

Criterion E: Product development

Complex techniques used:

- Three related tables
- Calculated fields/ use of queries
- Use of graphic fields
- Macros
- Proficient use of techniques to enable easy navigation such as menus or buttons

Structure Organization & Structure

The screenshot (Fig.1) below shows the organization of the database in terms of the tables and the relationships, which exist, between the fields of the Tables

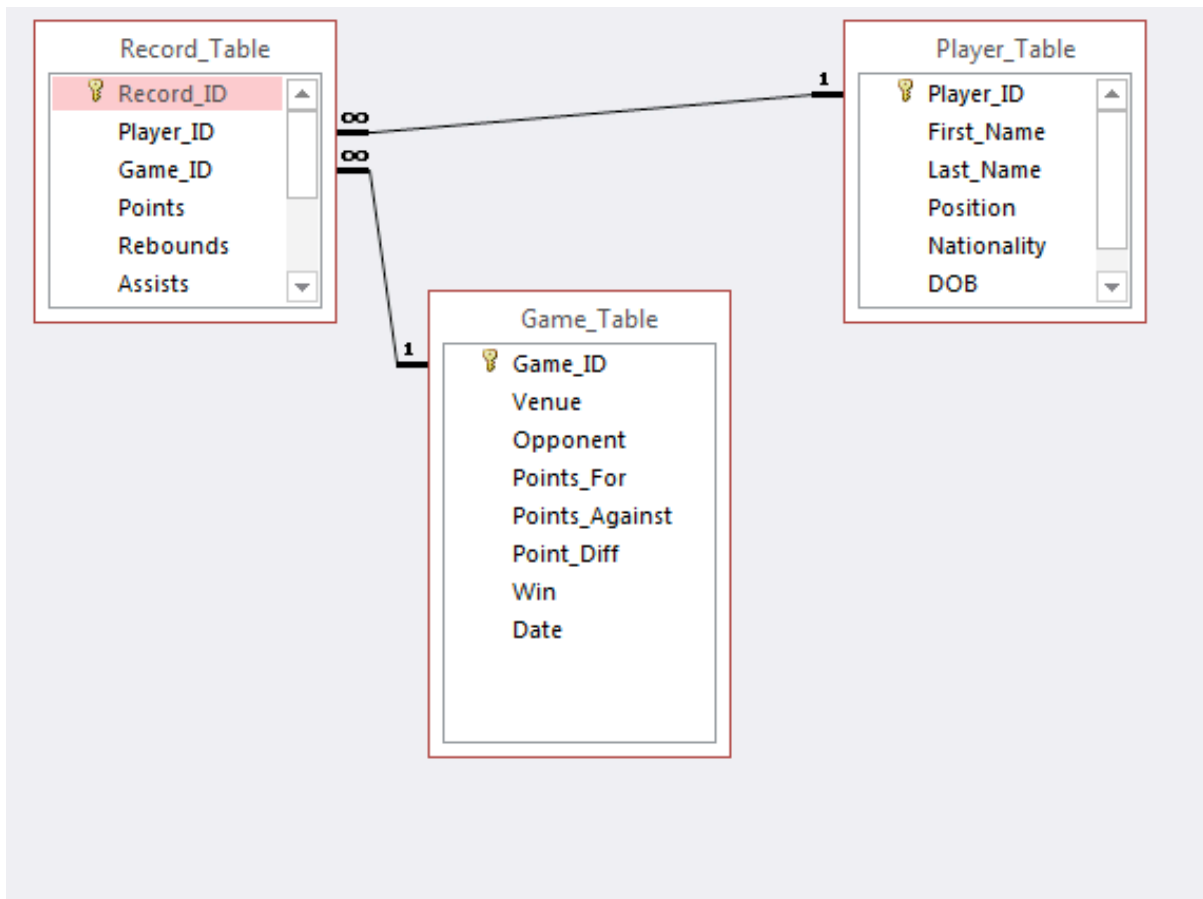


Fig.1 Relationship of Tables

Data Types and Validation Checks

The structure of the tables ensure that if data invalid data is entered then it is rejected, this is to prevent accidental miss-enter of data which would then result invalid reports and decrease the usefulness of the database, therefore to ensure that this does not happen it is important to only have a specific data type and validation rule/s. This is also important as invalid inputted data can also affect other data in it is used in an expression. For example for the field Points, the validation rule is Null between 0 and 99, this exist to ensure that the value is a non-negative, if the data is negative the message shown below in fig. 2 is shown and so that the value does not exceed an unreasonable amount of digits. The data type is chosen as a being numeric, therefore anything none numeric as shown below in fig. 3 is rejected.

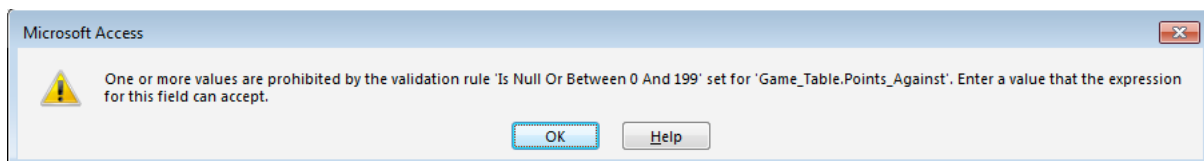


Fig.2 Showing error message for negative numbers

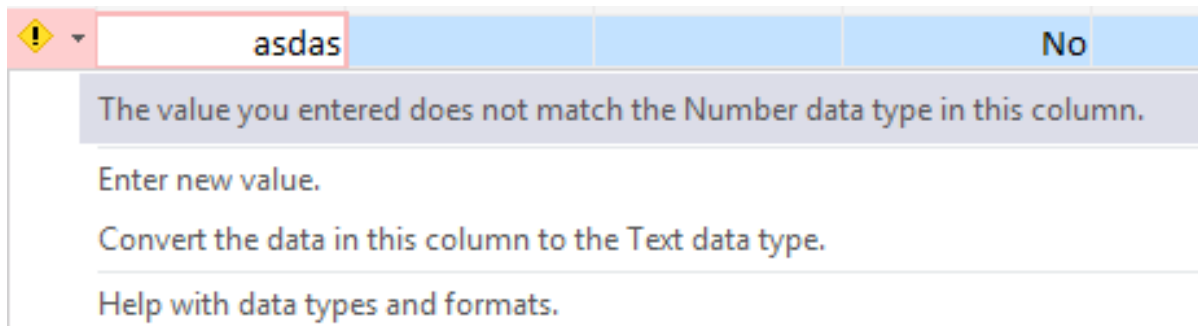


Fig.3 Showing error message for different data type

Sources below were used

<https://support.office.com/en-us/article/Restrict-data-input-by-using-validation-rules-b91c6b15-bcd3-42c1-90bf-e3a0272e988d>

<https://support.office.com/en-us/article/Validation-rules-ae5df363-ef15-4aa1-9b45-3c929314bd33>

Techniques

Calculated fields and Complex Queries

Field Name	Data Type
Game_ID	AutoNumber
Venue	Long Text
Opponent	Short Text
Points_For	Number
Points_Against	Number
Point_Diff	Calculated
Win	Calculated
Date	Date/Time

Fig.4 Calculated Fields

The reason for doing this is to avoid the possible issue of miscalculating the values, this has occurred in the previous system as shown in criterion A. Therefore it is important to have automatically calculated fields so that the issue of wrongly calculating statistics by manually doing the mathematics is eliminated. This not only improves the accuracy, but also reduces the time needed to get the final statistics, as it is done automatically as long as the data needed is inputted, which takes less as the time needed to actually calculate the average and totals is removed.

In order for my to do create the expression as seen below in fig. 5 I needed to learn the syntax of the language used in Access 2016, thus I searched for the official documentation and found Guide to expression syntax by Microsoft from the support office website.

Source: <https://support.office.com/en-us/article/Guide-to-expression-syntax-ebc770bc-8486-4adc-a9ec-7427cce39a90>

General	Lookup
Expression	[Points_For]-[Points_Against]
Result Type	Double

Fig.5 Expression for calculation for the calculated field of Point_Diff

Source: <https://support.office.com/en-us/article/Guide-to-expression-syntax-ebc770bc-8486-4adc-a9ec-7427cce39a90>

For the calculation of statistics the query based on the pseudo code (in Criterion D) was created as shown below, in SQL.

```

SELECT Record_Table.Player_ID, Count(Record_Table.Player_ID) AS Games_Played, Sum(Record_Table.Points) AS Total_Points, Sum(Record_Table.Rebounds) AS Total_Rebounds, Sum(Record_Table.Assists) AS Total_Assists, Sum(Record_Table.Steals) AS Total_Steals, Sum(Record_Table.Blocks) AS Total_Blocks, Sum(Record_Table.FTM) AS Total_FTM, Sum(Record_Table.FTA) AS Total_FTA, Sum(Record_Table.[3PTM]) AS Total_3PTM, ROUND(((Sum(Record_Table.Points))/Games_Played),1) AS Average_Points, ROUND(((Sum(Record_Table.Rebounds))/Games_Played),1) AS Average_Rebounds, ROUND(((Sum(Record_Table.Assists))/Games_Played),1) AS Average_Assists, ROUND(((Sum(Record_Table.Steals))/Games_Played),1) AS Average_Steals, ROUND(((Sum(Record_Table.Blocks))/Games_Played),1) AS Average_Blocks, ROUND(((Sum(Record_Table.FTM))/Games_Played),1) AS Average_FTM, ROUND(((Sum(Record_Table.FTA))/Games_Played),1) AS Average_FTA, ROUND(((Sum(Record_Table.[3PTM])/Games_Played),1) AS Average_3PTM, Format((Total_FTM/Total_FTA),"Percent") AS FT_Percentage
FROM Record_Table
GROUP BY Record_Table.Player_ID;

```

Fig.6 Showing the SQL for the calculations

Source used for SQL: <http://www.w3resource.com/sql/aggregate-functions/sum-with-group-by.php>

Use of graphic fields

The nba website along (as seen from Fig. 7) with other professional statistics websites have profiles of player, similar to my player table, these normally include images of the player. Therefore in order to make the database look more professional, I proposed to my client to include an image of the player so that it can be used in the Player Report, to make it easier to identify the person as normally we can identify face faster than names and also the main point of this is to make the report look as professional as possible with my abilities and the resources which I have.

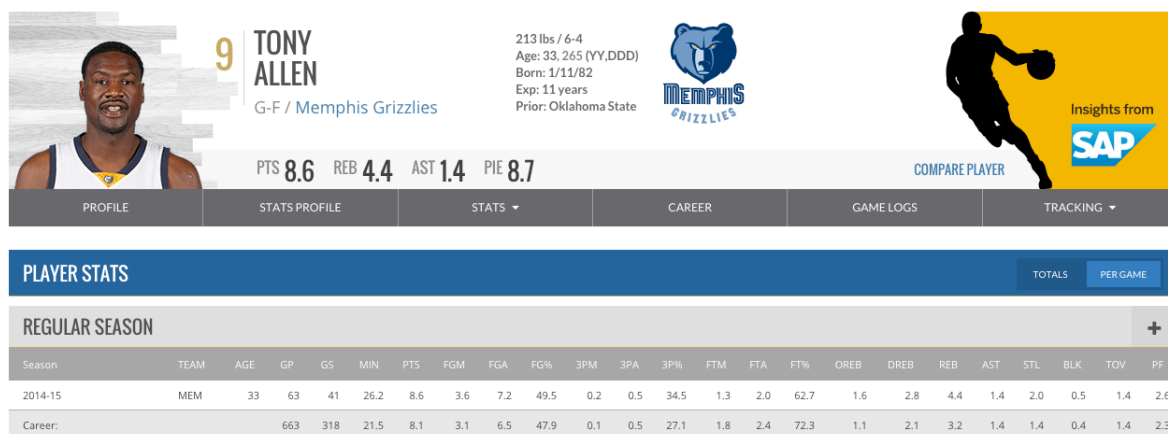


Fig. 7 NBA Profile of Tony Allen

Source: <http://stats.nba.com/player/#!/2754/?p=Tony%20Allen>

Therefore I created a field named Image, which has the specific field type of OLE Object (as shown in Fig.8), this allows it to attach files, such as images, hence making it a graphic field. We simply have to insert the correct object by selecting the correct type as seen from the fig 9.

Player_Table		
	Field Name	Data Type
🔑	Player_ID	AutoNumber
	First_Name	Long Text
	Last_Name	Long Text
	Position	Short Text
	Nationality	Long Text
	DOB	Date/Time
	Games_Played	Short Text
	Image	OLE Object

Fig. 8 showing data type of the field Image

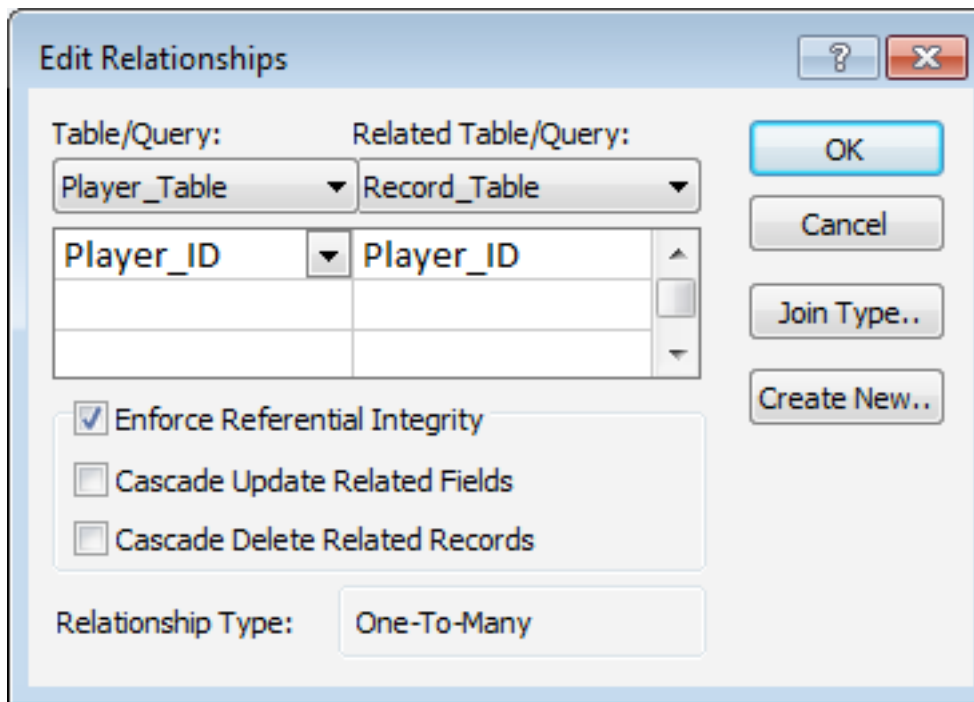


Fig.11 Enforcing relationship

Macros

The reason for Macros being used is to reduce the time needed to use the product, this as a macro reduce repetitive process, by doing all steps in the press of a button. The Macro used was for ensuring that when a record is completed that it is submitted and a new record can be inputted into the form. So time is saved, as only one button is needed. The macro is created by selecting the performances it produces as seen in Fig 13.

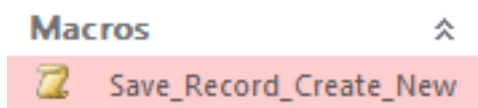


Fig. 12 Showing Macro

```

RunMenuCommand
  Command SaveRecord
GoToRecord
  Object Type
  Object Name
    Record Next
  Offset

```

Fig 13. Showing Contents of Macro

Proficient use of techniques to enable easy navigation

Buttons and a Switchboard were used to allow easy navigation between sections of the database. The Switchboard (essentially a form with buttons for navigation) is there to give the user a general interface allowing them to navigate easily through all parts of the database and even quit the applications (as seen below), the reason for having a Switchboard is that it is more user-friendly for end-user as they do not have to access all individual items, and they are all connect with buttons, which makes it far easier to access different parts.

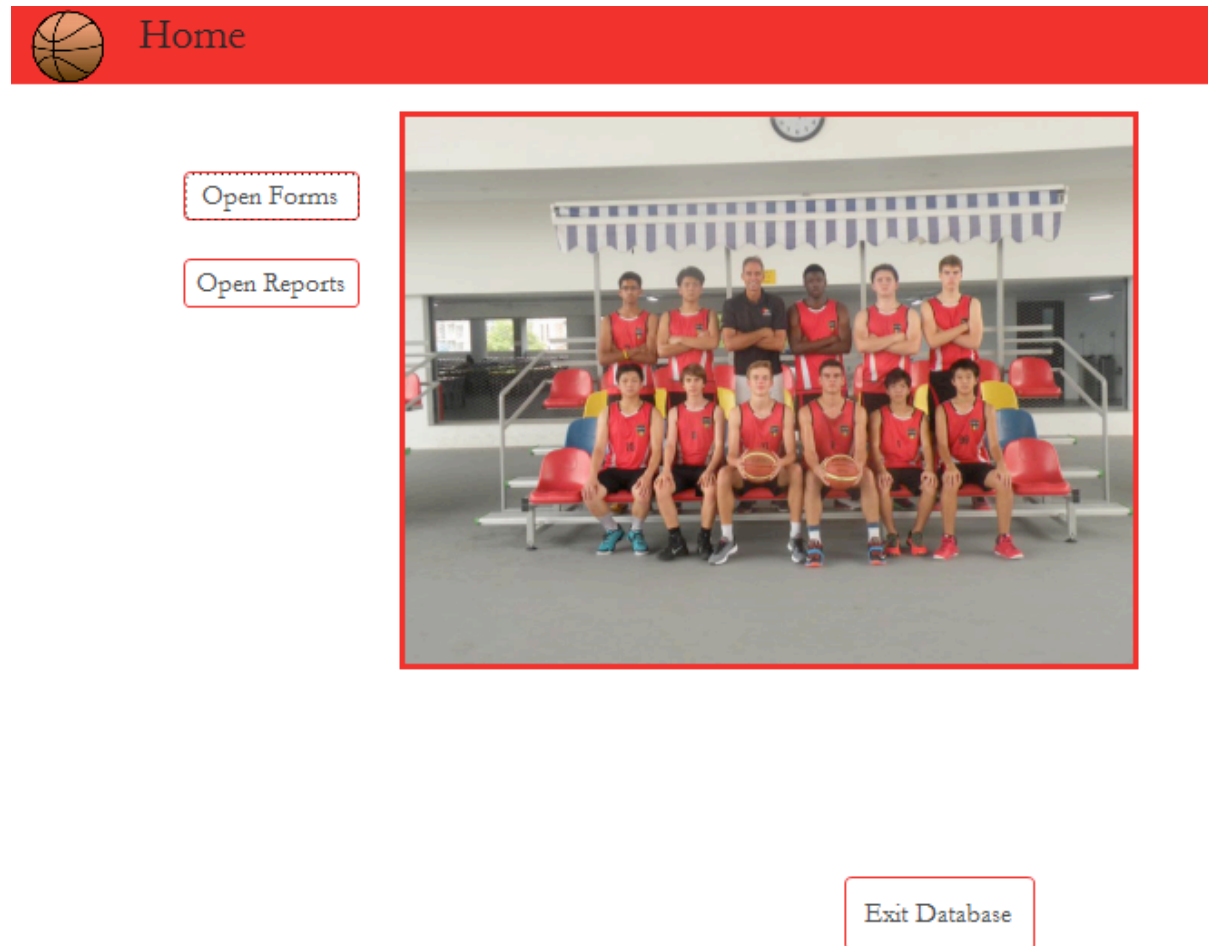


Fig.14 Showing the Main page from the switchboard

The Home page leads to a switchboard for forms or reports, as seen in fig. 15 this allows for easier and faster navigation, therefore reducing time needed.

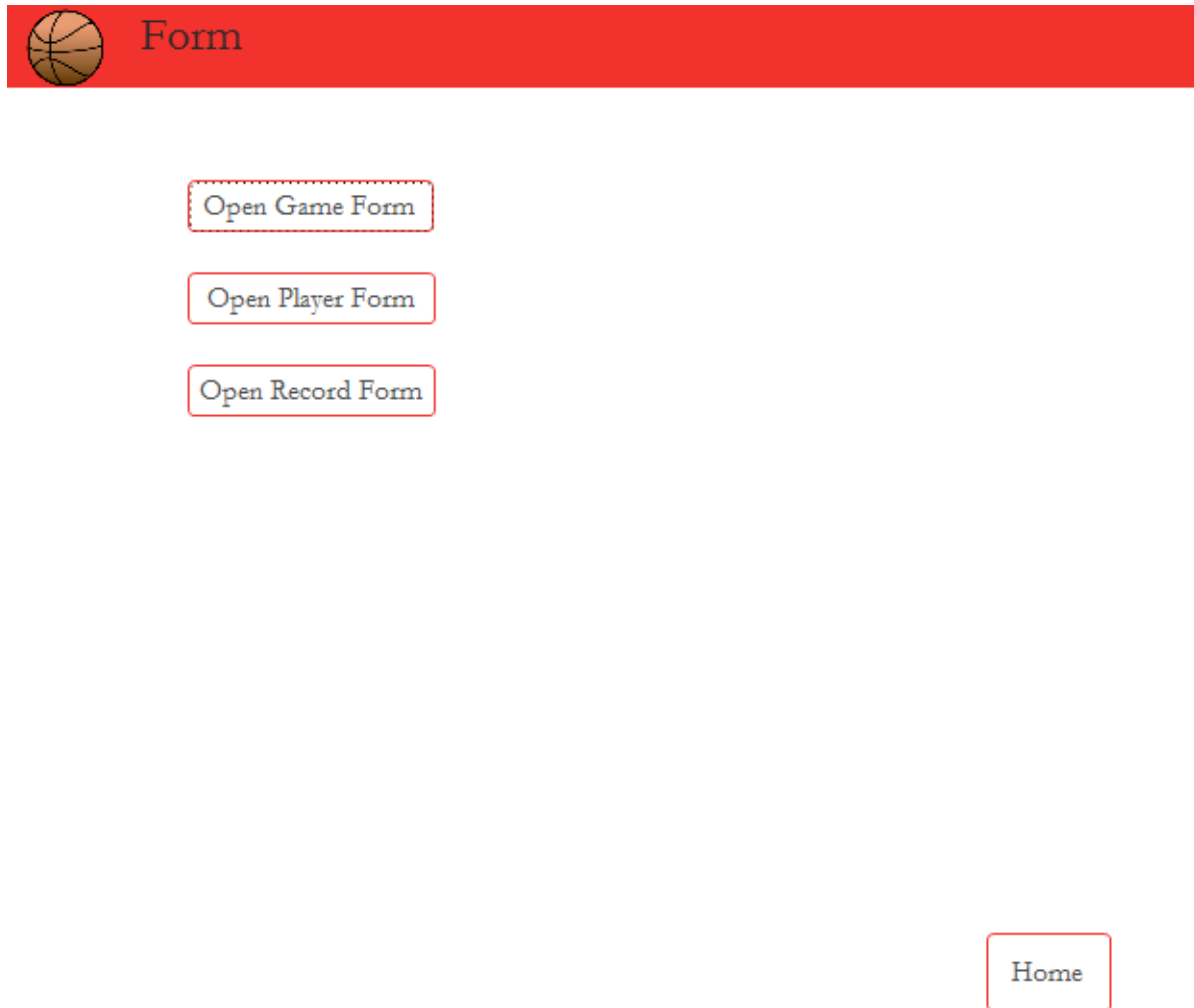



Fig.15 Showing the form switchboard page.

Buttons were also used in forums just like in the switchboards, to allow for easy navigation.

This can be seen on the next page for which buttons are used to navigate between records. Note buttons are created similarly to the macros but only include one step.



Player Form


Player_ID	<input type="text" value="10"/>
First_Name	<input type="text" value="Adam"/>
Last_Name	<input type="text" value="Beilin"/>
Position	<input type="text" value="SF"/>
Nationality	<input type="text" value="Danish"/>
DOB	<input type="text" value="5/30/1998"/>
Image	

Fig.16 Showing form with a lot of options for navigation.

Password

To ensure that the data In the database is never edited by anyone without permission, a password is needed, as otherwise anyone could edit the database if my client leaves his computer on without anyone ensuring that only people with permission. Hence there is a need for a password. This ensure from the start that only users with permission can use it, if they are not using it they should close it to prevent unauthorized access. The screen if the database is opened is shown below in Fig. 17

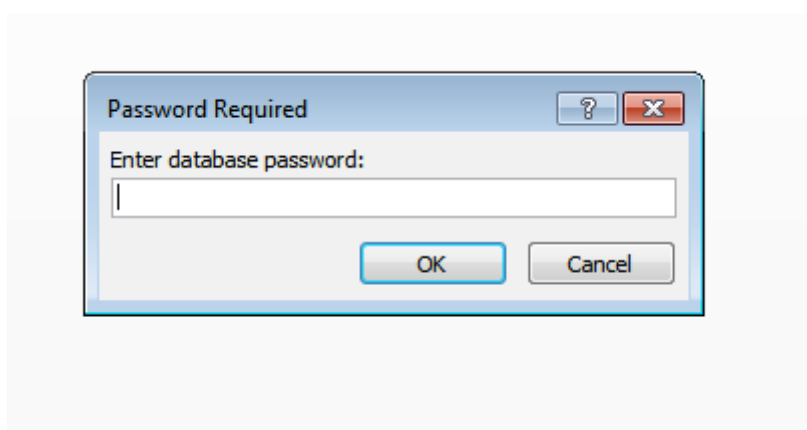


Fig. 17 Password required before even entering the database

Word count : 1055