**Hull University in association with**

Surname: Tindall

Forename(s): Kai

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date submitted: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**East Riding College**

**Foundation Degree in Computing**

**ASSIGNMENT COVER SHEET**

**2018/2019 ACADEMIC SESSION**

|  |  |
| --- | --- |
| Reception anonymous marking code |  |

|  |  |  |
| --- | --- | --- |
| **Registration University Number:**  **(Starts with year of registration eg. 200199999):**  **College Number: 20278932** | | |
| **Location:**  **St Marys Walk** | **Tutor: Ellen Smutts** | |
| **Module:**  **Databases** | | |
| **Assignment Title:** | | **Word Count:** |

|  |
| --- |
| **Under the University scheme for anonymous marking, your name will remain concealed until after your assignment has been marked. Please print your name clearly in the box at the top right hand corner, then sign, fold and seal. *Your student registration number from your student card must be entered clearly in the box provided and written on each sheet of your assignment.*** |

**For work submitted by hand**, read the following statement and sign and date in the box above. This signature is understood as compliance with the statement. **For work submitted online** … read the following statement, complete the details required and attach this document to your work. Submission of this completed form as an attachment to your work will be understood to be equivalent to signing the form in person.

|  |
| --- |
| I declare that the work I am submitting for assessment contains no section copied in whole or in part from any other source unless it is explicitly identified by means of quotation marks, or in the case of very long quotations, by means of wholly indented paragraphs. I declare that I have also acknowledged such quotations by providing detailed references in an approved format. I understand that unidentified and unreferenced copying both constitute plagiarism, which is one of a number of very serious offences under the University of Hull’s Code of Practice on the Use of Unfair Means - www.hull.ac.uk/handbook |

Students submitting work by hand are asked to submit two copies of assignments, one copy of which will be returned after assessment. Students submitting work within Merlin should use the Portfolio and send work to “Submitted Work”. It is advisable, however, to retain a copy of your work as insurance against loss or damage.

Databases

Assignment 2

Kai Tindall

2020

Table of Contents

[1 Entity Relationship Model 4](#_Toc35523133)

[2 Normalising Data 6](#_Toc35523134)

[2.1 Un-NOrmalised Data 6](#_Toc35523135)

[2.2 First Normal Form 6](#_Toc35523136)

[2.3 Second Normal Form 7](#_Toc35523137)

[2.4 Third NOrmal Form 7](#_Toc35523138)

[2.5 Higher Normal Form 7](#_Toc35523139)

[3 Interfaces 7](#_Toc35523140)

[3.1 Viewing multiple tables 8](#_Toc35523141)

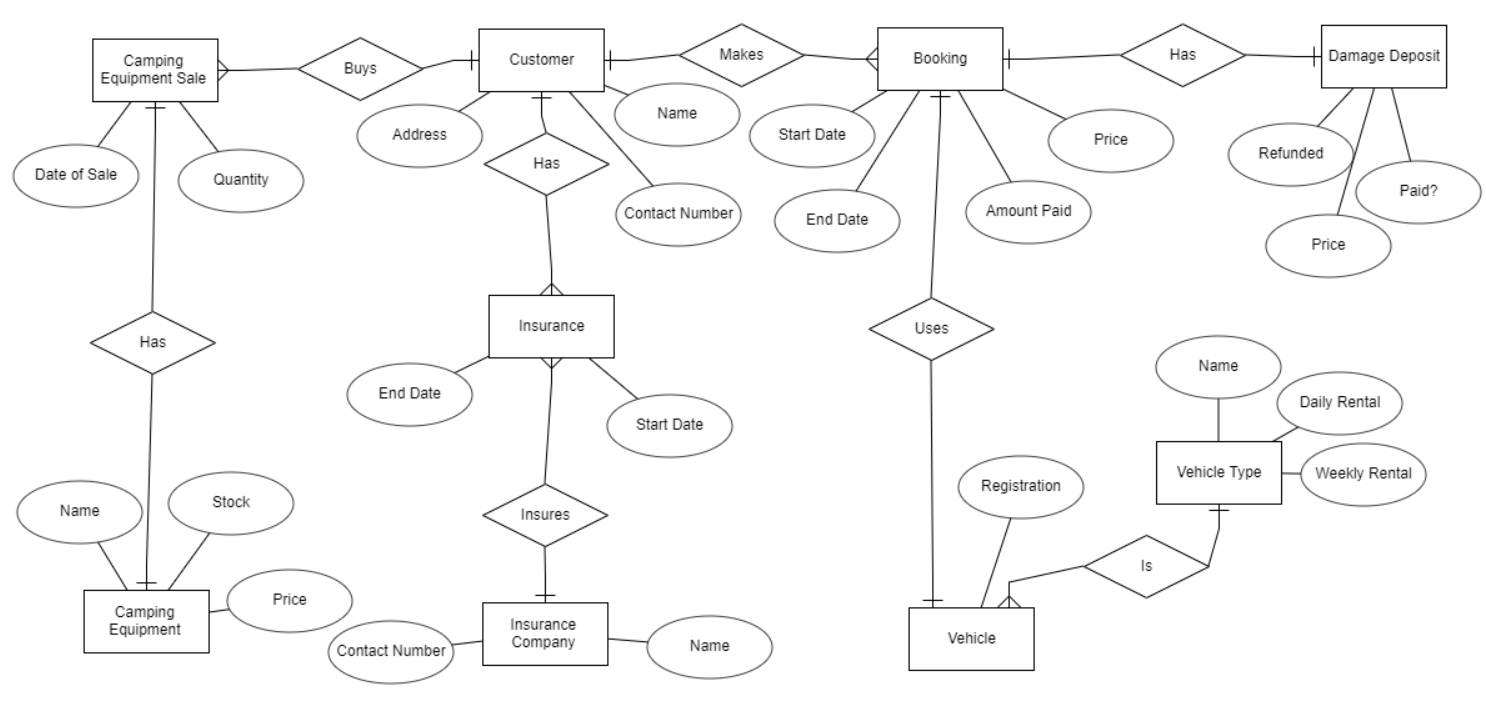
[3.2 Making Ad-HOc queries 8](#_Toc35523142)

[3.3 Create a new Booking 9](#_Toc35523143)

[3.4 Adding Stock to the Database 10](#_Toc35523144)

# Entity Relationship Model

To begin to plan this relational database, it is important to create an entity relationship model diagram, all this is, is a diagram that portrays all the entities you think you’ll need and how they relate to each other. It also shows all the attributes you think the entities will need (not including primary or foreign keys). I have created one for the given scenario below.



This also, to make a good amount of sense, needs to be accompanied by a data dictionary. A data dictionary will just go through the attributes and describe any details about them, such as data type or any notes about it. I have gone through each attribute and described the attribute. I also included primary and foreign keys in the data dictionary as even though they aren’t listed in the diagram, they’re still important pieces of information.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Attribute Name* | *Required?* | *Type* | *Field Length* | *Default Values* | *Notes* |
| **Customer** | | | | | |
| Customer\_Id | Yes | Integer | n/a | n/a | Auto incremented, Primary Key |
| First Name | Yes | Varchar | 20 | n/a |  |
| Surname | Yes | Varchar | 40 | n/a |  |
| Address Line 1 | Yes | Varchar | 50 | n/a |  |
| Address Line 2 | No | Varchar | 50 | Blank / empty |  |
| Post Code | Yes | Varchar | 10 | n/a |  |
| Contact Number | Yes | Varchar | 20 | n/a | Varchar instead of integer because they may include country extensions such as “+44”. |
| **Vehicles** | | | | | |
| Vehicle\_Id | Yes | Integer | n/a | n/a | Auto incremented, Primary key |
| VehicleType\_Id | Yes | Integer | n/a | n/a | Foreign key |
| Registration | Yes | Varchar | 15 | n/a | Not considered unique enough as if the business went international there could be two vehicles with the same registration. |
| **Vehicle Types** | | | | | |
| VehicleType\_Id | Yes | Integer | n/a | n/a | Auto incremented, Primary key |
| Name | Yes | Varchar | 30 | n/a |  |
| Rental charge per day | Yes | Decimal | n/a | n/a |  |
| Rental charge per week | Yes | Decimal | n/a | n/a |  |
| **Damage Deposits** | | | | | |
| DamageDeposit\_Id | Yes | Integer | n/a | n/a | Auto incremented, Primary key |
| Price | Yes | Decimal | n/a | n/a |  |
| Paid? | Yes | Bit | n/a | 0 |  |
| Refunded? | Yes | Bit | n/a | 0 |  |
| **Camping Equipment** | | | | | |
| CampEquipment\_Id | Yes | Integer | n/a | n/a | Auto incremented, Primary key |
| Name | Yes | Varchar | 30 | n/a |  |
| Price | Yes | Decimal | n/a | n/a |  |
| Stock | Yes | Integer | n/a | n/a |  |
| **Insurance Companies** | | | | | |
| InsuranceCompany\_Id | Yes | Integer | n/a | n/a | Auto incremented, Primary key |
| Name | Yes | Varchar | 50 | n/a |  |
| Contact Number | Yes | Varchar | 15 | n/a | Varchar so it can have extension number |
| **Insurance** | | | | | |
| Insurance\_Id | Yes | Integer | n/a | n/a | Auto incremented, Primary key |
| Customer\_Id | Yes | Integer | n/a | n/a | Foreign key |
| InsuranceCompany\_Id | Yes | Integer | n/a | n/a | Foreign key |
| StartDate | Yes | Date | n/a | n/a |  |
| EndDate | Yes | Date | n/a | n/a |  |
| **Equipment Sales** | | | | | |
| Sale\_Id | Yes | Integer | n/a | n/a | Auto incremented, Primary key |
| Customer\_Id | Yes | Integer | n/a | n/a | Foreign key |
| CampEquipment\_Id | Yes | Integer | n/a | n/a | Foreign key |
| DateOfSale | Yes | Date | n/a | n/a |  |
| Quantity | Yes | Integer | n/a | n/a |  |
| **Bookings** | | | | | |
| Booking\_Id | Yes | Integer | n/a | n/a | Auto incremented, Primary key |
| Customer\_Id | Yes | Integer | n/a | n/a | Foreign key |
| Vehicle\_id | Yes | Integer | n/a | n/a | Foreign key |
| DamageDeposit\_Id | Yes | Integer | n/a | n/a | Foreign key |
| StartDate | Yes | Date | n/a | n/a |  |
| EndDate | Yes | Date | n/a | n/a |  |
| Price | Yes | Decimal | n/a | n/a |  |
| Amount Paid | Yes | Decimal | n/a | 0 |  |

## Explanations

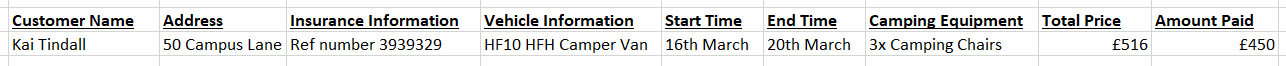
Here I will be explaining some of my design decisions.

I will not be using registration plates for the primary key for the vehicle table for the simple fact of registration plates can be altered. So, if Henry decides to change the registration plates of his vehicles to have them all line up with branding, then he’d have to go through the records and change that registration plate to the new one everywhere.

Contact numbers too will be stored as text so that you can have extension numbers present within the number. This wouldn’t be possible if I stored it as an integer as the “+” character would not be recognisable as a number.

# Normalising Data

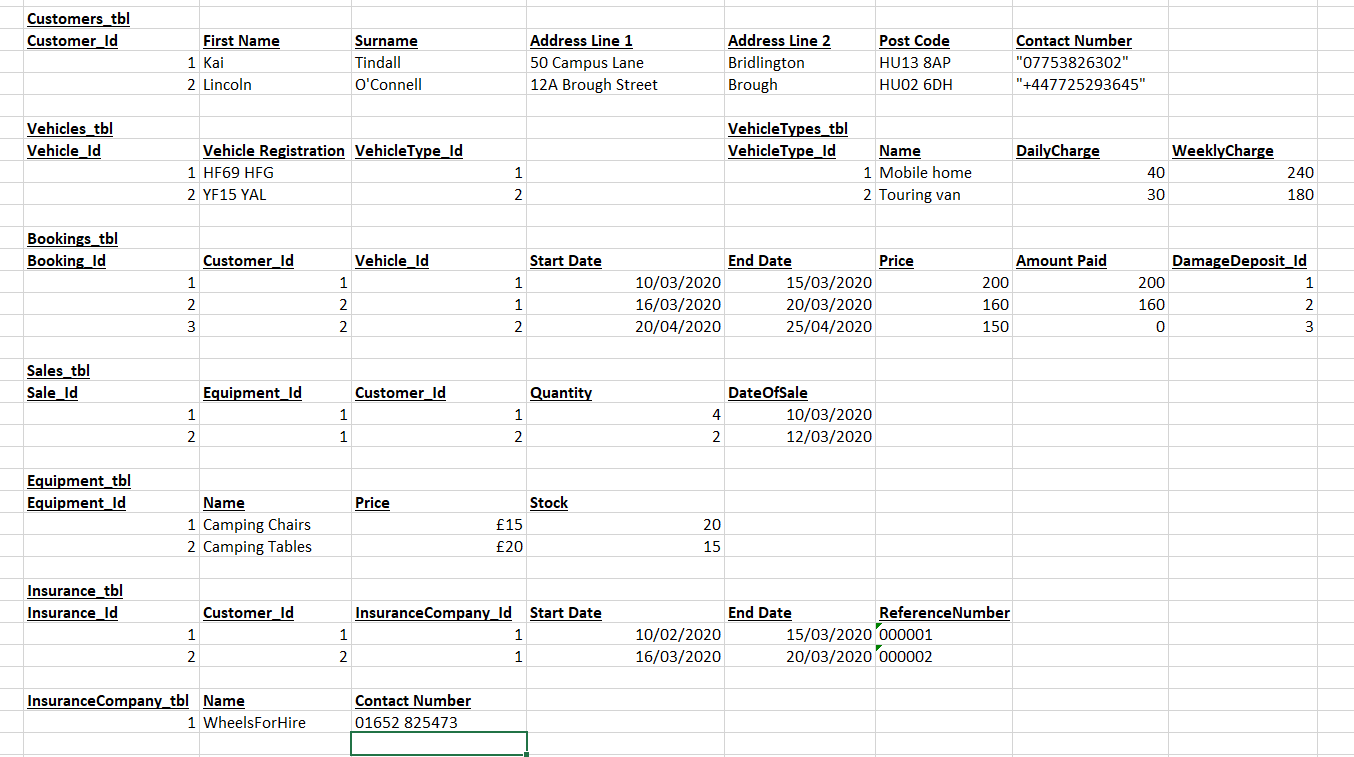
## Un-NOrmalised Data



This is a basic understanding of what does the customer want to know about each booking. Un-normalised data is just a list of data that the customer thinks they need to have. You can see things like names, addresses and such are all grouped together.

## First Normal Form

First normal form gets rid of repeating data, this means that if we still used the database from the un-normalised section then if Kai Tindall made multiple bookings then his name and address would be repeated for the next record. We don’t want this; we want to be able to put in Kai Tindall and his address and just be able to use it multiple times. That is First Normal form.



## Second Normal Form

Second normal form means that there are no dependencies on part of a key. This means that if you have a composite key of two fields, that there isn’t another field that depends on only one of the fields of the composite key.

From what I have already done to my database to make it comply with first normal form, it also now complies with second normal form. This means I don’t have to do anything else to my database.

## Third NOrmal Form

Third normal form means that there are no transitive dependencies. A transitive dependency is when “A determines B, B determines C”. Instead this should be broken down into so that “A determines B, and A determines C”. Fortunately, there are no transitive dependencies within my database, which means my database is in third normal form.

## Higher Normal Form

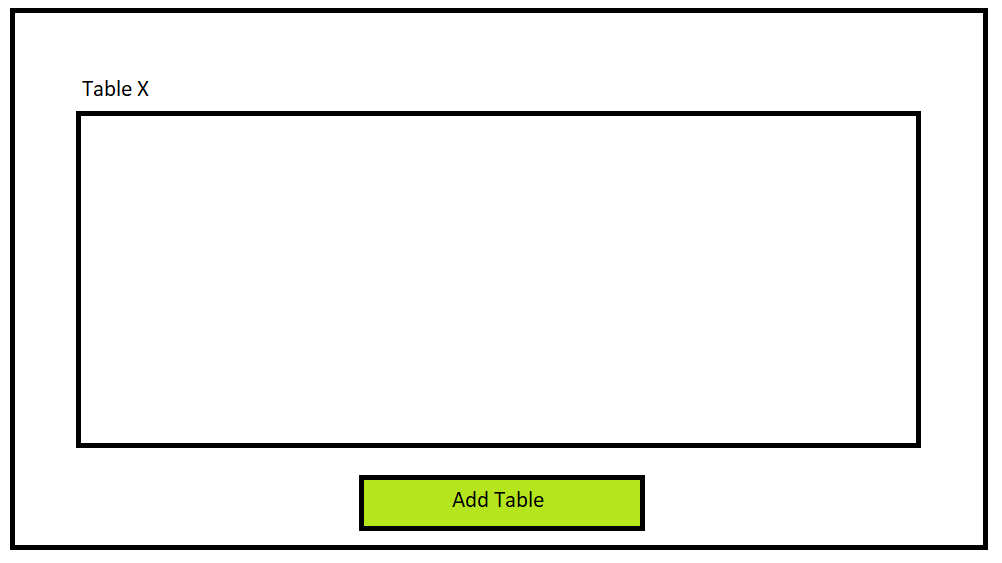
There are higher normal forms available, and while these do typically make the database more efficient there are diminishing returns for this. So, it is not worth putting in the effort to make the database more efficient if the database isn’t going to be big enough for it to matter.

# Interfaces

## Viewing multiple tables

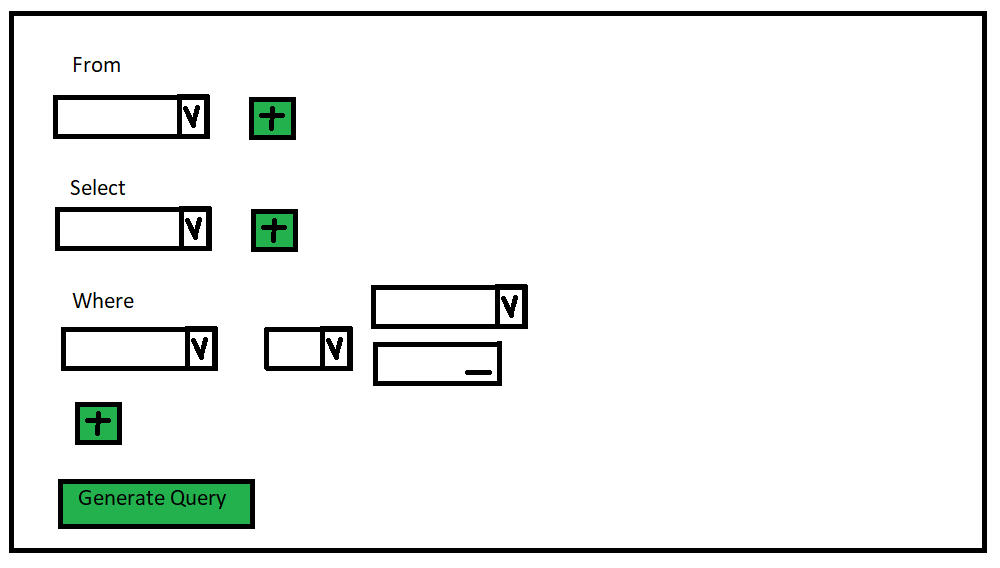
This will be a view that has one function, to display data for the user. I will have it with options to “expand data” This means it will expand foreign keys to include and data you’d like to see with it. This means that the user using the view can customise a table output to whatever they need to see.

I have made this screen very simple with just an “Add table” button at the bottom of all the current tables. This is good for HCI because it means that people won’t be getting confused with a lot of UI elements.



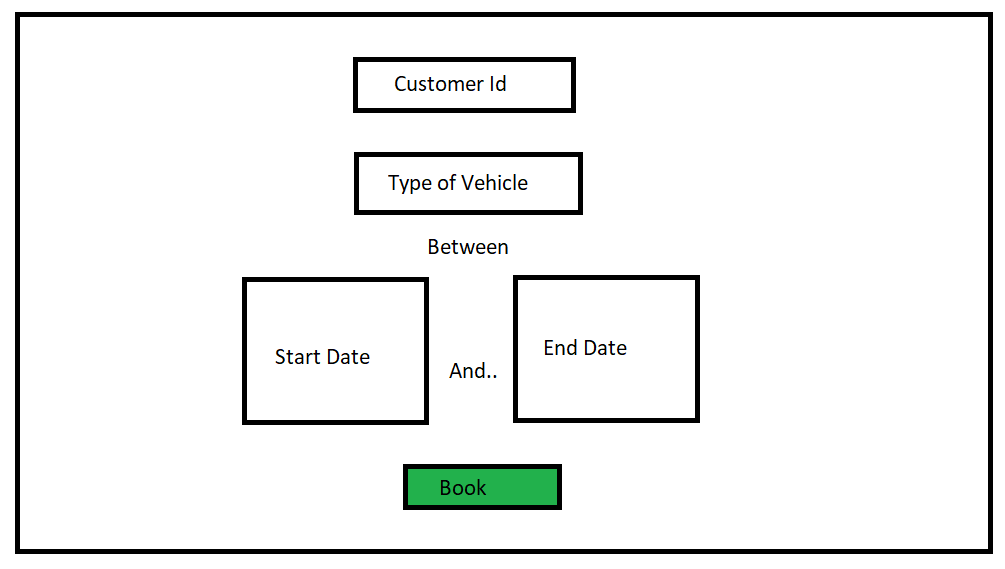
## Making Ad-HOc queries

The user would like to be able to make ad-hoc queries; I will implement this by using a “query builder” tool that will generate and run queries in the database. I have used colours (green) that’re associates with doing the right thing, so the user knows you should be pressing the buttons to build and then generate the query.



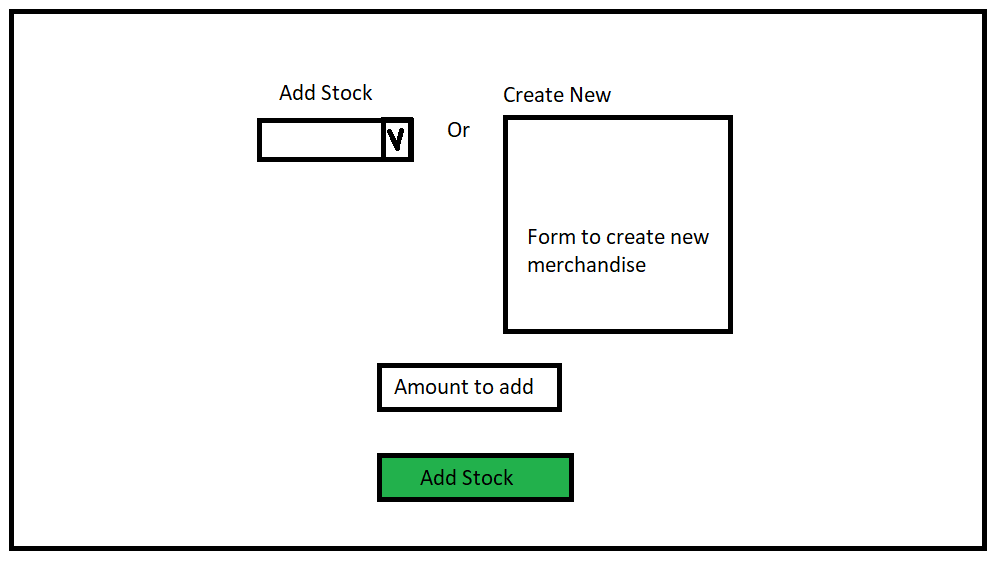
The from line will allow the user to select the tables they’re wanting to look into, the select line will be able to select attributes of the table you can choose to select. And the where line will allow users to filter results based on certain conditions. There are four boxes per “where” as you need to select the attribute, you’re filtering first, then the operation, then decide if you’re comparing it to another attribute or a hard value.

## Create a new Booking



In this view you can select a customer Id, I will have it where you can select customer Id from another view where you can search for names and address or create a new customer

## Adding Stock to the Database



In this view you can add stock to existing items ready for purchase or you can create entirely new merchandise options and add initial stock to that.