Applied Information and Communication Technology

Unit 7: Using Database Software
Summer 2010
Example Solution and Principal Examiner Advice and Guidance

The pack includes:

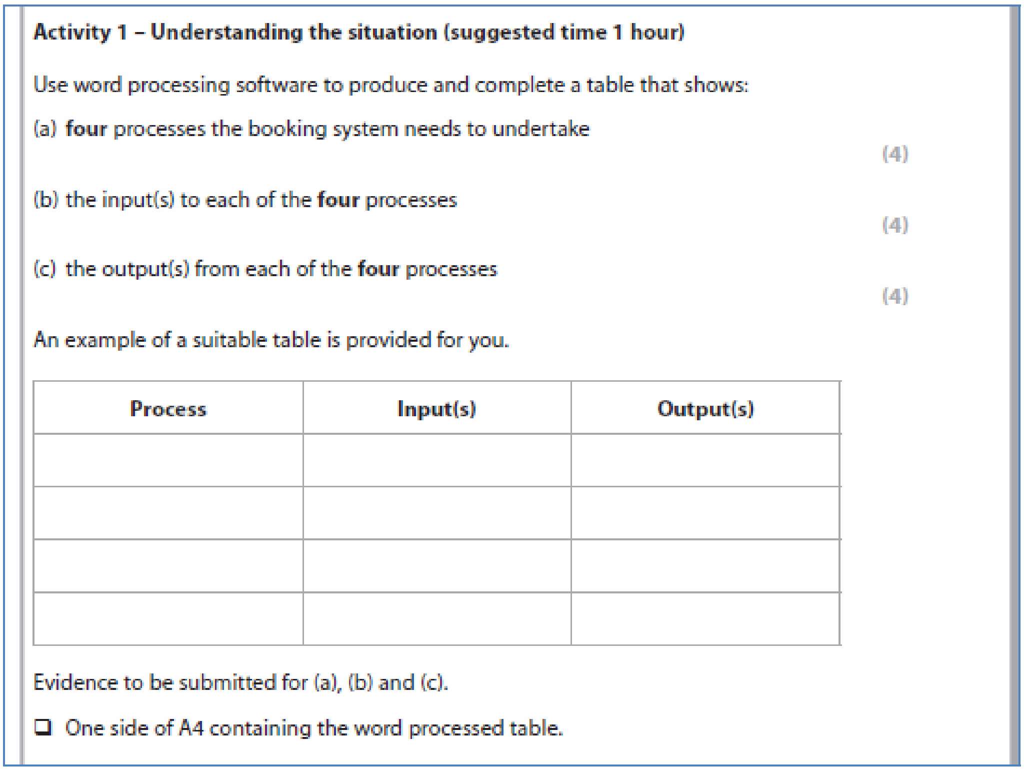
- **one** example answer for each question in the examination paper
- the database from which the answers are based – Version1
- another two database solutions:
  - Version2 a macro based solution
  - Version 3 a coded solution
- updated data files with the anomalies removed

Please note there were many different ways candidates could have implemented the solution and that this is in no way put forward as the best. It is purely designed to be an aid for both teachers and students completing unit 7.

Activity 1

Introduction

Activity 1 is designed to test 7.2 of the unit specification i.e. the Functional Specification. Any aspect of 7.2 can be tested in any one examination.



Activity 1 – Understanding the situation (suggested time 1 hour)

Use word processing software to produce and complete a table that shows:

(a) **four** processes the booking system needs to undertake (4)

(b) the input(s) to each of the **four** processes (4)

(c) the output(s) from each of the **four** processes (4)

An example of a suitable table is provided for you.

Process	Input(s)	Output(s)

Evidence to be submitted for (a), (b) and (c).

❑ One side of A4 containing the word processed table.

In this instance candidates were tested on their ability to recognise inputs, processes and outputs.

An Example Answer

Process	**Input(s)**	**Output(s)**
Create Customer ID	Highest existing Customer ID	Highest existing customer ID + 1
Find available seats	Performance date and type	Vacant seat details
Add new customer	Customer details including Name and address, town, postcode, card number	New customer details stored in customer table
Create Ticket Number	Highest existing ticket number	Highest existing ticket number +1

Notes

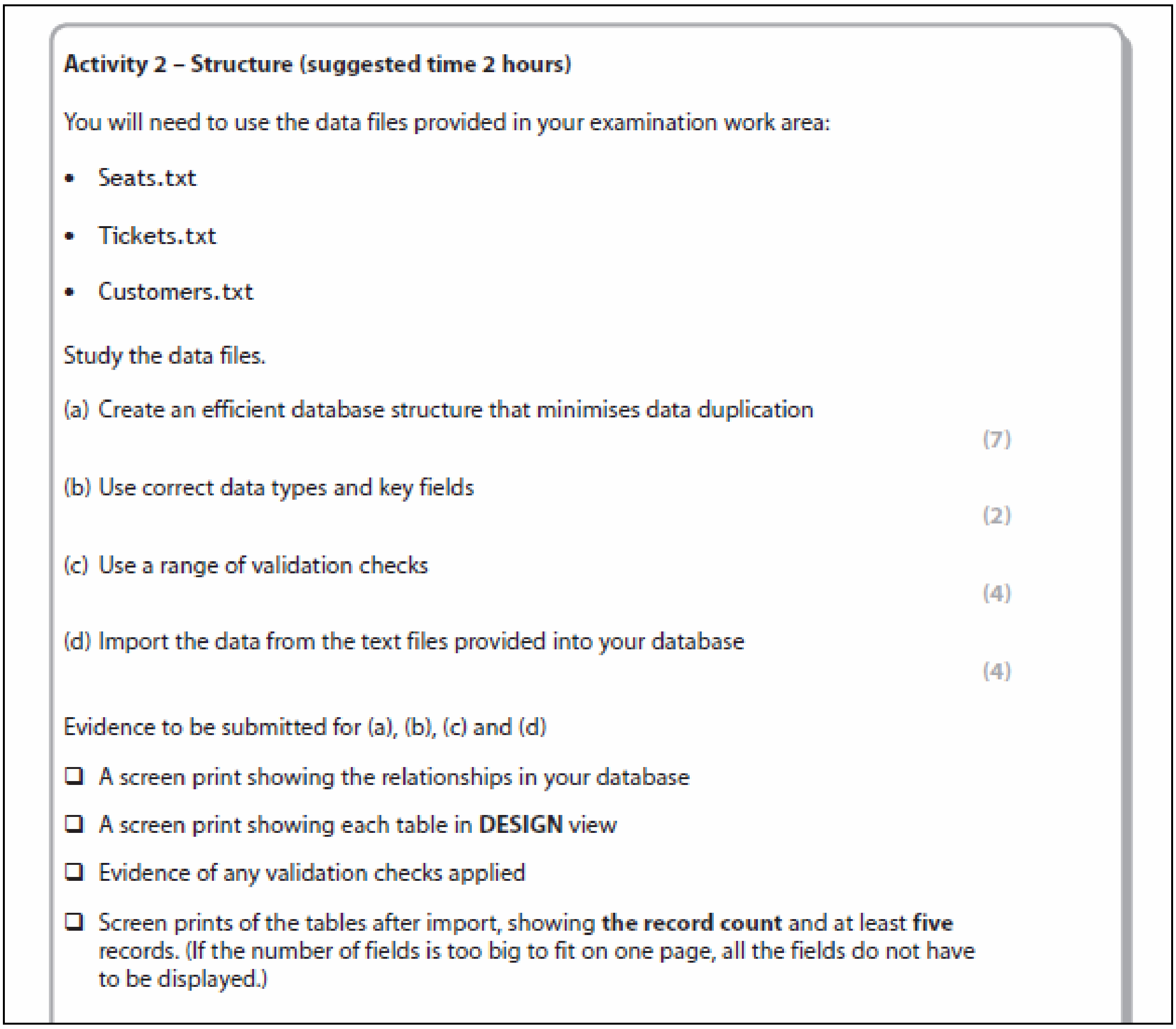
At times candidates tend to provide long winded commentaries as opposed to process descriptions. For example:

Process
When the details of the customer have been entered by the user the customer ID will need to be worked out and the customer ID will need to go up by one
A query will be built and used to try and find seats that are free this will look for those that are empty using Is Null
When the customer details have been put in the form the save button will be clicked and the new customer will be added to the customer table. A query will run that will append all of the details.

Activity 2

Introduction

Activity 2 is designed to test 7.4, 7.5 and 7.6 of the unit specification ie the Functional Specification. Any aspect of these can be tested.



Activity 2 – Structure (suggested time 2 hours)

You will need to use the data files provided in your examination work area:

- Seats.txt
- Tickets.txt
- Customers.txt

Study the data files.

(a) Create an efficient database structure that minimises data duplication (7)

(b) Use correct data types and key fields (2)

(c) Use a range of validation checks (4)

(d) Import the data from the text files provided into your database (4)

Evidence to be submitted for (a), (b), (c) and (d)

□ A screen print showing the relationships in your database

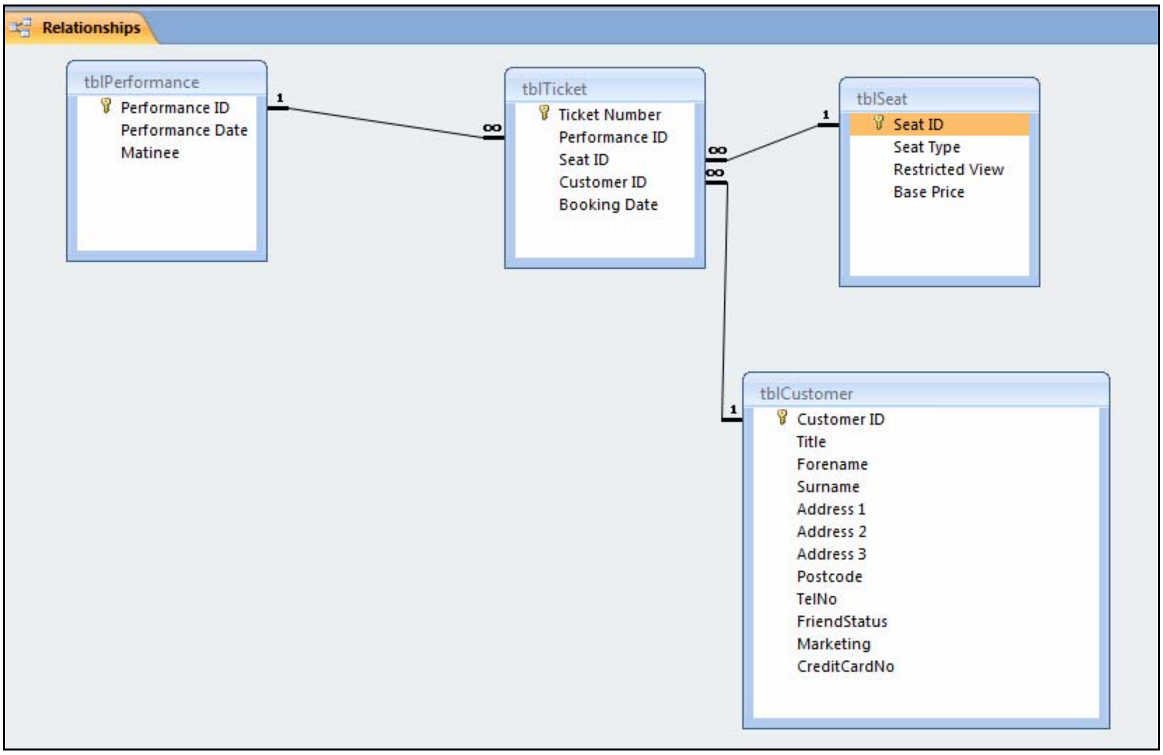
□ A screen print showing each table in **DESIGN** view

□ Evidence of any validation checks applied

□ Screen prints of the tables after import, showing **the record count** and at least **five** records. (If the number of fields is too big to fit on one page, all the fields do not have to be displayed.)

An Example Answer

1. Database Structure



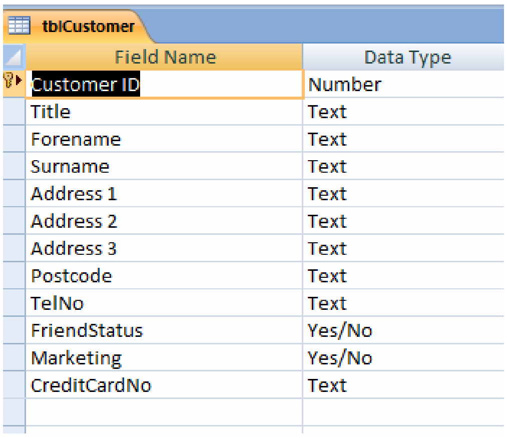
tblPerformance
- Performance ID
- Performance Date
- Matinee

tblTicket
- Ticket Number
- Performance ID
- Seat ID
- Customer ID
- Booking Date

tblSeat
- Seat ID
- Seat Type
- Restricted View
- Base Price

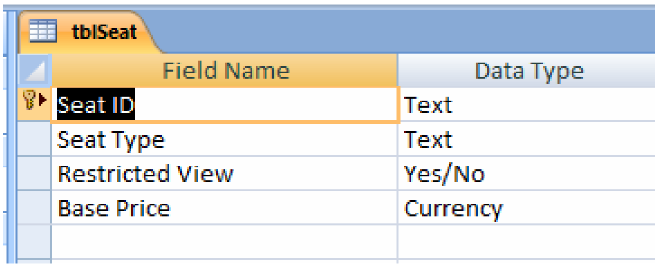
tblCustomer
- Customer ID
- Title
- Forename
- Surname
- Address 1
- Address 2
- Address 3
- Postcode
- TelNo
- FriendStatus
- Marketing
- CreditCardNo

2. Data Types and Keys



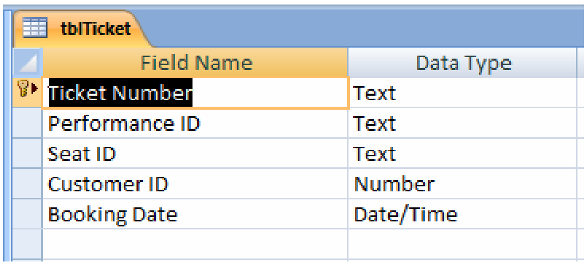
tblCustomer

Field Name	Data Type
Customer ID	Number
Title	Text
Forename	Text
Surname	Text
Address 1	Text
Address 2	Text
Address 3	Text
Postcode	Text
TelNo	Text
FriendStatus	Yes/No
Marketing	Yes/No
CreditCardNo	Text



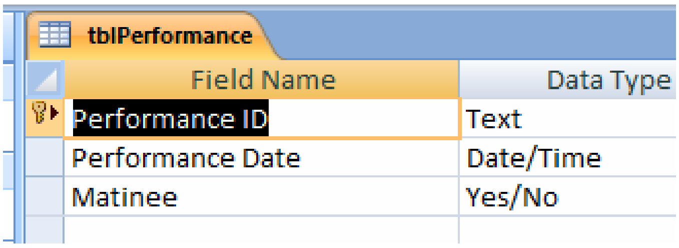
tblSeat

Field Name	Data Type
Seat ID	Text
Seat Type	Text
Restricted View	Yes/No
Base Price	Currency



tblTicket

Field Name	Data Type
Ticket Number	Text
Performance ID	Text
Seat ID	Text
Customer ID	Number
Booking Date	Date/Time

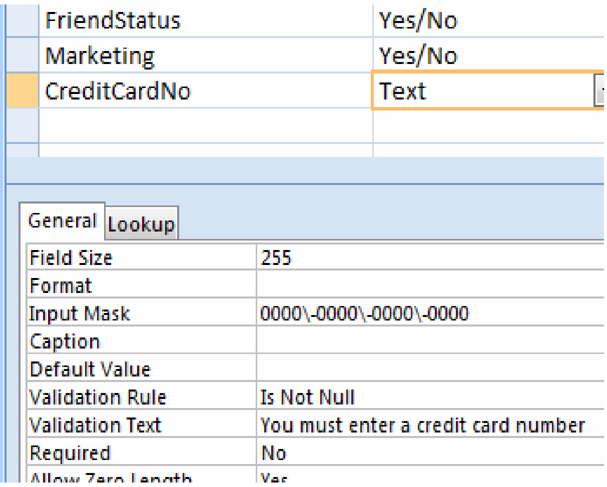


tblPerformance

Field Name	Data Type
Performance ID	Text
Performance Date	Date/Time
Matinee	Yes/No

3. Validation Checks

Format and Presence Check

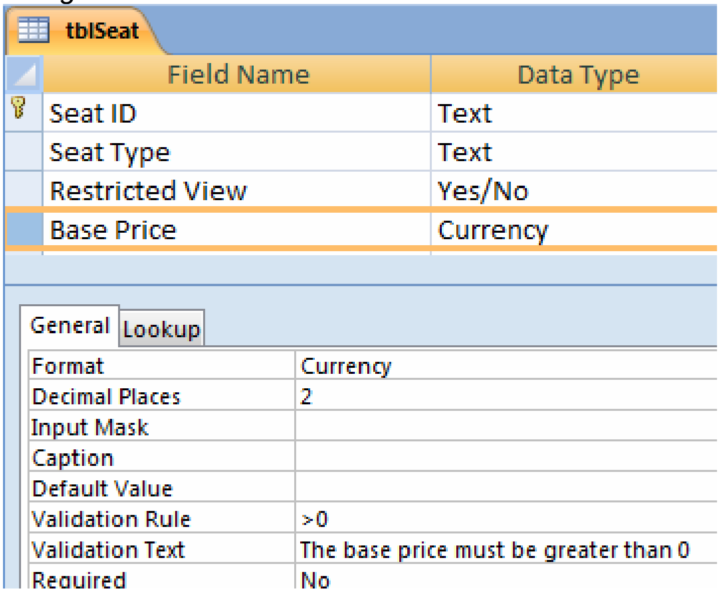


FriendStatus	Yes/No
Marketing	Yes/No
CreditCardNo	Text

General | Lookup

Field Size	255
Format	
Input Mask	0000\-0000\-0000\-0000
Caption	
Default Value	
Validation Rule	Is Not Null
Validation Text	You must enter a credit card number
Required	No
Allow Zero Length	Yes

Range Check



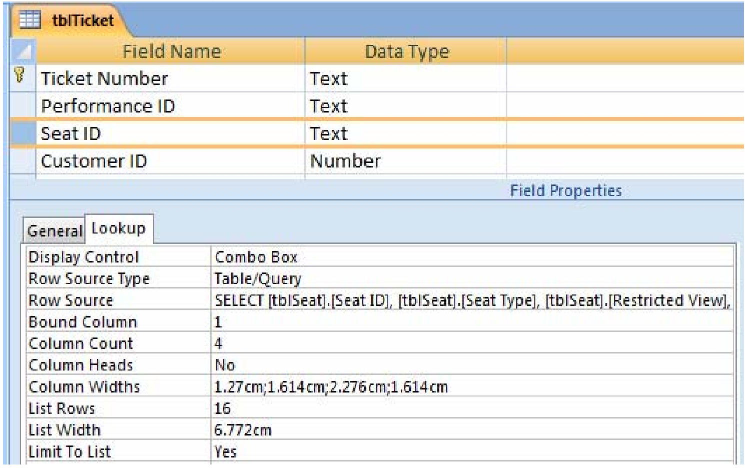
tblSeat

Field Name	Data Type
Seat ID	Text
Seat Type	Text
Restricted View	Yes/No
Base Price	Currency

General | Lookup

Format	Currency
Decimal Places	2
Input Mask	
Caption	
Default Value	
Validation Rule	>0
Validation Text	The base price must be greater than 0
Required	No

List or Table Lookup Check



tblTicket

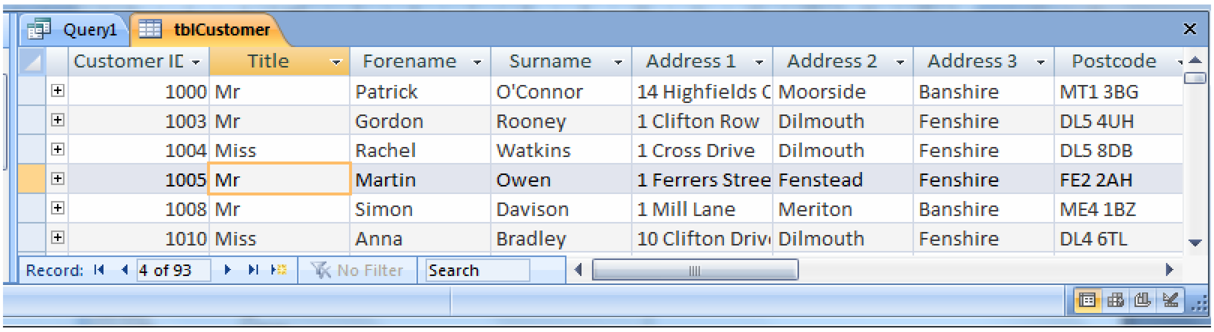
	Field Name	Data Type
🔑	Ticket Number	Text
	Performance ID	Text
	Seat ID	Text
	Customer ID	Number

Field Properties

General | Lookup

Display Control	Combo Box
Row Source Type	Table/Query
Row Source	SELECT [tblSeat].[Seat ID], [tblSeat].[Seat Type], [tblSeat].[Restricted View],
Bound Column	1
Column Count	4
Column Heads	No
Column Widths	1.27cm;1.614cm;2.276cm;1.614cm
List Rows	16
List Width	6.772cm
Limit To List	Yes

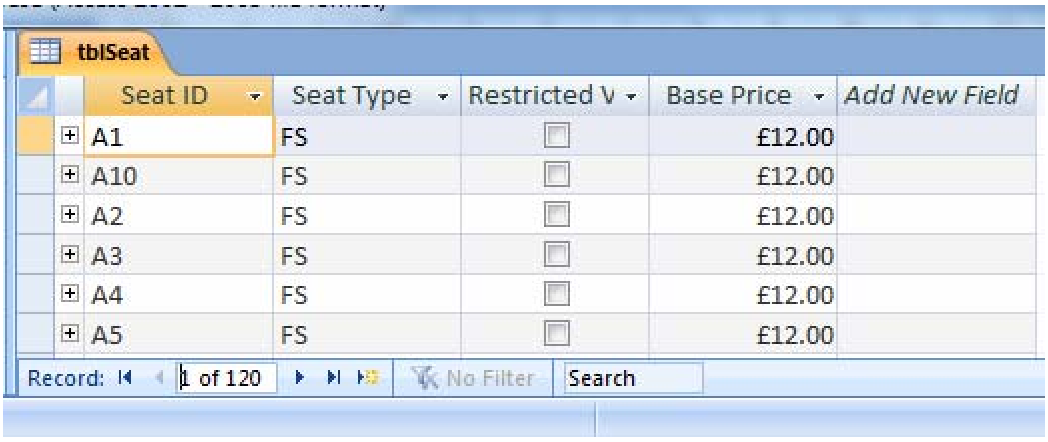
(d)



Query1 | tblCustomer

Customer IC	Title	Forename	Surname	Address 1	Address 2	Address 3	Postcode
1000	Mr	Patrick	O'Connor	14 Highfields C	Moorside	Banshire	MT1 3BG
1003	Mr	Gordon	Rooney	1 Clifton Row	Dilmouth	Fenshire	DL5 4UH
1004	Miss	Rachel	Watkins	1 Cross Drive	Dilmouth	Fenshire	DL5 8DB
1005	Mr	Martin	Owen	1 Ferrers Stree	Fenstead	Fenshire	FE2 2AH
1008	Mr	Simon	Davison	1 Mill Lane	Meriton	Banshire	ME4 1BZ
1010	Miss	Anna	Bradley	10 Clifton Driv	Dilmouth	Fenshire	DL4 6TL

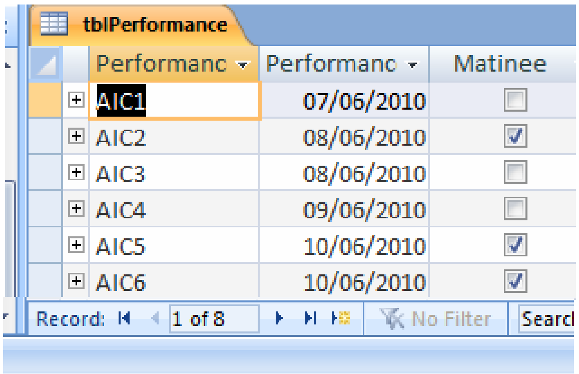
Record: 4 of 93 No Filter Search



tblSeat

Seat ID	Seat Type	Restricted V	Base Price	Add New Field
A1	FS	☐	£12.00	
A10	FS	☐	£12.00	
A2	FS	☐	£12.00	
A3	FS	☐	£12.00	
A4	FS	☐	£12.00	
A5	FS	☐	£12.00	

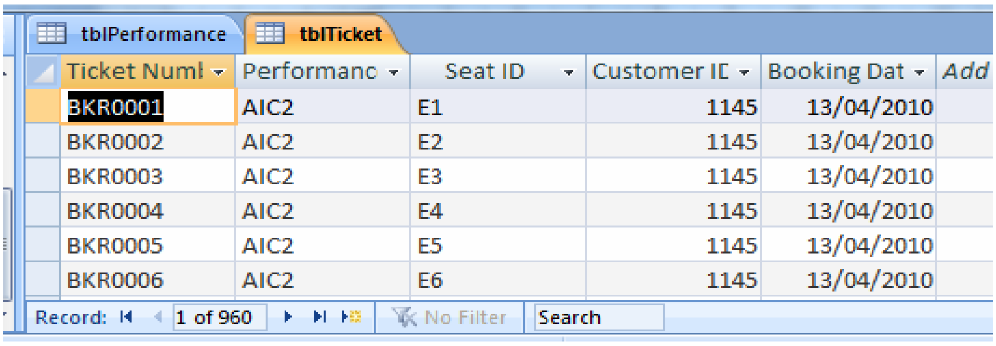
Record: 1 of 120 No Filter Search



tblPerformance

Performanc	Performanc	Matinee
AIC1	07/06/2010	☐
AIC2	08/06/2010	☑
AIC3	08/06/2010	☐
AIC4	09/06/2010	☐
AIC5	10/06/2010	☑
AIC6	10/06/2010	☑

Record: 1 of 8 No Filter Searcl



Ticket Numl	Performanc	Seat ID	Customer IC	Booking Dat	Add
BKR0001	AIC2	E1	1145	13/04/2010	
BKR0002	AIC2	E2	1145	13/04/2010	
BKR0003	AIC2	E3	1145	13/04/2010	
BKR0004	AIC2	E4	1145	13/04/2010	
BKR0005	AIC2	E5	1145	13/04/2010	
BKR0006	AIC2	E6	1145	13/04/2010	

Notes

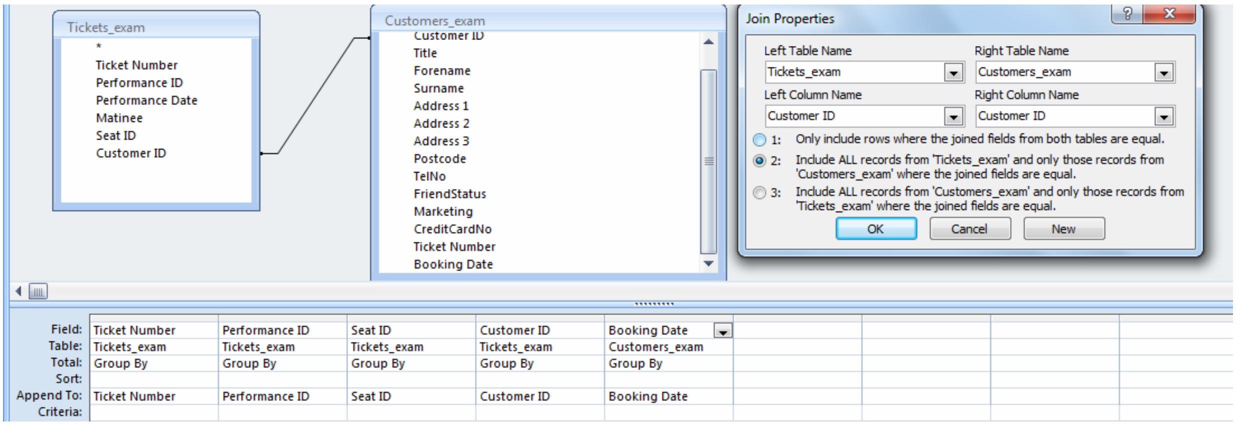
You will see that there are no annotations what-so-ever in the answer provided. None are required in this activity and it is usually possible to answer this question using three to four sheets of paper.

Candidates can go wrong in this activity in a number of places:

1. They do not provide evidence in the order they have been asked to which can make it extremely difficult, at times impossible, for the examiner to find the correct evidence.
2. They do not make sure all of the tables etc can be seen in the relationship diagram or they have not enforced referential integrity.
3. They show how they have built tables, relationships, append queries, input masks, changed data types, created range checks etc. They are only asked to show the final evidence.
4. You cannot see the table names in the screenshots.
5. You cannot see validation etc clearly as the screenshots have been cut off or the field the validation belongs to is not in the screenshot.
6. They cut off the number of records at the bottom of the tables when they crop the screenshots, they do not show at least five records or they take a screenshot before the number of records is shown.
7. They show how they have normalised or include entity relationship diagrams.

The mark scheme was flexible in the number of tables created and keys etc due to how candidates approached the problem of anomalies in the data sets and the need for a Customer ID to be linked to a ticket.

The append query used in order to build the ticket table in this example is shown below. A link has been added between the Ticket and Customer data set. The join type was altered by double clicking the link and changing it to ***include all records from Ticket_exam and only those records from Customer_exam where the joined fields are equal***. This allows the addition of records where the Customer ID is empty.



Left Table Name: Tickets_exam
Right Table Name: Customers_exam
Left Column Name: Customer ID
Right Column Name: Customer ID

○ 1: Only include rows where the joined fields from both tables are equal.
● 2: Include ALL records from 'Tickets_exam' and only those records from 'Customers_exam' where the joined fields are equal.
○ 3: Include ALL records from 'Customers_exam' and only those records from 'Tickets_exam' where the joined fields are equal.

Field:	Ticket Number	Performance ID	Seat ID	Customer ID	Booking Date
Table:	Tickets_exam	Tickets_exam	Tickets_exam	Tickets_exam	Customers_exam
Total:	Group By	Group By	Group By	Group By	Group By
Sort:					
Append To:	Ticket Number	Performance ID	Seat ID	Customer ID	Booking Date
Criteria:					

Activity 3

Introduction

Activity 3 is designed to test 7.3 and 7.7 of the unit specification ie the Functional Specification. Any aspect of these can be tested.

Activity 3 – The booking system (suggested time 3 hours and 30 minutes)

Creating the interface

(a) The new booking system must be able to add new customers and make bookings efficiently.

(i) Produce a query to check if the person on the telephone is an existing customer.

Screenprint the query in **DESIGN** view.

(1)

(ii) Produce a query to generate a unique customer ID.

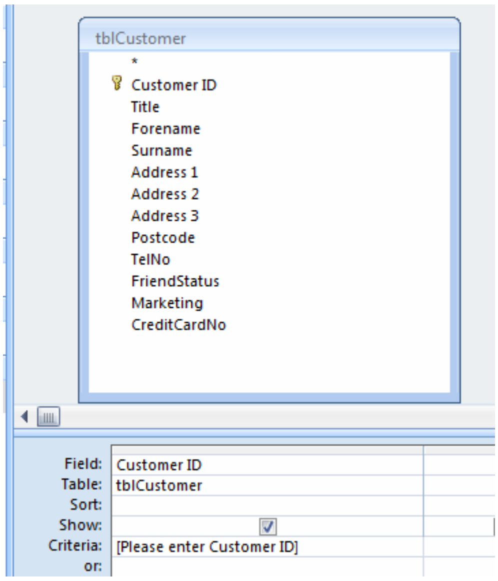
Screenprint the query in **DESIGN** view.

(1)

Part A

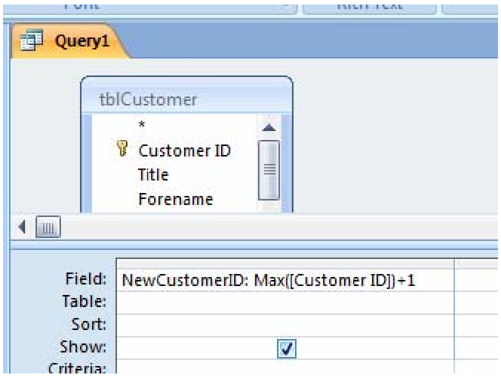
An Example Answer

(i)



Field:	Customer ID
Table:	tblCustomer
Sort:	
Show:	☑
Criteria:	[Please enter Customer ID]
or:	

(ii)



Field:	NewCustomerID: Max([Customer ID])+1
Table:	
Sort:	
Show:	☑
Criteria:	

Notes

You will see that there are no annotations what-so-ever in part A. None are required as the examiner can clearly see whether the task has been carried out correctly as long as the screenshots are clear. Please note the generation of primary keys are always a part of the examination and can be generated in a number of different ways. In this part of the activity the candidates were specifically directed to create a query to generate the new primary key. This is not always the case so they do need to read the question carefully.

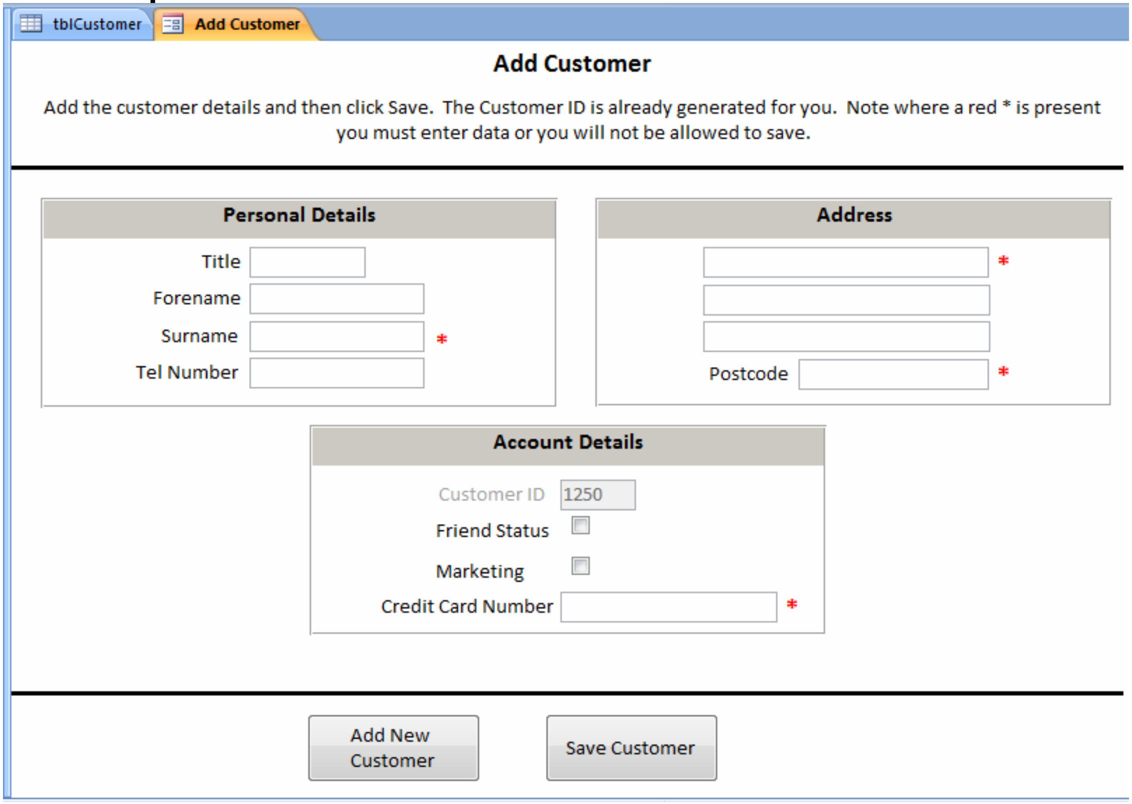
Part B

(b) The receptionist will need to be able to use the system to add new customers.

Create a form to enable the receptionist to add a new customer.

Screenprint the form in **FORM** view.

An Example Answer



Add Customer

Add the customer details and then click Save. The Customer ID is already generated for you. Note where a red * is present you must enter data or you will not be allowed to save.

Personal Details

Title
Forename
Surname *
Tel Number

Address

*

Postcode *

Account Details

Customer ID 1250
Friend Status □
Marketing □
Credit Card Number *

Notes

You will see that there are no annotations what-so-ever in answer provided. None are required for this part of the activity.

Part C

(c) The receptionist will need to be able to use the booking system to choose a specific performance and to search for seats that have not yet been booked.

(i) Create a form to enable the receptionist to enter the date and time of the performance.

Screenprint the form in **DESIGN** view.

(1)

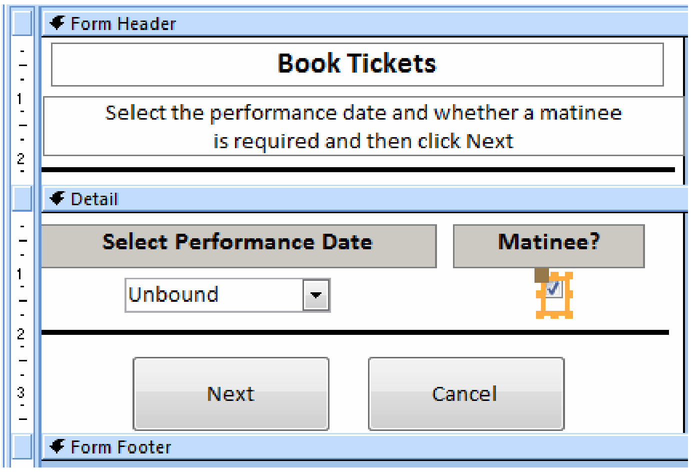
(ii) Create a query to search for seats that have not been booked.

Screenprint the query in **DESIGN** view.

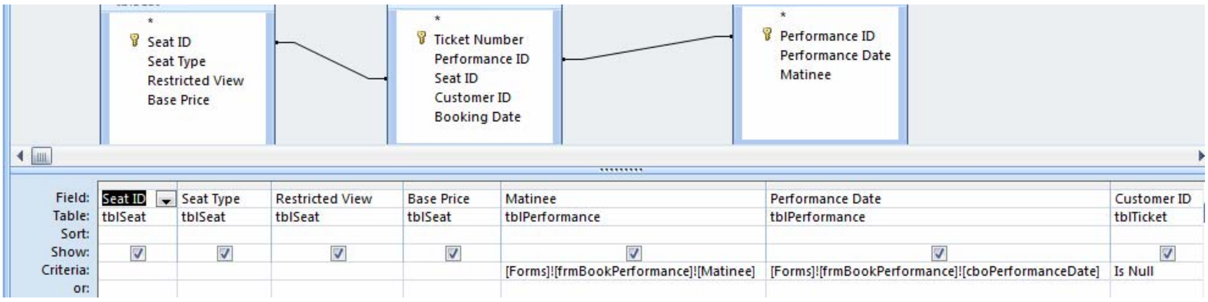
(2)

An Example Answer

(i)



(ii)

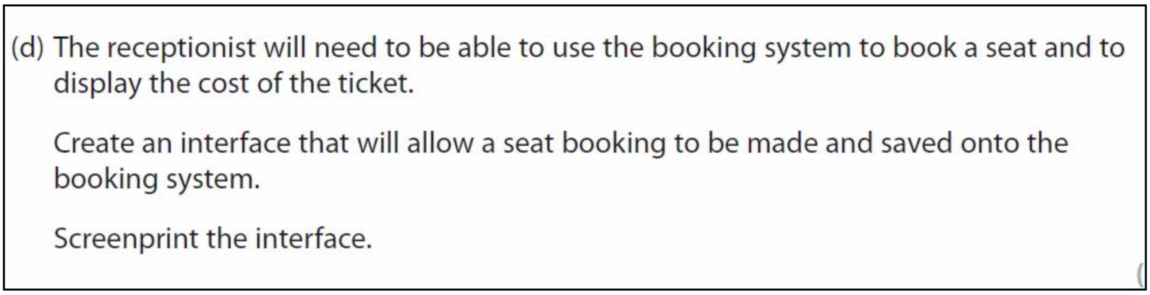


Field:	Seat ID	Seat Type	Restricted View	Base Price	Matinee	Performance Date	Customer ID
Table:	tblSeat	tblSeat	tblSeat	tblSeat	tblPerformance	tblPerformance	tblTicket
Sort:							
Show:	☑	☑	☑	☑	☑	☑	☑
Criteria:					[Forms]![frmBookPerformance]![Matinee]	[Forms]![frmBookPerformance]![cboPerformanceDate]	Is Null
or:							

Notes

You will see that there are no annotations what-so-ever in answer provided. None are required for this part of the activity.

Part D

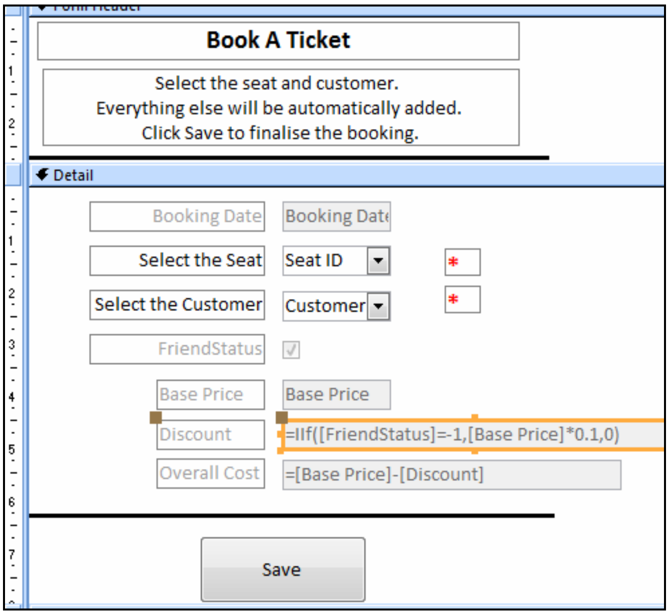


(d) The receptionist will need to be able to use the booking system to book a seat and to display the cost of the ticket.

Create an interface that will allow a seat booking to be made and saved onto the booking system.

Screenprint the interface.

An Example Answer



Form Header

Book A Ticket

Select the seat and customer.
Everything else will be automatically added.
Click Save to finalise the booking.

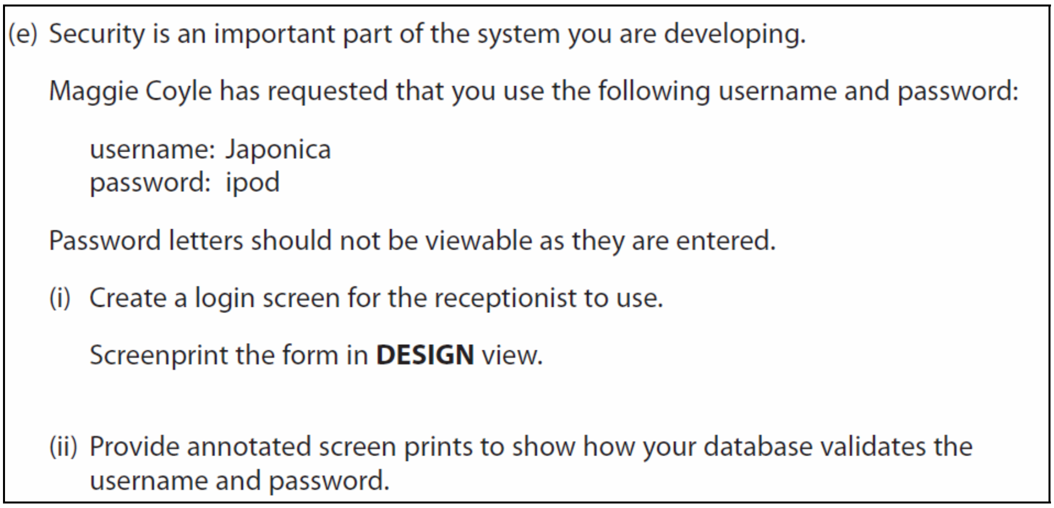
Detail

Booking Date	Booking Date
Select the Seat	Seat ID
Select the Customer	Customer
FriendStatus	☑
Base Price	Base Price
Discount	=IIf([FriendStatus]=-1,[Base Price]*0.1,0)
Overall Cost	=[Base Price]-[Discount]

Notes

You will see that there are no annotations what-so-ever in answer provided. None are required for this part of the activity.

Part E



(e) Security is an important part of the system you are developing.

Maggie Coyle has requested that you use the following username and password:

username: Japonica
password: ipod

Password letters should not be viewable as they are entered.

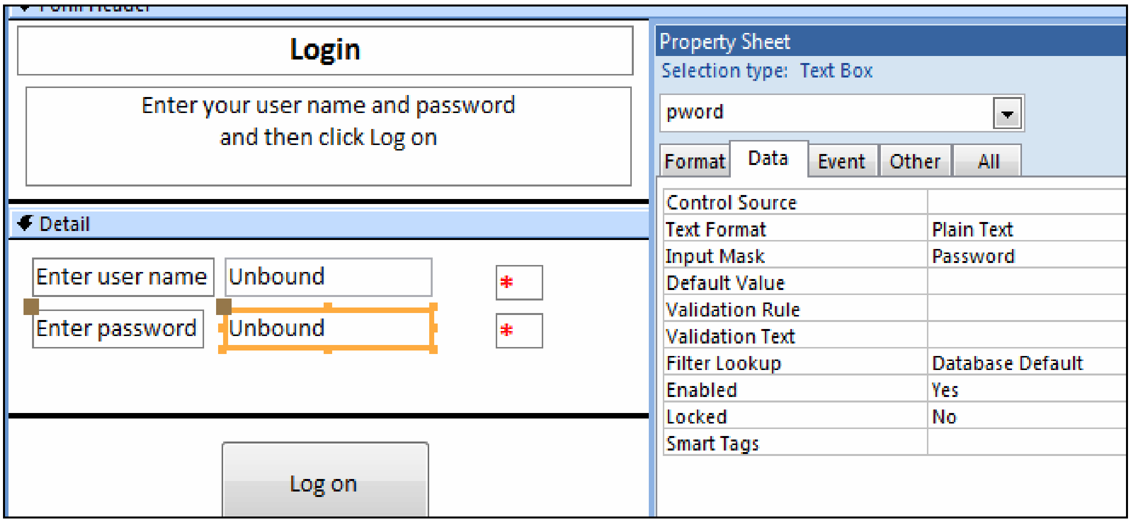
(i) Create a login screen for the receptionist to use.

Screenprint the form in **DESIGN** view.

(ii) Provide annotated screen prints to show how your database validates the username and password.

An Example Answer

(i)



Login

Enter your user name and password and then click Log on

Detail

Enter user name: Unbound *

Enter password: Unbound *

Log on

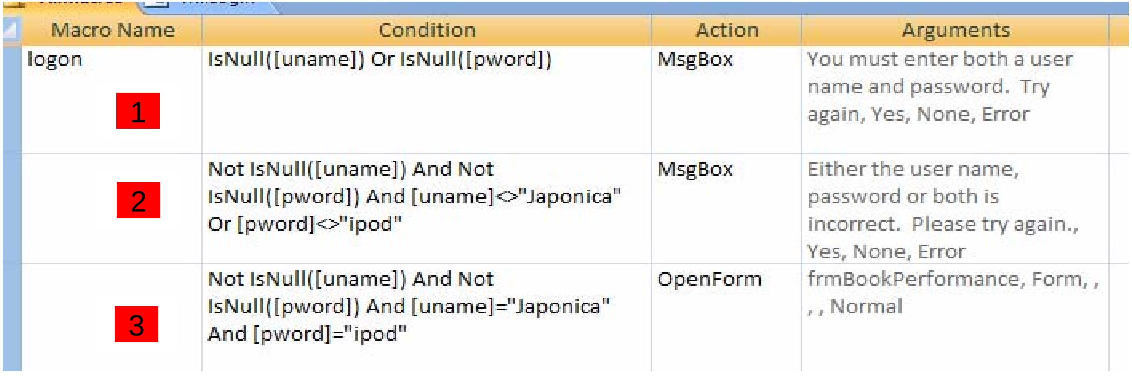
Property Sheet

Selection type: Text Box

pword

Control Source	
Text Format	Plain Text
Input Mask	Password
Default Value	
Validation Rule	
Validation Text	
Filter Lookup	Database Default
Enabled	Yes
Locked	No
Smart Tags	

(ii)



Macro Name	Condition	Action	Arguments
logon 1	IsNull([uname]) Or IsNull([pword])	MsgBox	You must enter both a user name and password. Try again, Yes, None, Error
2	Not IsNull([uname]) And Not IsNull([pword]) And [uname]<>"Japonica" Or [pword]<>"ipod"	MsgBox	Either the user name, password or both is incorrect. Please try again., Yes, None, Error
3	Not IsNull([uname]) And Not IsNull([pword]) And [uname]="Japonica" And [pword]="ipod"	OpenForm	frmBookPerformance, Form, , , , Normal

1. The first line of the macro uses IsNull to check to see whether the username or password has been left empty. If it has an error message is displayed to tell the user
2. The second line of the macro checks to see that the username and password have been entered then checks to see if the user name is not Japonica or the password is not ipod. If either are wrong then an error message is displayed tell the user they have a problem
3. The third line of the macro checks to see if the user name and password have been entered and the user name is Japonica and password ipod. If they are the find performance form opens.

Notes

This screenshot was accompanied by annotations purely because the candidates were specifically asked to include annotations. Encourage your candidates to annotate only when they are asked to do so or think that clarification is needed for the examiner.

Candidates can go wrong in this activity in a number of places:

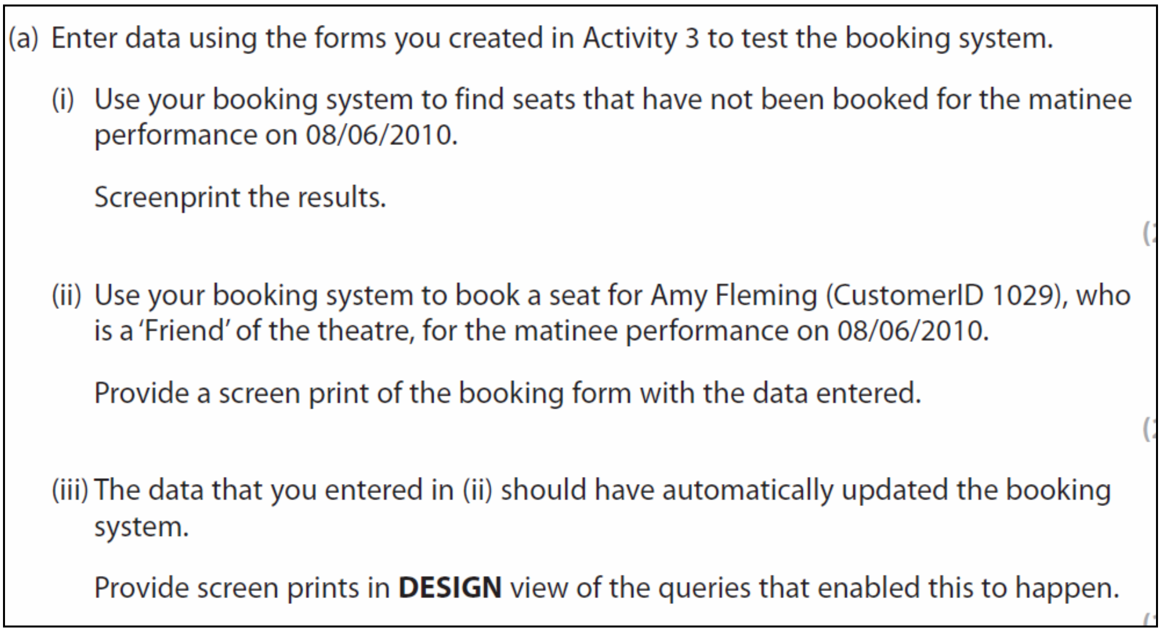
1. They do not provide the correct view of the form ie if asked for design view they provide form view and vice versa
2. They do not show the properties of the fields out in full eg when a calculation is carried out the field is not wide enough to show the formulae used etc
3. They do far too many screenshots and far too many annotations.
4. They truncate macro screenshots
5. They truncate query screenshots so criterion cannot be clearly seen.

Activity 4

Introduction

Activity 4 is designed to test 7.8 of the unit specification Testing. Any aspect of this can be tested.

Part A



(a) Enter data using the forms you created in Activity 3 to test the booking system.

(i) Use your booking system to find seats that have not been booked for the matinee performance on 08/06/2010.

Screenprint the results.

(ii) Use your booking system to book a seat for Amy Fleming (CustomerID 1029), who is a 'Friend' of the theatre, for the matinee performance on 08/06/2010.

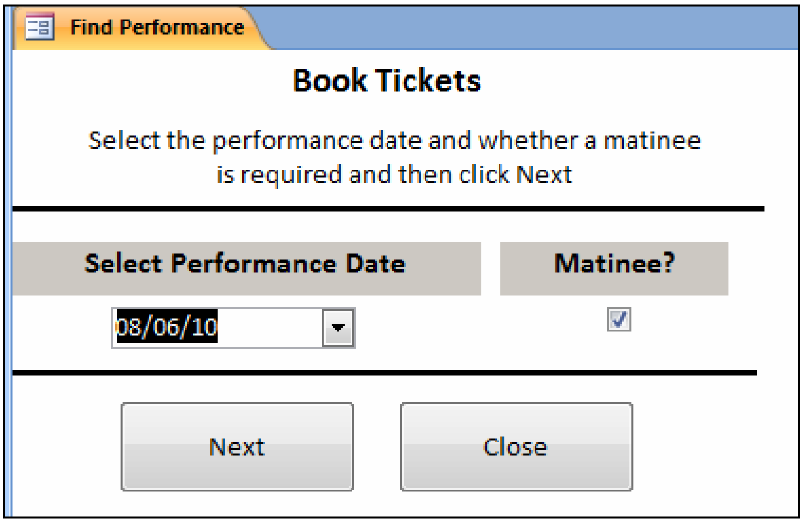
Provide a screen print of the booking form with the data entered.

(iii) The data that you entered in (ii) should have automatically updated the booking system.

Provide screen prints in **DESIGN** view of the queries that enabled this to happen.

An Example Answer

(i)

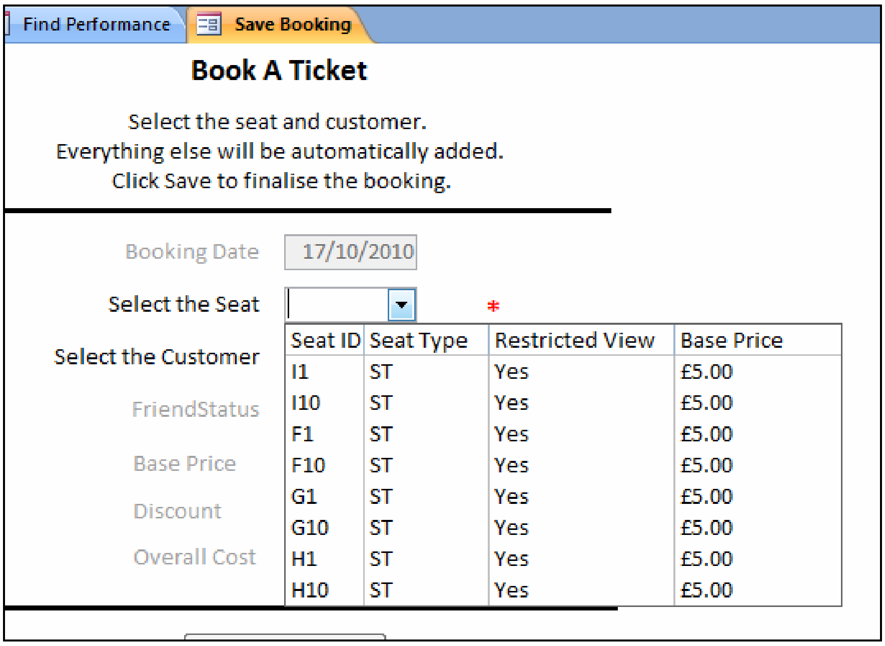


Book Tickets

Select the performance date and whether a matinee is required and then click Next

Select Performance Date: 08/06/10

Matinee? ☑



Book A Ticket

Select the seat and customer.
Everything else will be automatically added.
Click Save to finalise the booking.

Booking Date 17/10/2010

Select the Seat *

Select the Customer

FriendStatus

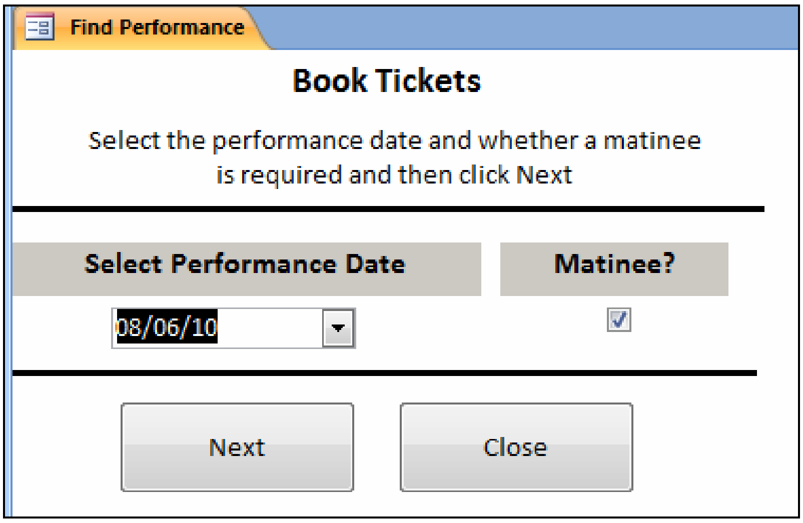
Base Price

Discount

Overall Cost

Seat ID	Seat Type	Restricted View	Base Price
I1	ST	Yes	£5.00
I10	ST	Yes	£5.00
F1	ST	Yes	£5.00
F10	ST	Yes	£5.00
G1	ST	Yes	£5.00
G10	ST	Yes	£5.00
H1	ST	Yes	£5.00
H10	ST	Yes	£5.00

(ii)

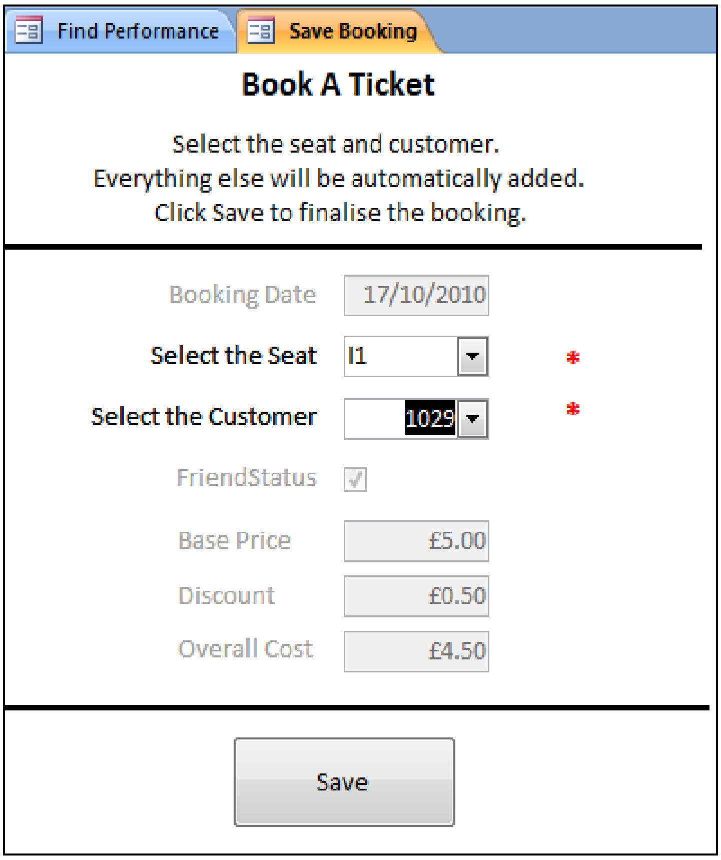


Book Tickets

Select the performance date and whether a matinee
is required and then click Next

Select Performance Date: 08/06/10

Matinee? ☑

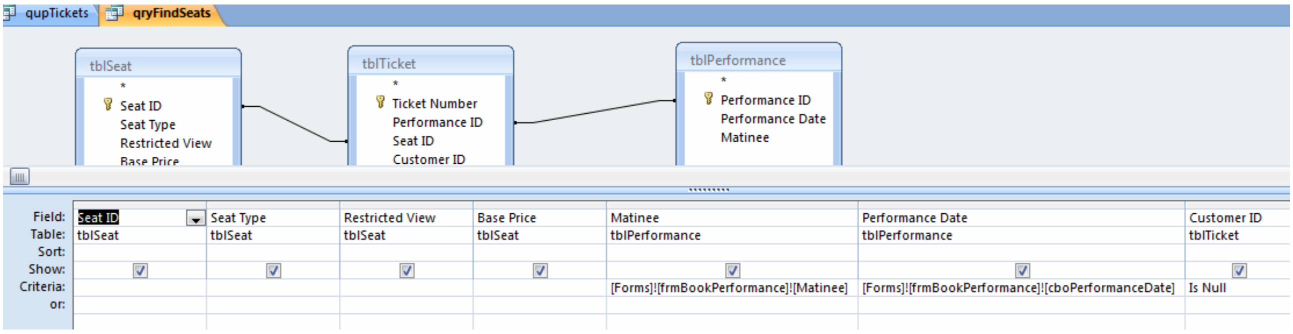


Book A Ticket

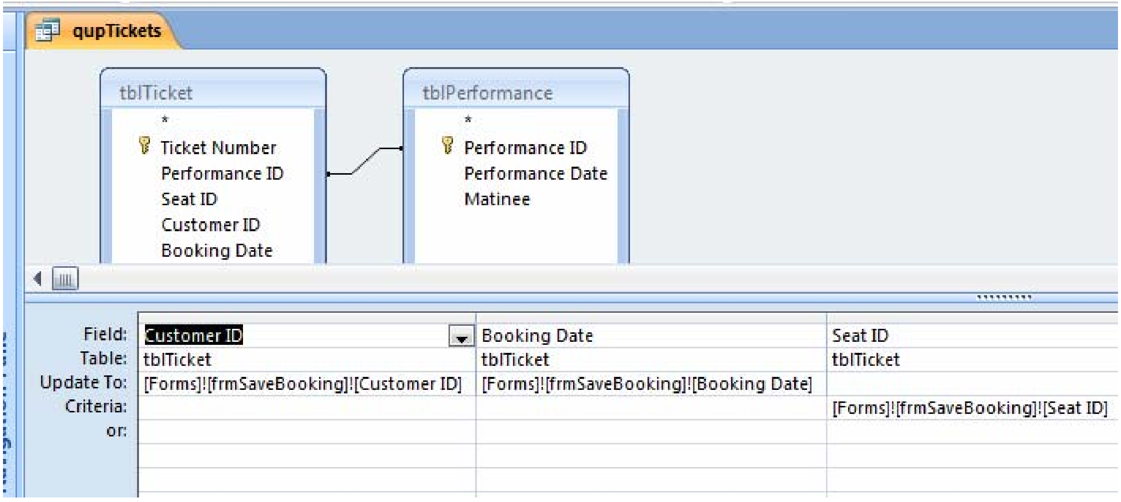
Select the seat and customer.
Everything else will be automatically added.
Click Save to finalise the booking.

Booking Date	17/10/2010
Select the Seat	I1 *
Select the Customer	1029 *
FriendStatus	☑
Base Price	£5.00
Discount	£0.50
Overall Cost	£4.50

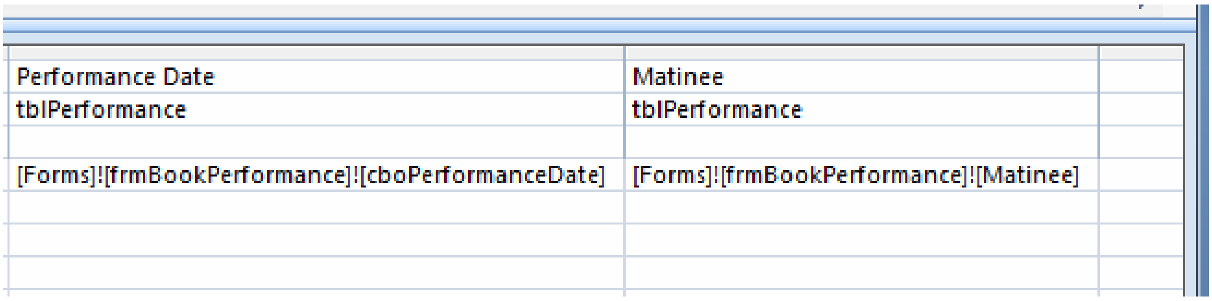
(iii)



Field:	Seat ID	Seat Type	Restricted View	Base Price	Matinee	Performance Date	Customer ID
Table:	tblSeat	tblSeat	tblSeat	tblSeat	tblPerformance	tblPerformance	tblTicket
Sort:							
Show:	☑	☑	☑	☑	☑	☑	☑
Criteria:					[Forms]![frmBookPerformance]![Matinee]	[Forms]![frmBookPerformance]![cboPerformanceDate]	Is Null
or:							



Field:	Customer ID	Booking Date	Seat ID
Table:	tblTicket	tblTicket	tblTicket
Update To:	[Forms]![frmSaveBooking]![Customer ID]	[Forms]![frmSaveBooking]![Booking Date]	
Criteria:			[Forms]![frmSaveBooking]![Seat ID]
or:			

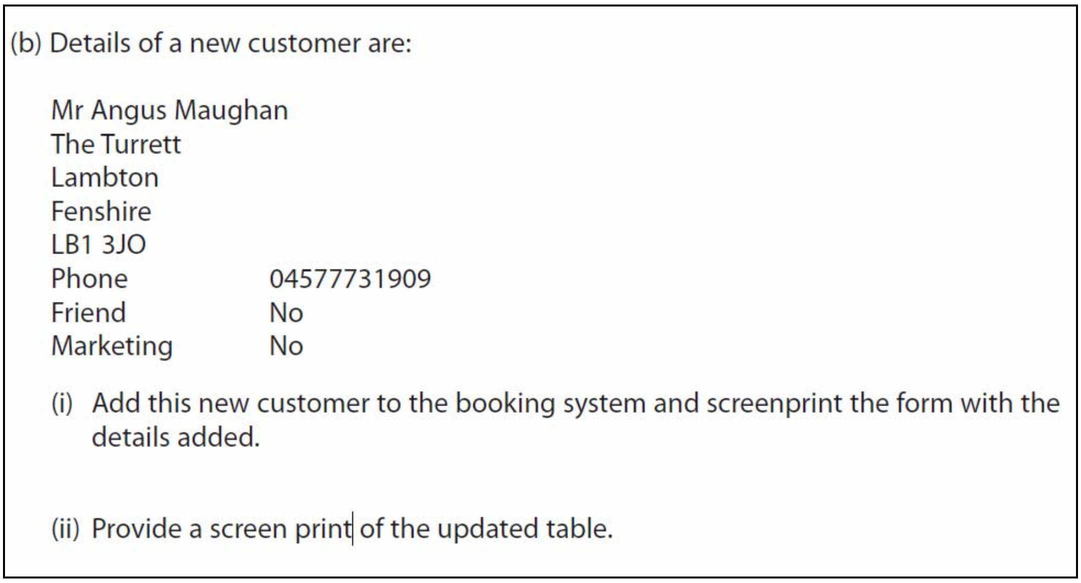


Performance Date	Matinee
tblPerformance	tblPerformance
[Forms]![frmBookPerformance]![cboPerformanceDate]	[Forms]![frmBookPerformance]![Matinee]

Notes

You will see that there are no annotations what-so-ever in answer provided. None are required for this part of the activity.

Part B



(b) Details of a new customer are:

Mr Angus Maughan
The Turrett
Lambton
Fenshire
LB1 3JO

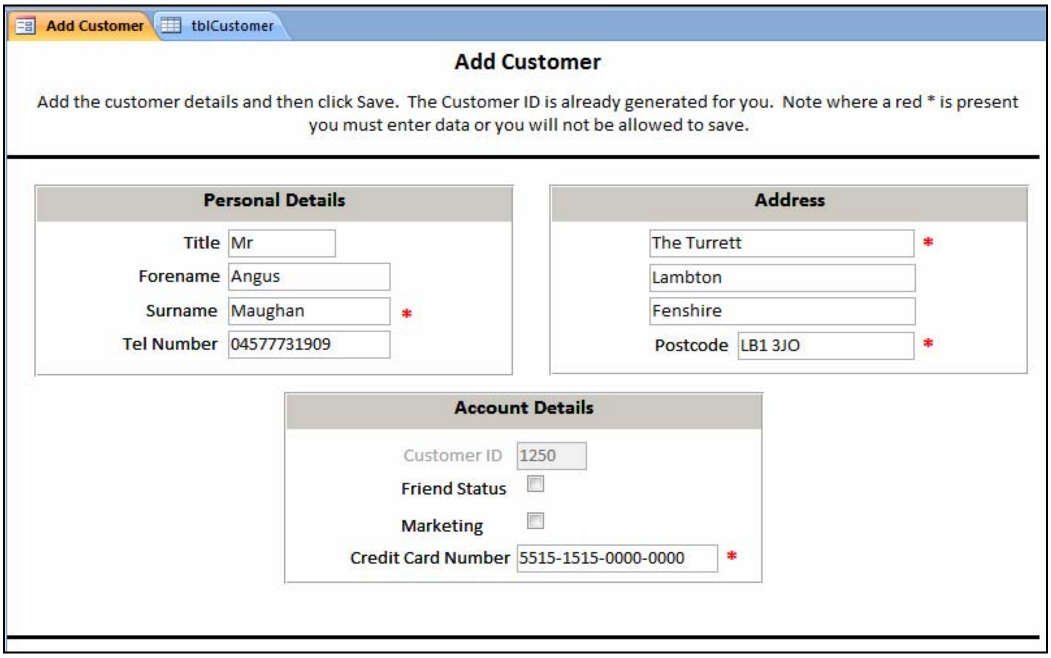
Phone	04577731909
Friend	No
Marketing	No

(i) Add this new customer to the booking system and screenprint the form with the details added.

(ii) Provide a screen print of the updated table.

An Example Answer

(i)



Add Customer

Add the customer details and then click Save. The Customer ID is already generated for you. Note where a red * is present you must enter data or you will not be allowed to save.

Personal Details

Title: Mr
Forename: Angus
Surname: Maughan *
Tel Number: 04577731909

Address

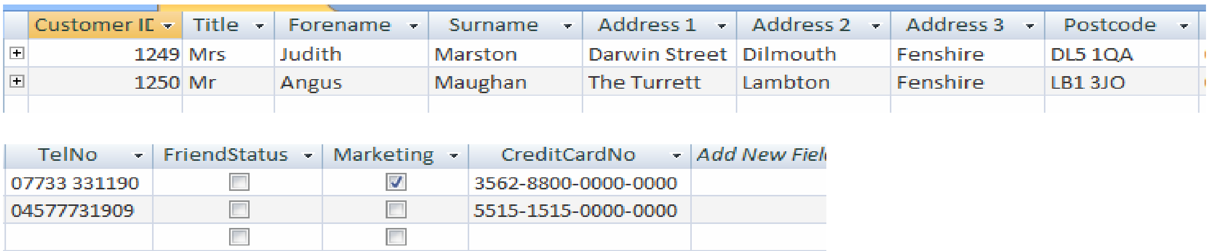
The Turrett *
Lambton
Fenshire
Postcode: LB1 3JO *

Account Details

Customer ID: 1250
Friend Status: □
Marketing: □
Credit Card Number: 5515-1515-0000-0000 *

I have added a credit card number to the test. The scenario did not say this had to be a required field but I have set it to a required field which is why I have added it.

(ii)



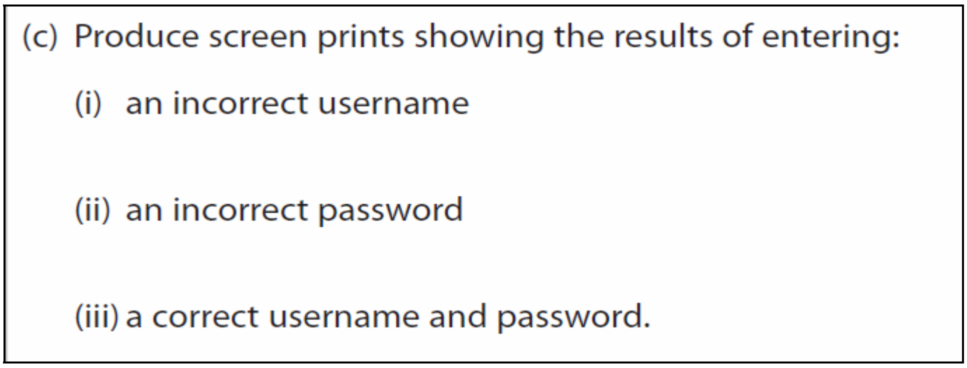
	Customer IC	Title	Forename	Surname	Address 1	Address 2	Address 3	Postcode
⊞	1249	Mrs	Judith	Marston	Darwin Street	Dilmouth	Fenshire	DL5 1QA
⊞	1250	Mr	Angus	Maughan	The Turrett	Lambton	Fenshire	LB1 3JO

TelNo	FriendStatus	Marketing	CreditCardNo	Add New Fiel
07733 331190	□	☑	3562-8800-0000-0000	
04577731909	□	□	5515-1515-0000-0000	
	□	□		

Notes

You will see that I chose to include a credit card number in the example given. The reason this is acceptable is because it was not a required field in the scenario therefore, it did not change what was being tested ie a valid record that should be saved into the database. If the credit card number had been specified as a required field in the scenario then I would not have included it as it would be trying to test an invalid record which should not have been saved into the database.

Part C



(c) Produce screen prints showing the results of entering:

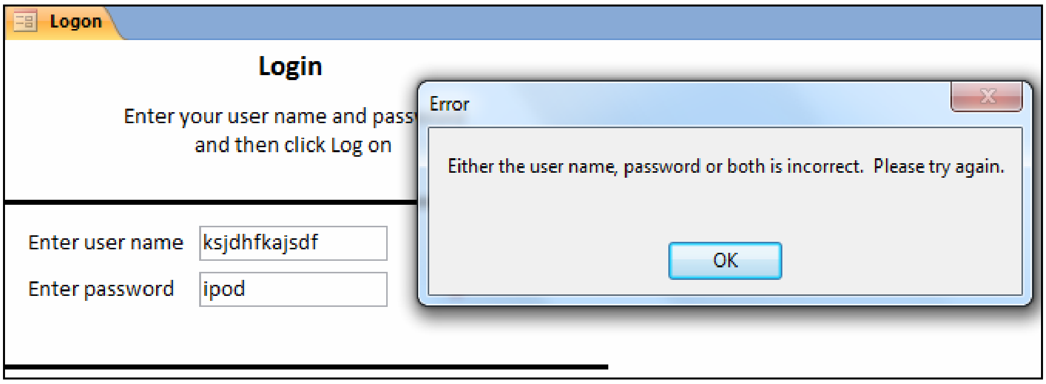
(i) an incorrect username

(ii) an incorrect password

(iii) a correct username and password.

An Example Answer

(i)



Login

Enter your user name and pass
and then click Log on

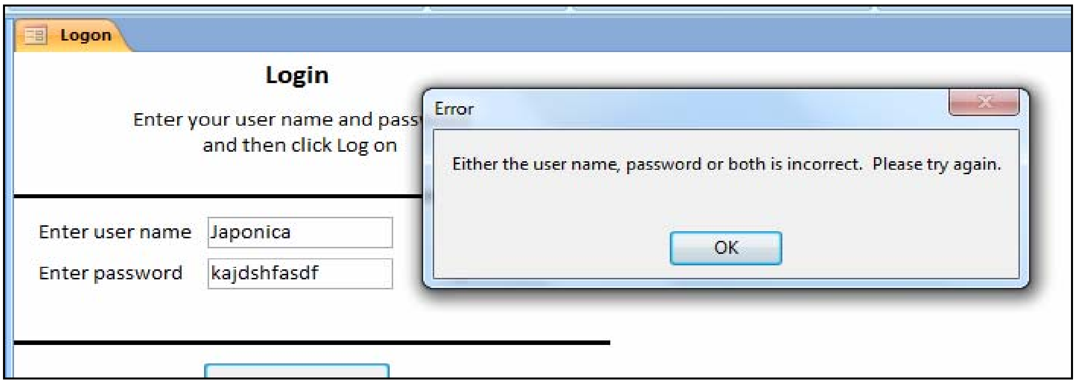
Enter user name: ksjdhfkajsdf

Enter password: ipod

Error

Either the user name, password or both is incorrect. Please try again.

(ii)



Login

Enter your user name and pass
and then click Log on

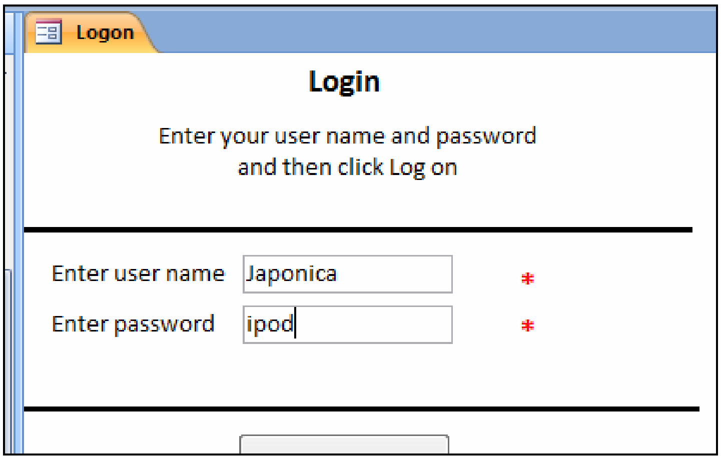
Enter user name: Japonica

Enter password: kajdshfasdf

Error

Either the user name, password or both is incorrect. Please try again.

(iii)

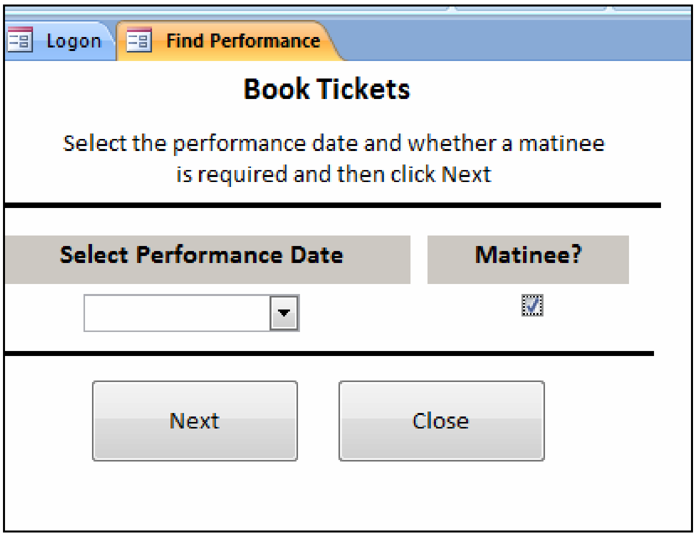


Login

Enter your user name and password
and then click Log on

Enter user name: Japonica *

Enter password: ipod *



Book Tickets

Select the performance date and whether a matinee
is required and then click Next

Select Performance Date	Matinee?
	☑

Notes

No annotations were included with this activity though the final screenshot could have been replaced with a written account of what happened when log on was successful as candidates were not specifically instructed to load a particular form.

Part D

(d) Create your own test data and enter it into the booking system.

On one sheet of A4 produce screen prints showing:

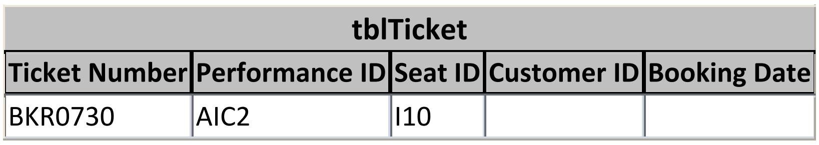
(i) an example of a successful booking

(ii) that the successful booking has been stored.

An Example Answer

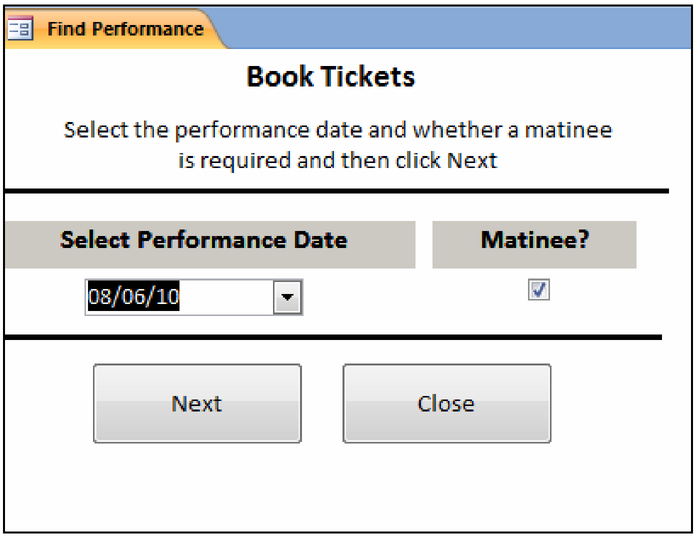
(i)

This is the ticket I am going to book as it is currently free. It is a performance on the 08/06/10 and is a matinee



tblTicket

Ticket Number	Performance ID	Seat ID	Customer ID	Booking Date
BKR0730	AIC2	I10		

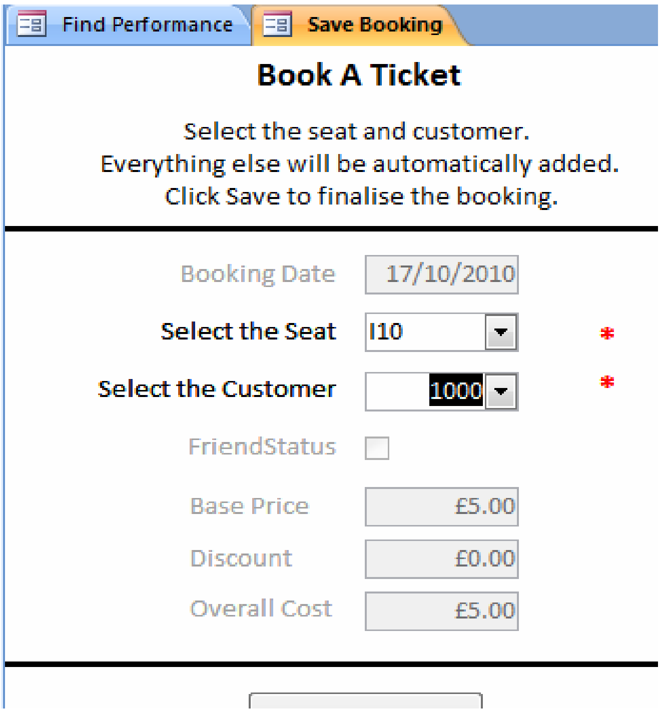


Book Tickets

Select the performance date and whether a matinee is required and then click Next

Select Performance Date: 08/06/10

Matinee? ☑

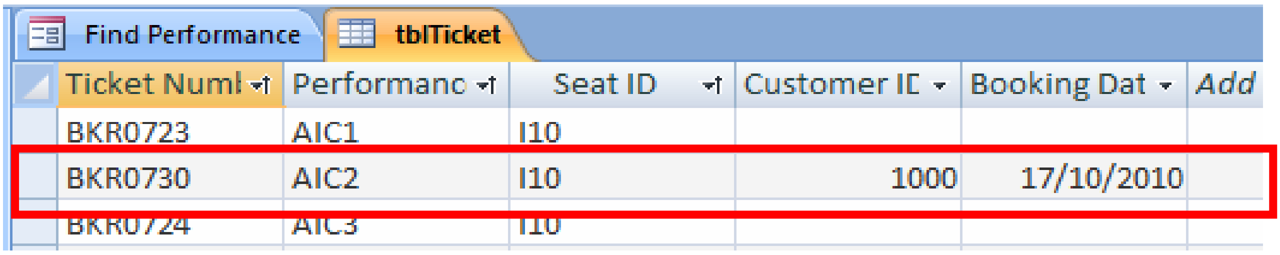


Book A Ticket

Select the seat and customer.
Everything else will be automatically added.
Click Save to finalise the booking.

Booking Date	17/10/2010
Select the Seat	I10 *
Select the Customer	1000 *
FriendStatus	□
Base Price	£5.00
Discount	£0.00
Overall Cost	£5.00

(ii)

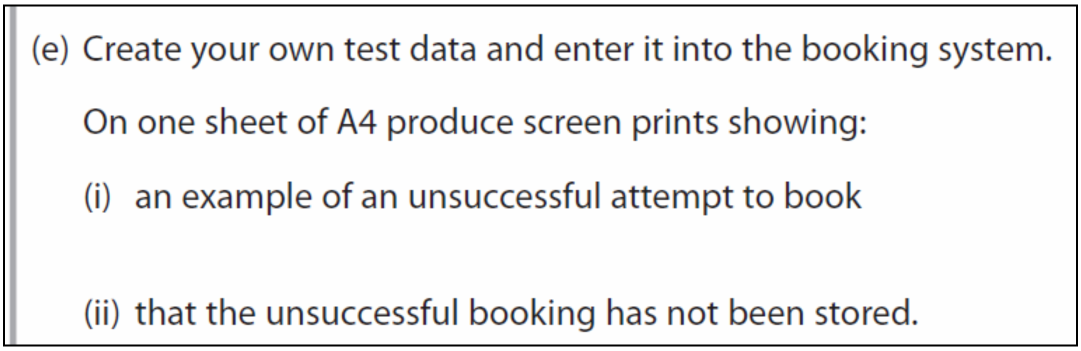


Ticket Numl	Performanc	Seat ID	Customer ID	Booking Dat	Add
BKR0723	AIC1	I10			
BKR0730	AIC2	I10	1000	17/10/2010	
BKR0724	AIC3	I10			

Notes

One small annotation was added to point out the purpose of the first screenshot.

Part E



(e) Create your own test data and enter it into the booking system.

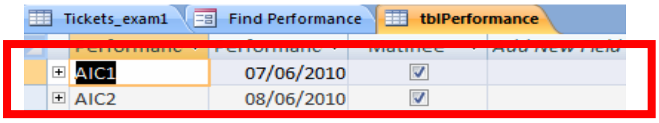
On one sheet of A4 produce screen prints showing:

(i) an example of an unsuccessful attempt to book

(ii) that the unsuccessful booking has not been stored.

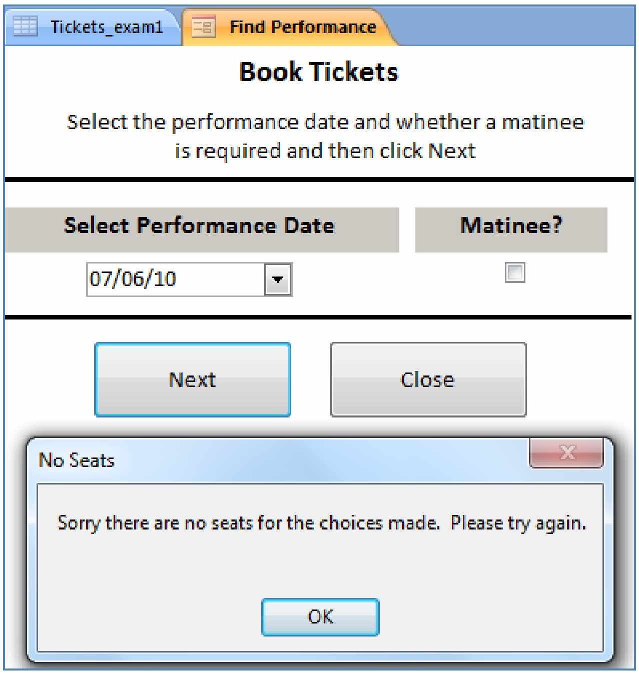
An Example Answer

(i)



Performanc	Performanc	Matinee	Add New Field
AIC1	07/06/2010	☑	
AIC2	08/06/2010	☑	

I am going to try and try to book a ticket for 07/06/10 and not for a matinee performance. As you can the performance is a matinee one so it should be unsuccessful.



Book Tickets

Select the performance date and whether a matinee is required and then click Next

Select Performance Date	Matinee?
07/06/10	□

No Seats

Sorry there are no seats for the choices made. Please try again.

The screenshot above shows that my database would not allow an unsuccessful booking. It did not open the save booking form so I could not proceed further.

Notes

Small annotations were added in this part of the activity to clarify what was being tested.

Candidates can go wrong in this activity in a number of places:

1. They do not enter the data as given
2. They do not provide clear screenshots
3. They turn what is clearly a test to ensure the database does not accept invalid data into valid data in order to make the record save.

Activity 5

Introduction

Activity 5 is designed to test the report aspect of 7.3 of the unit specification.

Booked tickets need to be printed as a database report.

Print the tickets purchased **on** 14 April (**ONLY**).

Each ticket should show:

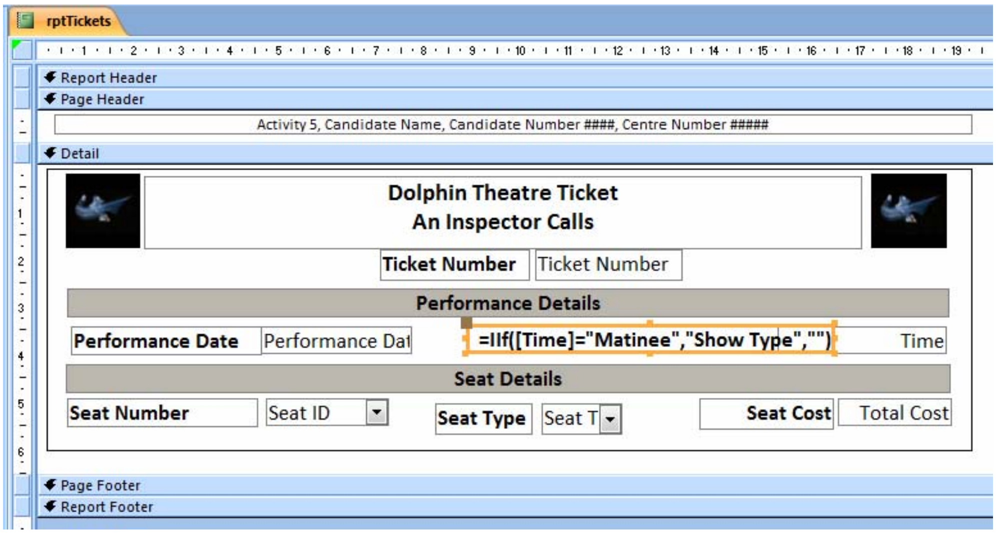
- the name of the production
- the date of the performance
- the word 'matinee' if it is for a matinee performance
- the seatID/number and seat type
- the ticketID/number
- the price paid.

The tickets should be printed **four** to a page of A4 (portrait).

Evidence to be submitted

❑ The printed database report and a screen print of the database report in **DESIGN** view.

An Example Answer



rptTickets

Report Header

Page Header

Activity 5, Candidate Name, Candidate Number ####, Centre Number #####

Detail

Dolphin Theatre Ticket
An Inspector Calls

Ticket Number | Ticket Number

Performance Details

Performance Date | Performance Dat | =IIf([Time]="Matinee","Show Type","") | Time

Seat Details

Seat Number | Seat ID | Seat Type | Seat T | Seat Cost | Total Cost

Page Footer

Report Footer

Notes

To view the report look at rptTicket in the database. No annotations were required in this activity.

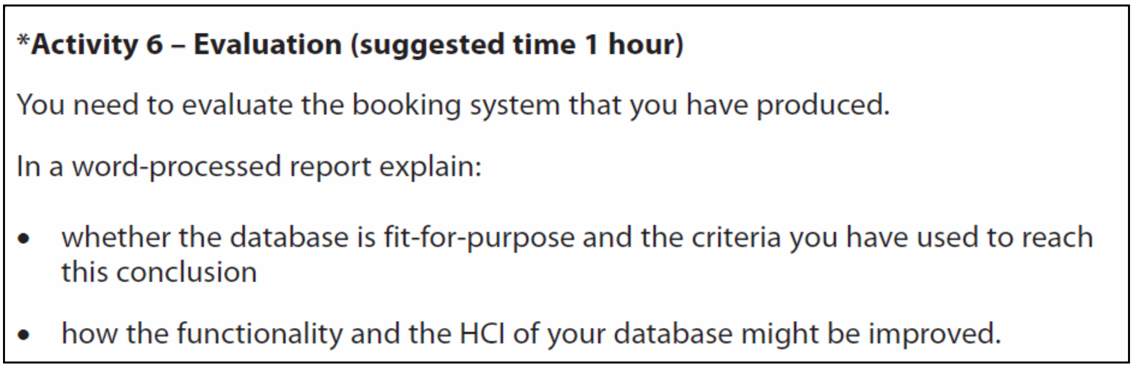
Candidates can go wrong in this activity in a number of places:

1. They do not put their candidate and centre details in the page header of the report so that the name only prints on the first page of the report
2. They do not put their candidates details on the report at all
3. They do not provide a design view screenshot.

Activity 6

Introduction

Activity 6 is designed to test the evaluation aspect of 7.8 of the unit specification.



***Activity 6 – Evaluation (suggested time 1 hour)**

You need to evaluate the booking system that you have produced.

In a word-processed report explain:

- whether the database is fit-for-purpose and the criteria you have used to reach this conclusion
- how the functionality and the HCI of your database might be improved.

An Example Answer

Evaluation against Requirements

The criteria I am going to use to judge whether my database is fit for purpose is the criteria I was given in the scenario by Maggie.

1. I think the security of the database is quite good and matches what Maggie asked for ie it does require and username and password in order to gain access. The username and password is exactly what Maggie asked for and access is not granted if either or both are incorrect. I think that this requirement has been met.

2. On the whole, I think the database is intuitive and simple to use and that Maggie will be happy with the HCI I have implemented. I have made sure all of the forms follow the same house style so that Maggie will easily become familiar with where particular objects should be eg save buttons etc. I have ensured that Maggie will know exactly what data is required by using red asterisks wherever fields cannot be left blank. This should mean that Maggie does not become frustrated if a save cannot take place. I have provided a good range of error messages that are user friendly so even if Maggie does make a mistake she will understand why and what to do to put it right. I think the database is efficient with regards to user input ie the only input is what could not be generated automatically or provided by a drop down box. Overall I think this requirement has been met very well.

3. The system does show which seats are available and I think it does this very well. The book performance form is easy to use only requiring the date of the performance to be selected and whether or not the customer requires a matinee performance. A query runs efficiently in the background taking its criteria from the book performance form. If there are no seats then Maggie

is not taken to the next form where customer details etc are added. I think this is good as Maggie may become frustrated and angry if she has to fill in that form only to find out at the end no seats are available.

4. The process for booking a seat is very straightforward. A maximum of three clicks on the find whether seats are available form and three on the next form where the booking actually takes place. This is a massive improvement on what it would have been like if I had not used drop down boxes or generated the overall cost of the ticket ie determined any discount automatically. So a maximum of six clicks and the seat is booked. I think that is efficient.

5. New customers can be added efficiently. The form is well laid out, grouping relevant information together to make it easier for Maggie to input. Required fields use the red asterisks and the CustomerID is automatically generated. The form does take into account the requirements of noting whether customers want their details passing on to the marketing department and whether they are a friend of the theatre or not. I have used Yes/No fields for both to make them more user friendly compared with 1 or 0 which may confuse Maggie. I think Maggie will be pleased with this aspect of the system.

6. I think the tickets looked professional incorporating the company logo. Maggie would have been able to split the tickets into four which would save a lot of paper in the long run.

Overall, all of Maggie's requirements have been met and, I think, very effectively. I am very happy with the system.

Improvements

Whilst I am happy with the system against the initial requirements it is clear to see that Maggie does not really understand the power of database systems. The system could be improved:

1. At this moment in time the database is for one particular theatre performance only ie 'An Inspector Calls'. It stands to reason that the theatre must offer other performances. Therefore, the database would be much more efficient if any number of performances could be stored, tickets booked etc.

2. Whilst the database does meet Maggie's requirements with regards to booking tickets it is not really as good as it should be. Only one seat can be booked at any one time which is not really efficient. The process would be much improved if more than one seat could be booked at a time.

3. At this moment in time there are no edit facilities eg you can add a new customer but cannot edit their details. This is something that would be a must for a more efficient system. Maggie should not have to go into the tables themselves which is what she would have to do now.

4. There are no facilities to delete data either. For example, if a customer cancels Maggie would have to physically go into the tables to amend the data or delete a record which is not really very good.

5. There is no main menu which means navigation between forms is very hard. It relies on Maggie knowing the names of the forms which is not really very good. A main menu would have aided navigation immensely.

Notes

The evaluation is split into two distinct parts. Candidates should be revisiting the scenario for the first part of the evaluation. They should be looking at what was asked of them and whether the system they have built includes it. It has to be evaluative ie has the requirement been met and to what degree. A running commentary of what they have done to build the system is not suitable for the higher mark bands.

The second part of the evaluation requires them to think of improvements. The databases that they are asked to create are always limited in some way or form. They should be thinking of how the functionality could be extended.