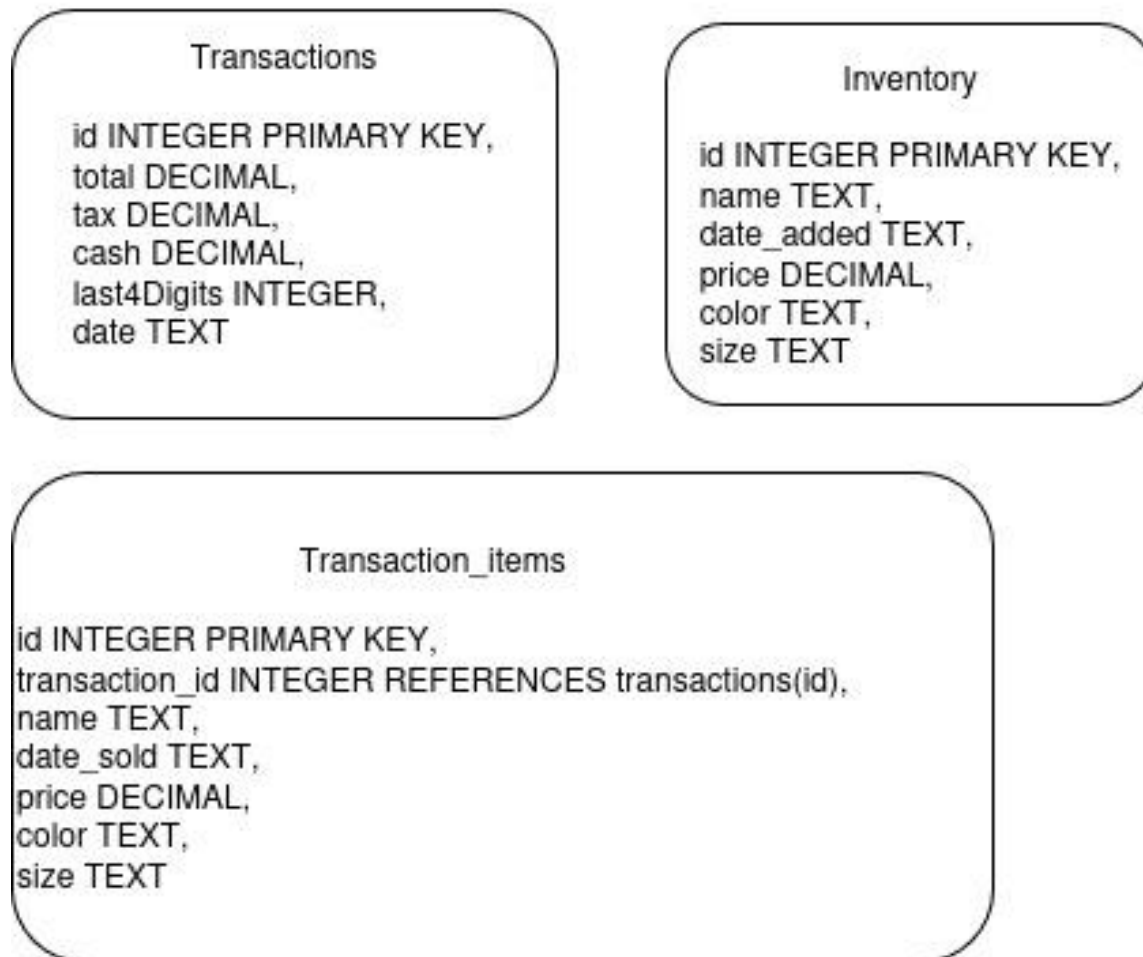


## MySQL



### Technology Choice

Since we are working with items and a straightforward collection of items, a relational database with SQL tables works well to store each item as a row. Therefore, we chose a SQL database hosted by MySQL due to its enterprise security and our need for safe storage due to holding transaction history.

### Tables

There is one database holding both our inventory items and our transactions. Each item is held in inventory or in transaction\_items, so that should a refund be issued, the item can be added

back to the inventory. Our transactions table holds information about each transaction with an id that can be referenced by each sold item for querying.

### **Potential Tradeoffs**

One of the tradeoffs for using SQL is that unlike a NoSQL database where each item can be freely moved between inventory and transactions, a few extra queries will need to be upheld in order to handle the transaction and refund logic. However, our sql queries for certain attributes of items will be much less complex than a NoSQL database when querying our inventory and sold items.

### **Possible Alternatives**

Should this system be migrated to a NoSQL approach, each item would have the same attributes so that it could easily be transferred between inventory and different transactions. Each transaction will directly have a nested list of each item sold, with its sales data relying on the values of the items in it, and refund fields will be added as needed.