Server-Side Development Project: FerrySYS

Submitted by: Lydia Sheehan

Date Submitted: 02/05/2024

Contents

[Project Manual 3](#_Toc165391760)

[Database Name 3](#_Toc165391761)

[.sql File Name 3](#_Toc165391762)

[FerrySYS: How it Works 3](#_Toc165391763)

[Database Function Elements 3](#_Toc165391764)

[Insert 3](#_Toc165391765)

[Delete 3](#_Toc165391766)

[Select 4](#_Toc165391767)

[Update 4](#_Toc165391768)

[Main Process 4](#_Toc165391769)

[Further Work 4](#_Toc165391770)

[Sources 4](#_Toc165391771)

# Project Manual

## Database Name

FerrySYS

## .sql File Name

FerrySYS.sql

## FerrySYS: How it Works

A customer can book a ticket for an upcoming departure, by filtering by day, port of departure and vehicle they are travelling in. Each vehicle type has an associated cost and therefore determines ticket price. Only one-way tickets are available for purchase.

An administrator can add a Vehicle Type to the system, meaning a vehicle with a description, an associated price, and it’s size in ‘units’. These ‘units’ are used to calculate the capacity of the ferry, in order to be able to stop selling tickets to a certain departure once it has reached capacity. One unit is based on the size of a car-park parking-space, so for example, a car would be 1 unit and a lorry would be 3.

An administrator can also amend an existing vehicle type’s attributes (all except the code, in order to not violate referential integrity).

An administrator can also delete a vehicle type.

## Database Function Elements

### Insert

The insert can be seen when creating a vehicle type as an administrator. Index -> Admin -> Create a Vehicle Type.

### Delete

The delete can be seen when deleting a vehicle type as an administrator. Index -> Admin -> Delete a Vehicle Type.

If the vehicle type exists in the tickets file, a ‘soft’ delete takes place: it isn’t deleted, it has a status which is set to unavailable, which means it is no longer available within the Book a Ticket function. A vehicle type can only be fully deleted when it does not appear in the tickets file i.e. a ticket with that vehicle type has not been previously sold.

### Select

A filtered select is present in Index -> Book A Ticket, where a timetable is created by selecting departures based on their date, time, and port of departure, and as to whether they have enough room for a selected vehicle type.

### Update

An update is visible in Index -> Admin -> Amend a Vehicle Type where an existing record in the vehicle types table can be altered.

### Main Process

The main processing occurs in Index -> Book a Ticket.

Departures are first filtered, as described above, to create a timetable. Then a ticket is inserted into the database.

## Database View

+--------------------+

| Tables\_in\_ferrysys |

+--------------------+

| departures |

| ports |

| tickets |

| vehicletypes |

+--------------------+

### VehicleTypes

The primary key of VehicleTypes is VCode, and it has no foreign keys.

+--------------+---------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+--------------+---------------+------+-----+---------+-------+

| VCode | char(2) | NO | PRI | NULL | |

| VDescription | varchar(25) | NO | | NULL | |

| Price | decimal(5,2) | YES | | NULL | |

| Units | tinyint(4) | NO | | NULL | |

| VStatus | enum('A','U') | NO | | NULL | |

+--------------+---------------+------+-----+---------+-------+

### Ports

The primary key for the Ports table is PCode, and it has no foreign keys.

+-------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------+-------------+------+-----+---------+-------+

| PCode | char(1) | NO | PRI | NULL | |

| PName | varchar(20) | YES | | NULL | |

+-------+-------------+------+-----+---------+-------+

### Departures

The primary key of Departures is DepID, and it has the foreign key of DepPort, referencing the Ports file.

+-----------+---------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+-----------+---------------+------+-----+---------+----------------+

| DepID | int(11) | NO | PRI | NULL | auto\_increment |

| DepTime | time | NO | | NULL | |

| ArrTime | time | NO | | NULL | |

| DepPort | char(1) | YES | MUL | NULL | |

| Capacity | tinyint(2) | YES | | NULL | |

| DepStatus | enum('A','I') | YES | | A | |

| Date | date | NO | | NULL | |

+-----------+---------------+------+-----+---------+----------------+

### Tickets

The primary key of the Tickets table is the TCode. It has two foreign keys, VCode which references VehicleTypes, and DepID which references Departures.

+-----------+--------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+-----------+--------------+------+-----+---------+----------------+

| TCode | int(11) | NO | PRI | NULL | auto\_increment |

| TDate | date | NO | | NULL | |

| TTime | time | NO | | NULL | |

| VCode | char(2) | NO | MUL | NULL | |

| SalePrice | decimal(5,2) | YES | | NULL | |

| DepID | int(11) | NO | MUL | NULL | |

+-----------+--------------+------+-----+---------+----------------+

## Further Work

There are several aspects of this project that are left unfinished, and that can be improved upon, if not for the scope of this project.

There is no verification for an admin logging in. In a real world scenario this would not be the case.

As things are, there needs to be an administrative way to insert departures into the database, and to set the departures that have already departed to inactive from the site. As things stand, a ticket cannot be purchased for a past departure, but they remain in the database as Active.

### Sources

The picture on the homepage is from [Pixabay](https://pixabay.com/photos/budapest-river-ferry-boat-hungary-2043113/), and compressed with kraken.io.