



Jonathan Corner
Information Engineering

Roll the Bones Database – Assignment 2

Contents

Introduction	3
Entity Relationship Diagram (ERD)	3
Relationship Annotations	4
Screenshots of Database build	5
Creating Command Buttons	6
Creating Tables	8
Creating Relationships	11
Creating Queries	15
Conclusion	16
User Guide	17

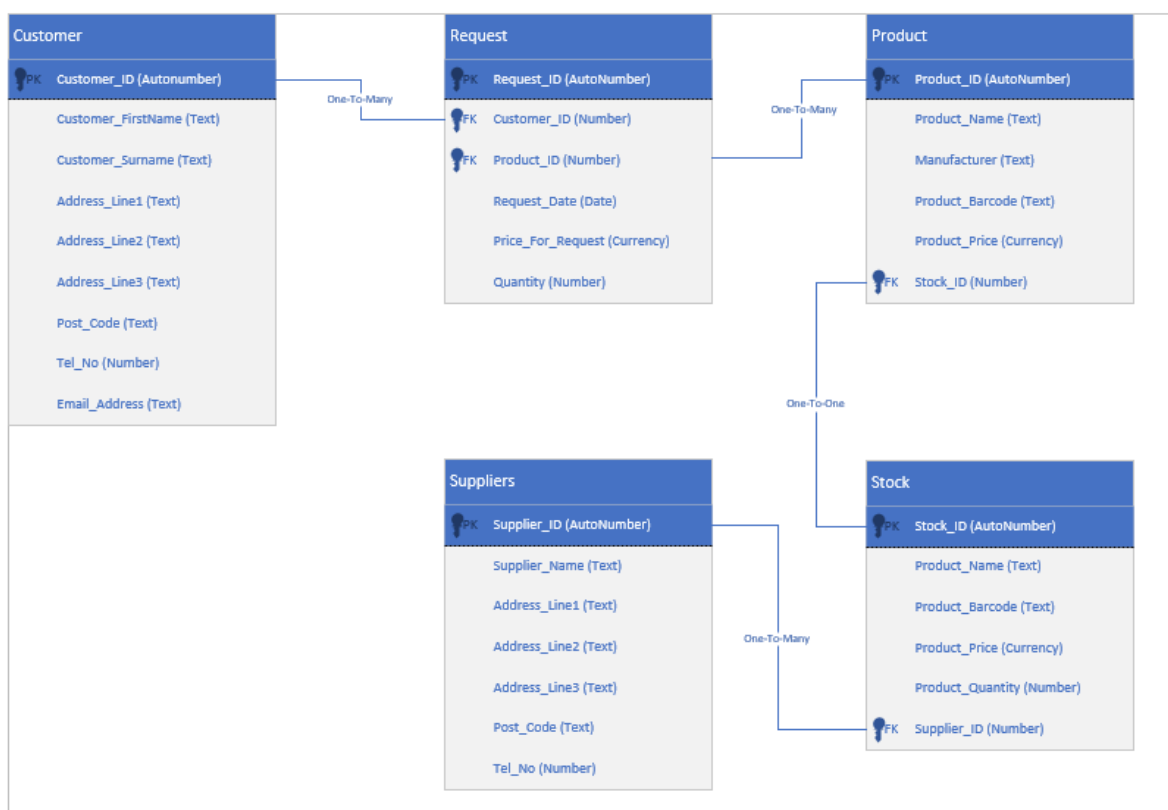
Introduction

In this report I have been tasked to create a database for Roll the Bones who are a top quality Board games retailer in the South Wales Area. Before I start the initial setup of the database I will first create the appropriate diagrams supporting my design, this includes the Entity Relationship Diagram (ERD)

To create my database I will be using Microsoft Access 2019, within this software I will create the required tables and relationships then run queries to test the outcome of the system I have created. Finally I will then go on to talk about the database as a whole and how it will be used by the client, with all supporting screenshots and documentation for the staff who will be using the system on a daily basis.

Entity Relationship Diagram (ERD)

For the first task I will be focusing on the Entity Relationship diagram where I will find all the entities and show the relationships between them from the database design perspective. As you can see from the design all the tables, fields, relationships and data types are shown, this design is what I'm looking to implement when creating the system for the client.



Relationship Annotations

Relationship 1

Key Type	Table	Field
Primary	Customer	Customer_ID
Foreign	Request	Customer_ID

When building my database, I decided to put these two fields together. Reason being is that when a customer requests a particular game which may not be in stock, the request table will populate the customer's information such as first name, surname, address etc. by the use of their membership ID (Customer_ID) into the form. I feel this is a One-To-Many Relationship as a customer can make many requests.

Relationship 2

Key Type	Table	Field
Primary	Product	Product_ID
Foreign	Request	Product_ID

The Request table has the Product_ID Field as a foreign key because when a customer makes a request, a member of staff will put the Product_ID in and it will populate all the fields such as Product name, Manufacturer, Price etc. which it gets from the Product table. This will be a One-To-Many Relationship as there is only one Product_ID per item but many products which can be requested from the Product table.

Relationship 3

Key Type	Table	Field
Primary	Stock	Stock_ID
Foreign	Product	Stock_ID

These two tables have the same relationship because the Product form requires the Stock_ID so it shows the Unique code on the Product form to refer to the Stock table if required, so this is why it's a Primary_Key within the Product Table. This is a One-To-One relationship as there is only one Stock_ID per item where each table will be pulling information in for.

Relationship 4

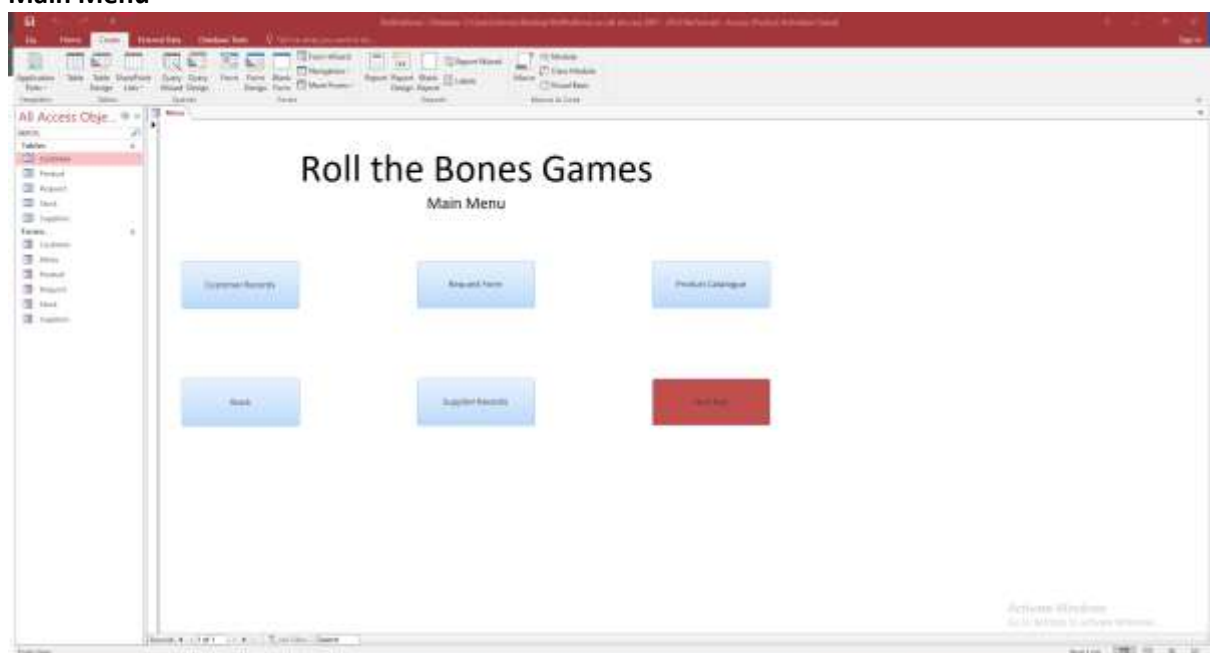
Key Type	Table	Field
Primary	Suppliers	Supplier_ID
Foreign	Stock	Supplier_ID

The Supplier_ID in the supplier's table is used to populate the supplier information into the form, this field is a foreign key in the Stock table, reason being is that it shows to the database user what supplier is going to provide that particular game. This is a One-To-Many relationship because when requesting from the stock form, a particular supplier may provide multiple games.

Screenshots of Database build

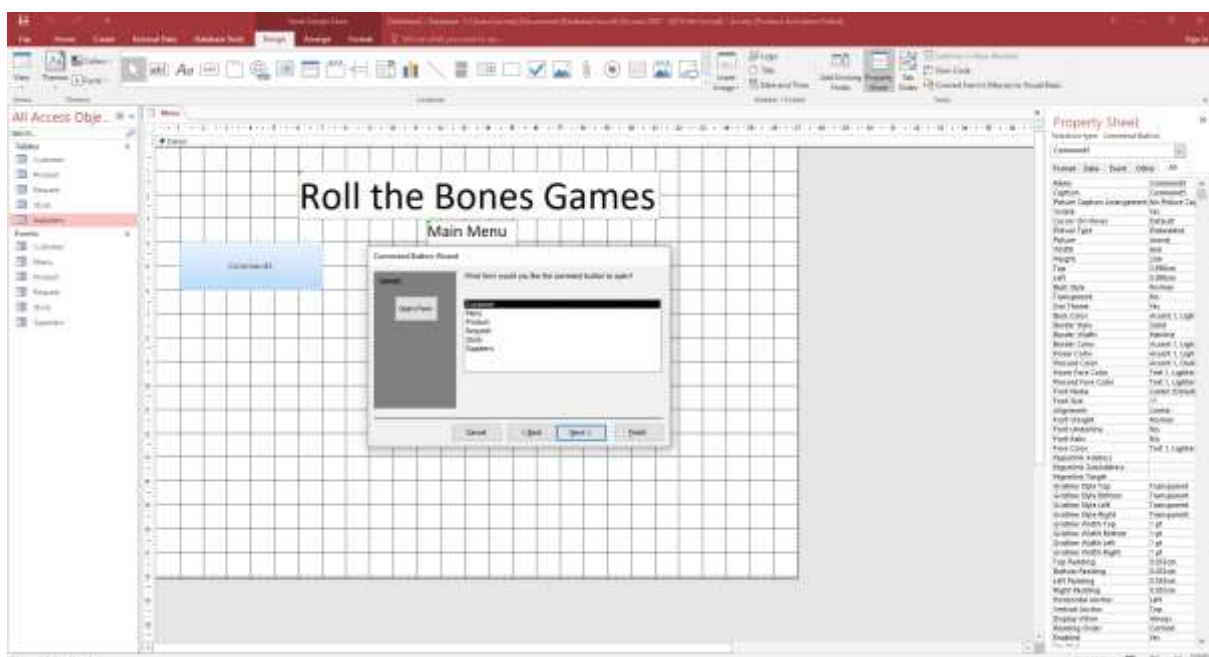
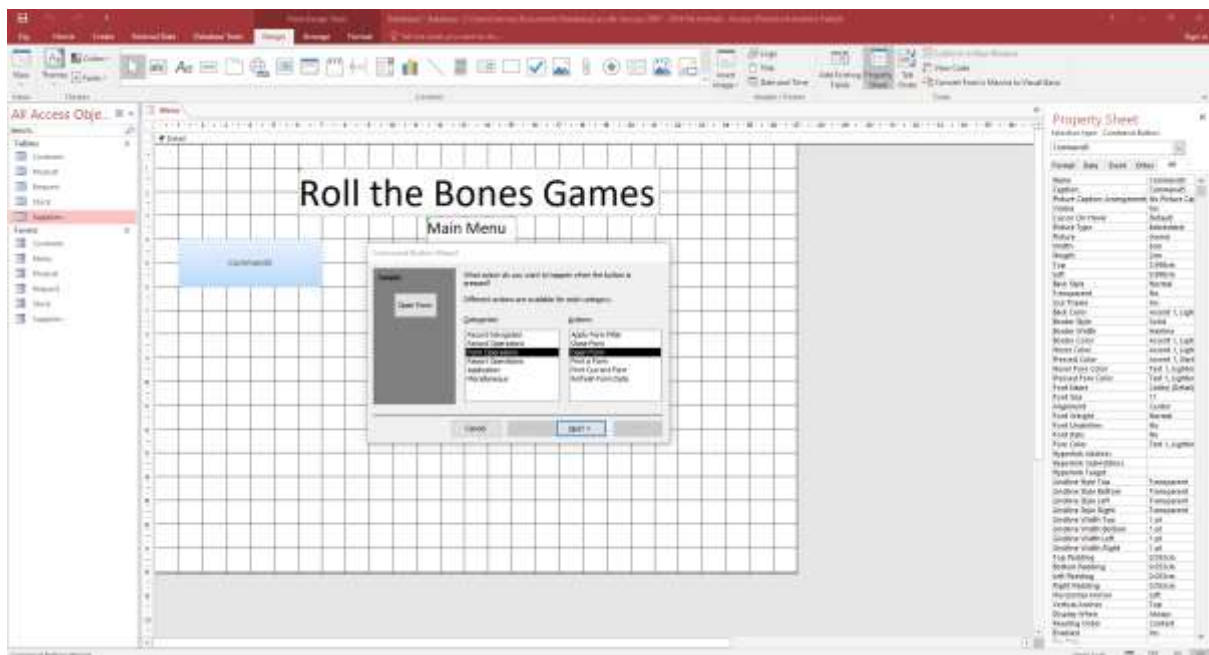
As part of my assignment I am tasked into creating my database for the customer using Microsoft Access and take relevant screenshots of the stages taken into building the system, my objective is to build the system based on my Entity Relationship Diagram above which I feel is to the specification. When the database is built I will make another check on the relationships to see if the system meets the client's needs.

Main Menu

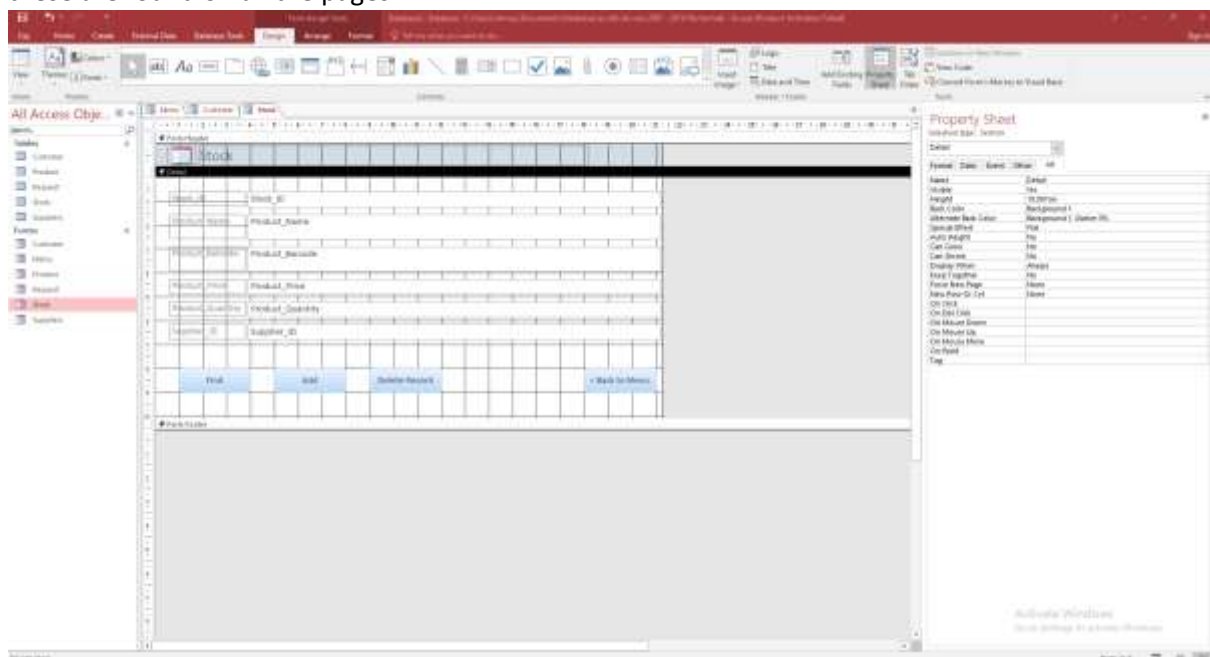


Above is my Main Menu for the database, here the staff member who will use the software will be able to access areas such as - Customer Records, Request Forms, Product Catalogue, Stock and Supplier Records. When you click on the command buttons it will open a form for that particular table where the staff member can add, delete or amend records, there is also a button on each form to allow the user to go back to the main menu. To create these command buttons I used the Command button wizard which is built into the Form Design toolkit on Microsoft Access, the screenshots on how to do this are found below.

Creating Command Buttons



Below are the buttons which have been created within each form to navigate around the system, these are found on all the pages.



Creating Tables

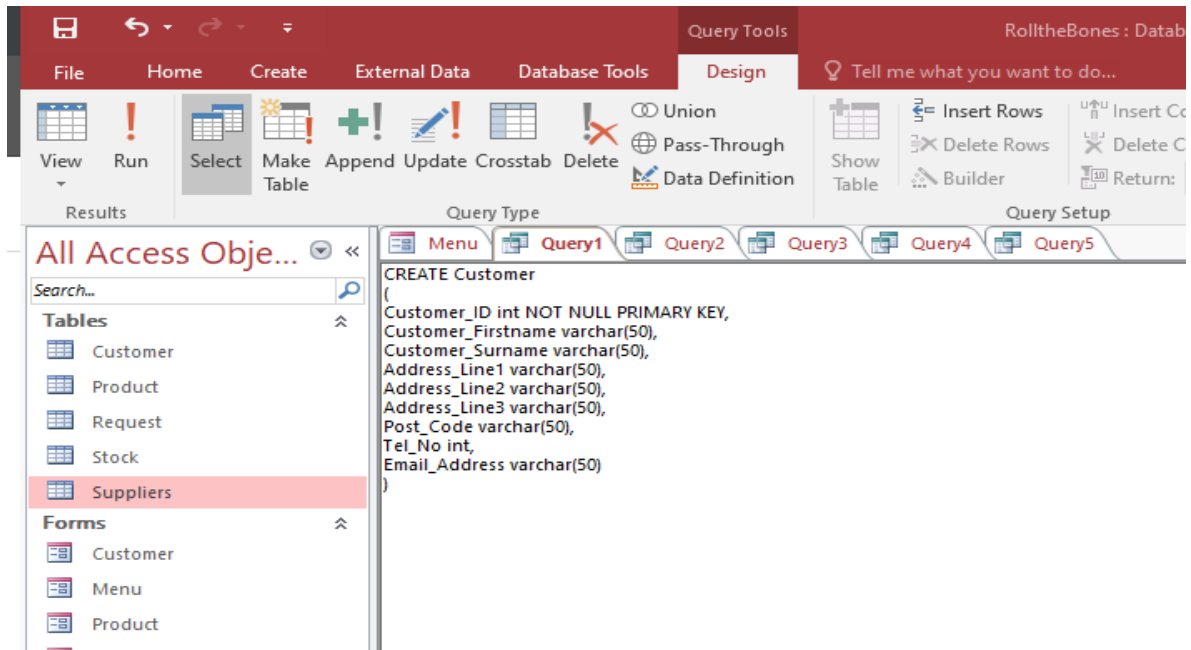
As part of my database I have created the following tables to store specific information. These include:

- Customer** - Holds customer/member records.
- Product** - Product information store, this is known as the product catalogue.
- Request** - Request form for a particular game required based on customer needs.
- Stock** - The stock database, this holds the numbers of all the products in storage.
- Suppliers** - Information of all the game suppliers.

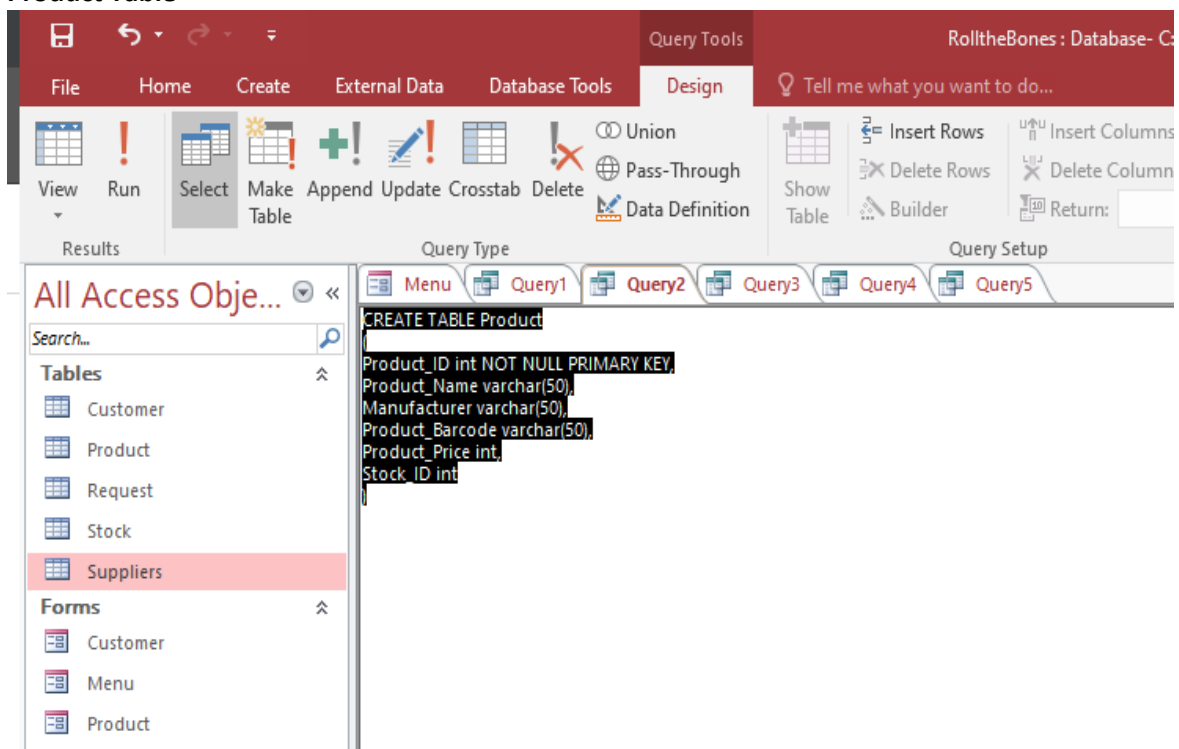
These tables could be created by using the table creation wizard within Microsoft Access but I felt that I would go into depth instead by creating these manually using SQL Code via the Query facility as I will be able to advance my knowledge on the data types and various Syntaxes.

The SQL code I put together to create my tables are screenshotted below:

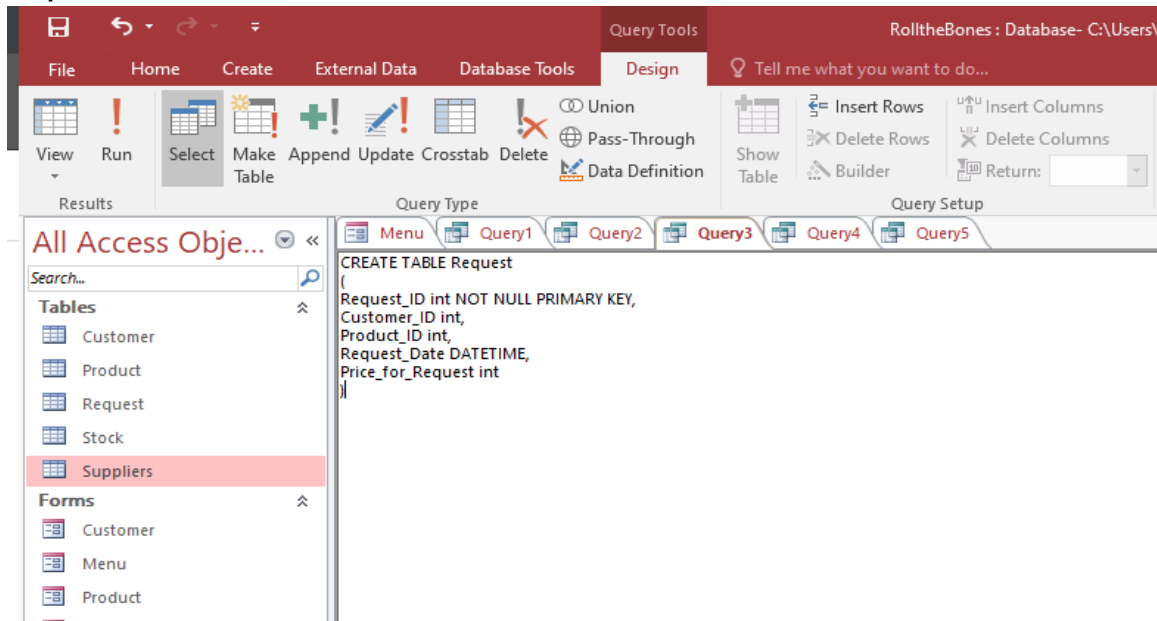
Customer Table



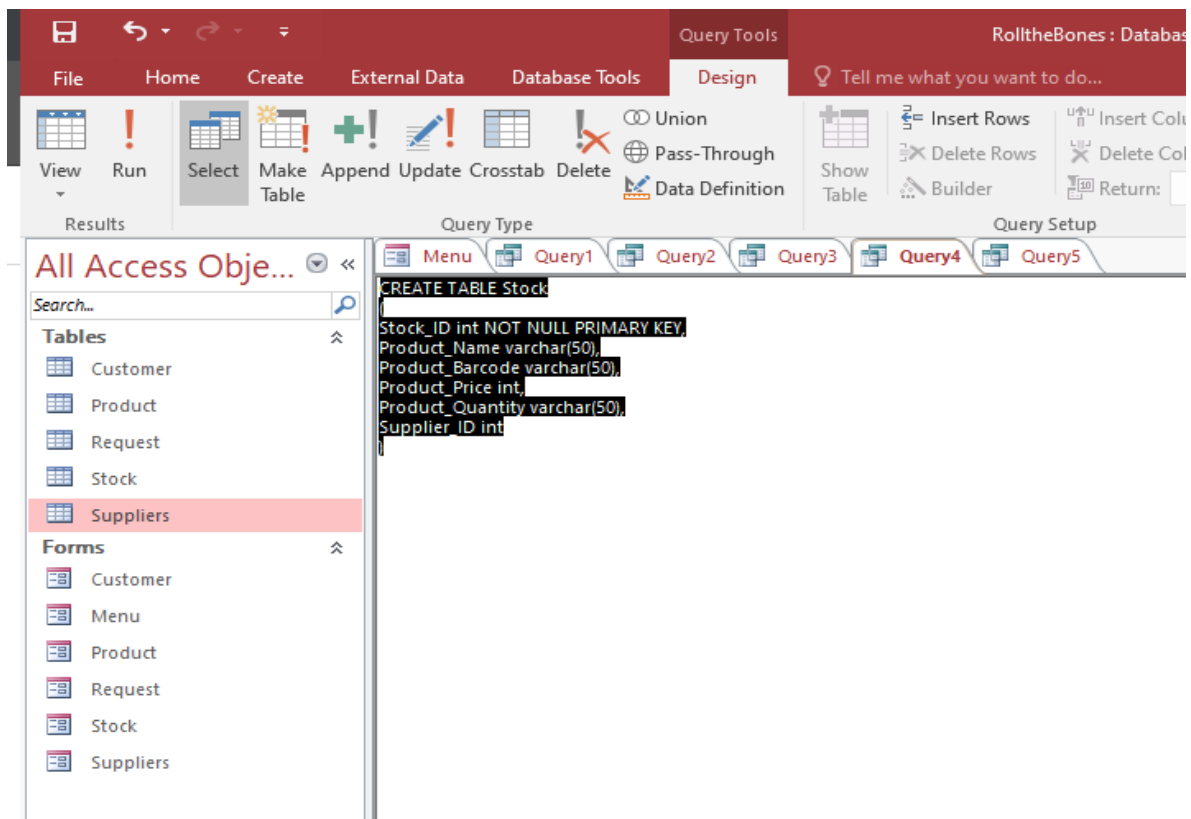
Product Table



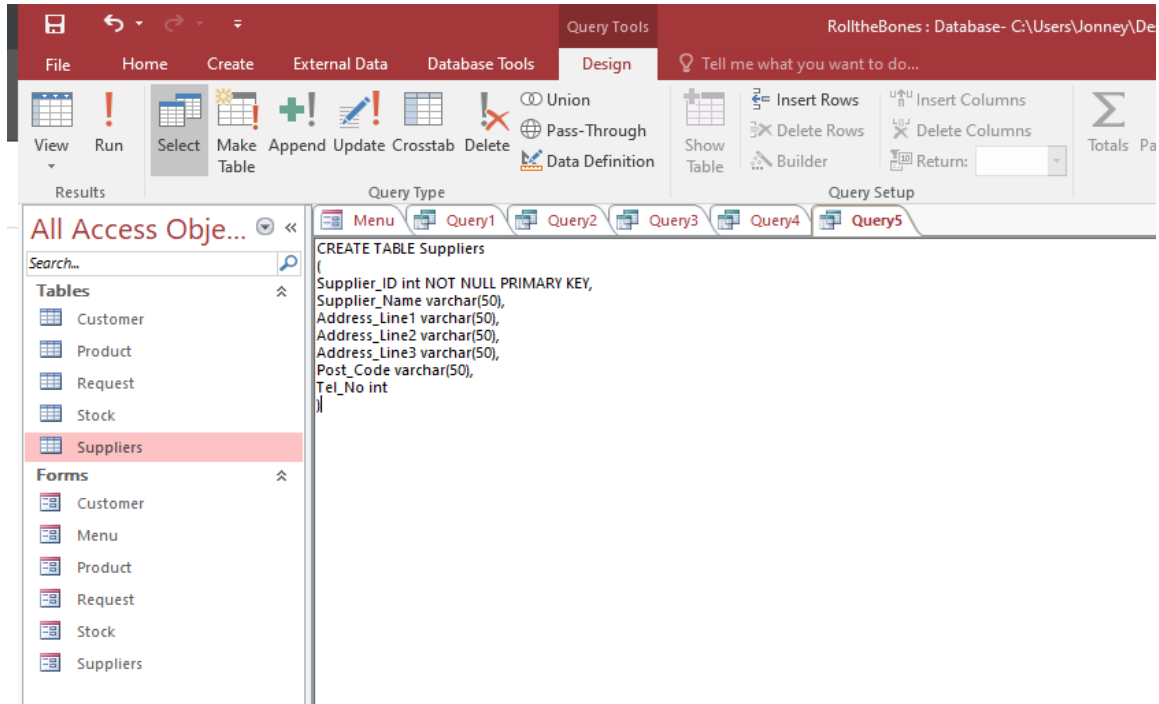
Request Table



Stock Table

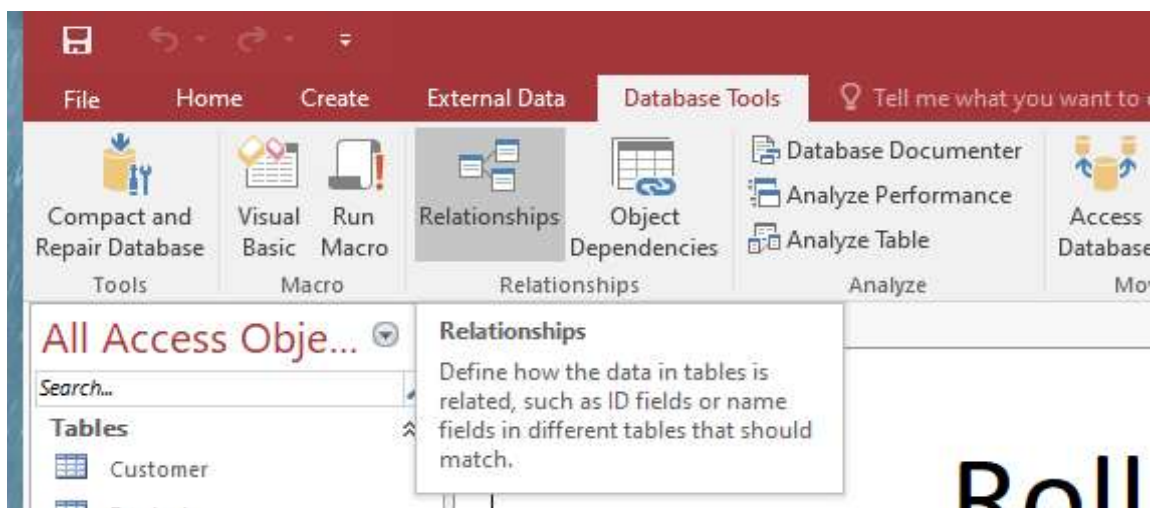


Suppliers Table



Creating Relationships

In order for the forms in my database to pull information from different tables I need to create relationships between them, this is crucial because if I relate the wrong field into the wrong table, incorrect information will be provided to the front end user. To create this, I did the following.



Relationship 1

Key Type	Table	Field
Primary	Customer	Customer_ID
Foreign	Request	Customer_ID

Edit Relationships

Table/Query: Customer Related Table/Query: Request

Customer_ID Customer_ID

☒ Enforce Referential Integrity

☒ Cascade Update Related Fields

☒ Cascade Delete Related Records

Relationship Type: One-To-Many

OK Cancel Join Type.. Create New..

Relationship 2

Key Type	Table	Field
Primary	Product	Product_ID
Foreign	Request	Product_ID

Edit Relationships ? X

Table/Query: Product Related Table/Query: Request

Product_ID Product_ID

☒ Enforce Referential Integrity

☒ Cascade Update Related Fields

☒ Cascade Delete Related Records

Relationship Type: One-To-Many

Create Cancel Join Type.. Create New..

Relationship 3

Key Type	Table	Field
Primary	Stock	Stock_ID
Foreign	Product	Stock_ID

Edit Relationships ? X

Table/Query: Stock Related Table/Query: Product

Stock_ID Stock_ID

☒ Enforce Referential Integrity

☒ Cascade Update Related Fields

☒ Cascade Delete Related Records

Relationship Type: One-To-Many

Create Cancel Join Type.. Create New..

Relationship 4

Key Type	Table	Field
Primary	Suppliers	Supplier_ID
Foreign	Stock	Supplier_ID

Edit Relationships

Table/Query: Suppliers Related Table/Query: Stock

Supplier_ID Supplier_ID

☒ Enforce Referential Integrity

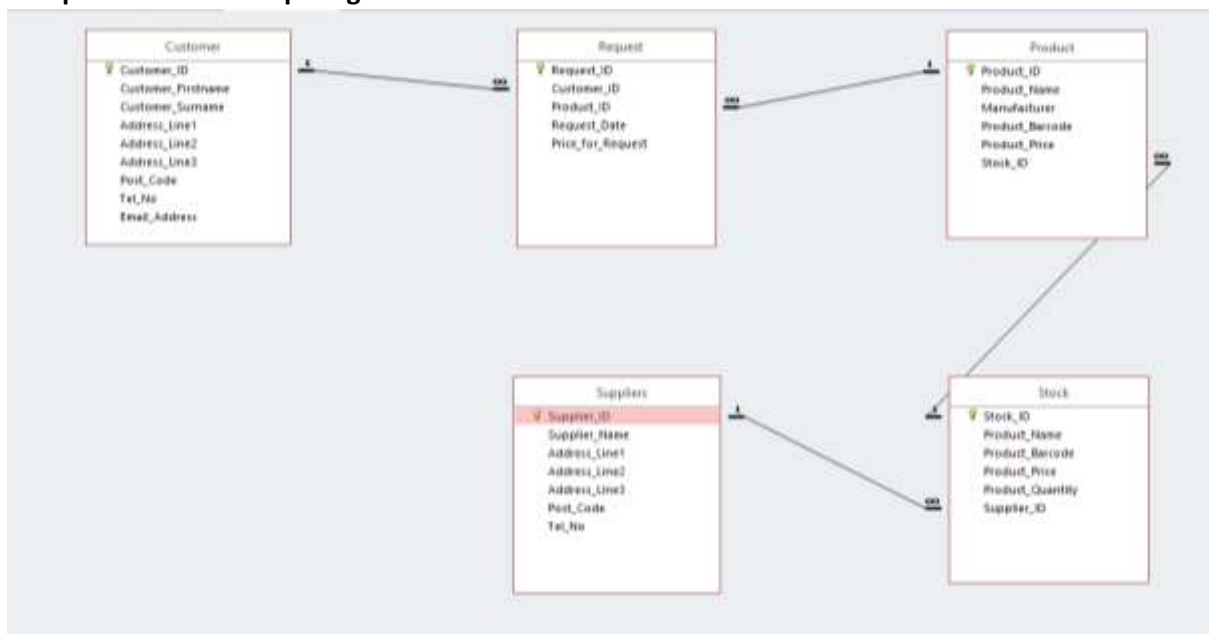
☒ Cascade Update Related Fields

☒ Cascade Delete Related Records

Relationship Type: One-To-Many

Buttons: Create, Cancel, Join Type.., Create New..

Completed Relationship Diagram

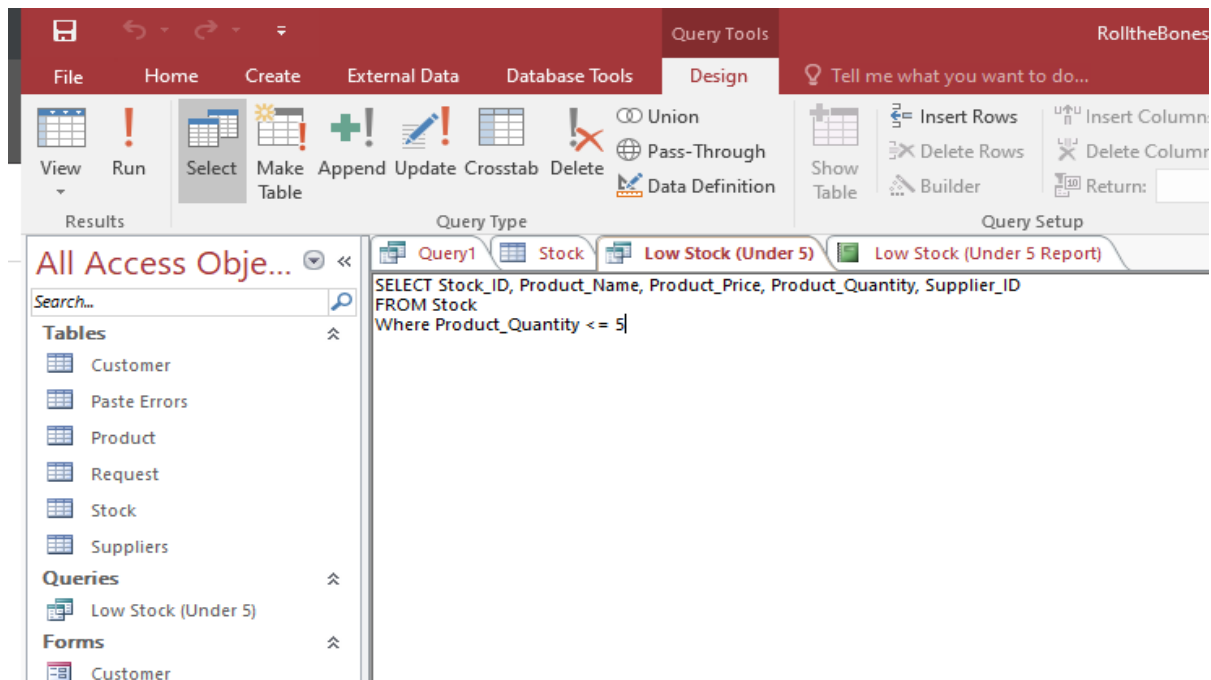


Above is the final relationship diagram of the database after creation. I feel that this is identical to my initial design using Microsoft Visio at the start of this document. To me I feel that my selection of Fields and tables has come together really well and the data displayed in all tables and forms is completely relevant to what the front end database user requires as part of the specification from Roll the Bones Games.

Creating Queries

Low Stock Query

As part of the new system for Roll the Bones, one of their requirements is to have a low stock system which has the ability to generate a report when they have the quantity of 5 games or less, using this they will be able to replenish their stock from their trusted suppliers to avoid going under the stock threshold.



I created this query by using the SQL view in Microsoft Access by using the SELECT Query and chose the field names wisely as these are what will be displayed when the query is run. For this example, I used - Stock_ID, Product_Name, Product_Price, Product_Quantity, Supplier_ID.



After I created the query and it generated the results of the games which fall under the quantity threshold, I then generated a report so the data can be more presentable.

Request Expiry (2 Weeks)

Another part of the new system requirement from Roll the Bones was to have the ability to see requests which expired 2 weeks from the creation date, in order to achieve this I created a Macro so the database users can see how many Customer's Game requests have expired. To do this I used the following command.

```
SELECT Request_ID, Customer_ID, Product_ID, Request_Date, Price_for_request  
FROM Request  
WHERE Request_Date <=(Date()-14)
```

When the staff member gets notified that the request is expiring, they then have the ability to action this by calling up the customer and asking if they want the store to source it any longer, if not the request will be closed.



The screenshot shows a Microsoft Access window titled "Requests Expired 2 Weeks ago". The window has a header bar with the title and a date/time stamp "07 January 2020 21:27:02". Below the header is a table with the following columns: Request_ID, Customer_ID, Product_ID, Request_Date, and Price_for_request. The table contains one row of data: Request_ID 2, Customer_ID 4, Product_ID 2, Request_Date 21/11/2019, and Price_for_request £20.00. At the bottom of the window, it says "Page 1 of 1".

Request_ID	Customer_ID	Product_ID	Request_Date	Price_for_request
2	4	2	21/11/2019	£20.00

Conclusion

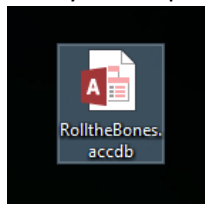
Overall I feel that this system would make daily tasks much easier for the staff at Roll the Bones as it's easy for the staff member to navigate between the forms and modify any records if necessary. With regards to the documentation, I have included full examples on how I created the relationships between the tables, showing all the connections between the fields as well as the membership classes and how everything will work prior to the system build. When working on Microsoft Access there are many wizards which can be used to create queries, macros, relationship links etc. I have done my utmost to use the SQL coding section instead so I can gain further knowledge into the Syntaxes for future use.

I feel if there was anything else I could improve on next time would be to merge my Product and Stock Tables together when creating Queries, as I would be able to pull in the additional data such as Product description etc. into the Stock Report, so the staff member would be able to identify the products at ease.

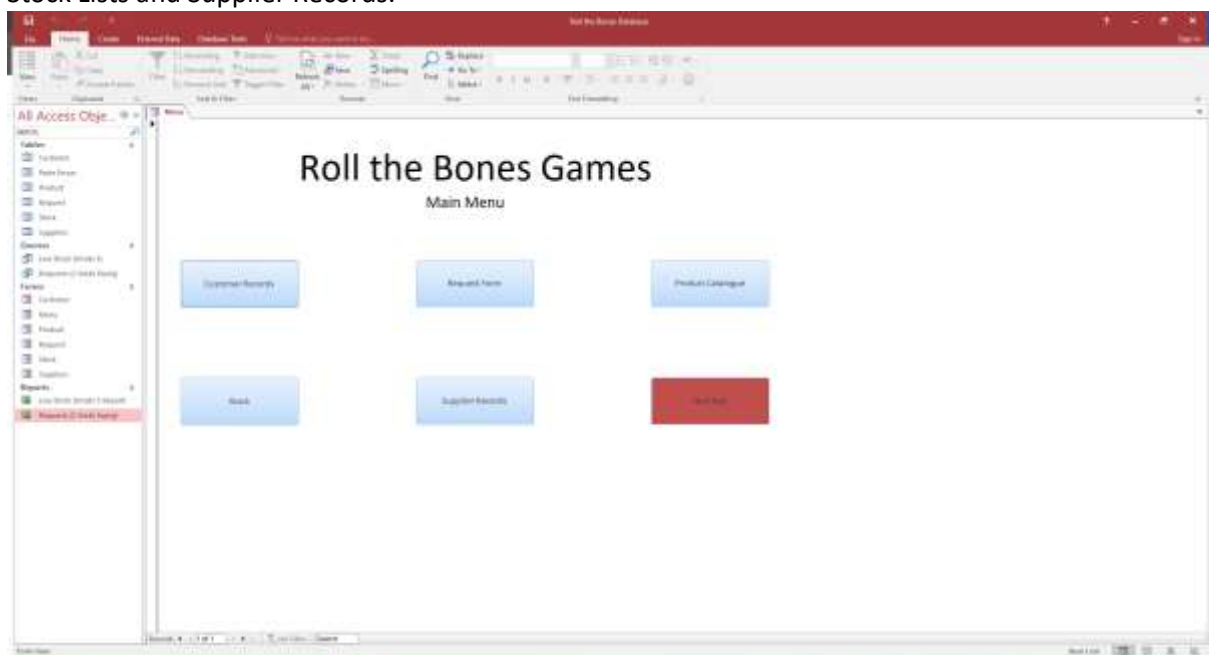
User Guide

As part of the system build I have created the guide below for the staff members of the game store, so they become familiar with the functionalities and basic use.

First off to access the database you will need to select the 'RolltheBones.accdb' to open the database within Microsoft Access from the location it's saved in, for this example the file is located on my desktop.



Once the database has been opened you will be presented by the Main Menu, here you will have access to various forms of the system such as Customer Records, Request Form, Product Catalogue, Stock Lists and Supplier Records.



To access a specific form, simply left click on the button and it will take you directly to the page desired. In this instance we will be going into the Customer Records form.

The screenshot shows a software interface with a red header bar. On the left is a sidebar with 'All Access Objects' and a tree view. The main area displays a 'Customer' form with the following fields: Customer ID (text), First Name (text), Surname (text), Mobile Number (text), Email (text), Country (text), Post Code (text), Telephone Number (text), and Street Address (text). At the bottom of the form are four buttons: 'Find', 'Add', 'Delete Record', and '< Back to Menu'.

Here you can see the fields that are available when accessing this form. Here you will have the ability to search for a customer via their ID number or name by using the find function, Add a customer by populating their details into the relevant fields or even delete a customer record when found.

We have added in the descriptions for each field in the form above so you are able to get a better understanding of what data should be inputted.

If you would like to go back, simply click on the '< Back to Menu' button where you will be returned to the Main Menu to make another selection.

The screenshots below have been taken from the other forms within the system, within these you will have the same functionality as the Customer Record section where you can find and amend any of the data.

Request Form

The screenshot shows a software interface similar to the first one. The main area displays a 'Request' form with the following fields: Request ID (text), Customer ID (text), Product ID (text), Request Date (text), and Request For (text). At the bottom of the form are four buttons: 'Find', 'Add', 'Delete Record', and '< Back to Menu'.

Product Catalogue

The screenshot shows the 'Product' form in the SAP S/4HANA 'All Access Objects' view. The form is titled 'Product' and contains the following fields:

- Product_ID: [Empty]
- Product_Name: [Empty]
- Product_Description: [Empty]
- Product_Supplier: [Empty]
- Product_Price: [Empty]
- Product_Stock: [Empty]

At the bottom of the form, there are four buttons: 'Find', 'Add', 'Update/Refresh', and 'Back to Menu'.

Stock List

The screenshot shows the 'Stock' form in the SAP S/4HANA 'All Access Objects' view. The form is titled 'Stock' and contains the following fields:

- Stock_ID: [Empty]
- Stock_Name: [Empty]
- Stock_Description: [Empty]
- Stock_Price: [Empty]
- Stock_Stock: [Empty]
- Stock_Stocking: [Empty]

At the bottom of the form, there are four buttons: 'Find', 'Add', 'Update/Refresh', and 'Back to Menu'.

Supplier Records

Suppliers

Supplier_ID: 1

Supplier_Name: Lithings and Co

Address_1: 1 Peter England Road

Address_2: Bingley

Address_3: West Yorkshire

Address_4: LS23 7BA

Email_Address: LITH@LITH.CO.UK

Phone_Number: 01937 541111

Web_Site: WWW.LITH.CO.UK

Buttons: Find, Add, Update/Refresh, Back to Map

Report Running

There are two types of reports the system is capable of creating, these include:

Low Stock - This report shows any Products which are currently in stock where the quantity is 5 or under, This will give you the supplier ID where you can get in touch with them to replenish the stock.

Stock ID	Product Name	Product Price	Product Quantity	Supplier ID
1	game the throne	42.40	4	1
6	The Grand Landing game	46.20	2	1

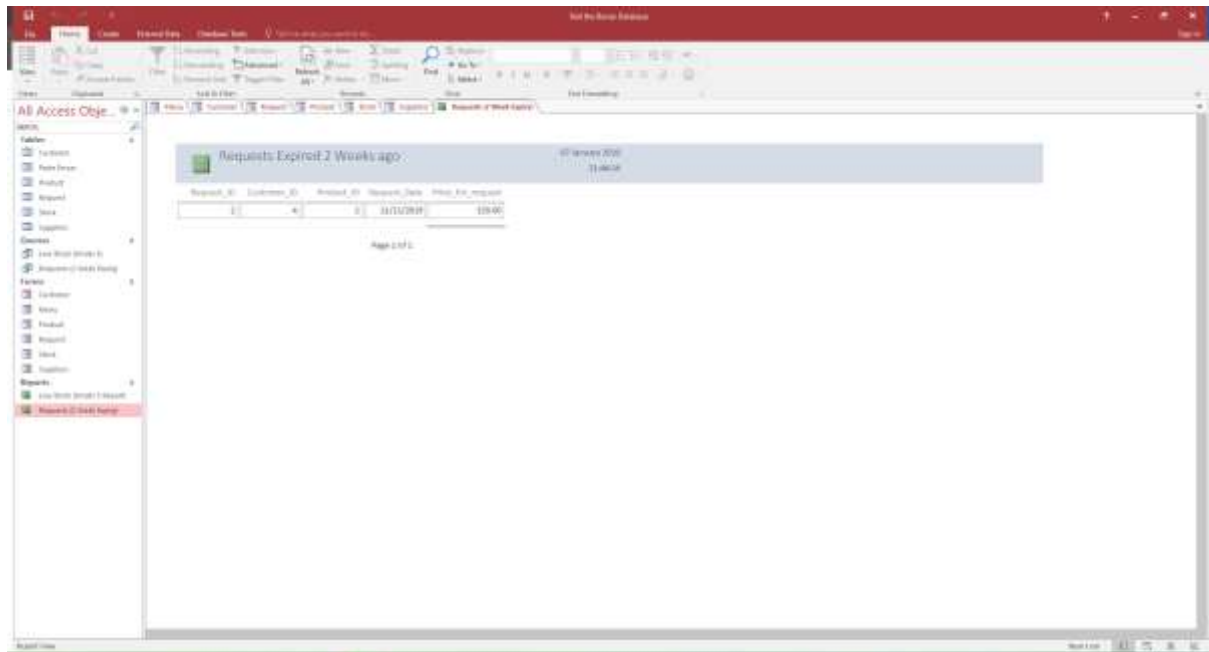
Page 2 of 2

If you would like to print out the report right click on the 'Low Stock (Under 5 Report)' tab and select 'Print Preview' where you will be presented with the following:

Stock ID	Product Name	Product Price	Product Quantity	Supplier ID
1	game the throne	42.40	4	1
6	The Grand Landing game	46.20	2	1

Page 2 of 2

Requests - This report will show any requests made by a customer which are over 2 weeks' old that need to be actioned as they are expiring on the system. Here you can get information such as the date and the customer ID where you can contact them to see if they would still like you to source a particular item for them.



Requests Expired 2 Weeks ago

Request_ID	Customer_ID	Product_ID	Request_Date	Price	Qty
1	4	2	11/11/2019	129.00	

Page 2 of 2