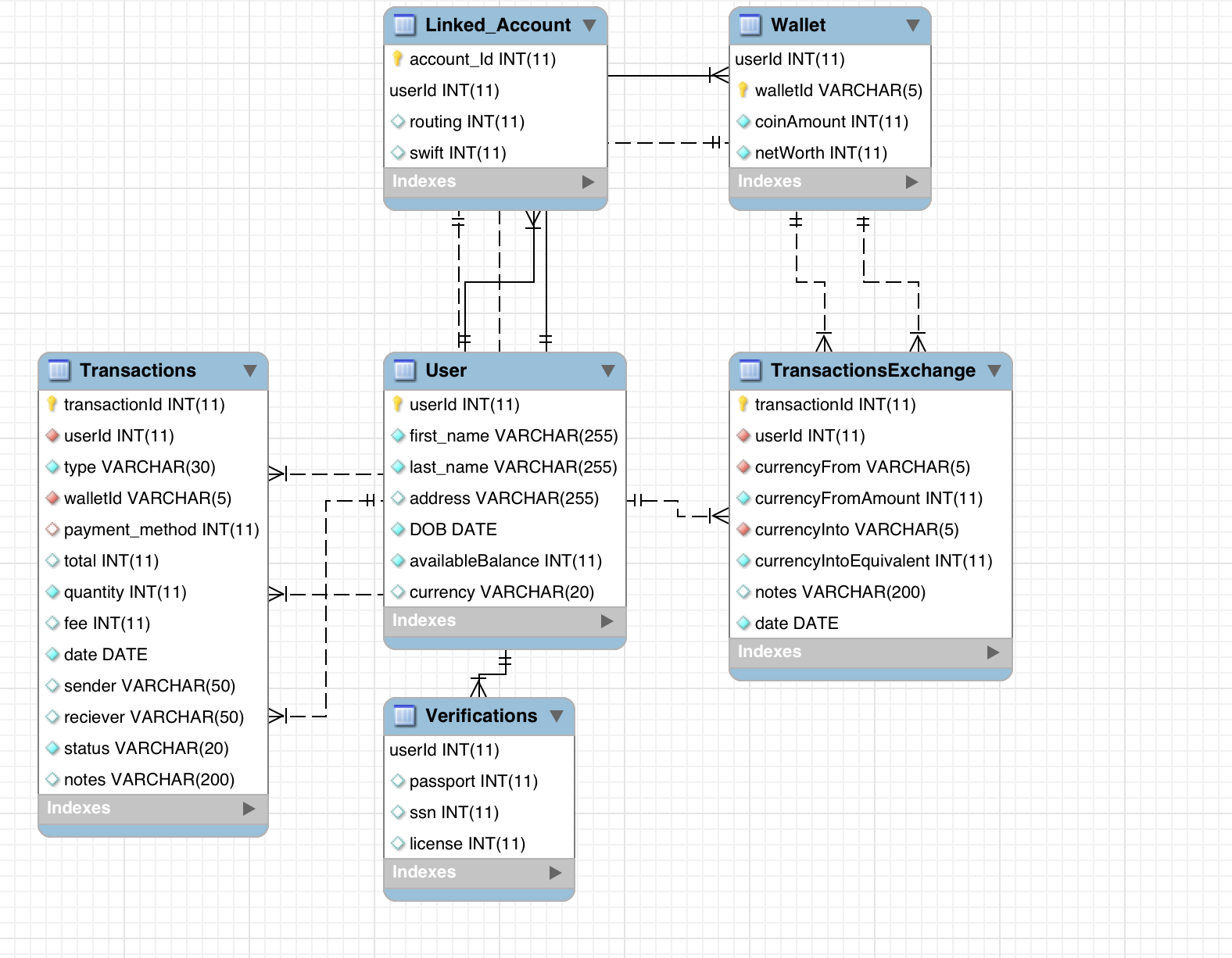


The decision behind separating exchange type transactions from the rest comes from the decision of separating internal and external transactions. An exchange is more of an internal transaction where the only numerical change is in coin quantity. No real capital change occurs. Thus, the need to log it with external (capital related) transactions seemed unnecessary/inconsistent.

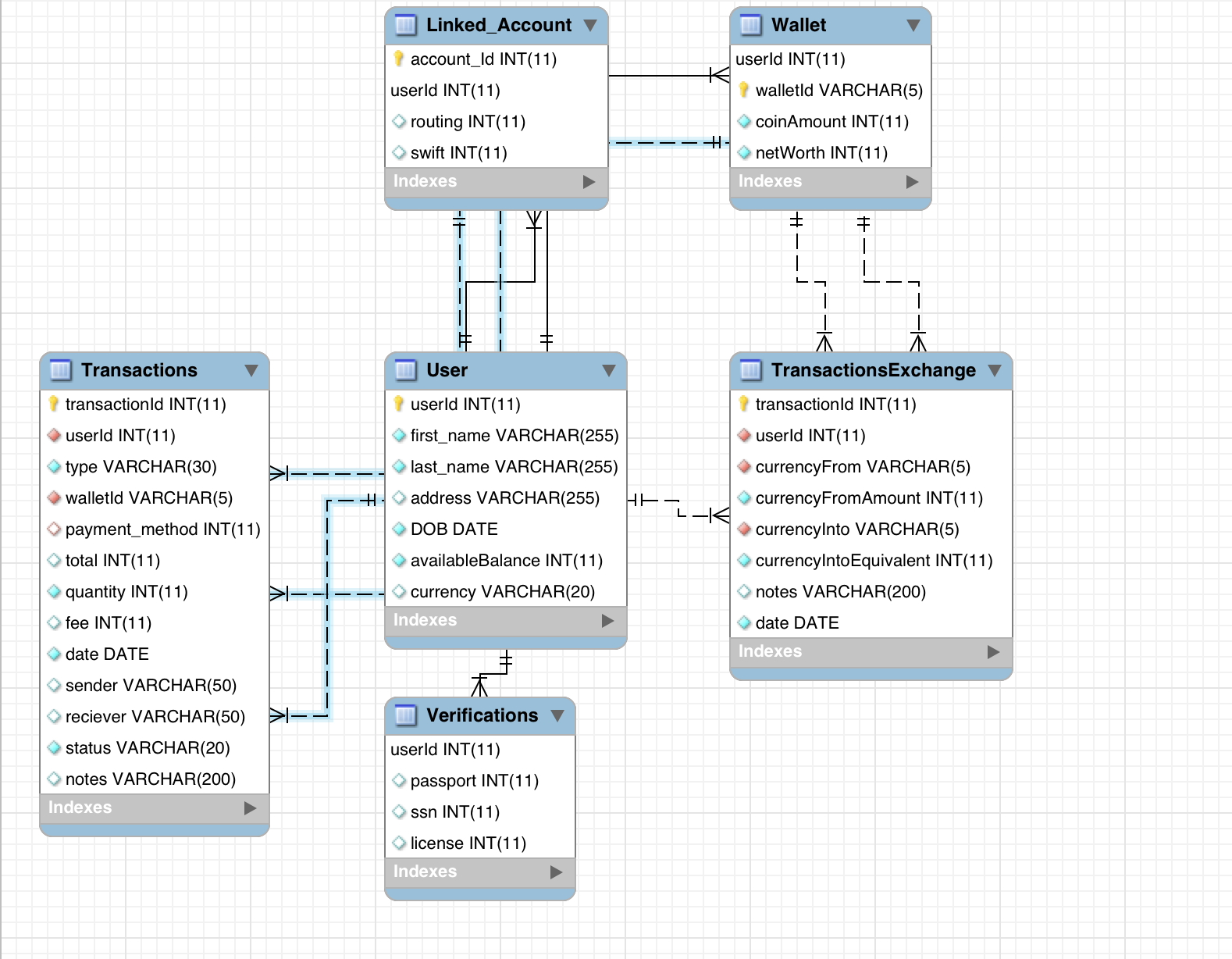


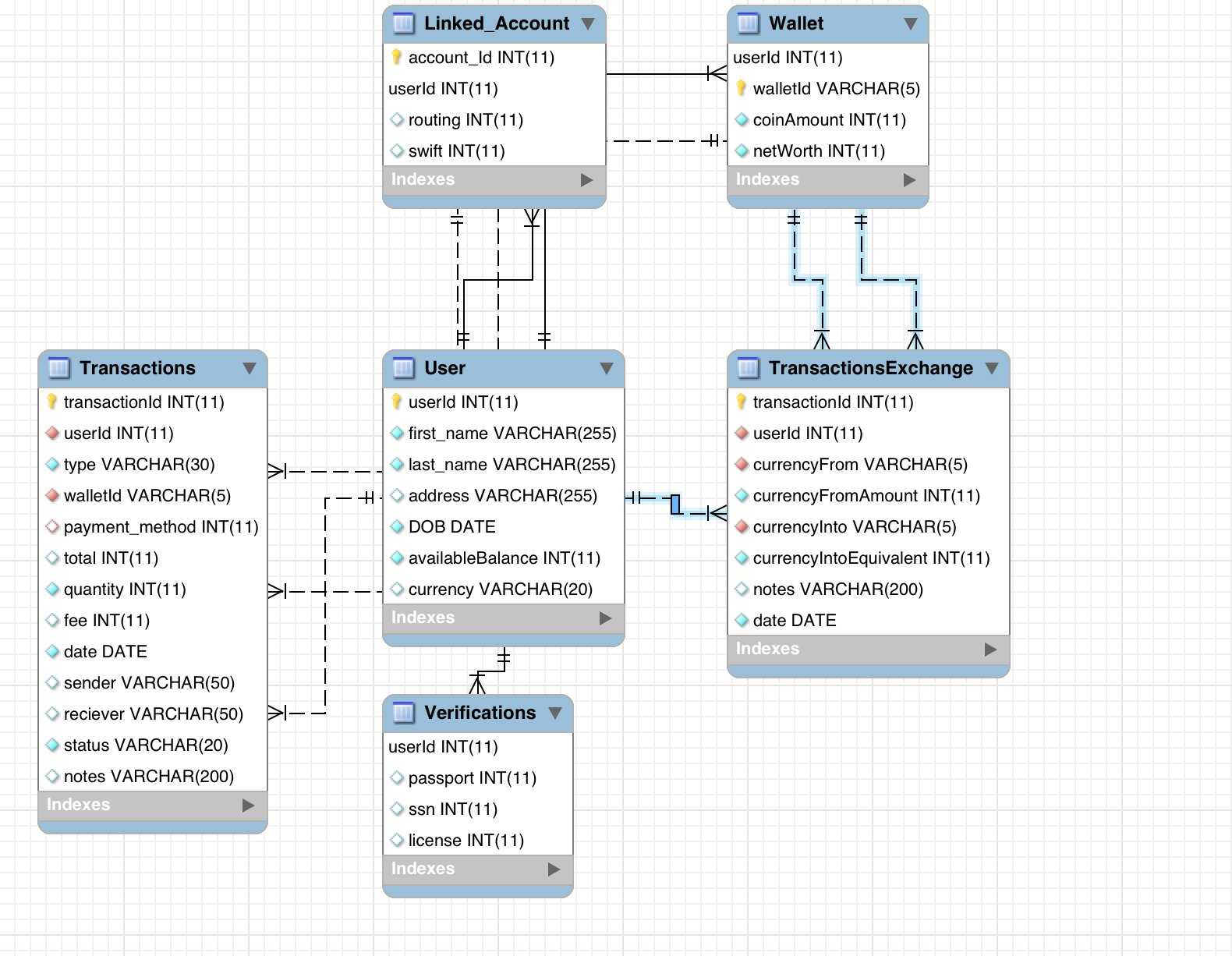
Basic creation query for a user’s wallets, all tied together under same userId. Above are also examples of how wallet balances would be displayed.

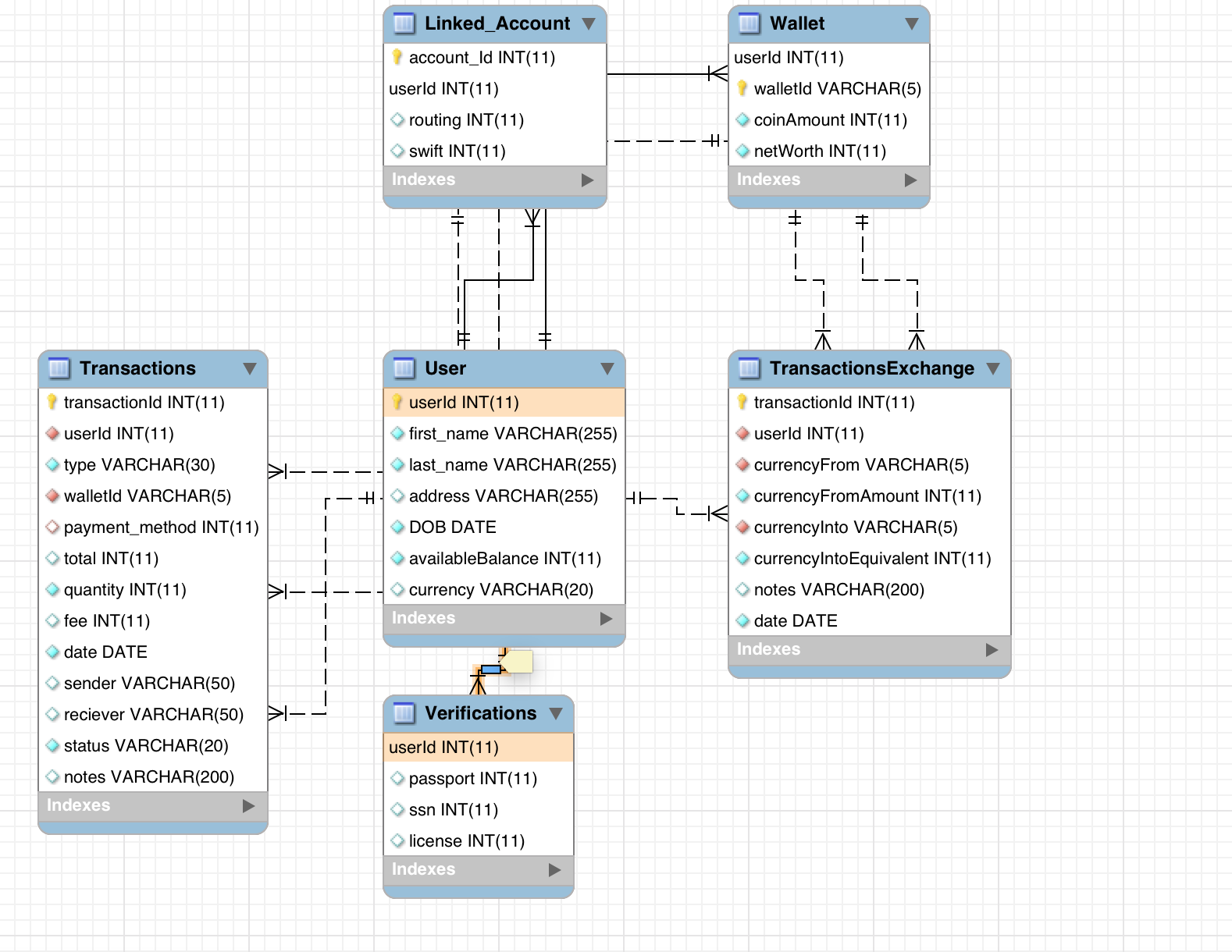
Sample queries for pulling up all transactions per wallet:

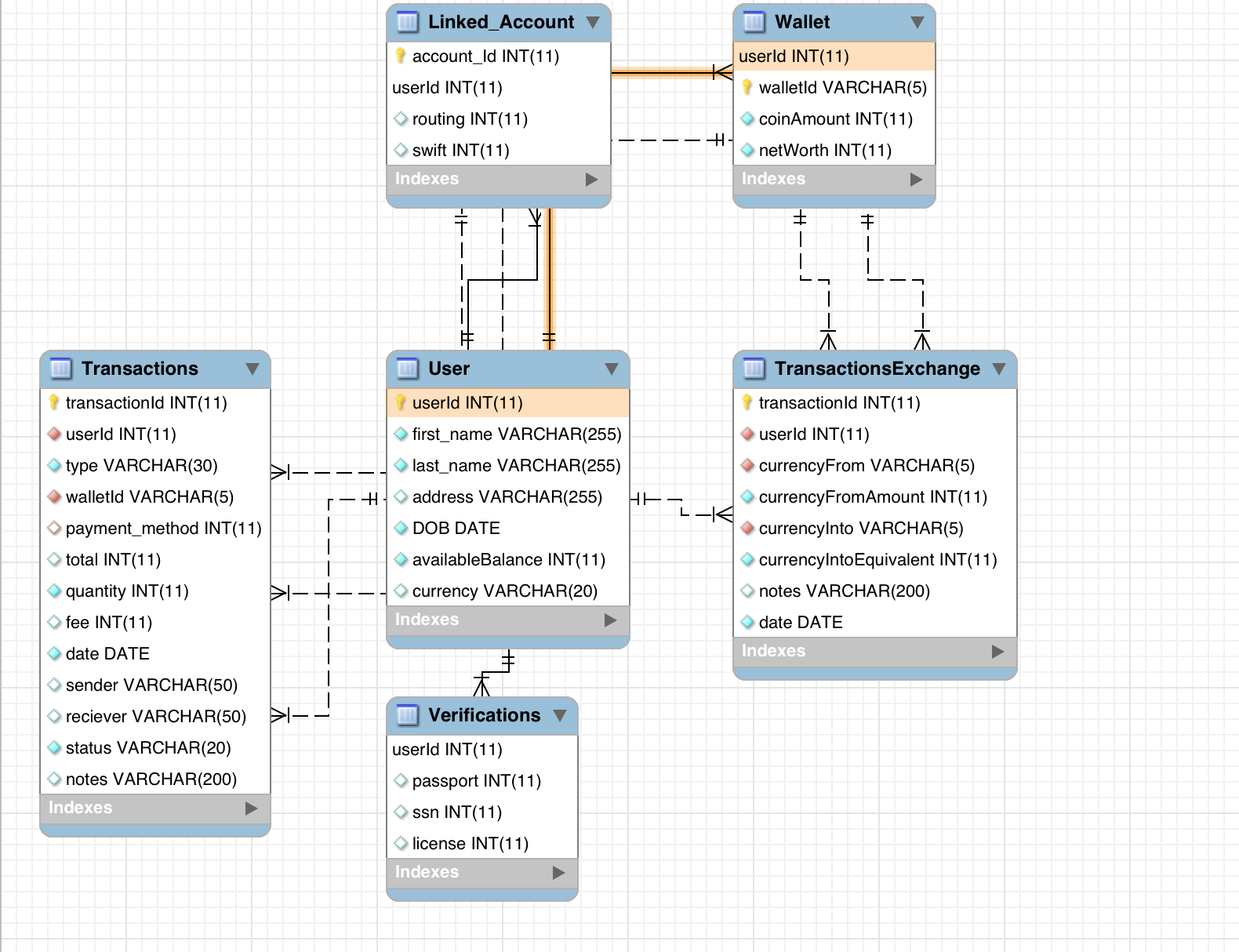
******EER Diagram:**

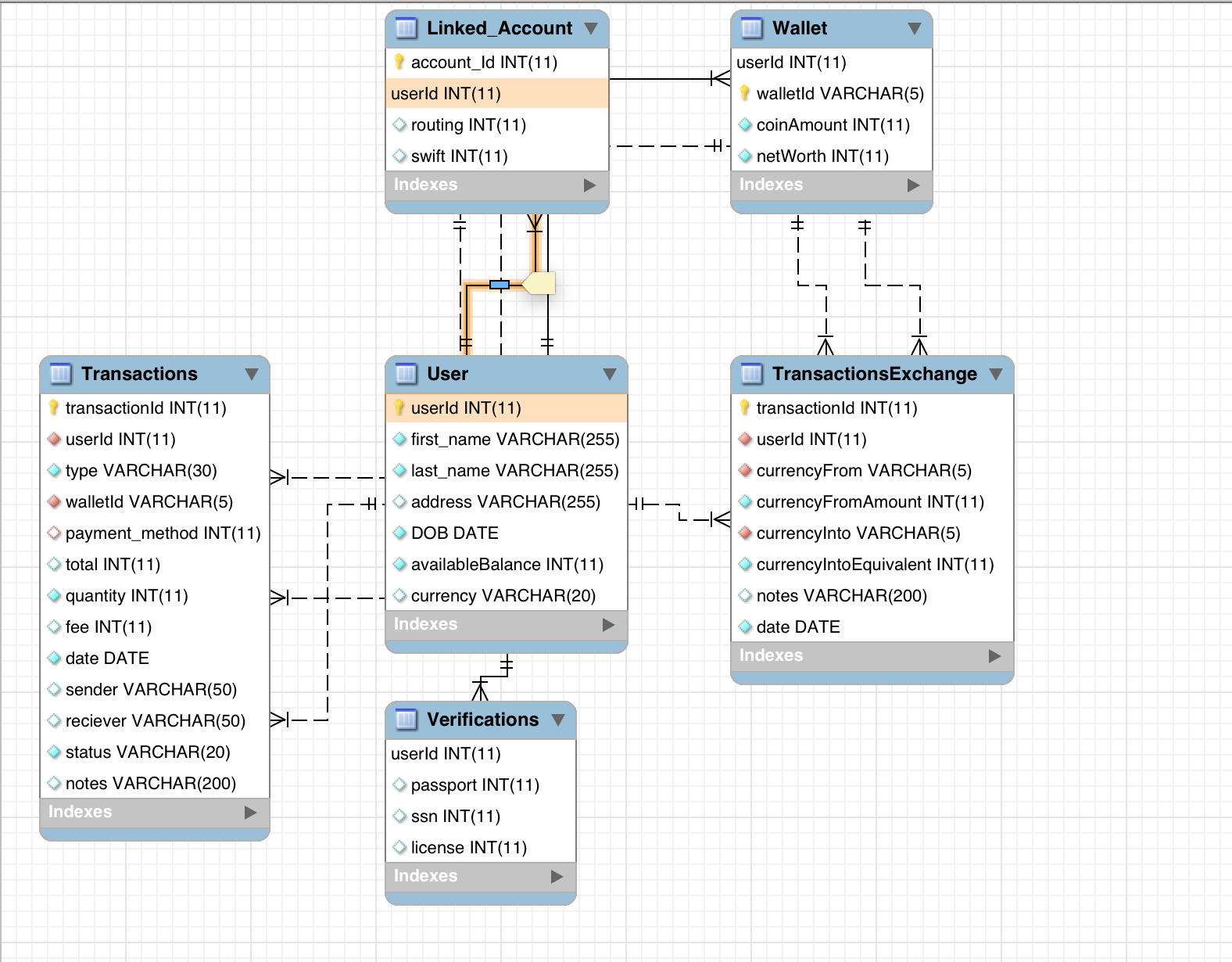
**Relationships:**







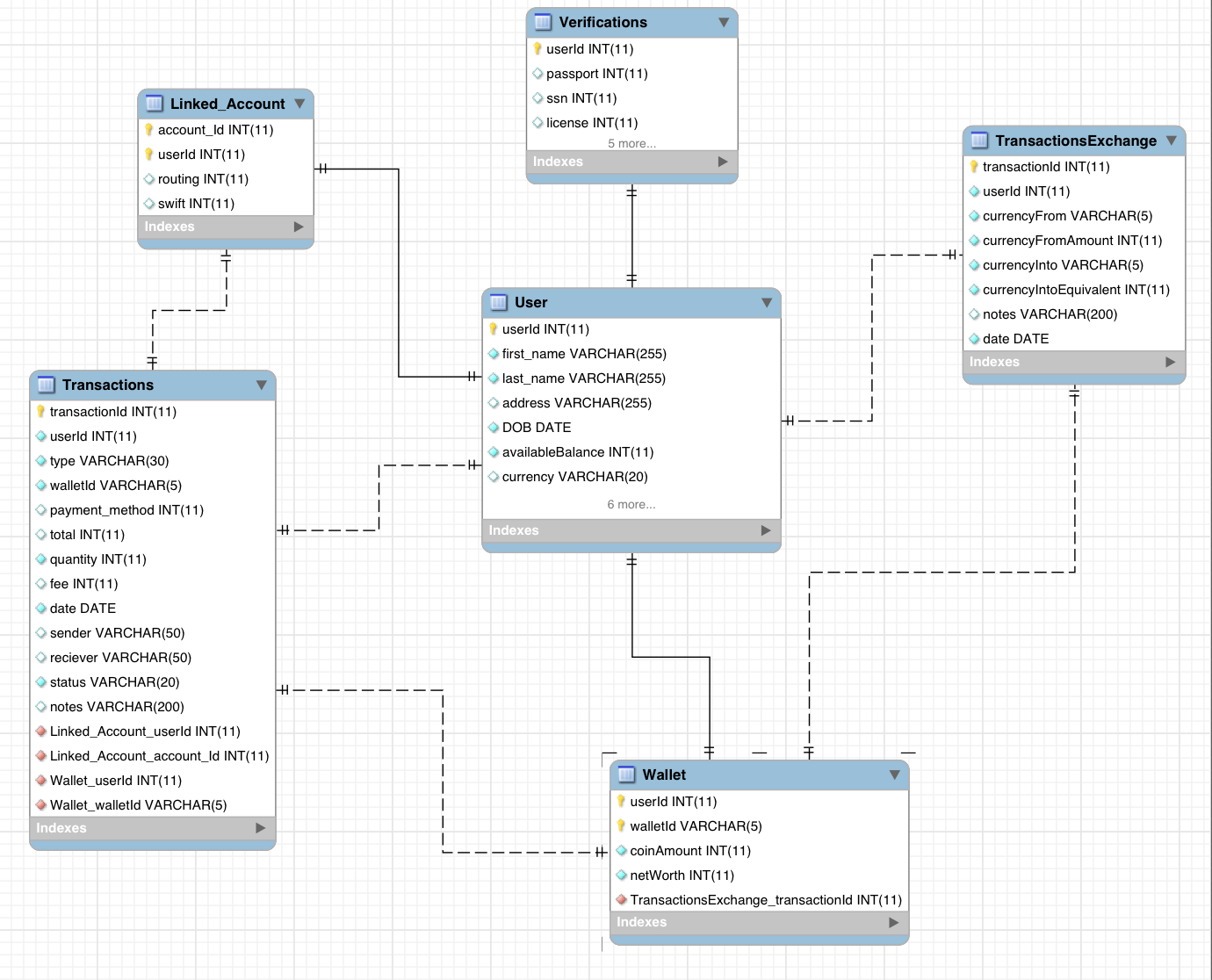




**Identifying/Non-Identifying Relationship Diagram:**

Solid Line – Identifying

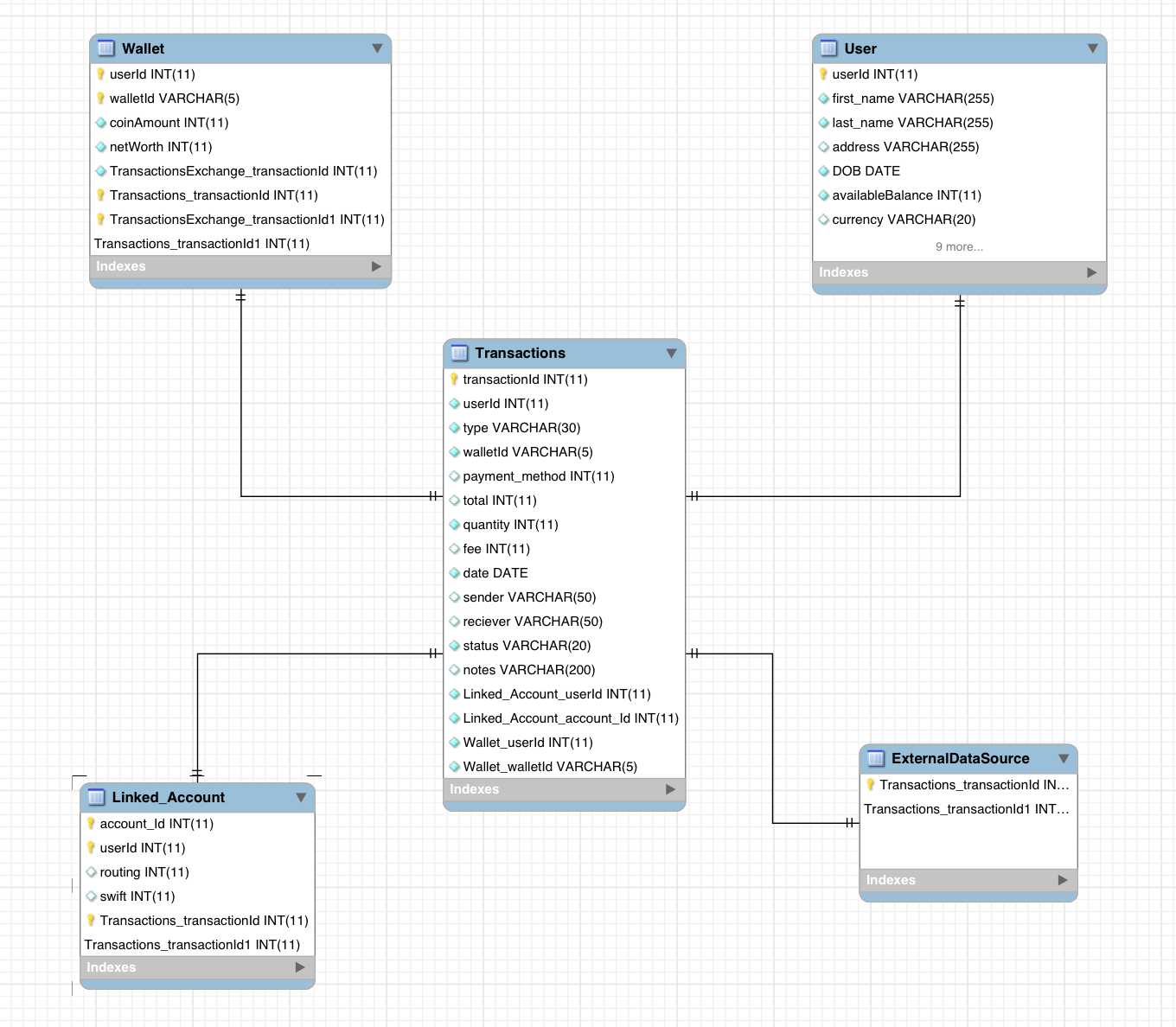
Dashed Line – Non Identifying

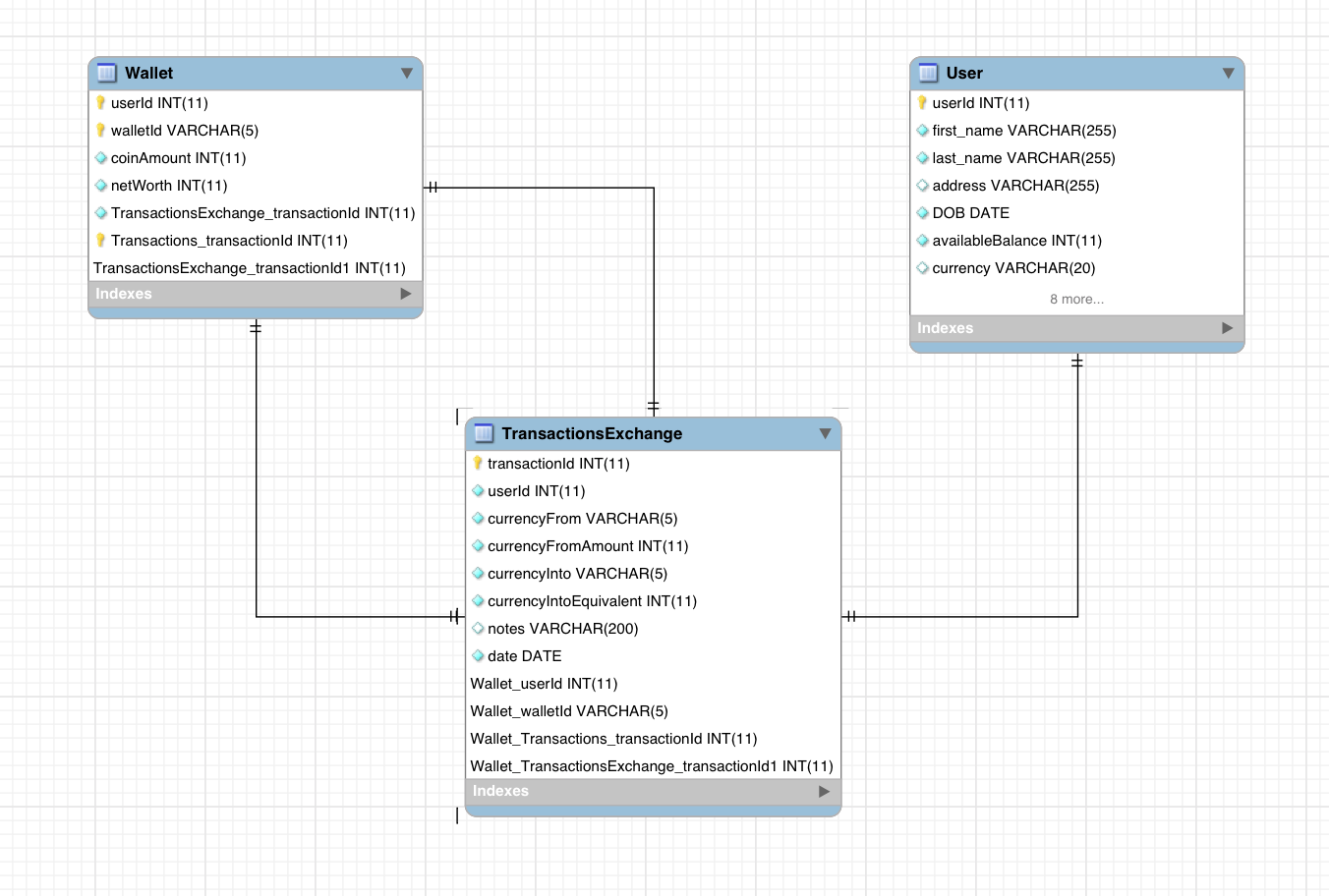
****

**Star Schema:**

I understand the concept of the star schema. A composition of low dimension tables each representing an aspect of a higher dimension relation. The higher dimensional object here is my transactions table, gathering information from lower dimensional objects (user, wallet, accounts). I also understand from the examples we observed in class that these low dimensional tables are typically simple single PK tables. The problem is that, for uniqueness/identification purposes, I can’t seem to find a way to split the composite keys in the Linked\_Account and Wallet tables without compromising the integrity of the data on application.

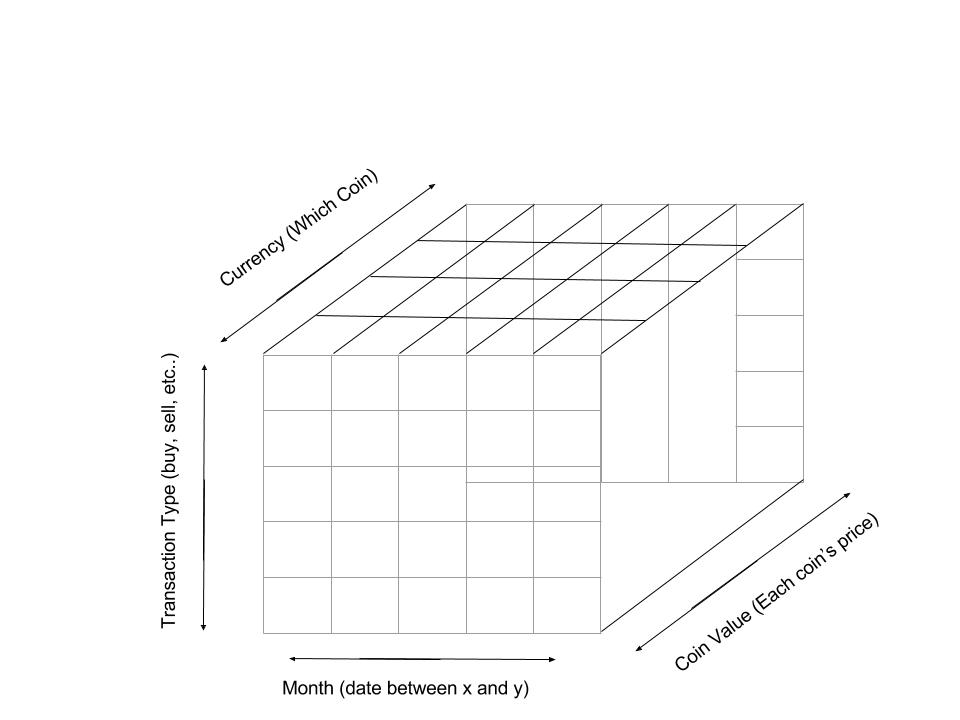
The same scenarios plays out for the TransactionExchange table.





As you can see in the first diagram, there is an ‘ExternalDataSource’ table, this table serves as a placeholder for integration with external data sources, as platforms like ‘Coinbase’ and others use for updating the latest pricing information. This is simply a placeholder, but it serves to show the possible implementation of not just one but many external data sources, whose information can be adapted to the transactions as needed.

**Business Intelligence Solution Questions:**

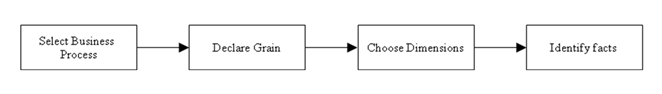


This cube design allows me to answer business intelligence question about purchasing patterns. Allows us analyze and produce predictive data, which can help pricing predictions, and can be offer as a service to the platform clients.

Questions answered by cube:

* + - 1. Which coin has most sell transactions this year?
      2. Which coin has most buys this year?
      3. Which coin was most sold in September.
      4. Which coin was most bought in December.
      5. How did people react to X event in June?
      6. Which month had most trades, what happened?
      7. Which coin was exchanged most?
      8. Which coin sells at the highest price?
      9. What month are people more likely to Buy in?
      10. Which months saw greatest price increase?
      11. Which coin has highest sustained price?
      12. Which coin fluctuates most?

Etc..



**The Business Process**

The business process here is the transactional aspect of the platform (The main function of the platform). Although this platform only serves as a staging area for cryptocurrency transactions, it can also serve as a data mining platform, data that can be used for sales predictions (for example).

**The Grain**

The fact table will be composed of the transaction types: to track buying and selling patterns as well as other transactional patterns, currency: (the type of coin) to track quantitatively transactional pattern specific to each coin, the value of each coin at time of transaction logging: to track valuation data as the currency evolves/devolves, and finally time: to plot all other data over time spans and get accurate readings on transactional patterns specific to each coin.

**Dimensions**

As stated above, 4 dimensions will be used, transaction type, currency, price, and time.

**Facts**The above questions serve as examples for some facts we might want derived from the data. For this sort of platform an aspect of interest for many is transactional patters (since the price of cryptocurrencies is determined by amount bought by users, similarly is the depreciation calculated).

**Data Mining**

The storage of transactional data is itself a data mining solution, and consequently one to be exploited. Platforms like Coinbase are primary operators of cryptocurrency transactions, their main source of revenue comes from transactional fees, however, they collect tons of transactional data that can be mined for cryptocurrency predictive data. Just like department store sales records are used to mine for purchasing patterns, and are quantitative factors that drive a company’s stock, similarly does Coinbase’s (using Coinbase as an example here) transactional data. Their transactional data serves a quantitative purpose in helping drive a coin price by either influencing or motivating users to buy or sell (for example).

The cube above offers an insight to just how much transactional data a platform like Coinbase collects, and for the purposes of this assignment I took a more simplistic approach, even more dimension could be added to the model (such as which countries led in transactions, for example) that would make the data even more complex and tantalizing.