

# Data Model Specification

DE BOER, Lucas (K2023556)

Kingston University

## 1 Introduction

This document will explain the proposed entity relational diagrams and data dictionaries for the system. The methodology applied to validate the relational model has also been provided. The section on entity relational diagrams will be divided into two parts. The first part will explain how a diagram was created for each user story. In contrast, the second part will cover the various relationship types found in the model, as well as how the many-to-many relationships were resolved. The data dictionary section provides an overview of how a dictionary was made for each user story, explanations for chosen data types, such as binary for passwords, and constraints, such as nullable.

Finally, the entity relational diagrams were validated by constructing an SQL SELECT statement that satisfied the needs of each user story. These SQL statements required multiple tables to be joined, and thus, the modelled relationships were thoroughly tested. If the SQL statement is feasible and should logically work, then the ER Diagram is valid.

## 2 Entity Relational Diagrams

Entity Relational diagrams, referred to as ER diagrams, are graphical representations of the relationships among entities such as people or objects. An entity is a thing of interest to a system about which data is kept. For the context of this system, these entities could be watercraft, bookings, employees, and equipment/items. These entities are depicted as a box with their name at the top. The line between the boxes represents the relationship between them. The multiplicities specify the cardinality or number of instances of an entity that can be associated with the instance of another entity. The possible types of multiplicity include zero-to-one, one-to-one, zero-to-many, and one-to-many. In the "View Fleet" Figure 2.1.1, the multiplicity between "WatercraftTypes" and "Models" is a zero-to-many relationship, as a watercraft type may never be allocated to a model; however, a single type may be assigned to many models. Many-to-many relationships should not be present in any ER diagram if normalised to the third normal form. These types of relationships are resolved, as discussed later in Section 2.2.

To construct the complete system ER diagram, Figure 2.0.1(Appendix A for larger image), all necessary entities were identified first. An ER diagram was independently made for each user story to accomplish this. This ensured that all tables and attributes could be identified.

### 2.1 Per user story

An ER diagram was constructed for each user story. These diagrams were created and updated during the "Updating Data Model" sprints to capture the necessary tables and attributes to accomplish a selected user story. Creating smaller ER diagrams focusing on a particular user story allows a developer to comprehend the requirements that must be implemented during a sprint more easily. The ER diagram shown in Figure 2.1.1 showcases the minimalist characteristic of these focused diagrams, as only four tables were needed to implement the "View Fleet" user story.

A comprehensive set of entity relationship diagrams for all user stories can be found in Appendix B.

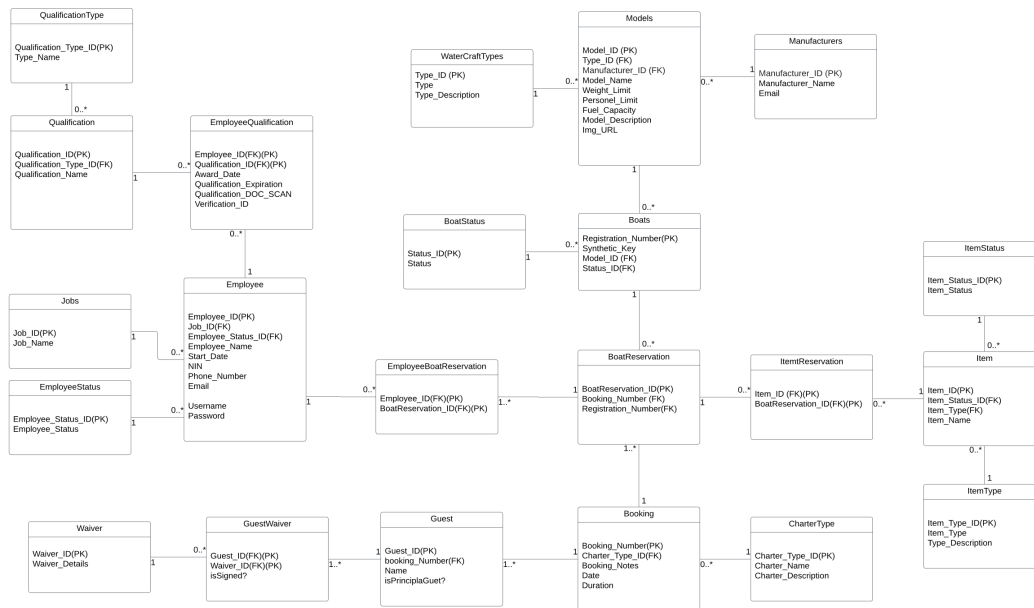


Figure 2.0.1 ERD for the entire System

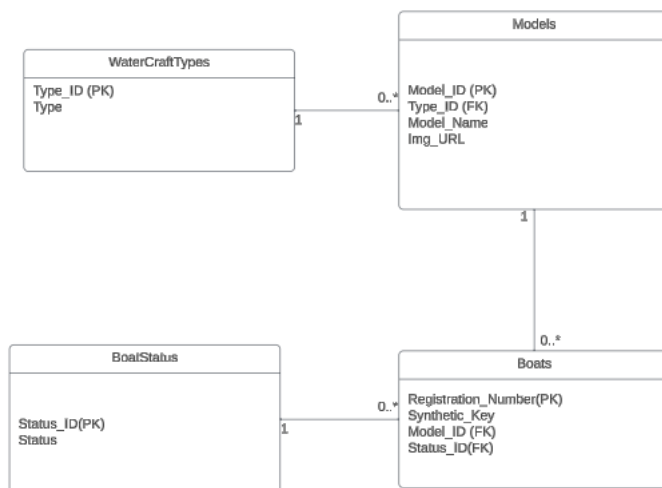


Figure 2.1.1 ERD for “View Fleet”.

## 2.2 Table to Table Relationships

This document section will explain the different parts of this system’s entity relation diagram, making the relationships between the tables clear.

Resolving the relationship between tables is highly important, especially many-to-many relationships. In a many-to-many relationship, more than one record in a table is related to more than one record in another table. These relationships can introduce update anomalies as the resultant table contains repeated values. Update anomalies include insertion, deletion, and modification anomalies. The process of normalisation eliminates these anomalies by reducing redundancy and dependency. Normalisation ensures that every table only has one purpose and has only the columns necessary to support that purpose.

By applying the normalisation process, intermediate tables were created between all many-to-many relationships. The “Employee-Qualification constellation” shown in Figure 2.2.1 is the result of

solving the many-to-many relationship between “Employee” and “Qualification”, as employees can have many qualifications, and many employees can have qualifications. The “EmployeeQualification” table, which sits between this many-to-many relationship, uses the ID from both the “Qualification” and “Employee” tables as a composite primary key. By doing this, even if many employees have the same qualification, the records in this table will always be unique due to the combination of both IDs.

The “watercraft constellation” shown in Figure 2.2.2 only has one-to-many relationships. A row in “Boats” must have only one “Model\_ID”, which must be present in the “Models” table due to the foreign key constraint. Conversely, the non-to-many multiplicity shows that the same “Model\_ID” number could be present many times in the “boats” table. The status and type attributes were separated from the “Boats” and “Models” tables. This ensures that if a modification were made to the naming scheme for either, every entry would not have to be manually updated. Furthermore, it must be highlighted that the “Boats” table is where all the charter company's watercraft have been stored. The “Models” table exists because a charter company may own more than one boat with the same model. This removes the need for the user to re-enter the model data every time.

To handle the booking functionality of the system, reservation tables for the boats, employees, and items/equipment have been made. To explain these tables, the concept of a booking will be established first. A booking can have many guests who belong to a single group, as shown in Figure 2.2.3. However, a booking can include many watercraft/boats; in other words, a guest could rent out multiple boats for the same booking. A boat can also be used on many bookings. The “BoatReservation” table was created to solve this many-to-many relationship, as shown in Figure 2.2.4. This table has the “Booking\_Number” and “Registration\_Number” of the boat used for the booking. These fields have a paired UNIQUE constraint, meaning that the same boat cannot be reserved twice for the same booking (A boat cannot exist twice simultaneously). The “BoatReservation\_ID” ID is the primary key of the “BoatReservation” table and is thus unique to the boat and booking pairing. This ID can allocate the employees, items/equipment to the specific boat and booking. This allocation is shown in Figure 2.2.5 and Figure 2.2.6.

In Figure 2.2.5, the “EmployeeBoatReservation” table allocates which boat and booking each employee is scheduled for. As you can see, the “BoatReservation\_ID” is paired with the “Employee\_ID” to form a composite primary key. The same logic is applied to the “ItemReservation” table, as shown in Figure 2.2.6. This table designates which boat the items/equipment are supposed to be on.

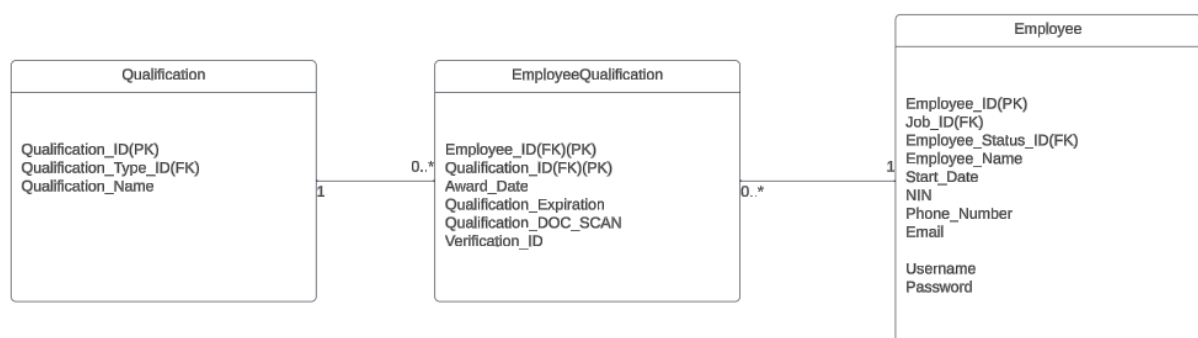


Figure 2.2.1 ERD “Employee-Qualification constellation” showing the resolved many-to-many.

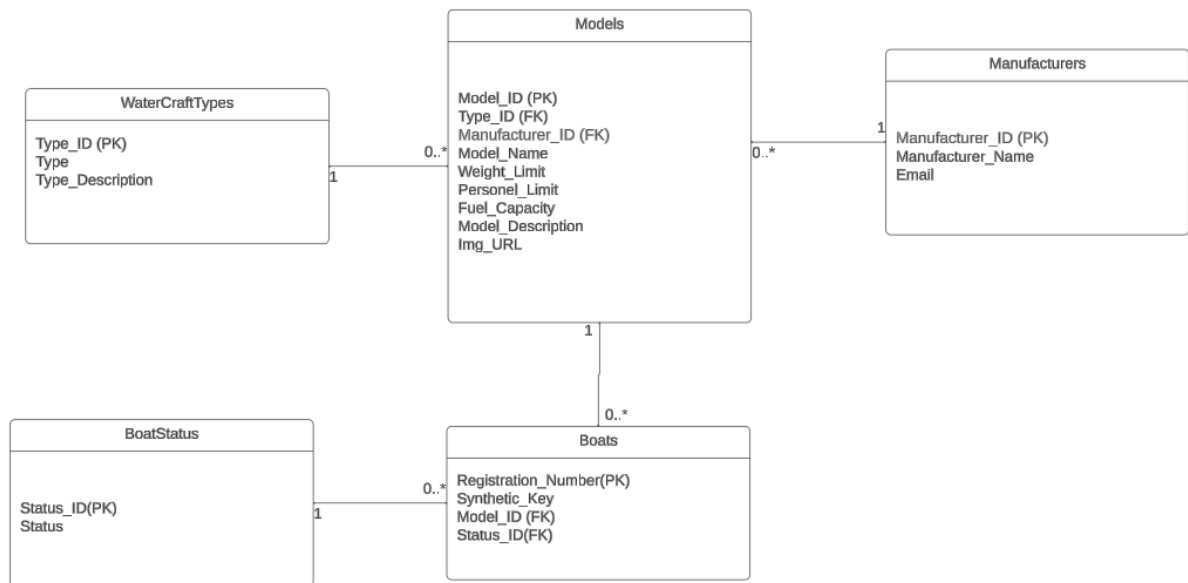


Figure 2.2.2 ERD “Watercraft constellation” shows the main tables for watercraft data.

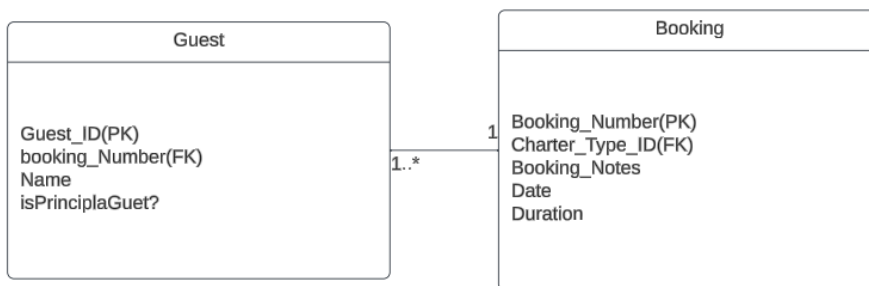


Figure 2.2.3 ERD “Booking-Guest constellation” showing bookings having many guests.

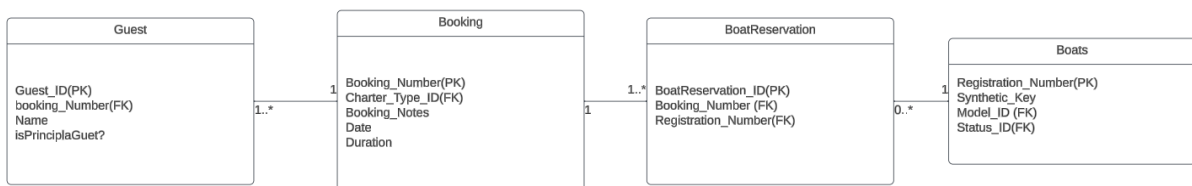


Figure 2.2.4 ERD “Booking-Boats constellation” showing the resolved many-to-many.

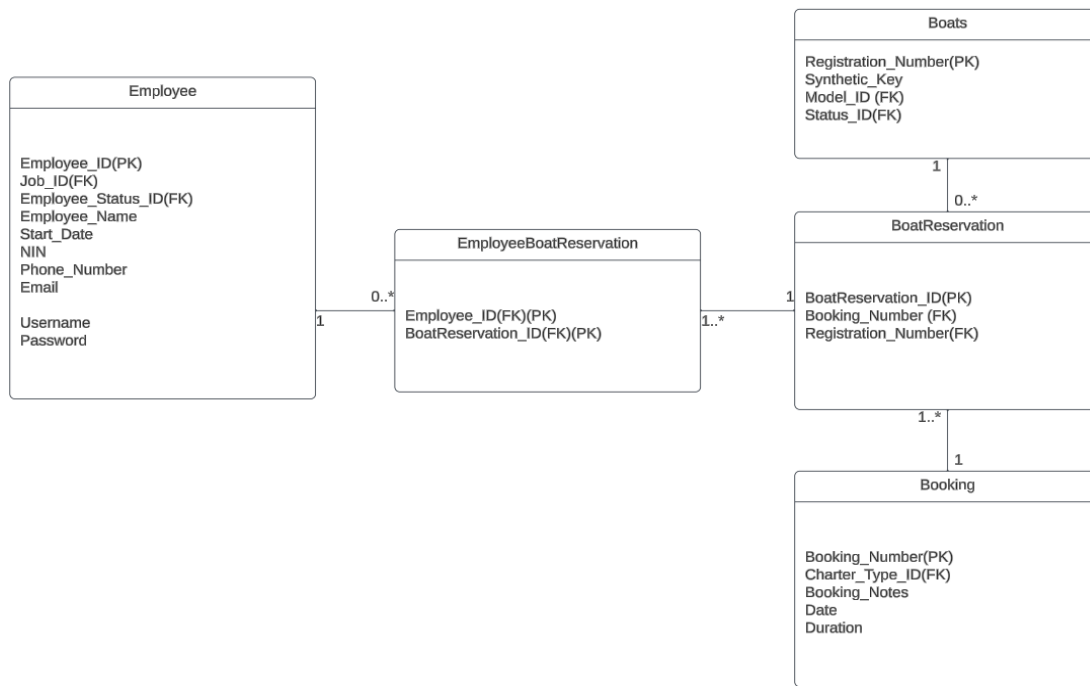


Figure 2.2.5 ERD “Employee-BoatReservation constellation” showing employees get allocated.

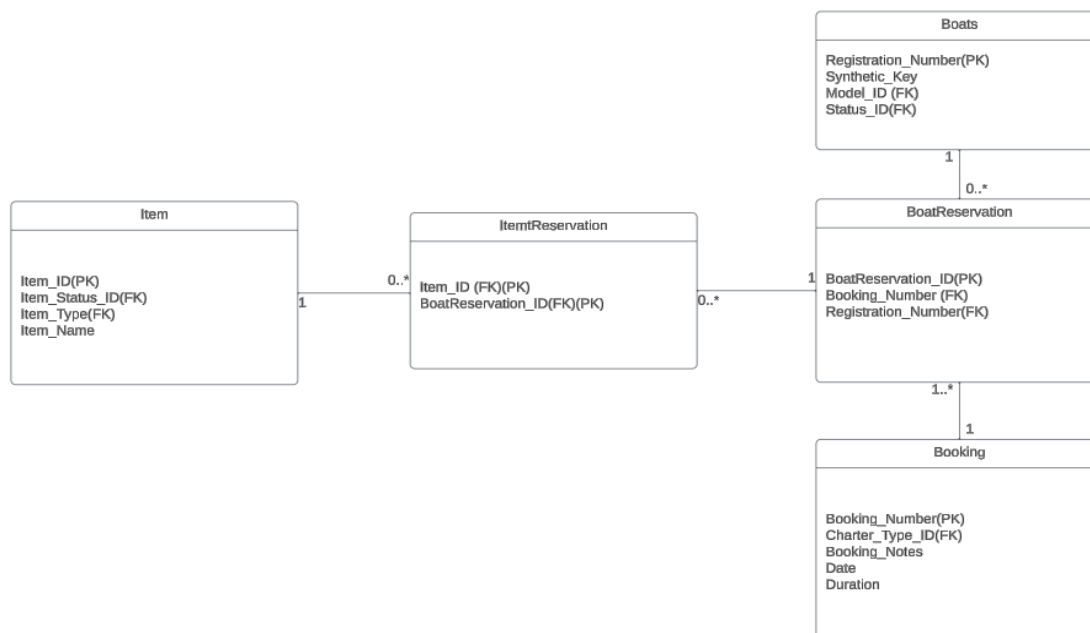


Figure 2.2.6 ERD “Item-BoatReservation constellation” showing employees get allocated.

### 3 Data Dictionaries

Data dictionaries are tables containing the attributes of a particular entity with their respective data type and constraints. Every table includes a primary key, and some have foreign key constraints. The foreign key constraint enforces referential integrity in the database. The “Job\_ID” in the “Employee” table shown in Figure 3.1 is a foreign key, meaning that the job ID value for the employee must also exist in the “Jobs” table shown in Figure 3.2.

Some Attributes like the “NIN” in the Employee table have the “NULLABLE” constraint. While null values should be kept to a minimum, sometimes it is essential as not every employee may have a

National Insurance Number. The attributes which do not have a constraint are “Not Null” by default. Extra attention must be given to the attributes that explicitly state the “Not Null” constraint, as these values, under no circumstances, should ever be Null.

In Figure 3.1, the attributes “Username” and “Password” have a binary data type. Both attributes will be hashed and converted to binary to ensure the database complies with user data security, confidentiality, and privacy requirements.

Data dictionaries were made for each user story. These data dictionaries contained only the necessary attributes for the designated user story and can be found in Appendix C. The data dictionary for the complete ERD Diagram can be found in Appendix D.

<b>Employee</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Employee_ID	Int(5)	Primary Key
Job_ID	Int(5)	Foreign Key
Employee_Status_ID	Int(5)	Foreign Key
Employee_Name	Varchar(100)	
Start_Date	Date	
NIN	Varchar(9)	NULLABLE
Phone_Number	Varchar(11)	
Email	Varchar(300)	
Username	Binary	Not Null
Password	Binary	Not Null

Figure 3.1 DATA DICTIONARY of “Employee” Table for “Edit My Profile” User Story

<b>Jobs</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Job_ID	Int(5)	Primary Key
Job_Name	Varchar(100)	

Figure 3.2 DATA DICTIONARY of “Jobs” Table

## 4 ERD Validation

An SQL SELECT statement was constructed for each User story. These statements satisfied the given requirement and necessitated using attributes from more than one table. When selecting from multiple tables, the relationships in the data model are tested; the SQL statement will not work if the relationships are invalid. If the SQL statement is feasible and should logically work, then the ER Diagram is valid. Figure 4.1 shows the validation of the “View Fleet” ER Diagram. The select statement must fetch data from multiple tables to satisfy the requirement. The SQL statement presented in Figure 4.1 is logical and works. Thus, it can be deduced that the ER diagram is correct. Figure 4.2 has a more complex ER diagram, yet the SQL statement required to validate it is relatively simple, making this an effective way of validating ER diagrams of all complexities.

A comprehensive list of ERD validation tables can be found in Appendix E.

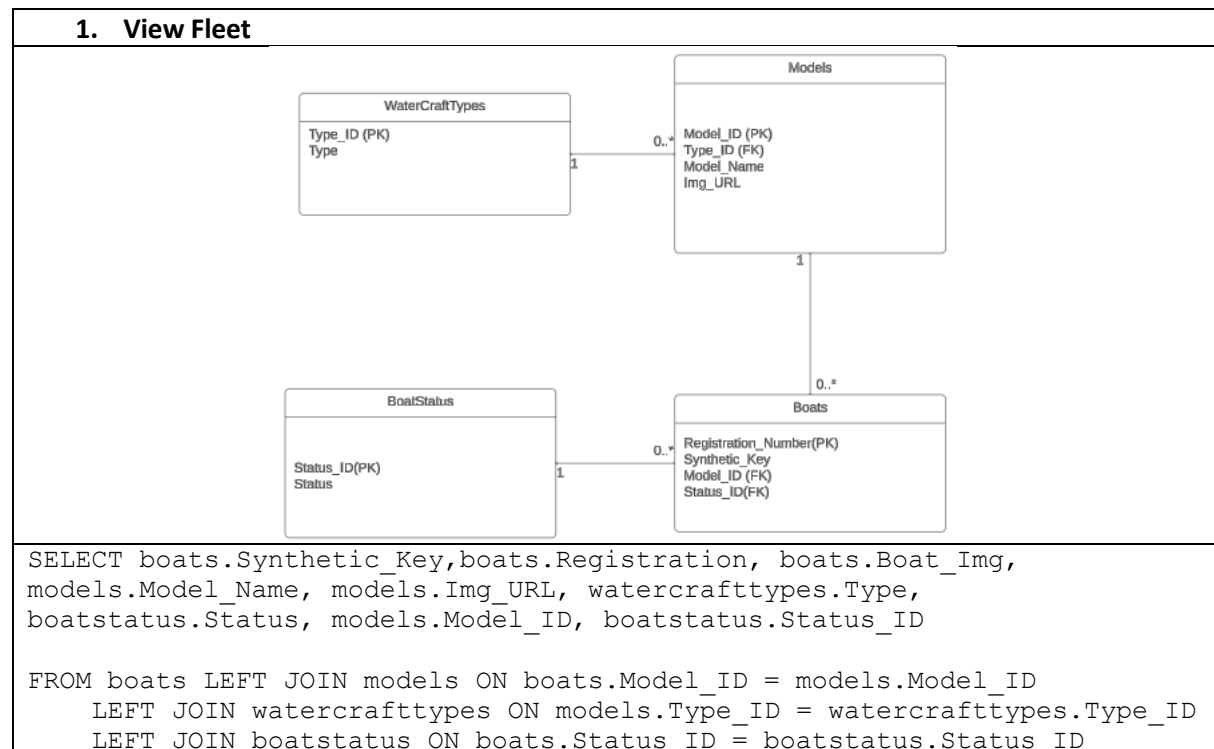
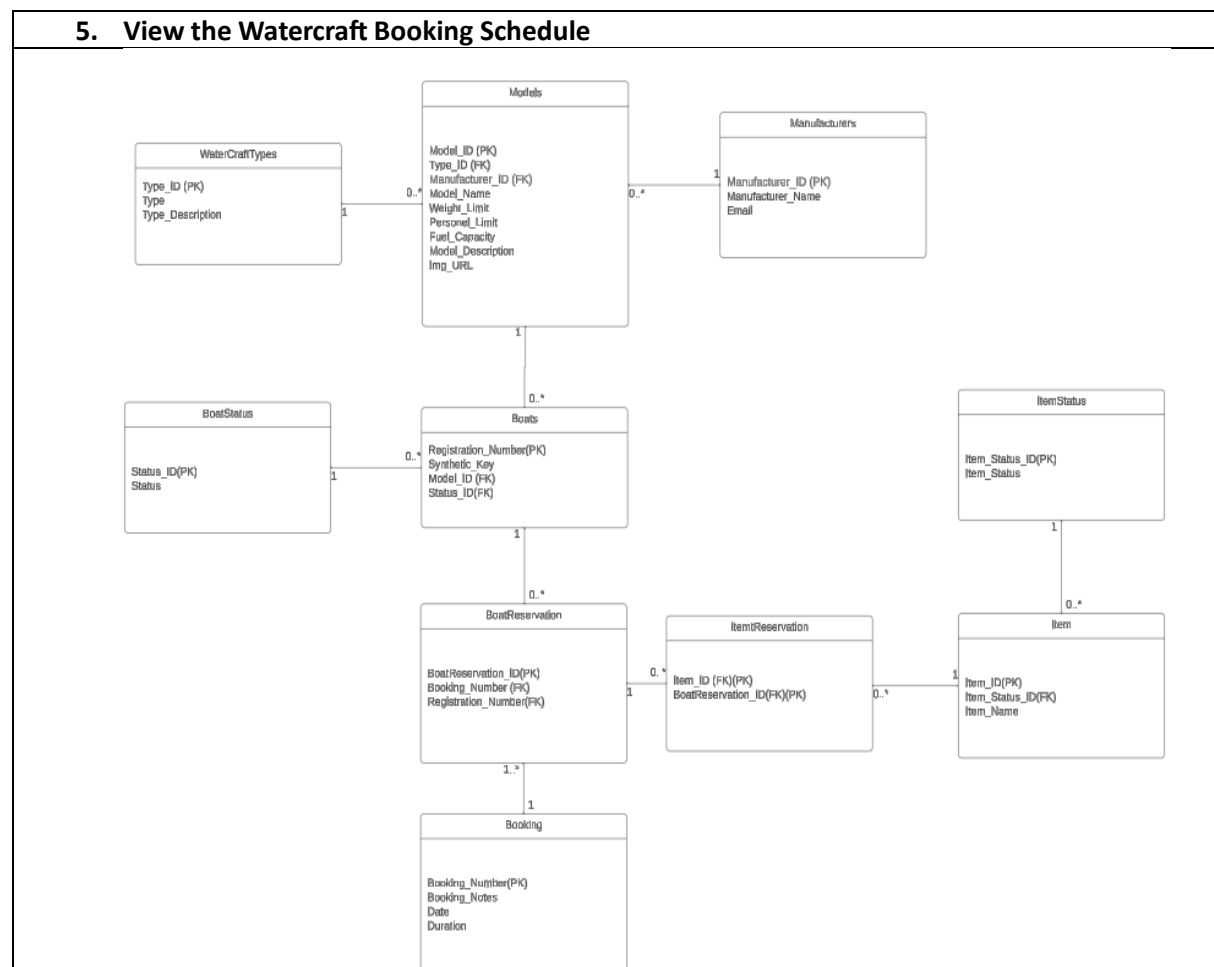


Figure 4.1 DATA VALIDATION Table for "View Fleet" User Story



```
SELECT bookings.Booking_Notes, bookings.Date, bookings.Duration,  
bookings.Charter_Type_ID  
FROM boats LEFT JOIN boatreservation ON boats.Registration =  
boatreservation.Registration  
LEFT JOIN bookings ON boatreservation.Booking_Number =  
bookings.Booking_Number  
LEFT JOIN chartertype ON bookings.Charter_Type_ID =  
chartertype.Charter_Type_ID  
WHERE boats.Synthetic_key = ?;
```

*Figure 4.2 DATA VALIDATION Table for “View Watercraft Booking Schedule” User Story*

## 5 Conclusion

This document has reviewed the system’s ER diagram and explained the relationships between the main entities. All attributes have their data types and constraints specified, along with the reasoning behind why the username and password attributes are set to binary. The ER diagram provided has been validated to ensure that it can provide the necessary functionality to support all system requirements.



## Appendix A – System ER Diagram

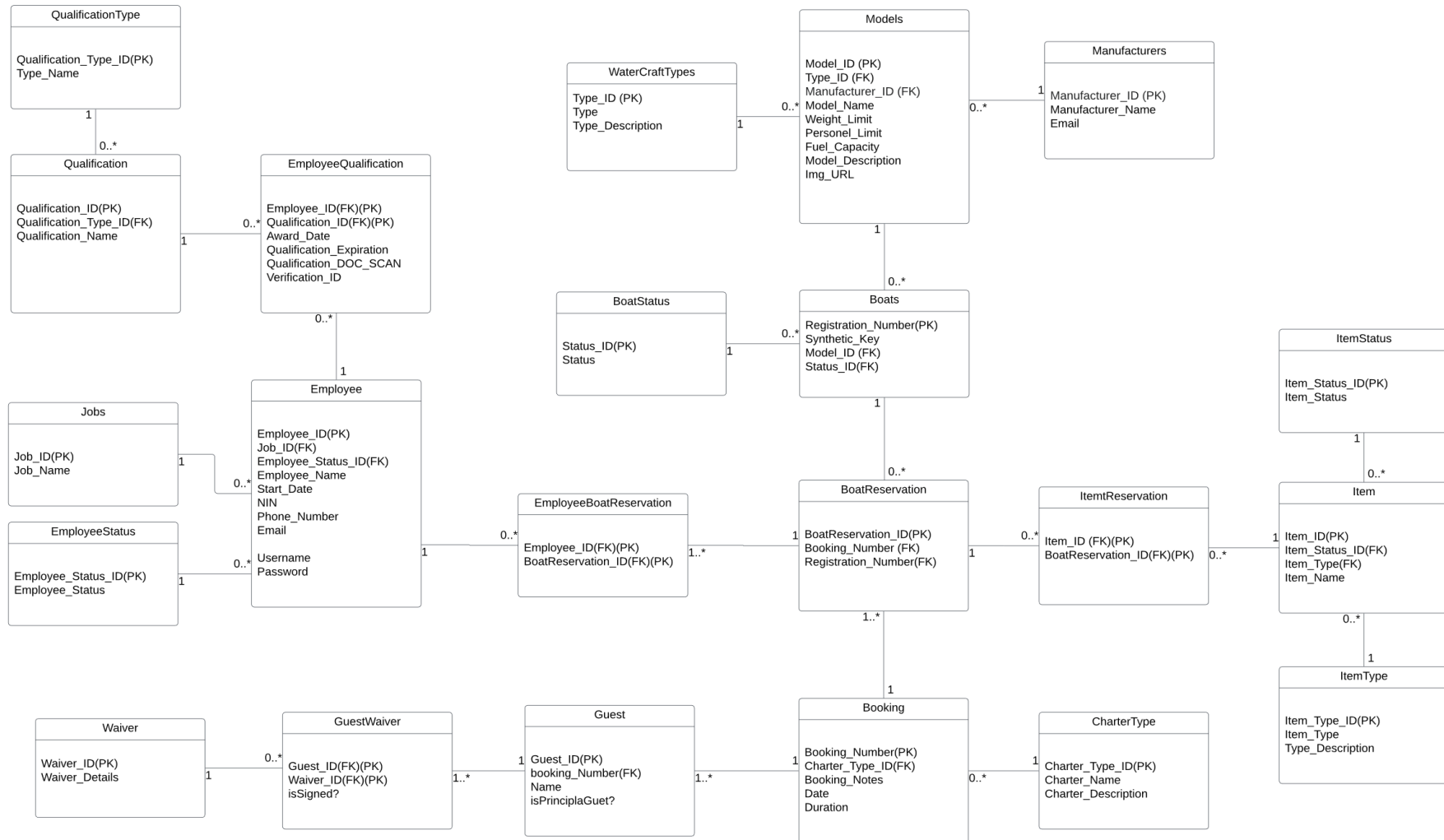


Figure A.1 ER Diagram for the entire System

## Appendix B – User Story ER Diagrams

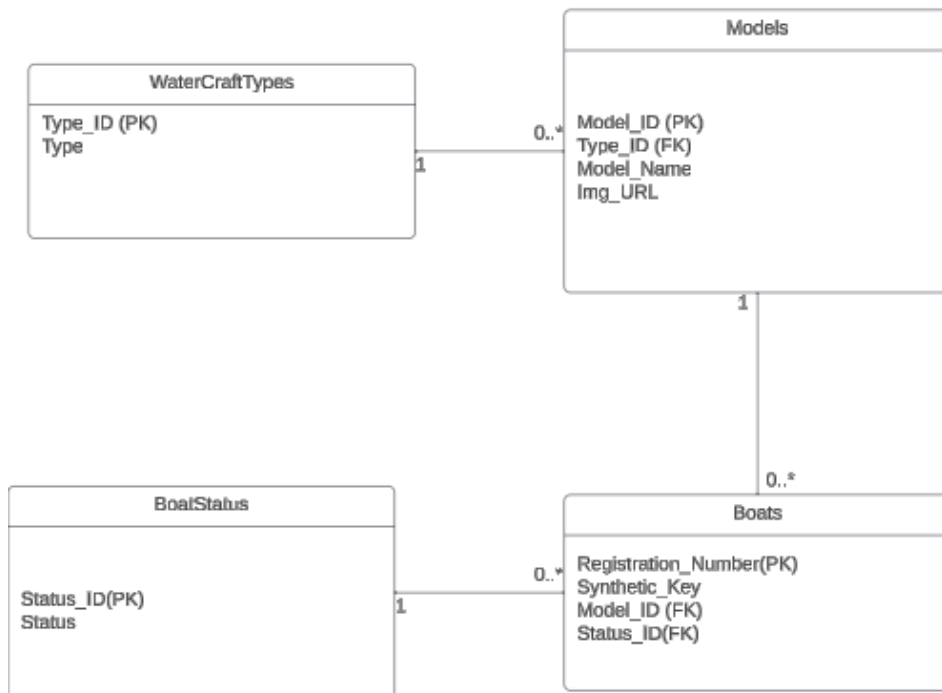


Figure B.1 ERD for "View Fleet".

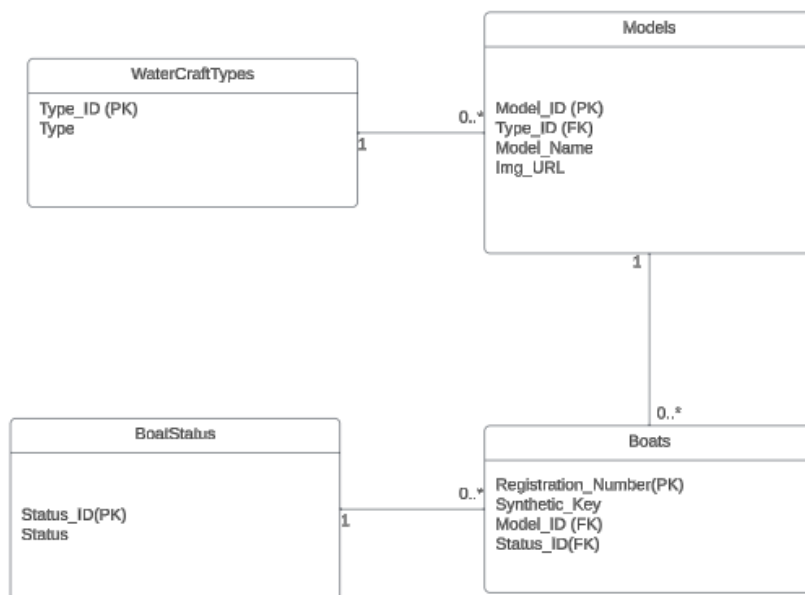


Figure B.2 ERD for "Add Watercraft".

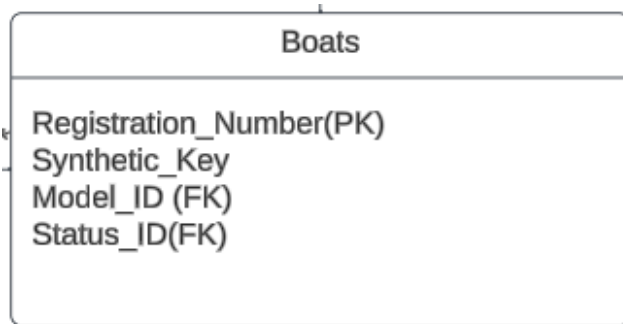


Figure B.3 ERD for “Delete Watercraft”.

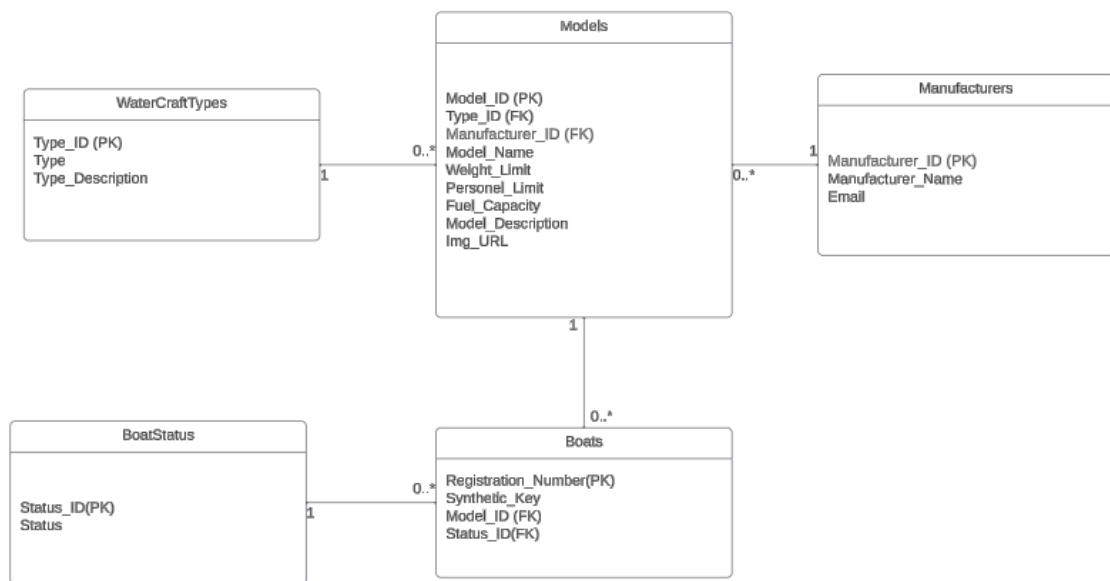


Figure B.4 ERD for “Edit Watercraft”.

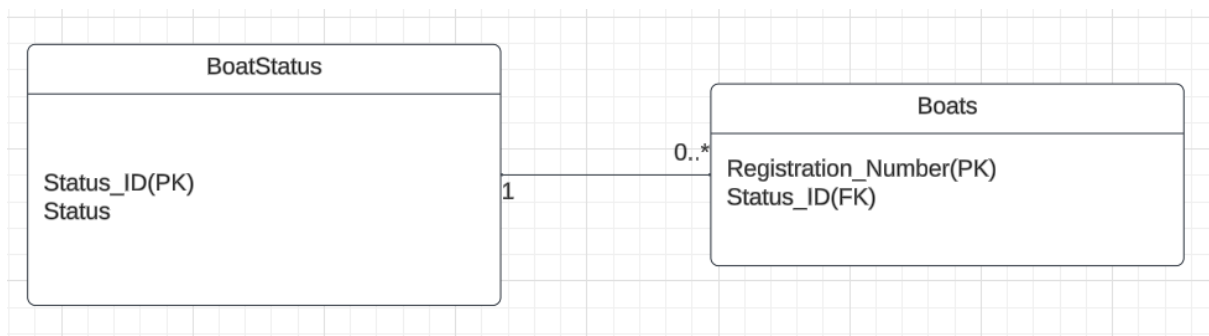


Figure B.5 ERD for “Group Watercraft by Status”.

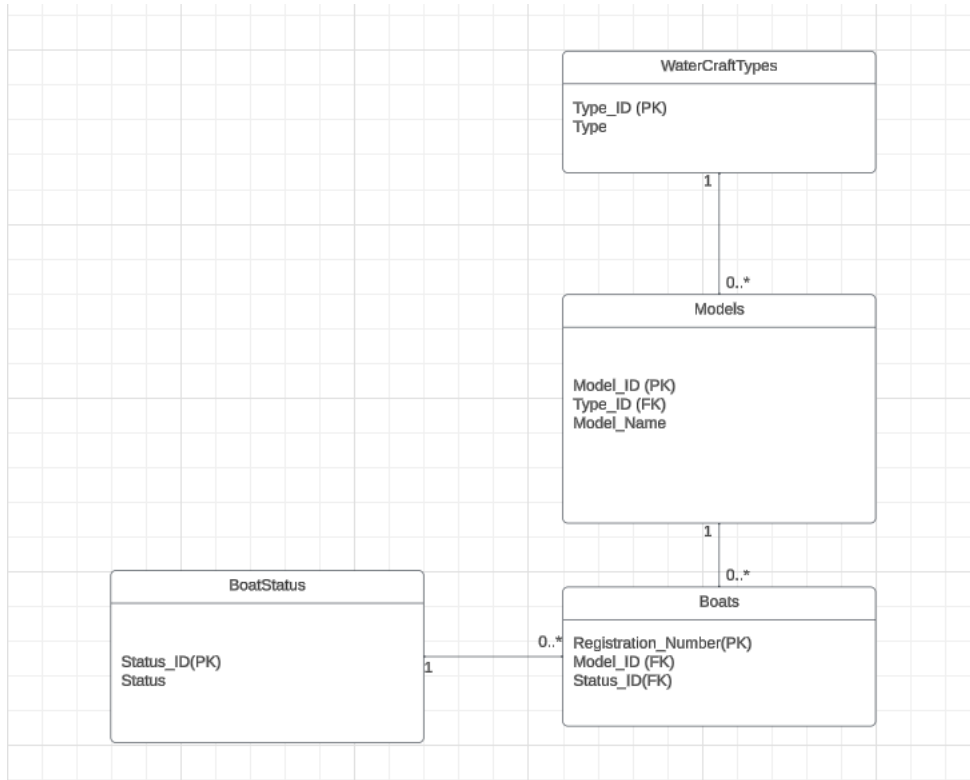


Figure B.6 ERD for “Search for Watercraft”.

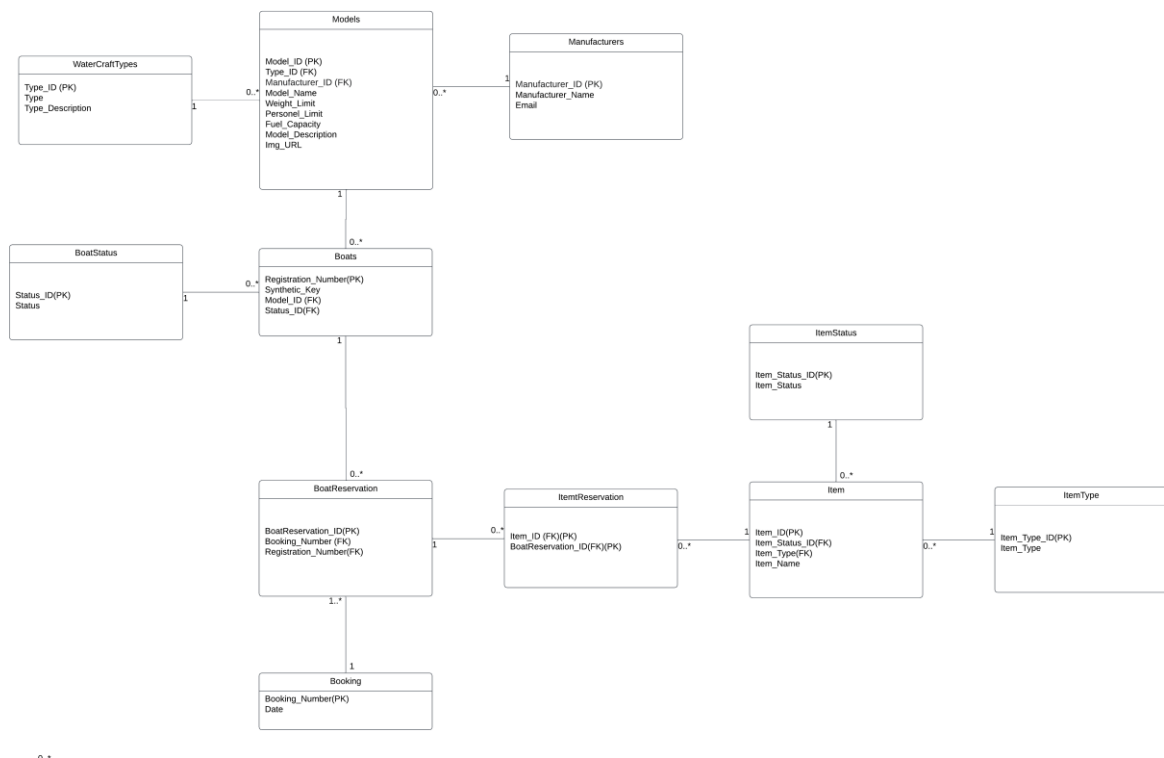


Figure B.7 ERD for “View Watercraft Details”.

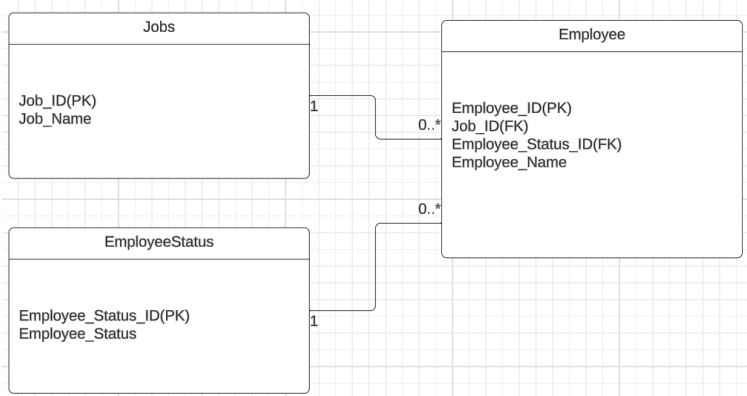


Figure B.8 ERD for “View All Employees”.

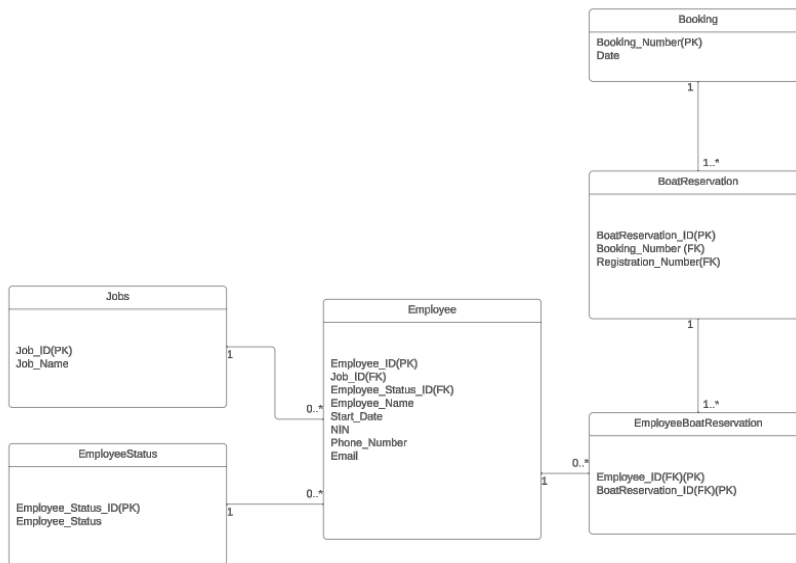


Figure B.9 ERD for “View Employee profiles/details” and “View my Profile”.

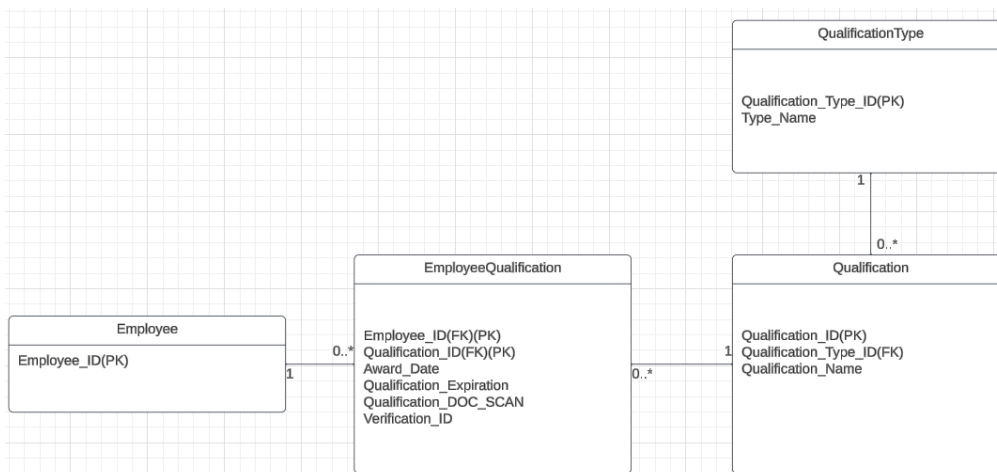


Figure B.10 ERD for “Add Qualification to Profile”.

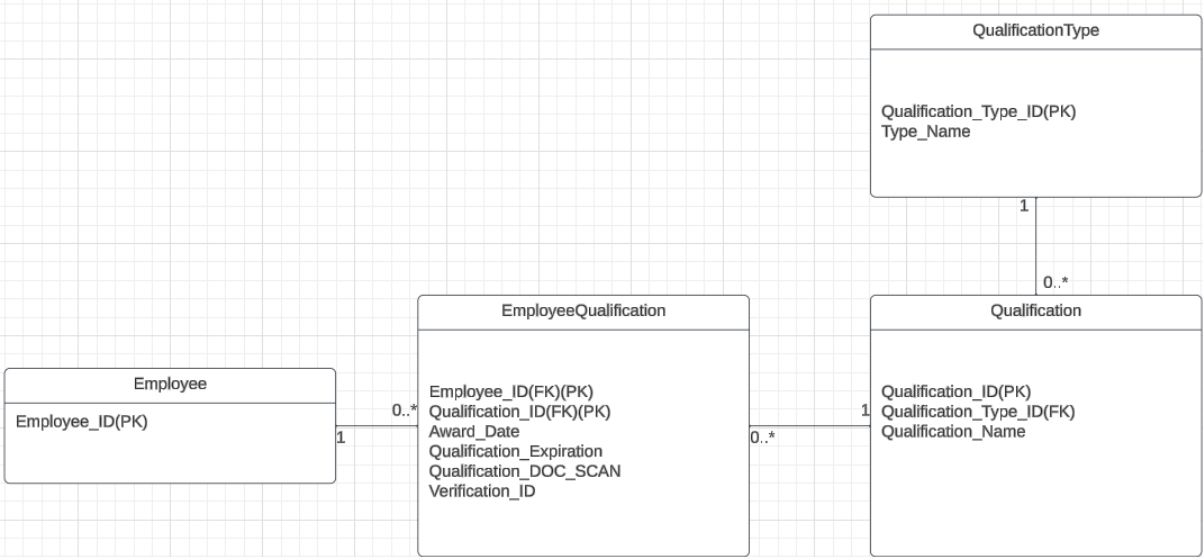


Figure B.11 ERD for “View Employee Qualifications” and “View my Qualifications”.

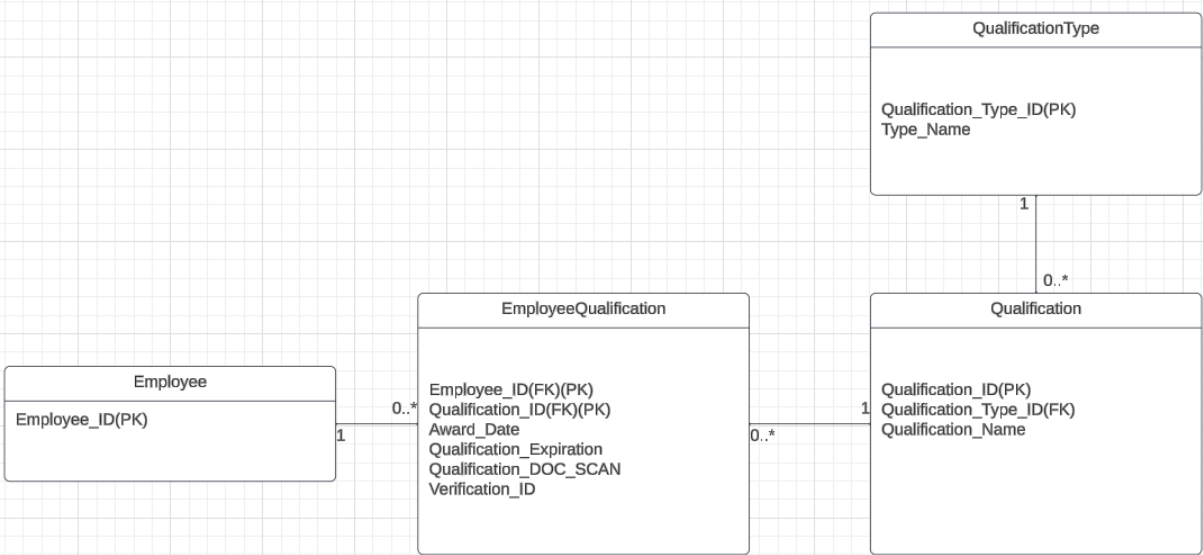


Figure B.12 ERD for “Edit my Qualifications”.

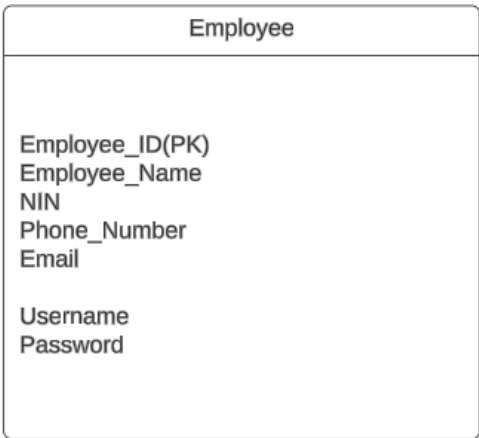


Figure B.11 ERD for “Edit my Profile”.

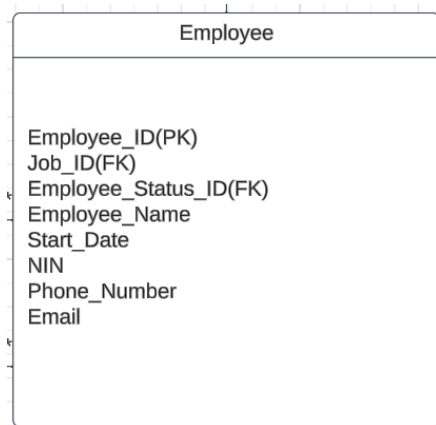


Figure B.11 ERD for "Search for Employee".

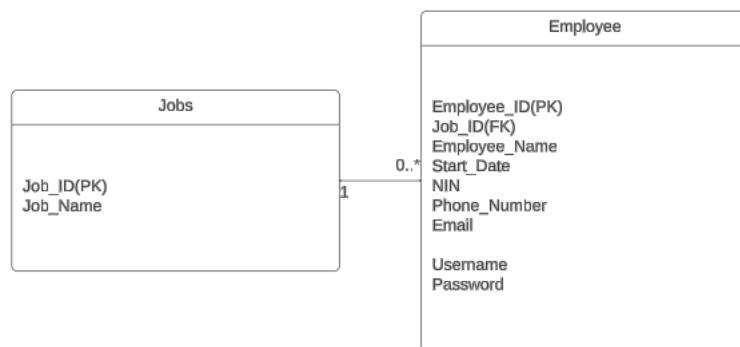


Figure B.12 ERD for "Create Employee Account".

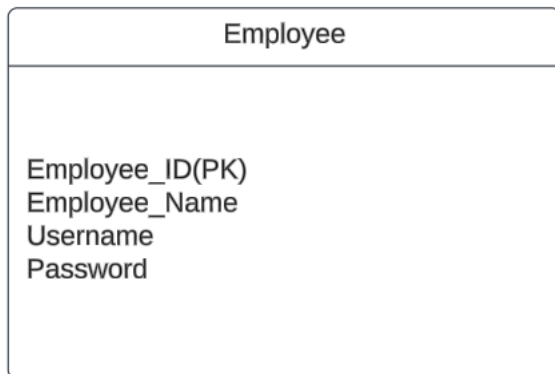


Figure B.12 ERD for "Delete Employee Account".

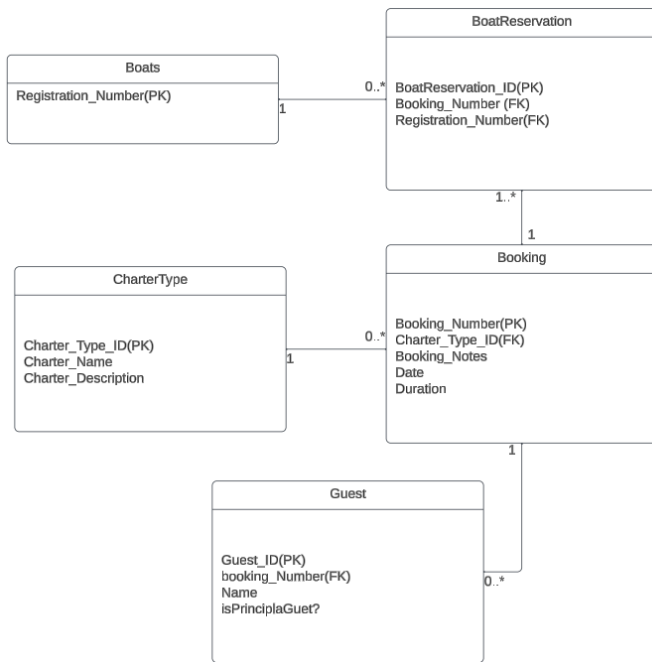


Figure B.15 ERD for “View all Bookings”.

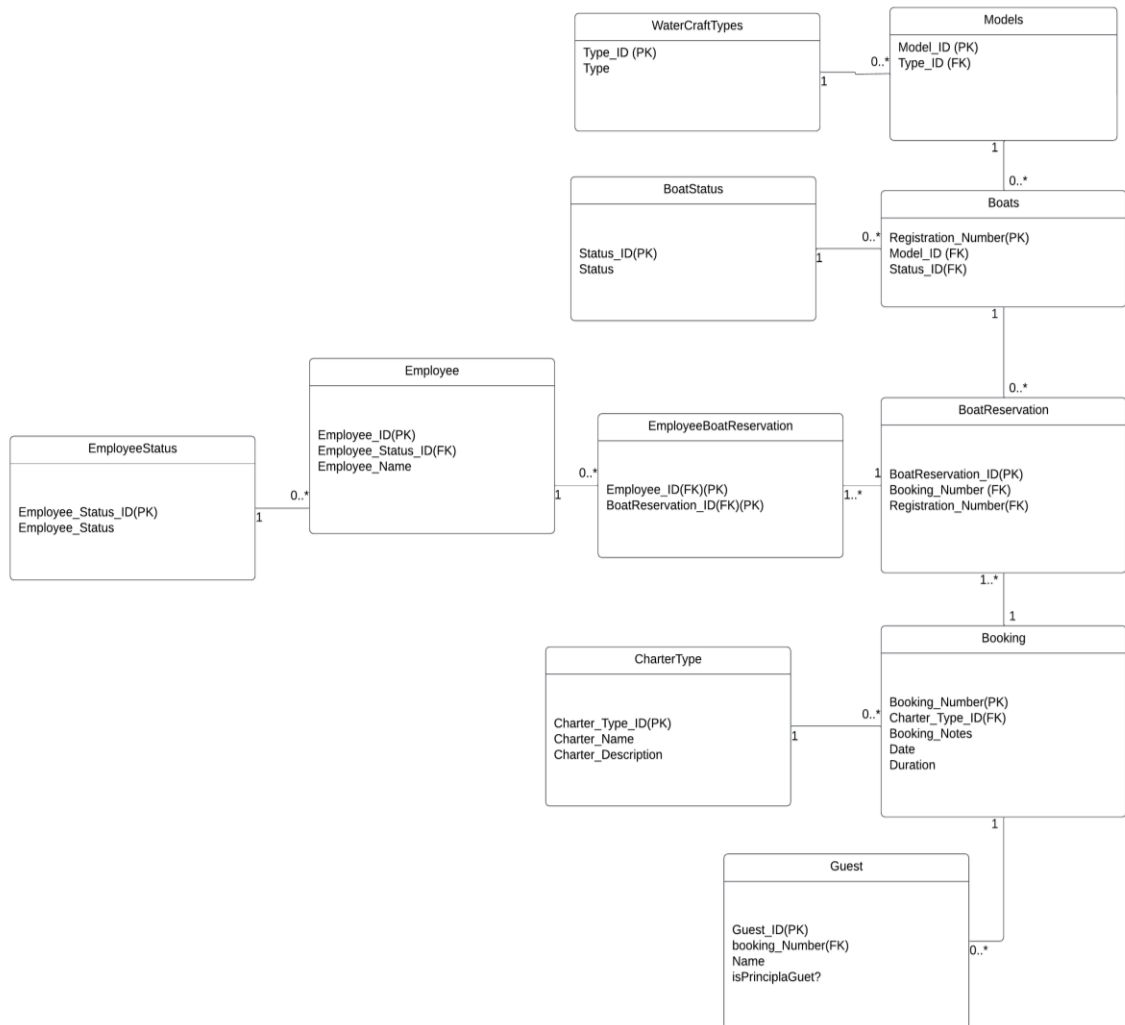




Figure B.16 ERD for “View Booking Details”.

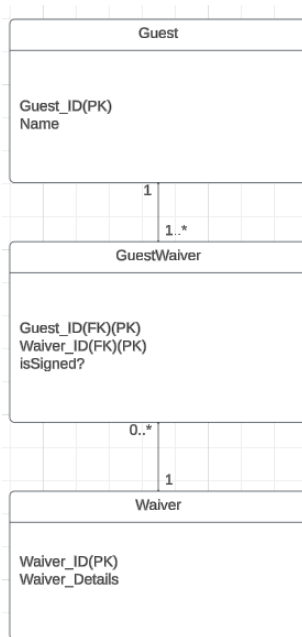


Figure B.17 ERD for “View Waivers”.

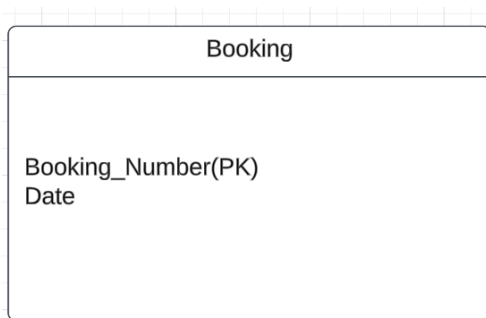


Figure B.18 ERD for “Search Booking by ID”.

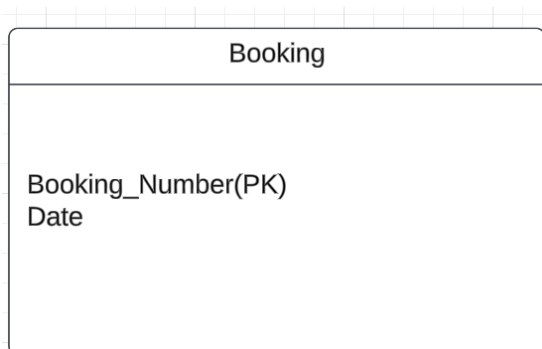


Figure B.19 ERD for “Search Booking by Date”.

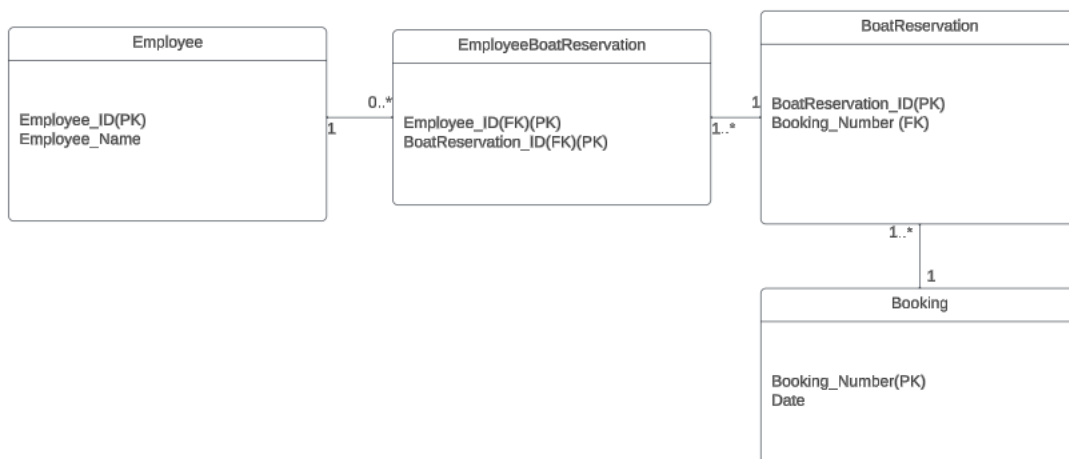


Figure B.20 ERD for “Remove/ Add Crew Member from (to) Booking”.

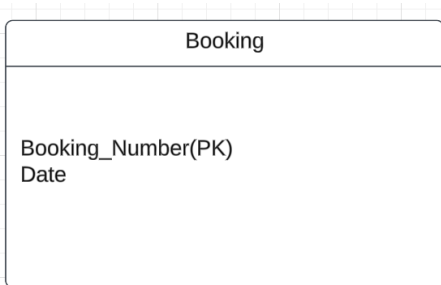


Figure B.21 ERD for “Request Change Booking Time(Owner)”.

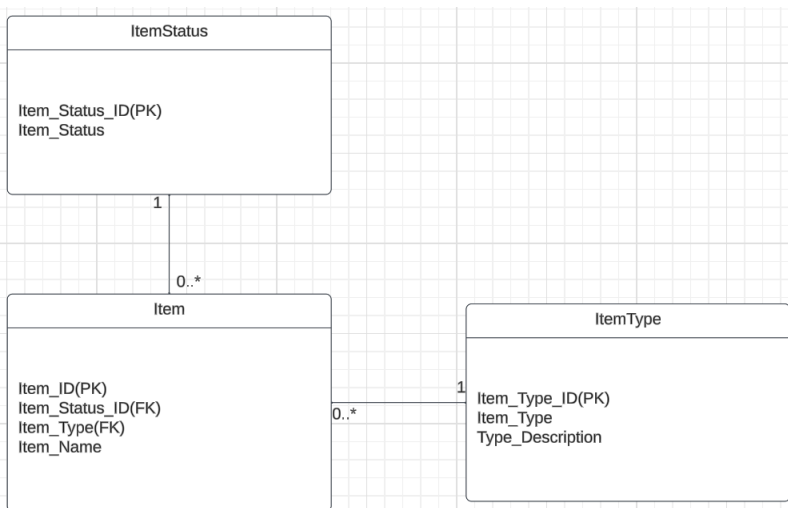


Figure B.22 ERD for “View Inventory”.

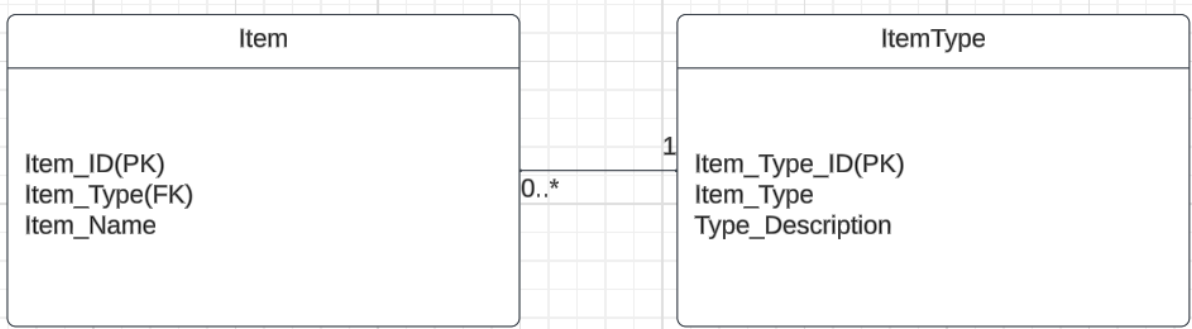


Figure B.23 ERD for “Add Item to Inventory”.

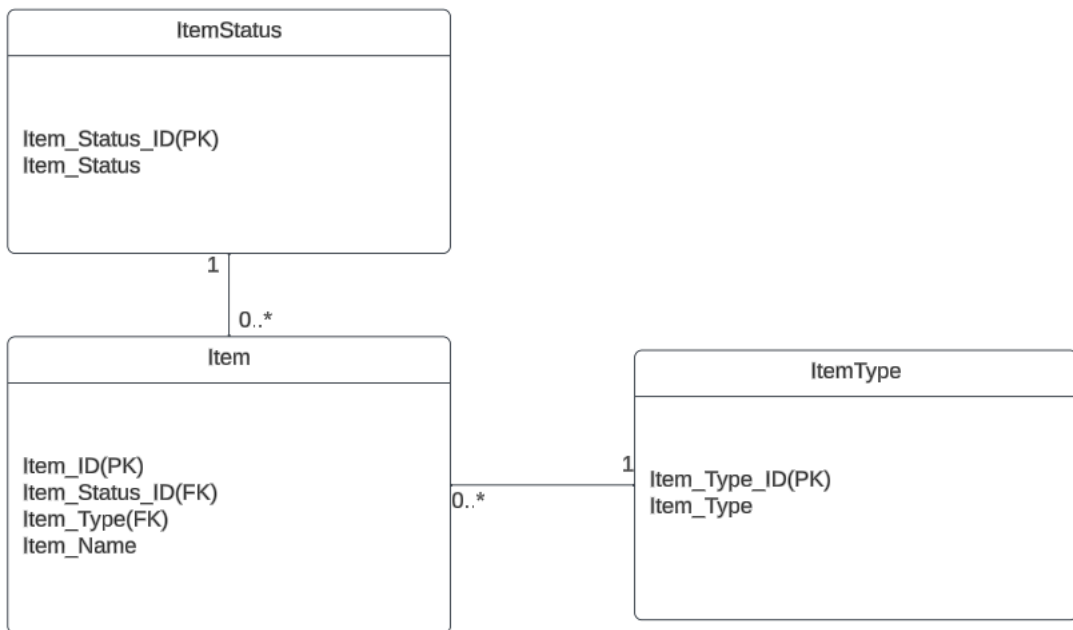


Figure B.24 ERD for “Remove Item from Inventory”.

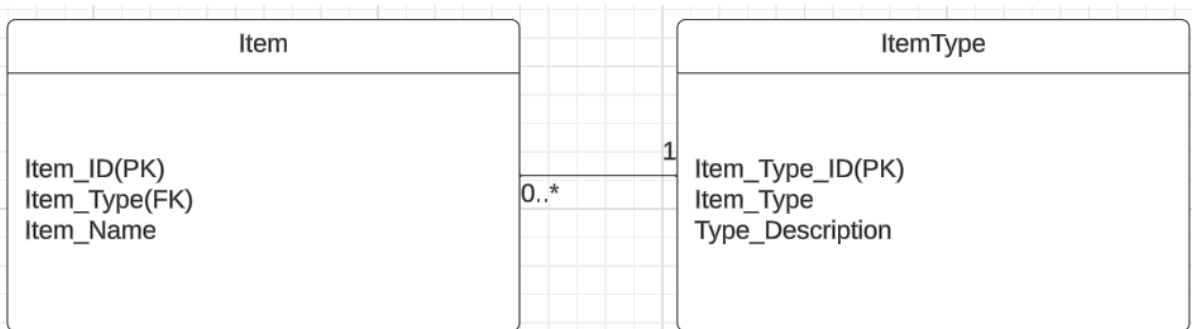


Figure B.25 ERD for “Edit Item”.

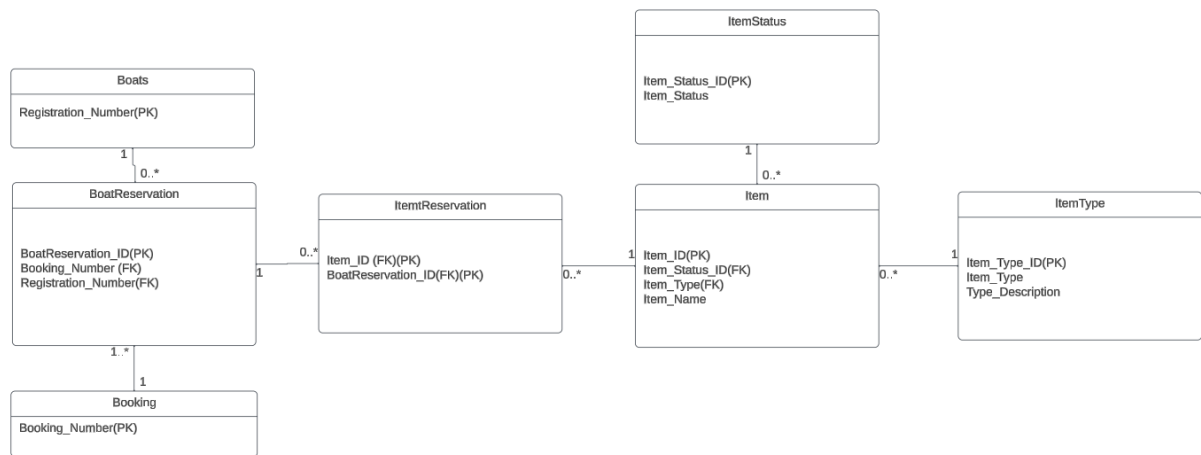


Figure B.26 ERD for “Assign Item to Watercraft”.

## Appendix C – Data Dictionaries

<b>WaterCraftTypes</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Type_ID	int(5)	Primary Key
Type	Varchar(20)	NOT NULL
<b>Models</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Model_ID	Int(5)	Primary Key
Type_ID	int(5)	Foreign Key
Model_Name	Varchar(100)	NOT NULL
Img_URL	Varchar(500)	DEFAULT ( <a href="https://idea7.co.uk/wp-content/uploads/2021/02/placeholder-250x250-1.png">https://idea7.co.uk/wp-content/uploads/2021/02/placeholder-250x250-1.png</a> )
<b>BoatStatus</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Status_ID	Int(5)	Primary Key
Status	Varchar(50)	NOT NULL
<b>Boats</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Registration	Varchar(100)	Primary Key
Synthetic_Key	Int(11)	Unique
Model_ID	Int(5)	Foreign Key
Status_ID	Int(5)	Foreign Key

Figure C.1 DATA DICTIONARY for “View Fleet”.

<b>WaterCraftTypes</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Type_ID	int(5)	Primary Key
Type	Varchar(20)	NOT NULL
<b>Models</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Model_ID	Int(5)	Primary Key
Type_ID	int(5)	Foreign Key
Model_Name	Varchar(100)	NOT NULL

Img_URL	Varchar(500)	DEFAULT ( <a href="https://idea7.co.uk/wp-content/uploads/2021/02/placeholder-250x250-1.png">https://idea7.co.uk/wp-content/uploads/2021/02/placeholder-250x250-1.png</a> )
<b>BoatStatus</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Status_ID	Int(5)	Primary Key
Status	Varchar(50)	NOT NULL
<b>Boats</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Registration	Varchar(100)	Primary Key
Synthetic_Key	Int(11)	Unique
Model_ID	Int(5)	Foreign Key
Status_ID	Int(5)	Foreign Key

Figure C.2 DATA DICTIONARY for “Add Watercraft”.

<b>Boats</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Registration	Varchar(100)	Primary Key
Synthetic_Key	Int(11)	Unique
Model_ID	Int(5)	Foreign Key
Status_ID	Int(5)	Foreign Key

Figure C.3 DATA DICTIONARY for “Delete Watercraft”.

<b>WaterCraftTypes</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Type_ID	int(5)	Primary Key
Type	Varchar(20)	NOT NULL
Type_Description	Varchar(500)	
<b>Manufacturers</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Manufacturer_ID	Int(5)	Primary Key
Manufacturer_Name	Varchar(50)	
Email	Varchar(300)	
<b>Models</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Model_ID	Int(5)	Primary Key

Type_ID	int(5)	Foreign Key
Manufacturer_ID	Int(5)	Foreign Key
Model_Name	Varchar(100)	NOT NULL
Weight_Limit_lbs	Int(20)	
Seating	Int(3)	
Fuel_Capacity_Gal	Int(4)	NULLABLE
Model_Description	Varchar(9000)	
Img_URL	Varchar(500)	DEFAULT ( <a href="https://idea7.co.uk/wp-content/uploads/2021/02/placeholder-250x250-1.png">https://idea7.co.uk/wp-content/uploads/2021/02/placeholder-250x250-1.png</a> )
<b>BoatStatus</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Status_ID	Int(5)	Primary Key
Status	Varchar(50)	NOT NULL
<b>Boats</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Registration	Varchar(100)	Primary Key
Synthetic_Key	Int(11)	Unique
Model_ID	Int(5)	Foreign Key
Status_ID	Int(5)	Foreign Key

Figure C.4 DICTIONARY for “Edit Watercraft”.

<b>BoatStatus</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Status_ID	Int(5)	Primary Key
Status	Varchar(50)	NOT NULL
<b>Boats</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Registration_Number	Varchar(100)	Primary Key
Status_ID	Int(5)	Foreign Key

Figure C.5 DATA DICTIONARY for “Group Watercraft by Status”.

<b>WaterCraftTypes</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Type_ID	int(5)	Primary Key
Type	Varchar(20)	NOT NULL

<b>Models</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Model_ID	Int(5)	Primary Key
Type_ID	int(5)	Foreign Key
Model_Name	Varchar(100)	NOT NULL
<b>BoatStatus</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Status_ID	Int(5)	Primary Key
Status	Varchar(50)	NOT NULL
<b>Boats</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Registration_Number	Varchar(100)	Primary Key
Model_ID	Int(5)	Foreign Key
Status_ID	Int(5)	Foreign Key

Figure C.6 DATA DICTIONARY for “Search for Watercraft”.

<b>WaterCraftTypes</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Type_ID	int(5)	Primary Key
Type	Varchar(20)	NOT NULL
<b>Manufacturers</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Manufacturer_ID	Int(5)	Primary Key
Manufacturer_Name	Varchar(50)	NOT NULL
Email	Varchar(300)	
<b>Models</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Model_ID	Int(5)	Primary Key
Type_ID	int(5)	Foreign Key
Manufacturer_ID	Int(5)	Foreign Key
Model_Name	Varchar(100)	NOT NULL
Weight_Limit(lbs)	Int(20)	
Seating	Int(3)	
Fuel_Capacity (Gal)	Int(4)	Nullable



Model_Description	Varchar(9000)	
Img_URL	Varchar(500)	
<b>BoatStatus</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Status_ID	Int(5)	Primary Key
Status	Varchar(50)	NOT NULL
<b>Boats</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Registration_Number	Varchar(100)	Primary Key
Model_ID	Int(5)	Foreign Key
Status_ID	Int(5)	Foreign Key
Synthetic_Key	Int(11)	Unique
<b>Booking</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Booking_Number	Int(5)	Primary Key
Date	DateTime	
<b>BoatReservation</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
BoatReservation_ID	Int(5)	Primary Key
Registration_Number	Varchar(100)	Foreign Key
Booking_Number	Int(5)	Foreign Key
<b>ItemStatus</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Item_Status_ID	Int(5)	Primary Key
Item_Status	Varchar(50)	
<b>ItemType</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Item_Type_ID	Int(5)	Primary key
Item_Type	Varchar(100)	
<b>Item</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Item_ID	Int(5)	Primary Key
Item_Status_ID	Int(5)	Foreign Key

Item_Type_ID	Int(5)	Foreign Key
Item_Name	Varchar(100)	
<b>ItemReservation</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Item_ID	Int(5)	Foreign Key (PK)
BoatReservation_ID	Int(5)	Foreign Key (PK)

Figure C.7 DATA DICTIONARY for “View Watercraft Details”.

<b>Jobs</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Job_ID	Int(5)	Primary Key
Job_Name	Varchar(100)	
<b>EmployeeStatus</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Employee_Status_ID	Int(5)	Primary Key
Employee_Status	Varchar(50)	
<b>Employee</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Employee_ID	Int(5)	Primary Key
Job_ID	Int(5)	Foreign Key
Employee_Status_ID	Int(5)	Foreign Key
Employee_Name	Varchar(100)	

Figure C.8 DATA DICTIONARY for “View All Employees”.

<b>Jobs</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Job_ID	Int(5)	Primary Key
Job_Name	Varchar(100)	
<b>EmployeeStatus</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Employee_Status_ID	Int(5)	Primary Key
Employee_Status	Varchar(50)	
<b>Employee</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Employee_ID	Int(5)	Primary Key

Job_ID	Int(5)	Foreign Key
Employee_Status_ID	Int(5)	Foreign Key
Employee_Name	Varchar(100)	
Start_Date	Date	
NIN	Varchar(9)	NULLABLE
Phone_Number	Varchar(11)	
Email	Varchar(300)	
<b>Booking</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Booking_Number	Int(5)	Primary Key
Date	DateTime	
<b>BoatReservation</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
BoatReservation_ID	Int(5)	Primary Key
Registration_Number	Varchar(100)	Foreign Key
Booking_Number	Int(5)	Foreign Key
<b>EmployeeBoatReservation</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Employee_ID	Int(5)	Foreign Key (PK)
BoatReservation_ID	Int(5)	Foreign Key (PK)

Figure C.9 DATA DICTIONARY for “View Employee profiles/details” and “View my Profile”.

<b>Employee</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Employee_ID	Int(5)	Primary Key
<b>QualificationType</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Qualification_Type_ID	Int(5)	Primary Key
Type_Name	Varchar(100)	
<b>Qualification</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Qualification_ID	Int(5)	Primary key
Qualification_Type_ID	Int(5)	Foreign Key
Qualification_Name	Varchar(100)	

<i>EmployeeQualification</i>		
<i>Field Name</i>	<i>Data Type</i>	<i>Constraints</i>
Employee_ID	Int(5)	Foreign Key (PK)
Qualification_ID	Int(5)	Foreign Key (PK)
Award_Date	Date	
Qualification_Expiration	Date	
Qualification_DOC_SCAN	PDF	
Verification_ID	Varchar(100)	

Figure C.10 DATA DICTIONARY for “Add Qualification to Profile”.

<i>QualificationType</i>		
<i>Field Name</i>	<i>Data Type</i>	<i>Constraints</i>
Qualification_Type_ID	Int(5)	Primary Key
Type_Name	Varchar(100)	

<i>Qualification</i>		
<i>Field Name</i>	<i>Data Type</i>	<i>Constraints</i>
Qualification_ID	Int(5)	Primary key
Qualification_Type_ID	Int(5)	Foreign Key
Qualification_Name	Varchar(100)	

<i>Employee</i>		
<i>Field Name</i>	<i>Data Type</i>	<i>Constraints</i>
Employee_ID	Int(5)	Primary Key

<i>EmployeeQualification</i>		
<i>Field Name</i>	<i>Data Type</i>	<i>Constraints</i>
Employee_ID	Int(5)	Foreign Key (PK)
Qualification_ID	Int(5)	Foreign Key (PK)
Award_Date	Date	
Qualification_Expiration	Date	
Qualification_DOC_SCAN	PDF	
Verification_ID	Varchar(100)	

Figure C.11 DATA DICTIONARY for “View Employee Qualifications” and “View my Qualifications”.

<i>Employee</i>		
<i>Field Name</i>	<i>Data Type</i>	<i>Constraints</i>
Employee_ID	Int(5)	Primary Key

<b>QualificationType</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Qualification_Type_ID	Int(5)	Primary Key
Type_Name	Varchar(100)	
<b>Qualification</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Qualification_ID	Int(5)	Primary key
Qualification_Type_ID	Int(5)	Foreign Key
Qualification_Name	Varchar(100)	
<b>EmployeeQualification</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Employee_ID	Int(5)	Foreign Key (PK)
Qualification_ID	Int(5)	Foreign Key (PK)
Award_Date	Date	
Qualification_Expiration	Date	
Qualification_DOC_SCAN	PDF	
Verification_ID	Varchar(100)	

Figure C.12 DATA DICTIONARY for “Edit my Qualifications”.

<b>Employee</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Employee_ID	Int(5)	Primary Key
Job_ID	Int(5)	Foreign Key
Employee_Status_ID	Int(5)	Foreign Key
Employee_Name	Varchar(100)	
Start_Date	Date	
NIN	Varchar(9)	NULLABLE
Phone_Number	Varchar(11)	
Email	Varchar(300)	
Username	Binary	Not Null
Password	Binary	Not Null

Figure C.13 DATA DICTIONARY for “Edit my Profile”.

<b>Employee</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>

Employee_ID	Int(5)	Primary Key
Job_ID	Int(5)	Foreign Key
Employee_Status_ID	Int(5)	Foreign Key
Employee_Name	Varchar(100)	
Start_Date	Date	
NIN	Varchar(9)	NULLABLE
Phone_Number	Varchar(11)	
Email	Varchar(300)	

Figure C.14 DATA DICTIONARY for “Search for Employee”.

<b>Jobs</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Job_ID	Int(5)	Primary Key
Job_Name	Varchar(100)	
<b>Employee</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Employee_ID	Int(5)	Primary Key
Job_ID	Int(5)	Foreign Key
Employee_Name	Varchar(100)	
Start_Date	Date	
NIN	Varchar(9)	NULLABLE
Phone_Number	Varchar(11)	
Email	Varchar(300)	
Username	Binary	Not Null
Password	Binary	Not Null

Figure C.15 DATA DICTIONARY for “Create Employee Account”.

<b>Employee</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Employee_ID	Int(5)	Primary key
Employee_Name	Varchar(100)	
Username	Binary	Not Null
Password	Binary	Not Null

Figure C.16 DATA DICTIONARY for “Delete Employee Account”.

<b>CharterType</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Charter_Type_ID	Int(5)	Primary Key
Charter_Name	Varchar(100)	
Charter_Description	Varchar(9000)	
<b>Booking</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Booking_Number	Int(5)	Primary Key
Charter_Type_ID	Int(5)	Foreign Key
Date	DateTime	
<b>Guest</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Guest_ID	Int(5)	Primary key
Booking_Number	Int(5)	Foreign Key
Guest_Name	Varchar(50)	
isPrincipalGuest?	Boolean	Default(false)
<b>Boats</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Registration_Number	Varchar(100)	Primary Key
<b>BoatReservation</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
BoatReservation_ID	Int(5)	Primary Key
Registration_Number	Varchar(100)	Foreign Key
Booking_Number	Int(5)	Foreign Key

Figure C.17 DATA DICTIONARY for “View all Bookings”.

<b>WaterCraftTypes</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Type_ID	int(5)	Primary Key
Type	Varchar(20)	NOT NULL
<b>Models</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Model_ID	Int(5)	Primary Key
Type_ID	int(5)	Foreign Key

<b>BoatStatus</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Status_ID	Int(5)	Primary Key
Status	Varchar(50)	NOT NULL
<b>Boats</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Registration_Number	Varchar(100)	Primary Key
Model_ID	Int(5)	Foreign Key
Status_ID	Int(5)	Foreign Key
<b>CharterType</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Charter_Type_ID	Int(5)	Primary Key
Charter_Name	Varchar(100)	
Charter_Description	Varchar(9000)	
<b>Booking</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Booking_Number	Int(5)	Primary Key
Charter_Type_ID	Int(5)	Foreign Key
Booking_Notes	Varchar(3000)	
Duration	Time	
Date	DateTime	
<b>Guest</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Guest_ID	Int(5)	Primary key
Booking_Number	Int(5)	Foreign Key
Guest_Name	Varchar(50)	
isPrincipalGuest?	Boolean	Default(false)
<b>BoatReservation</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
BoatReservation_ID	Int(5)	Primary Key
Registration_Number	Varchar(100)	Foreign Key
Booking_Number	Int(5)	Foreign Key
<b>EmployeeStatus</b>		



<i>Field Name</i>	<i>Data Type</i>	<i>Constraints</i>
Employee_Status_ID	Int(5)	Primary Key
Employee_Status	Varchar(50)	
<b>Employee</b>		
<i>Field Name</i>	<i>Data Type</i>	<i>Constraints</i>
Employee_ID	Int(5)	Primary Key
Employee_Status_ID	Int(5)	Foreign Key
Employee_Name	Varchar(100)	
<b>EmployeeBoatReservation</b>		
<i>Field Name</i>	<i>Data Type</i>	<i>Constraints</i>
Employee_ID	Int(5)	Foreign Key (PK)
BoatReservation_ID	Int(5)	Foreign Key (PK)

Figure C.18 DATA DICTIONARY for “View Booking Details”.

<b>Waiver</b>		
<i>Field Name</i>	<i>Data Type</i>	<i>Constraints</i>
Waiver_ID	Int(5)	Primary Key
Waiver_Details	Varchar(3000)	
<b>Guest</b>		
<i>Field Name</i>	<i>Data Type</i>	<i>Constraints</i>
Guest_ID	Int(5)	Primary key
Guest_Name	Varchar(50)	
<b>GuestWaiver</b>		
<i>Field Name</i>	<i>Data Type</i>	<i>Constraints</i>
Guest_ID	Int(5)	Foreign Key (PK)
Waiver_ID	Int(5)	Foreign Key (PK)
isSigned?	Boolean	

Figure C.19 DATA DICTIONARY for “View Waivers”.

<b>Booking</b>		
<i>Field Name</i>	<i>Data Type</i>	<i>Constraints</i>
Booking_Number	Int(5)	Primary Key
Date	DateTime	

Figure C.20 DATA DICTIONARY for “Search Booking by ID”.

<b>Booking</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Booking_Number	Int(5)	Primary Key
Date	DateTime	

Figure C.21 DATA DICTIONARY for “Search Booking by Date”.

<b>Employee</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Employee_ID	Int(5)	Primary Key
Employee_Name	Varchar(100)	
<b>Booking</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Booking_Number	Int(5)	Primary Key
Date	DateTime	
<b>BoatReservation</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
BoatReservation_ID	Int(5)	Primary Key
Booking_Number	Int(5)	Foreign Key
<b>EmployeeBoatReservation</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Employee_ID	Int(5)	Foreign Key (PK)
BoatReservation_ID	Int(5)	Foreign Key (PK)

Figure C.22 DATA DICTIONARY for “Remove/ Add Crew Member from (to) Booking”.

<b>Booking</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Booking_Number	Int(5)	Primary Key
Date	DateTime	

Figure C.23 DATA DICTIONARY for “Request Change Booking Time(Owner)”.

<b>ItemStatus</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Item_Status_ID	Int(5)	Primary Key
Item_Status	Varchar(50)	
<b>ItemType</b>		

<i><b>Field Name</b></i>	<i><b>Data Type</b></i>	<i><b>Constraints</b></i>
Item_Type_ID	Int(5)	Primary key
Item_Type	Varchar(100)	
Type_Description	Varchar(3000)	
<i><b>Item</b></i>		
<i><b>Field Name</b></i>	<i><b>Data Type</b></i>	<i><b>Constraints</b></i>
Item_ID	Int(5)	Primary Key
Item_Status_ID	Int(5)	Foreign Key
Item_Type_ID	Int(5)	Foreign Key
Item_Name	Varchar(100)	

Figure C.24 DATA DICTIONARY for “View Inventory”.

<i><b>ItemType</b></i>		
<i><b>Field Name</b></i>	<i><b>Data Type</b></i>	<i><b>Constraints</b></i>
Item_Type_ID	Int(5)	Primary key
Item_Type	Varchar(100)	
Type_Description	Varchar(3000)	
<i><b>Item</b></i>		
<i><b>Field Name</b></i>	<i><b>Data Type</b></i>	<i><b>Constraints</b></i>
Item_ID	Int(5)	Primary Key
Item_Status_ID	Int(5)	Foreign Key
Item_Type_ID	Int(5)	Foreign Key
Item_Name	Varchar(100)	

Figure C.25 DATA DICTIONARY for “Add Item to Inventory”.

<i><b>ItemStatus</b></i>		
<i><b>Field Name</b></i>	<i><b>Data Type</b></i>	<i><b>Constraints</b></i>
Item_Status_ID	Int(5)	Primary Key
Item_Status	Varchar(50)	
<i><b>ItemType</b></i>		
<i><b>Field Name</b></i>	<i><b>Data Type</b></i>	<i><b>Constraints</b></i>
Item_Type_ID	Int(5)	Primary key
Item_Type	Varchar(100)	
<i><b>Item</b></i>		
<i><b>Field Name</b></i>	<i><b>Data Type</b></i>	<i><b>Constraints</b></i>

Item_ID	Int(5)	Primary Key
Item_Status_ID	Int(5)	Foreign Key
Item_Type_ID	Int(5)	Foreign Key
Item_Name	Varchar(100)	

Figure C.26 DATA DICTIONARY for “Remove Item from Inventory”.

<i>ItemType</i>		
<i>Field Name</i>	<i>Data Type</i>	<i>Constraints</i>
Item_Type_ID	Int(5)	Primary key
Item_Type	Varchar(100)	
Type_Description	Varchar(3000)	
<i>Item</i>		
<i>Field Name</i>	<i>Data Type</i>	<i>Constraints</i>
Item_ID	Int(5)	Primary Key
Item_Status_ID	Int(5)	Foreign Key
Item_Type_ID	Int(5)	Foreign Key
Item_Name	Varchar(100)	

Figure C.27 DATA DICTIONARY for “Edit Item”.

<i>ItemStatus</i>		
<i>Field Name</i>	<i>Data Type</i>	<i>Constraints</i>
Item_Status_ID	Int(5)	Primary Key
Item_Status	Varchar(50)	
<i>ItemType</i>		
<i>Field Name</i>	<i>Data Type</i>	<i>Constraints</i>
Item_Type_ID	Int(5)	Primary key
Item_Type	Varchar(100)	
<i>Item</i>		
<i>Field Name</i>	<i>Data Type</i>	<i>Constraints</i>
Item_ID	Int(5)	Primary Key
Item_Status_ID	Int(5)	Foreign Key
Item_Type_ID	Int(5)	Foreign Key
Item_Name	Varchar(100)	
<i>Boats</i>		
<i>Field Name</i>	<i>Data Type</i>	<i>Constraints</i>

Registration_Number	Varchar(100)	Primary Key
<b>Booking</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Booking_Number	Int(5)	Primary Key
Date	DateTime	
<b>BoatReservation</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
BoatReservation_ID	Int(5)	Primary Key
Registration_Number	Varchar(100)	Foreign Key
Booking_Number	Int(5)	Foreign Key
<b>ItemReservation</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Item_ID	Int(5)	Foreign Key (PK)
BoatReservation_ID	Int(5)	Foreign Key (PK)

Figure C.28 DATA DICTIONARY for "Assign Item to Watercraft".

## Appendix D – System Data Dictionary

<b>WaterCraftTypes</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Type_ID	int(5)	Primary Key
Type	Varchar(20)	NOT NULL
Type_Description	Varchar(500)	
<b>Manufacturers</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Manufacturer_ID	Int(5)	Primary Key
Manufacturer_Name	Varchar(50)	
Email	Varchar(300)	
<b>Models</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Model_ID	Int(5)	Primary Key
Type_ID	int(5)	Foreign Key
Manufacturer_ID	Int(5)	Foreign Key
Model_Name	Varchar(100)	NOT NULL
Weight_Limit_lbs	Int(20)	
Seating	Int(3)	
Fuel_Capacity_Gal	Int(4)	NULLABLE
Model_Description	Varchar(9000)	
Img_URL	Varchar(500)	DEFAULT ( <a href="https://idea7.co.uk/wp-content/uploads/2021/02/placeholder-250x250-1.png">https://idea7.co.uk/wp-content/uploads/2021/02/placeholder-250x250-1.png</a> )
<b>BoatStatus</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Status_ID	Int(5)	Primary Key
Status	Varchar(50)	NOT NULL
<b>Boats</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Registration	Varchar(100)	Primary Key
Synthetic_Key	Int(11)	Unique
Model_ID	Int(5)	Foreign Key
Status_ID	Int(5)	Foreign Key
<b>BoatReservation</b>		

<i><b>Field Name</b></i>	<i><b>Data Type</b></i>	<i><b>Constraints</b></i>
BoatReservation_ID	Int(5)	Primary Key
Registration_Number	Varchar(100)	Foreign Key
Booking_Number	Int(5)	Foreign Key
<i><b>CharterType</b></i>		
<i><b>Field Name</b></i>	<i><b>Data Type</b></i>	<i><b>Constraints</b></i>
Charter_Type_ID	Int(5)	Primary Key
Charter_Name	Varchar(100)	
Charter_Description	Varchar(9000)	
<i><b>Booking</b></i>		
<i><b>Field Name</b></i>	<i><b>Data Type</b></i>	<i><b>Constraints</b></i>
Booking_Number	Int(5)	Primary Key
Charter_Type_ID	Int(5)	Foreign Key
Booking_Notes	Varchar(3000)	
Duration	Time	
Date	DateTime	
<i><b>Guest</b></i>		
<i><b>Field Name</b></i>	<i><b>Data Type</b></i>	<i><b>Constraints</b></i>
Guest_ID	Int(5)	Primary key
Booking_Number	Int(5)	Foreign Key
Guest_Name	Varchar(50)	
isPrincipalGuest?	Boolean	Default(false)
<i><b>Waiver</b></i>		
<i><b>Field Name</b></i>	<i><b>Data Type</b></i>	<i><b>Constraints</b></i>
Waiver_ID	Int(5)	Primary Key
Waiver_Details	Varchar(3000)	
<i><b>GuestWaiver</b></i>		
<i><b>Field Name</b></i>	<i><b>Data Type</b></i>	<i><b>Constraints</b></i>
Guest_ID	Int(5)	Foreign Key (PK)
Waiver_ID	Int(5)	Foreign Key (PK)
isSigned?	Boolean	
<i><b>ItemStatus</b></i>		
<i><b>Field Name</b></i>	<i><b>Data Type</b></i>	<i><b>Constraints</b></i>

Item_Status_ID	Int(5)	Primary Key
Item_Status	Varchar(50)	
<b>ItemType</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Item_Type_ID	Int(5)	Primary key
Item_Type	Varchar(100)	
<b>Item</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Item_ID	Int(5)	Primary Key
Item_Status_ID	Int(5)	Foreign Key
Item_Type_ID	Int(5)	Foreign Key
Item_Name	Varchar(100)	
<b>ItemReservation</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Item_ID	Int(5)	Foreign Key (PK)
BoatReservation_ID	Int(5)	Foreign Key (PK)
<b>Jobs</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Job_ID	Int(5)	Primary Key
Job_Name	Varchar(100)	
<b>EmployeeStatus</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Employee_Status_ID	Int(5)	Primary Key
Employee_Status	Varchar(50)	
<b>Employee</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Employee_ID	Int(5)	Primary Key
Job_ID	Int(5)	Foreign Key
Employee_Status_ID	Int(5)	Foreign Key
Employee_Name	Varchar(100)	
Start_Date	Date	
NIN	Varchar(9)	NULLABLE
Phone_Number	Varchar(11)	

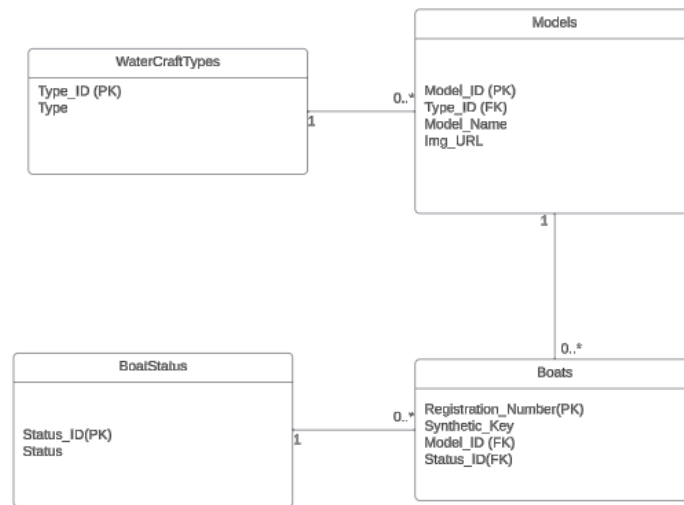


Email	Varchar(300)	
Username	Binary	Not Null
Password	Binary	Not Null
<b>QualificationType</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Qualification_Type_ID	Int(5)	Primary Key
Type_Name	Varchar(100)	
<b>Qualification</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Qualification_ID	Int(5)	Primary key
Qualification_Type_ID	Int(5)	Foreign Key
Qualification_Name	Varchar(100)	
<b>EmployeeQualification</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Employee_ID	Int(5)	Foreign Key (PK)
Qualification_ID	Int(5)	Foreign Key (PK)
Award_Date	Date	
Qualification_Expiration	Date	
Qualification_DOC_SCAN	PDF	
Verification_ID	Varchar(100)	
<b>EmployeeBoatReservation</b>		
<b>Field Name</b>	<b>Data Type</b>	<b>Constraints</b>
Employee_ID	Int(5)	Foreign Key (PK)
BoatReservation_ID	Int(5)	Foreign Key (PK)

Figure D.1 DATA DICTIONARY for The System

## Appendix E – ERD Validation

### 1. View Fleet



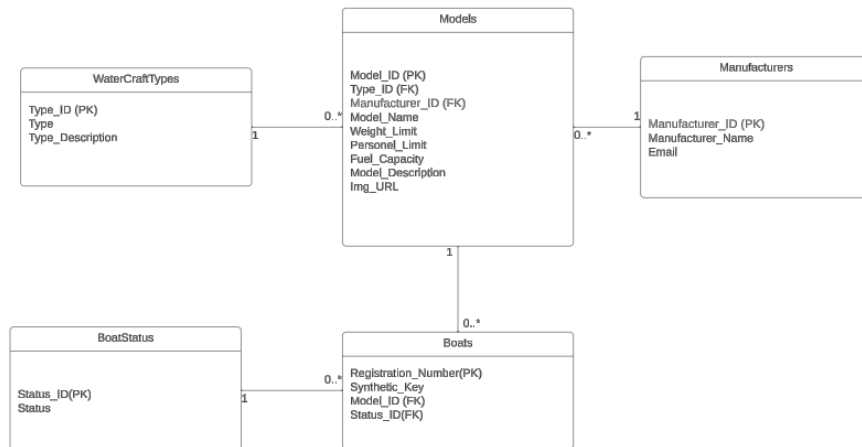
```

SELECT boats.Synthetic_Key,boats.Registration, boats.Boat_Img,
models.Model_Name, models.Img_URL, watercrafttypes.Type,
boatstatus.Status, models.Model_ID, boatstatus.Status_ID

FROM boats LEFT JOIN models ON boats.Model_ID = models.Model_ID
      LEFT JOIN watercrafttypes ON models.Type_ID = watercrafttypes.Type_ID
      LEFT JOIN boatstatus ON boats.Status_ID = boatstatus.Status_ID
    
```

Figure E.1 Validation for "View Fleet"

### 2. Search For Watercraft



```

SELECT boats.Synthetic_Key,boats.Registration, boats.Boat_Img,
models.Model_Name, manufacturers.Manufacturer_Name, models.Img_URL,
watercrafttypes.Type, boatstatus.Status, models.Model_ID,
boatstatus.Status_ID

FROM boats LEFT JOIN models ON boats.Model_ID = models.Model_ID
      LEFT JOIN manufacturers ON models.Manufacturer_ID =
manufacturers.Manufacturer_ID
      LEFT JOIN watercrafttypes ON models.Type_ID = watercrafttypes.Type_ID
      LEFT JOIN boatstatus ON boats.Status_ID = boatstatus.Status_ID

WHERE boats.Registration = ?
    
```

Figure E.2 Validation for “Search for Watercraft”

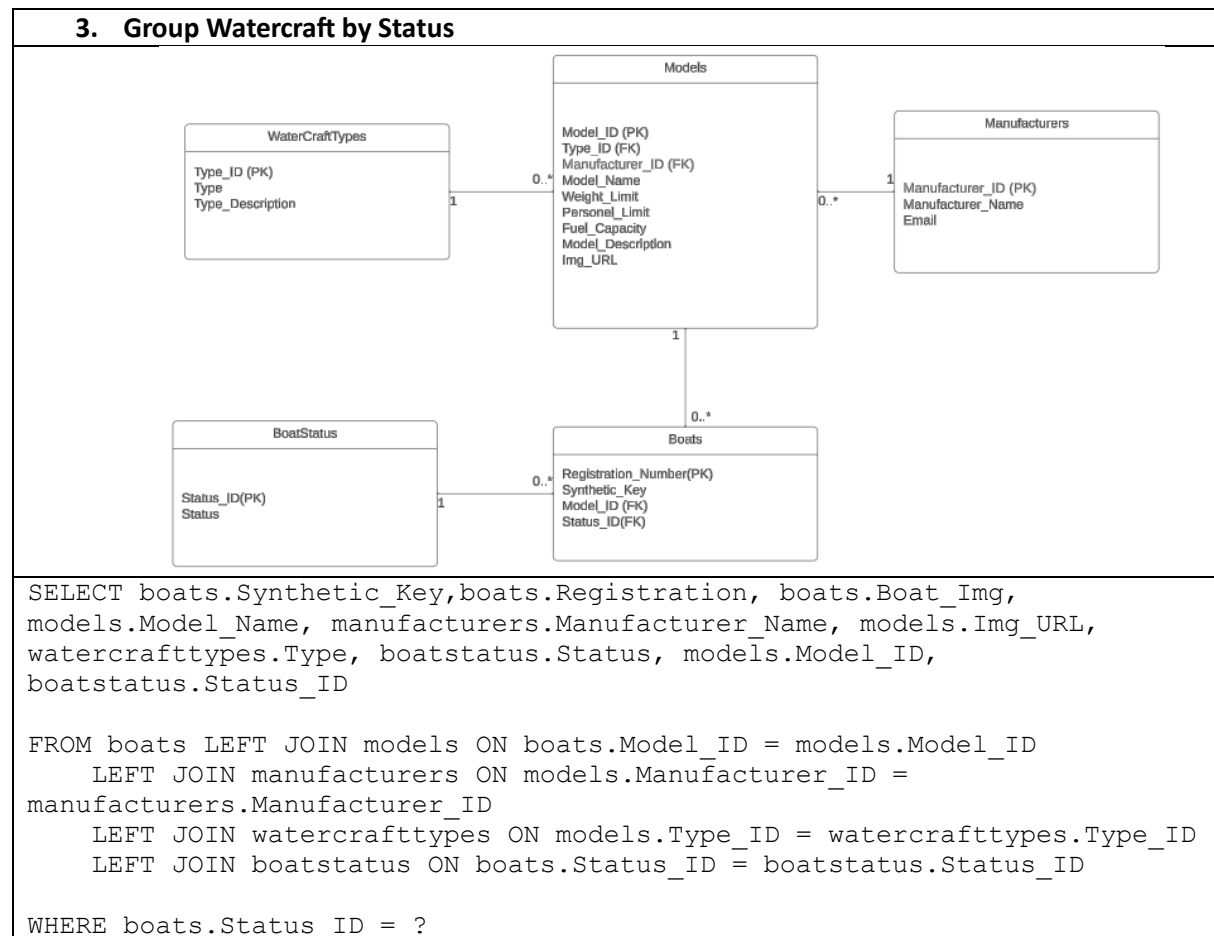
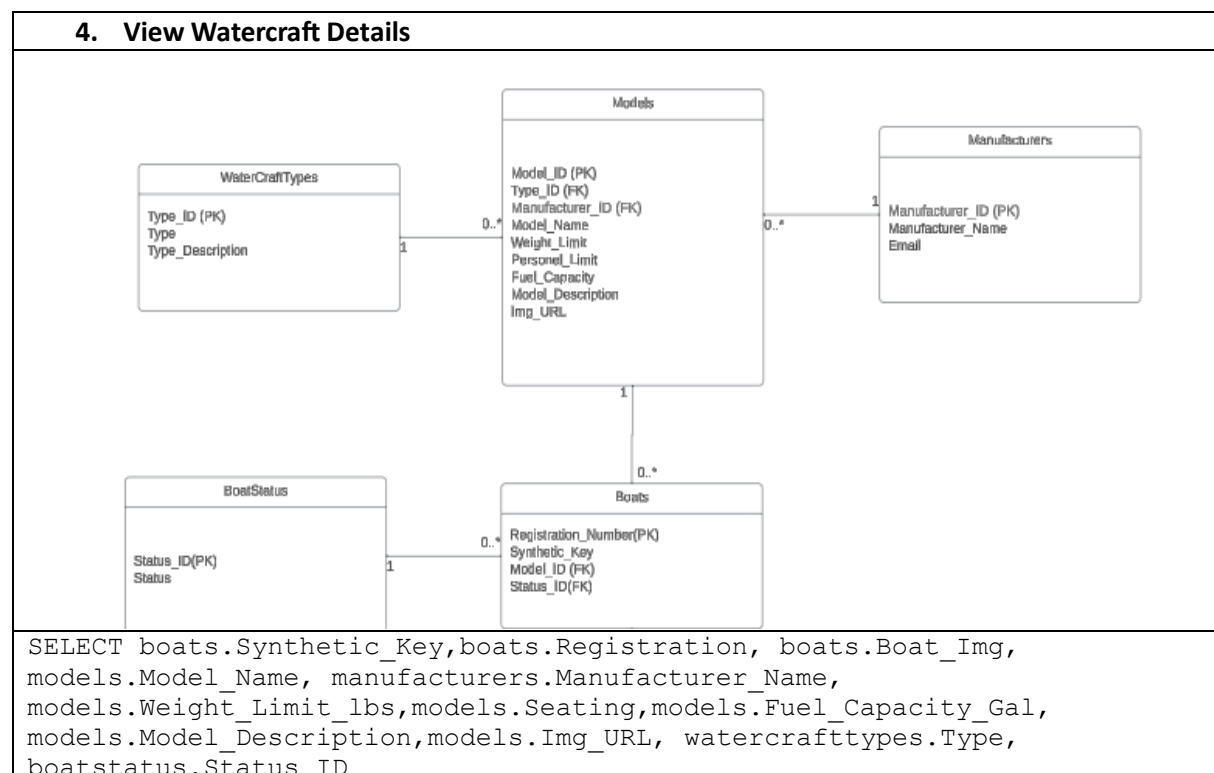


Figure E.3 Validation for “Group Watercraft by Status”



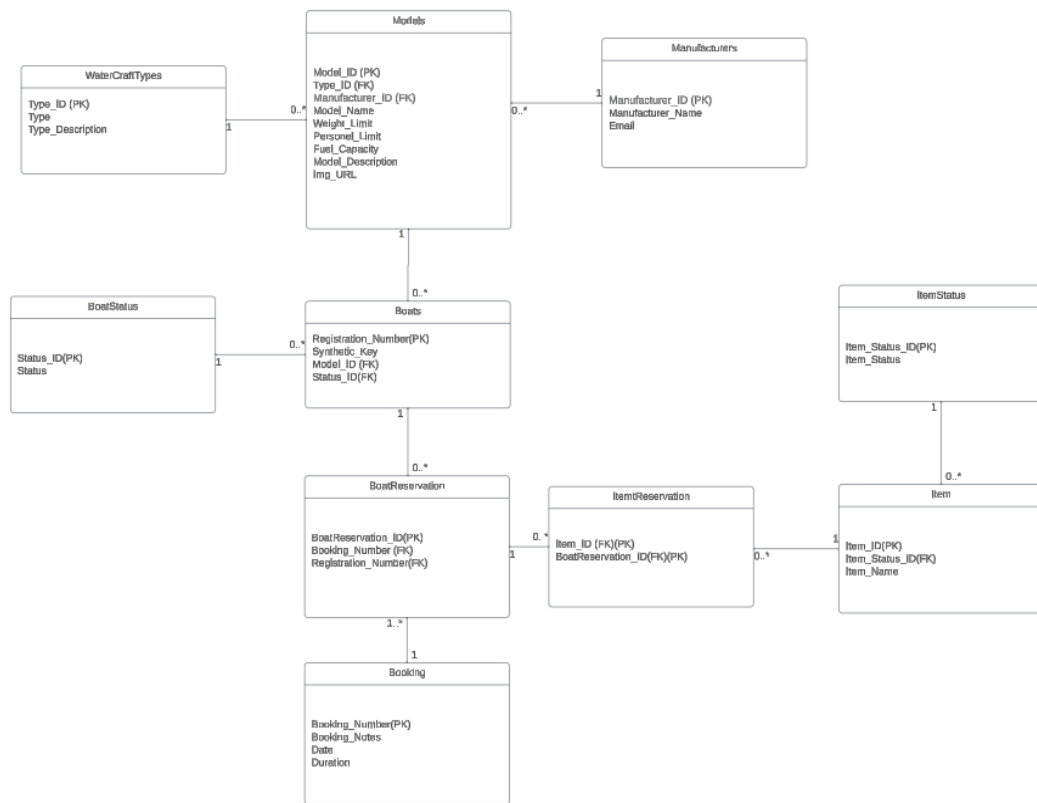
```

FROM boats LEFT JOIN models ON boats.Model_ID = models.Model_ID
      LEFT JOIN manufacturers ON models.Manufacturer_ID =
manufacturers.Manufacturer_ID
      LEFT JOIN watercrafttypes ON models.Type_ID = watercrafttypes.Type_ID
      LEFT JOIN boatstatus ON boats.Status_ID = boatstatus.Status_ID
WHERE boats.Synthetic_key = ?

```

Figure E.4 Validation for “View Watercraft Details”

## 5. View the Watercraft Booking Schedule



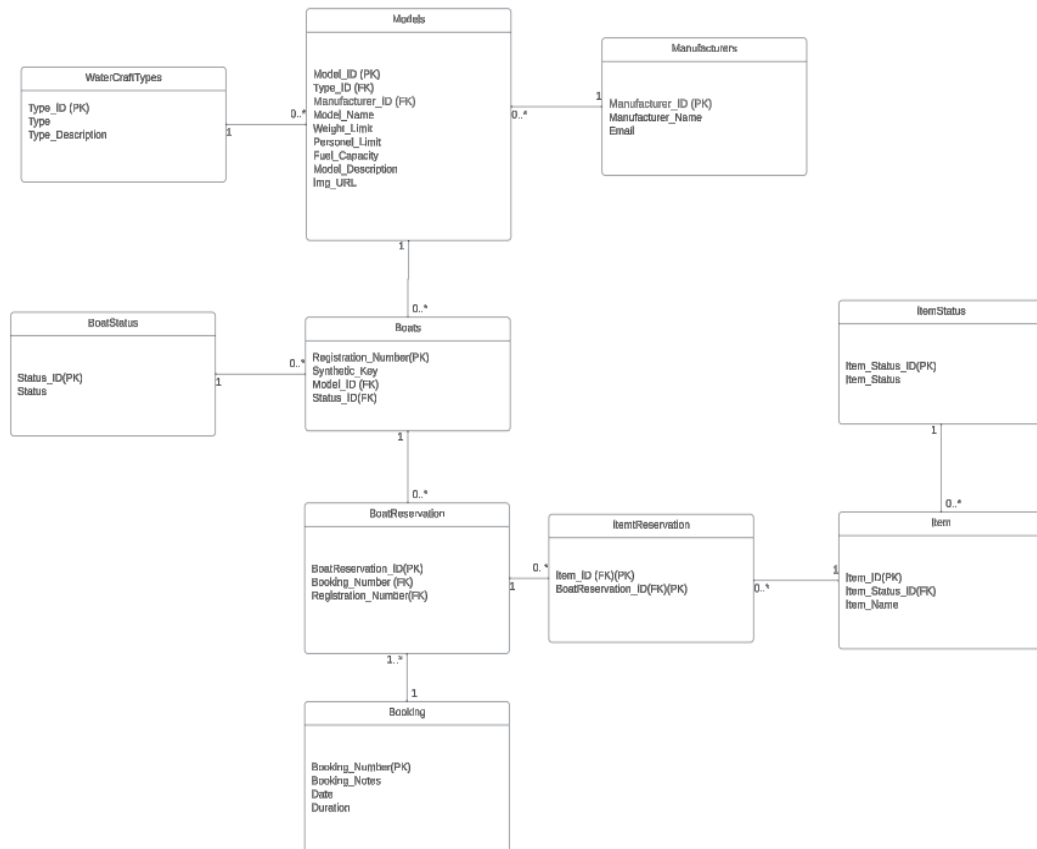
```

SELECT bookings.Booking_Notes, bookings.Date, bookings.Duration,
bookings.Charter_Type_ID
FROM boats LEFT JOIN boatreservation ON boats.Registration =
boatreservation.Registration
LEFT JOIN bookings ON boatreservation.Booking_Number =
bookings.Booking_Number
LEFT JOIN chartertype ON bookings.Charter_Type_ID =
chartertype.Charter_Type_ID
WHERE boats.Synthetic_key = ?;

```

Figure E.5 Validation for “Watercraft Booking Schedule”

## 6. View Items that are on Watercraft

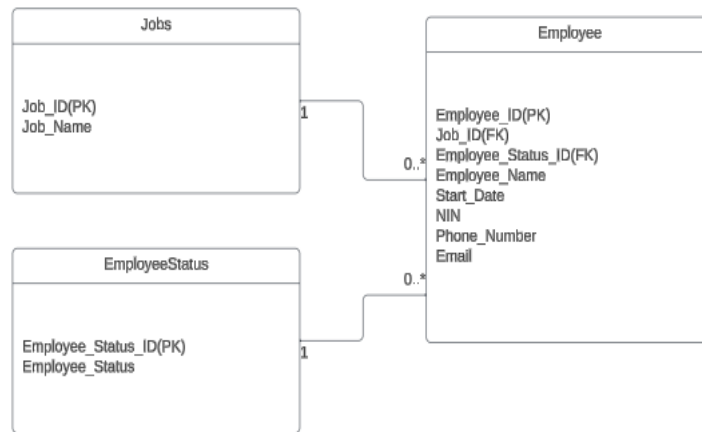


```

SELECT item.Item_Name,itemstatus.Item_Status
FROM boats LEFT JOIN boatreservation ON boats.Registration =
boatreservation.Registration
LEFT JOIN itemreservation ON boatreservation.BoatReservation_ID =
itemreservation.BoatReservation_ID
LEFT JOIN item ON itemreservation.Item_ID = item.Item_ID
LEFT JOIN itemstatus ON item.Item_Status_ID = itemstatus.Item_Status_ID
WHERE boats.Synthetic_key = 1; LEFT JOIN watercrafttypes ON
models.Type_ID = watercrafttypes.Type_ID
LEFT JOIN boatstatus ON boats.Status_ID = boatstatus.Status_ID
WHERE boats.Synthetic_key = ?
    
```

Figure E.6 Validation for “View Items that are on Watercraft”

## 7. View all Employees

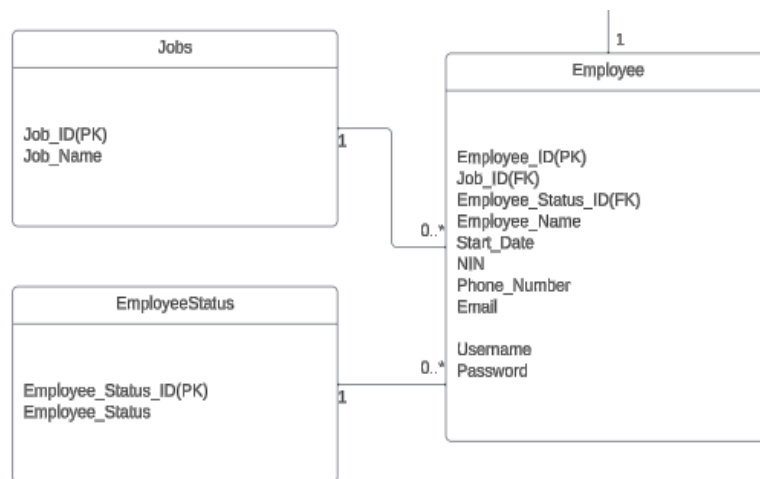


```
SELECT employee.Employee_ID, employee.Employee_Name,
employeeStatus.Employee_Status, jobs.Job_Name
```

```
FROM employee LEFT JOIN employeeStatus ON employee.Employee_Status_ID =
employeeStatus.Employee_Status_ID
LEFT JOIN jobs ON employee.Job_ID = jobs.Job_ID
```

Figure E.7 Validation for “View All Employees”

## 8. View Employee profiles/details and “View my Profile”

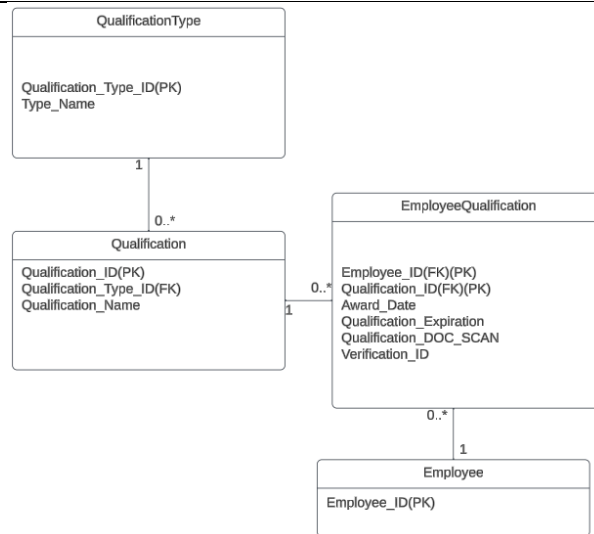


```
SELECT employee.Employee_ID, employee.Employee_Name, employee.Start_Date,
employee.NIN, employee.Phone_Number, employee.Email,
employeeStatus.Employee_Status, jobs.Job_Name
```

```
FROM employee LEFT JOIN employeeStatus ON employee.Employee_Status_ID =
employeeStatus.Employee_Status_ID
LEFT JOIN jobs ON employee.Job_ID = jobs.Job_ID
```

Figure E.8 Validation for “View Employee profiles/details” and “View my Profile”

## 9. View Employee Qualification



```

SELECT employeeQualification.Qualification_DOC_SCAN,
employeeQualification.Qualification_Expiration,
employeeQualification.Award_Date, employeeQualification.Veerification_ID
qualification.Qualification_Name, qualificationType.Type_Name
  
```

```

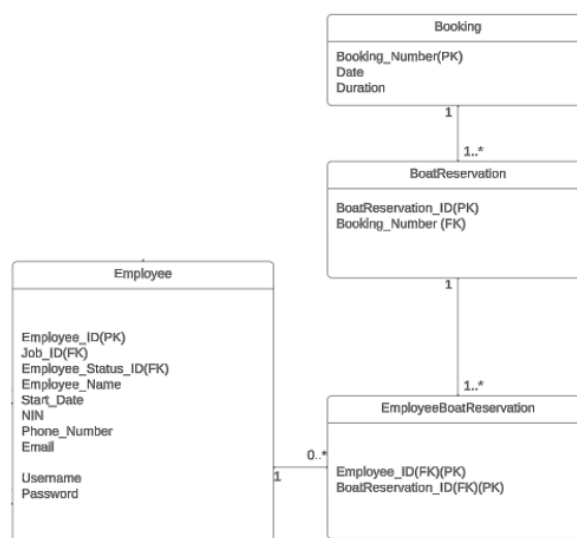
FROM employeeQualification LEFT JOIN employee ON
employeeQualification.Employee_ID = employee.Employee_ID LEFT JOIN
qualification ON employeeQualification.Qualification_ID =
qualification.Qualification_ID
LEFT JOIN qualificationType ON qualification.Qualification_Type_ID =
qualificationType.Qualification_Type_ID
  
```

```

WHERE employeeQualification.Employee_ID = ?
  
```

Figure E.9 Validation for "View Employee Qualification"

## 10. View Employee Schedule



```

SELECT bookings.Date, bookings.Booking_Number
  
```

```

FROM employeeBoatReservation LEFT JOIN employee ON
employeeBoatReservation.Employee_ID = employee.Employee_ID LEFT JOIN
  
```

```

boatReservation ON employeeBoatReservation.BoatReservation_ID =
boatReservation.BoatReservation_ID LEFT JOIN bookings ON
boatreservation.Booking_Number = bookings.Booking_Number

WHERE employeeBoatReservation.Employee_ID = ?

```

Figure E.10 Validation for “View Employee Schedule”

11. Search for Employee by Name	
	<div>Employee</div> <div> Employee_ID(PK)  Job_ID(FK)  Employee_Status_ID(FK)  Employee_Name  Start_Date  NIN  Phone_Number  Email    Username  Password </div>
<pre> SELECT employee.Employee_ID, employee.Employee_Name, employeeStatus.Employee_Status, jobs.Job_Name  FROM employee LEFT JOIN employeeStatus ON employee.Employee_Status_ID = employeeStatus.Employee_Status_ID LEFT JOIN jobs ON employee.Job_ID = jobs.Job_ID  WHERE employee.Employee_Name = ? </pre>	

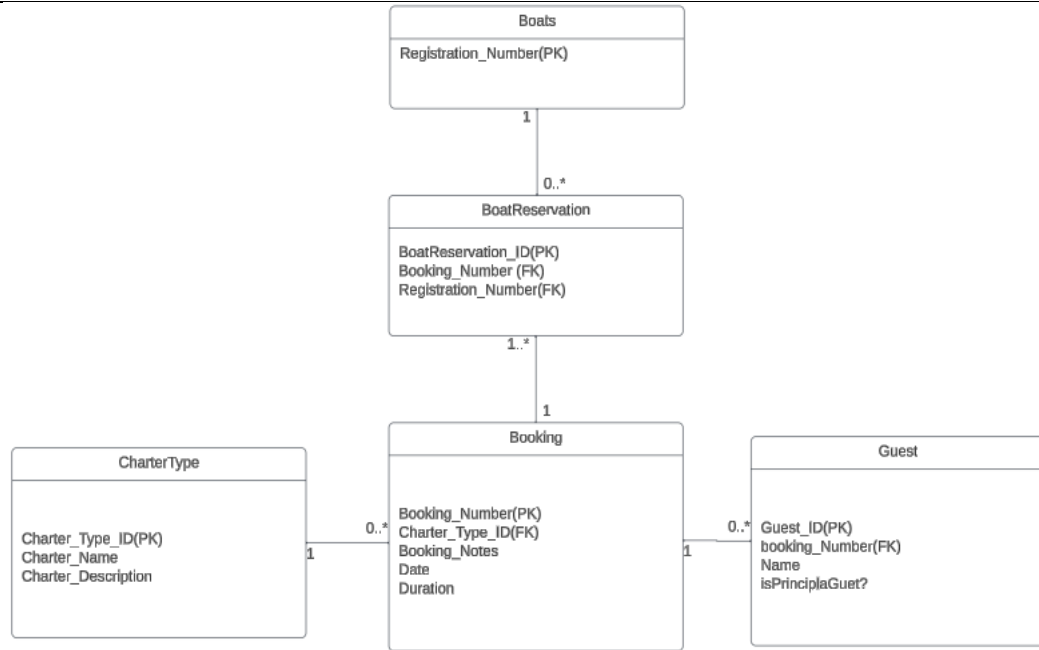
Figure E.11 Validation for “Search for Employee by Name”

12. Search for Employee by ID	
	<div>Employee</div> <div> Employee_ID(PK)  Job_ID(FK)  Employee_Status_ID(FK)  Employee_Name  Start_Date  NIN  Phone_Number  Email    Username  Password </div>
<pre> SELECT employee.Employee_ID, employee.Employee_Name, employeeStatus.Employee_Status, jobs.Job_Name  FROM employee LEFT JOIN employeeStatus ON employee.Employee_Status_ID = employeeStatus.Employee_Status_ID LEFT JOIN jobs ON employee.Job_ID = jobs.Job_ID  WHERE employee.Employee_ID = ? </pre>	

Figure E.12 Validation for “Search for Employee by ID”



### 13. View all Bookings



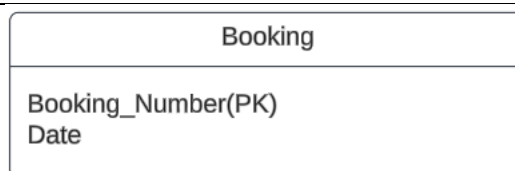
```

SELECT bookings.Date, bookings.Booking_Number, chartertype.Charter_Name,
boats.Registration, guest.Name

FROM boatreservation LEFT JOIN boats ON boatreservation.Registration =
boats.Registration
LEFT JOIN bookings ON boatreservation.Booking_Number =
bookings.Booking_Number
LEFT JOIN chartertype ON bookings.Charter_Type_ID =
chartertype.Charter_Type_ID
LEFT JOIN guest ON bookings.Booking Number = guest.Booking Number
    
```

Figure E.13 Validation for "View all Bookings"

### 14. Search Booking by ID



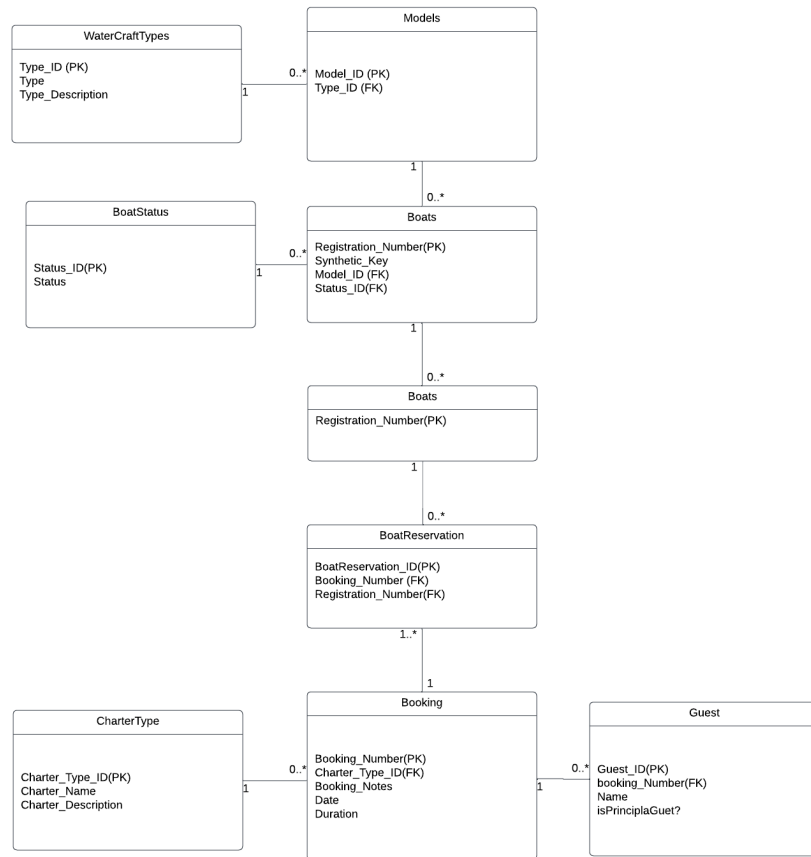
```

SELECT bookings.*

FROM bookings
WHERE bookings.Booking_Number = ?
    
```

Figure E.14 Validation for "Search Booking By ID"

## 15. View Booking Detail

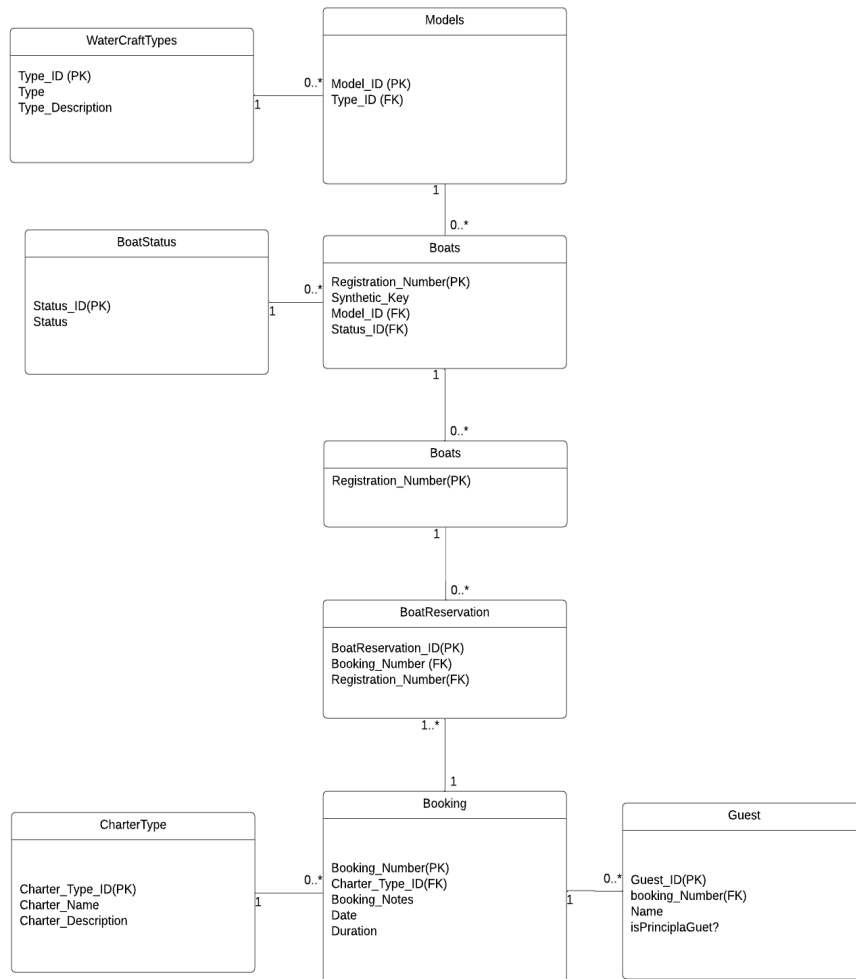


```
SELECT bookings.Date, bookings.Booking_Number, bookings.Duration,
bookings.Booking_Notes, chartertype.Charter_Name, boats.Registration,
guest.Name
```

```
FROM boatreservation LEFT JOIN boats ON boatreservation.Registration =
boats.Registration
LEFT JOIN bookings ON boatreservation.Booking_Number =
bookings.Booking_Number
LEFT JOIN chartertype ON bookings.Charter_Type_ID =
chartertype.Charter_Type_ID
LEFT JOIN guest ON bookings.Booking_Number = guest.Booking_Number
```

Figure E.15 Validation for "View Booking details"

## 16. View the watercraft reserved for a booking



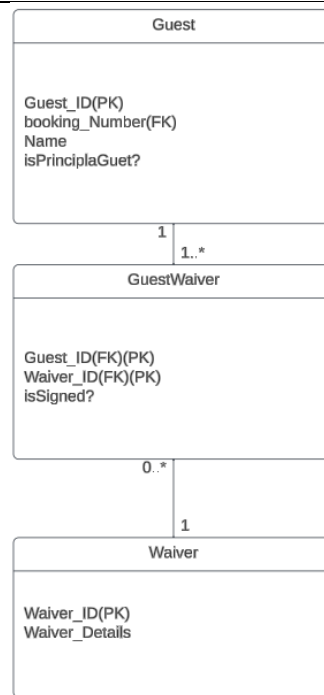
```
SELECT boats.Synthetic_Key,boats.Registration, boats.Boat_Img,
models.Model_Name, models.Img_URL, boatstatus.Status
```

```
FROM boats LEFT JOIN models ON boats.Model_ID = models.Model_ID
      LEFT JOIN watercrafttypes ON models.Type_ID =
watercrafttypes.Type_ID
      LEFT JOIN boatstatus ON boats.Status_ID = boatstatus.Status_ID
      LEFT JOIN boatReservation ON boats.Registration_Number =
boatReservation.Registration_Number
```

```
WHERE boatReservation.Booking_Number = ?
```

Figure E.16 Validation for "View the Watercraft reserved for the booking"

## 17. View Waivers

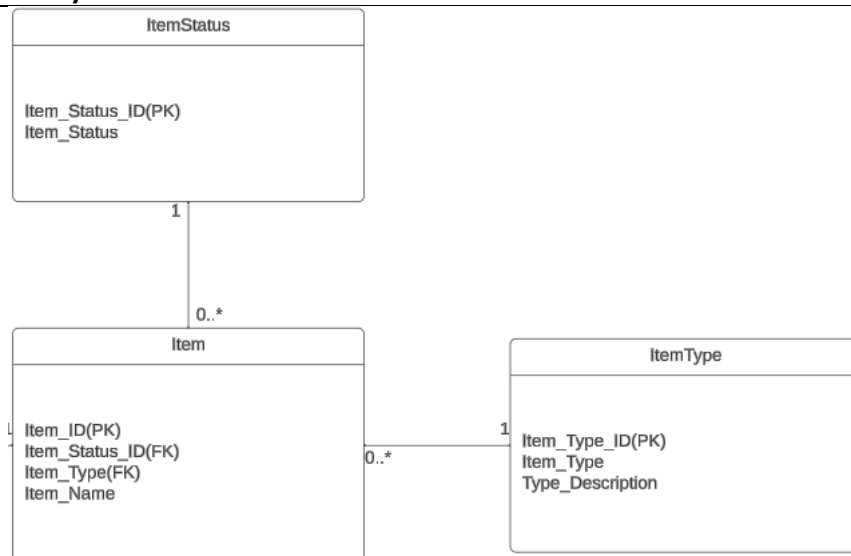


```

SELECT guest.Name, waiver.Waiver_Details
FROM guestWaiver LEFT JOIN guest ON guestWaiver.Guest_ID = guest.Guest_ID
LEFT JOIN waiver ON guestWaiver.Waiver_ID = waiver.Waiver_ID
  
```

Figure E.17 Validation for "View Waivers"

## 18. View Inventory



```

SELECT item.Item_Name, itemstatus.Item_Status, itemtype.Item_Type
FROM item LEFT JOIN itemstatus ON item.Item_Status_ID =
itemstatus.Item_Status_ID
LEFT JOIN itemtype ON item.Item_Type_ID = itemtype.Item_Type_ID
  
```

Figure E.18 Validation for "View Inventory"