

Rum Business

The Distillery

Setting up all you need to run a rum business

Arnold Abraham - 12. Januar 2021



Proposal

You managed to find yourself in an old & inactive distillery. Bought it from a doubtful man recently. The property deed seemed to be legit. It was for the right amount of money. Almost exactly the same amount you got as a cut-off from padre Luigi when the pirates had taken the rum.

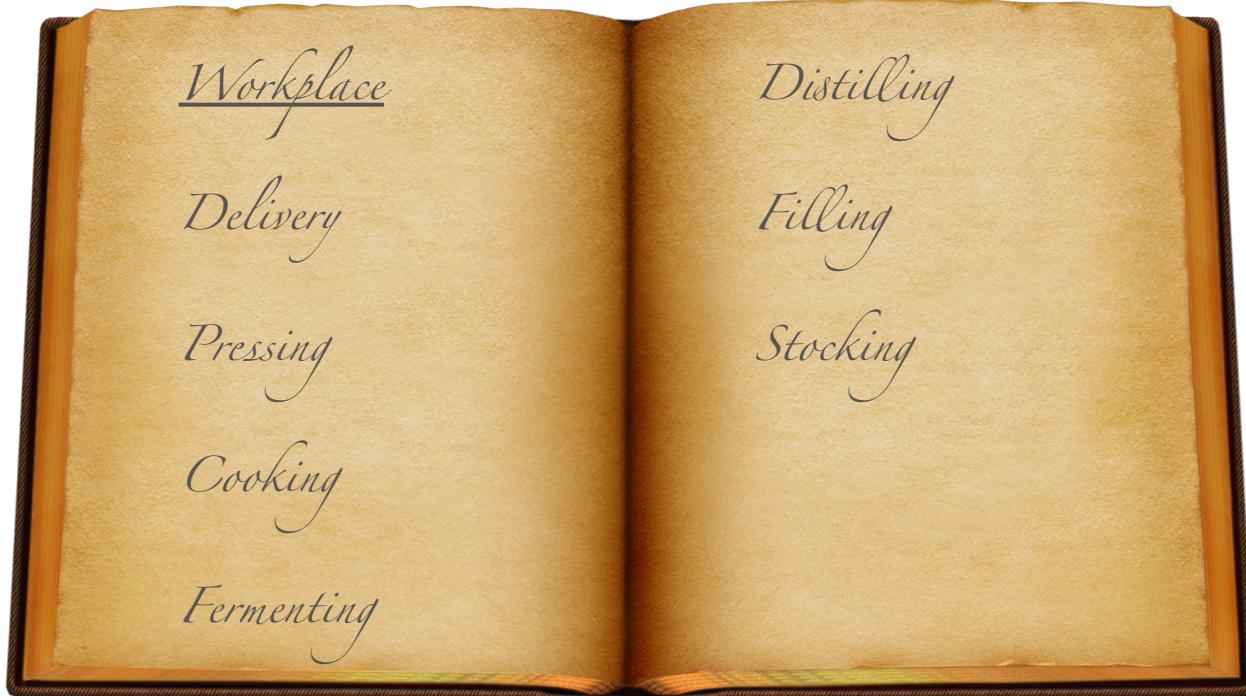


Once you've tested the assets of the facility, you sit down in the director's office and think about your plan how to build up the books (SQL Tables) to successfully run this business while you are on the sea as a true SQL pirate would be!

Task: Designing the Database

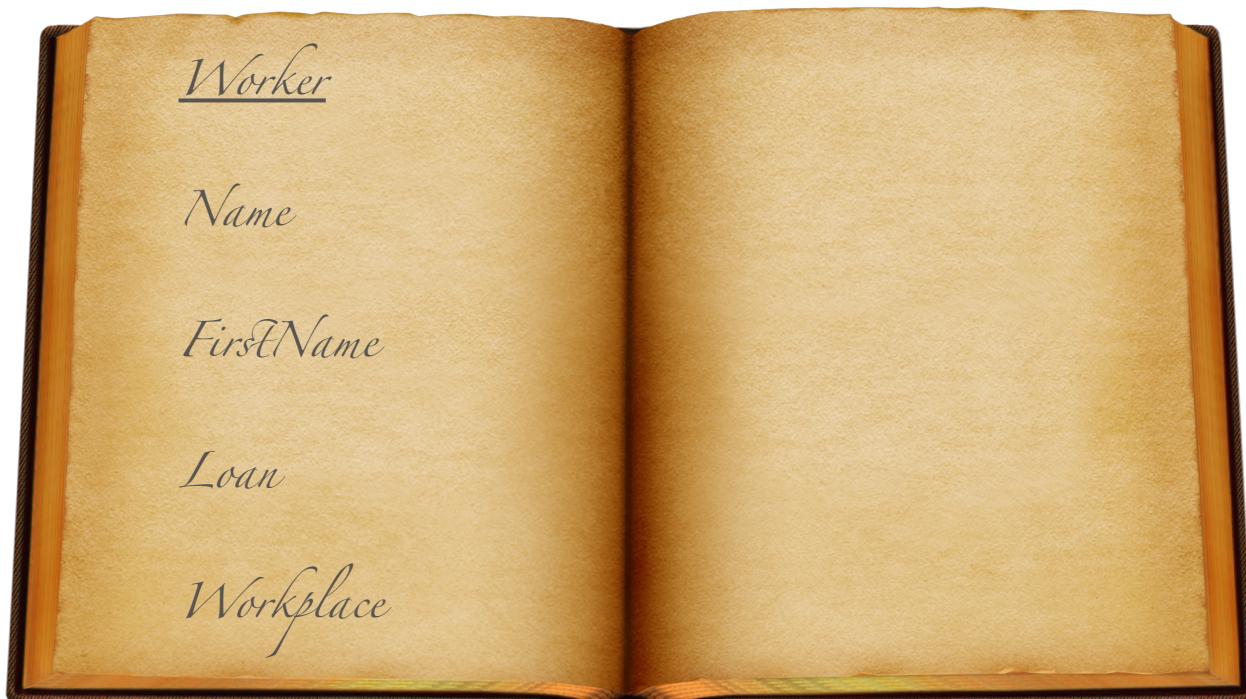
Workplaces

To make rum, you need the sugar bags. You got these already down at Luigis place. Thinking about the steps to make rum in a distillery you come to the conclusion in needing the following jobs to be done:



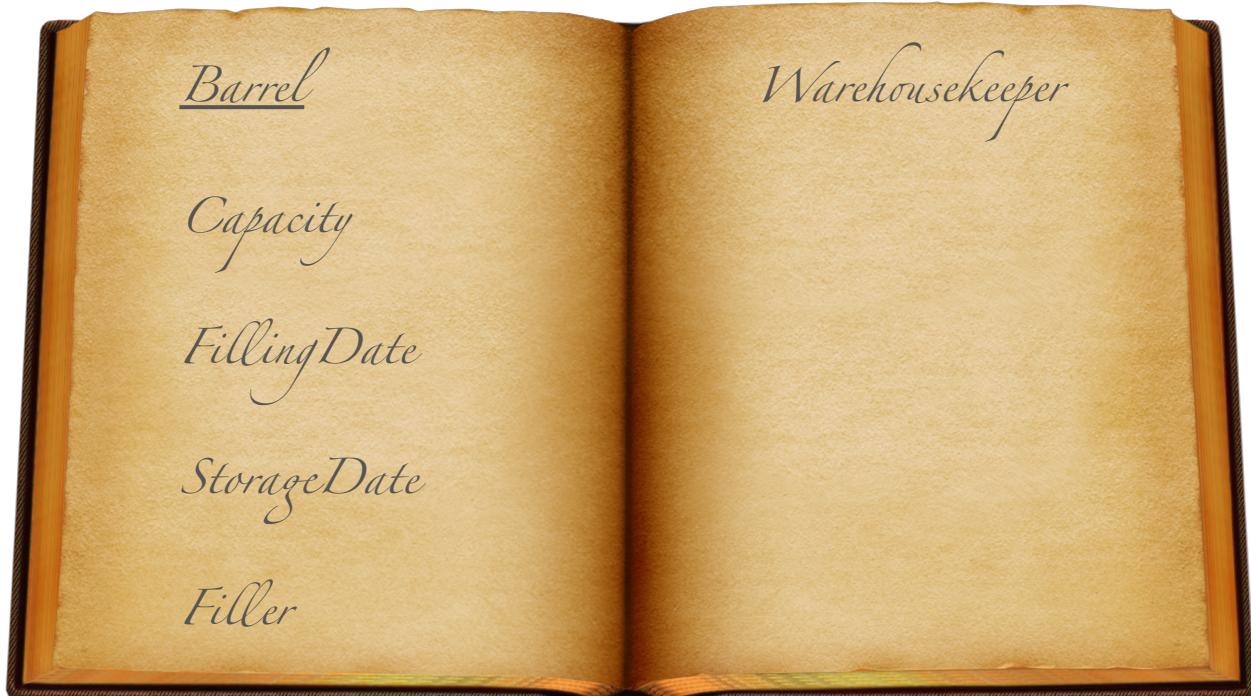
Workers

The workers need almost the same book (table) like the plantation does.



Warehouse

Since you want to sell your rum, you have to ripen the rum until the sale. Normally rum is stored in barrels and then filled into bottles. We only sell barrels. No tiny bottles in our business!



You looked at the three books and because of you are a smart one you detect immediately that you will need at least one more table. Think about the sugar bags and that they can rot. Also think about an extra table for the workplaces (but that is optional).

Task: Bring life into the distillery

I provide a database scheme here, but do yourself a favour: Try the last task on your own at first. Only then you will learn.

I provide the scheme for the next tasks and for the case that you didn't succeed the first task.

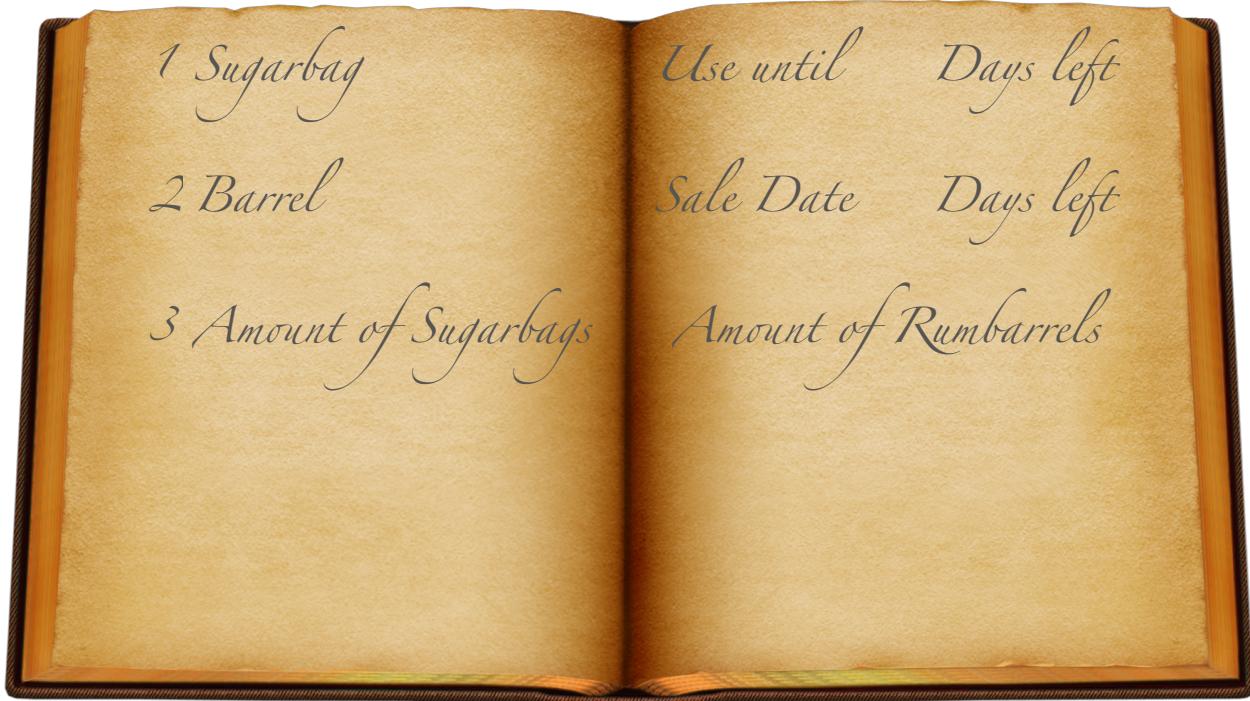
Corresponding files in folder "Interim Solution":

1. Rum Distillery Database Scheme.png
2. Rum Distillery Database Scheme.db
3. 0_Fill_Workplace_Description.sql
4. 1_Fill_Worker.sql
5. 2_Fill_Workplace.sql
6. 3_Fill_Barrel.sql
7. 4_Fill_Sugarbag.sql

Fill the database with the corresponding queries in order to resolve the following tasks. Remember you can't add data to empty tables with constraints. It will end in a constraint failed error. Either use PRAGMA or remove the foreign keys, add data and put in foreign keys again.

Task: Business Books

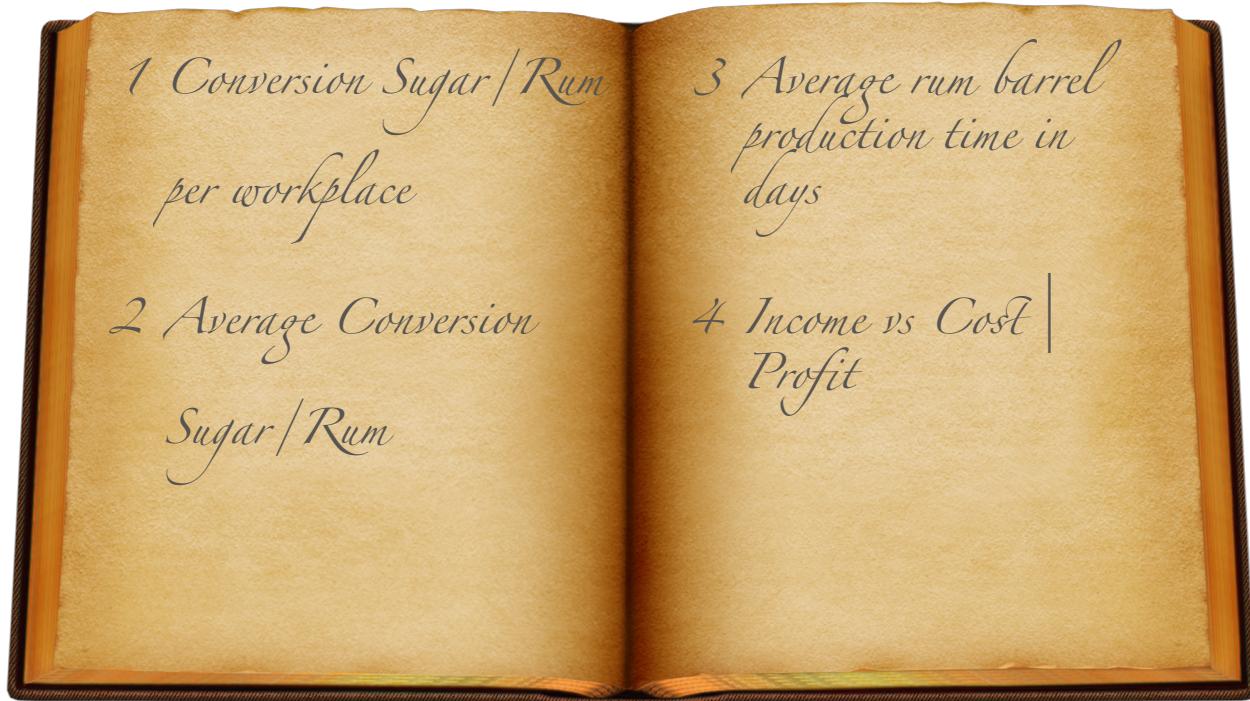
The business books - Wares



Per row one query.

1. List the „use until“ of the sugar bags so that the least freshest is on top. For „Days left“ take the date of the 19th of June 1703.
2. The rum has to be stored for at least 30 days before we can sell it. Make the earliest sellable on top.
3. Count the number of sugarbags in one column and in another one the amount of rum barrels. Clearly spoken: One query result in two columns of two tables.

The business books - Production Process



1. List the Workplace (in clear text, no integers) and the corresponding ratio with a „%“ sign inside the result table.
2. List the conversion ratio of the whole factory. In the format „1 : x“.
3. Calculate the average production time of each barrel of rum. Start with the delivery from the plantation to the distillery. Print the result with the unit days inside the result table column.
4. Calculate the income, the cost, and the estimated profit of rum sales. A barrel brings 303 doubloons and the loan has to be paid also. You found a customer that preordered barrels that are 18 days away from their storage time. Remember each rum barrel has to be stored for 30 days at least. Also, the loan of the workers is due in 18 days. Take as current date the 19th of June 1703.

Tipps

- Don't forget about primary keys (they are not given explicitly by this document, think of them on your own).
- Don't forget about your foreign keys.
- Think about how to find a common description for all work places. What do they all have in common?
- You are able to use a [tutorial](#) or the [official documentation](#) of Sqlite
- When you calculate with dates, google „SQLite julianday“ or the date()-function of SQLite.
- Think about nesting SELECT queries.