Mercantile Ships

Aly Shmahell
aly.shmahell@gmail.com

3rd Year Student
Department of Computer Science, University of L'Aquila
Databases - Lab Module
Prof. Pierluigi Pierini

Jan 9th 2018

Abstract

In this document I will showcase the 3 stages of design of the "Mercantile Ships" Database.
In this effort, and under permission from the requirements file, I made some decisions regarding the techniques and tools used in the design process.
In some extreme cases I had to settle for less than optimal solution either for conformity purposes (the solution conforms best to the design stage) or for practicality, or for highlighting the optimal solution.

Design and Tool Choices

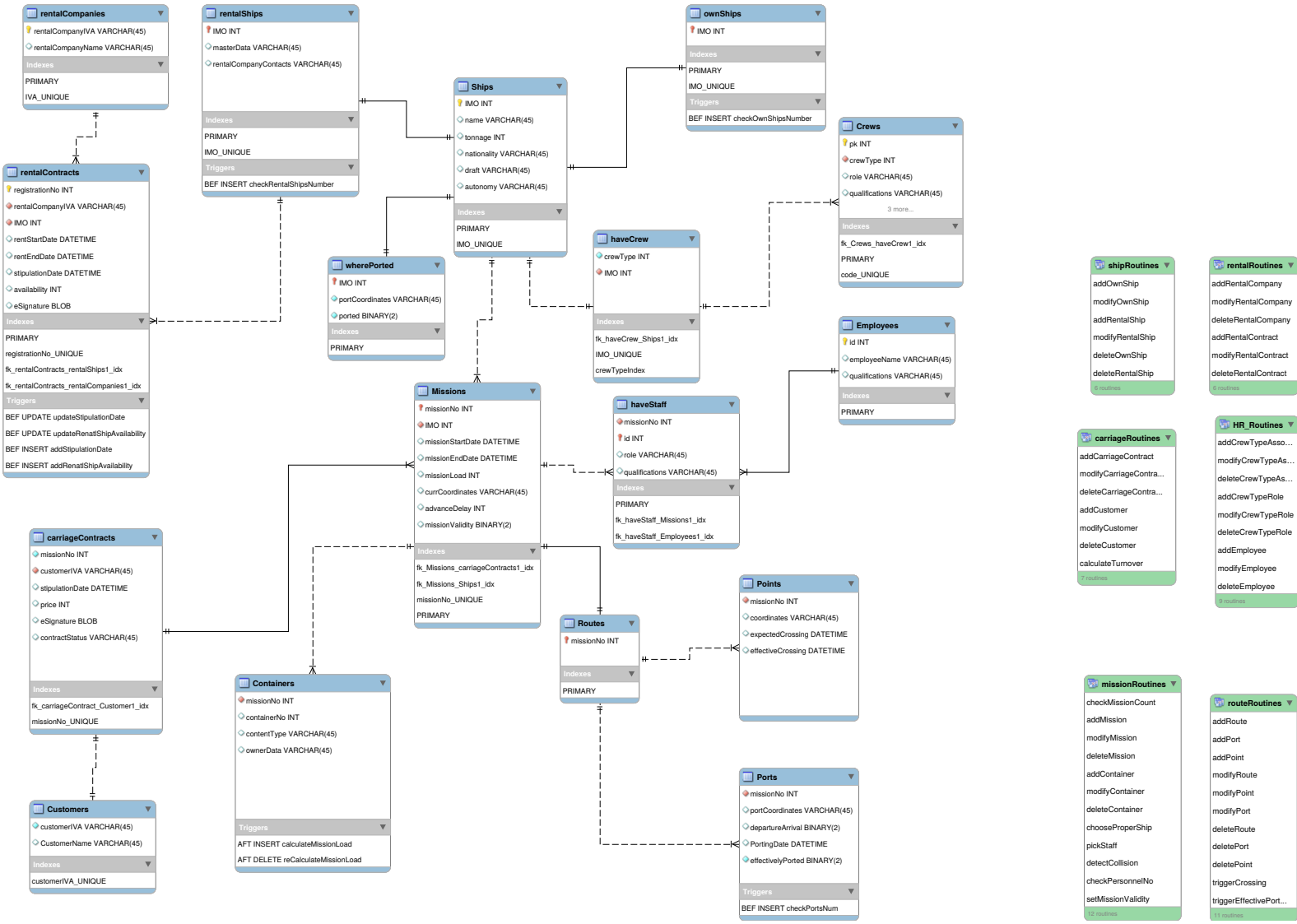
Having Ubuntu linux as my main operating system, I had a small set of tools to choose from, the obvious ones anyways.
For the Conceptual Schema Dia was the most suggested on general forums, but it is highly outdated and full of bugs (as tested on my system), also, but it didn't offer any integrated features.

This is why I chose **MySQL Workbench**, it comes the most recommended from DB professionals, doesn't offer ER design capability, but better, it comes equipped with an EER designer, it offers backward engineering, forward engineering, SQL scripting, python scripting, SQL database connection, and with its SQL linter/debugger, the three stages of the DB design get improved live and professionally.

One note on MySQL Workbench, on its default settings, it only cares about relationships that translate into Foreign Keys, all other operations can be added as SQL routines(procedures)/triggers/scripts. Which makes sense, because this is how we actually translate from Conceptual Design to Logical Design.
But, the EER model in MySQL Workbench can be tweaked a little if someone wants to to add non-key-type relations. However, I chose to stick with the defaults.

There are some other tools which I found (rather late), like draw.io, it's just a modeling tool, much like Dia, but it is more up to date and it has some nice features like adding your own shape (you have to script it in xml) so you can represent any special types of entities or relationships that are not present in the default palette.
Also, initially I only looked for tools concerning MySQL, but after doing a big portion of the project, I discovered pgModeler for PostgreSQL, which would have been better in terms of Conceptual Design because it represents non-key-type relations more out of the box and much better.

Conceptual Schema



Data Dictionary - Entities

Entity	Description	Attributes	Identifier
Ships	Represents all Ships.	IMO, name, tonnage, nationality, draft, autonomy	IMO
ownShips	Represents Company Ships.	IMO	IMO
rentalShips	Represents Ship Rented from Rental Companies.	IMO, masterData, rentalCompanyContacts	IMO
rentalCompanies	Companies we rent ships from	rentalCompanyIVA, rentalCompanyName	rentalCompanyIVA
rentalContracts	Contracts between our company and the companies we rent ships from.	registrationNo, rentalCompanyIVA, IMO, rentStartDate, rentEndDate, availability, esignature	registrationNo
haveCrew	Represents associations between certain ships and certain crew types.	IMO, crewType	IMO
Crews	represent the various roles of the various crew types.	Pk, crewType, role, qualifications, certificate, vacancies, salary	pk
wherePorted	represents the current port of each ship in the present, independent of mission ports.	IMO, portCoordinates, ported	IMO
Missions	represents the missions corresponding to carriage contracts.	missionNo, IMO, missionStartDate, missionEndDate, missionLoad, advanceDelay, currCoordinates, missionValidity	missionNo
Employees	represents all the employees of the company, regardless of their current specific occupation at the company.	id, employeeName, qualifications	id
haveStaff	represents the staff of each mission, who are picked from the employees according to crew type.	missionNo, id, role, qualifications.	missionNo
Customers	represents the customers who want to request carriage missions.	customerIVa, customerName	customerIVA
carriageContracts	carriage contracts between our company and a customer.	missionNo, customerIVA, stipulationDate, price, esignature, contractStatus	missionNo
Containers	describes each individual container of each mission.	missionNo, containerNo, contentType, ownerData	missionNo
routes	identifies the route of each mission	missionNo	missionNo
Points	represents the geographic points of each route.	missionNo, coordinates, expectedCrossing, effectiveCrossing	missionNo
Ports	represents ports associated with missions.	missionNo, portCoordinates, departureArrival, portingDate, effectivelyPorted	missionNo

Code Repository

A Github repository exists for this project under:

`https://github.com/AlyShmahell/Navibus-Mercatoriis`