

**Coursework Task**

**Higher Information Systems**

*Valid for session 2005/2006 only*

## **Coursework Task: Triathlon**

### **Higher Information Systems Coursework Task 2005-2006**

#### **Background**

A triathlon is a competition in which competitors complete 3 contests in swimming, cycling and running. The contests take place consecutively without a break, and the winner of a triathlon is the competitor who completes all three contests in the shortest time.

Every year, competitors take part in a series of five triathlon events across Scotland, known as the Scottish Triathlon Championship. Competitors in each triathlon gain points equivalent to their position in the event, and the male and female competitors with the lowest number of points at the end of the series are the overall champions.

Some of the athletes also compete as a team of two men and two women, so that competitor positions in any triathlon also count towards the prestigious Team Challenge Cup. The team which has the lowest total positions at the end of the series wins the cup.

The governing body for triathlons in Scotland is TriScot, who have a website and produce a monthly newsletter for athletes. The website and the newsletter publish general information about triathlons as well as details of forthcoming events and a results service. This will include statistics on races, competitors, teams and articles on training techniques, the history of the event and so on.

## Part 2—Database Systems

In order to produce the latest Triathlon championship table and appropriate statistics on competitors and teams, TriScot uses a relational database. This database stores information about competitors, teams, events and results.

The database stores the results of the five triathlon events in the Scottish Triathlon Championship. Each triathlon is given a unique Event Number. The database records the competitors position and time taken in each triathlon.

At the start of each season, competitors are given a unique number for all triathlons in the season. There are two types of championship: individual and team. All competitors compete for the individual championship, but some are also members of a team, and their results are also included in the team championship. Each team comprises four competitors, two male and two female. For the 2006 season, there are 12 teams, as follows:

Team No	Team Name
1	(no team)
2	Lothian Flyers
3	Glasgow Kestrels
4	West Highland Terriers
5	Border Collies
6	Fife Finches
7	Tayside Tempests
8	Viking Marauders
9	Lanarkshire Lynchers
10	Grampian Grouchos
11	Forth Flitters
12	Campsie Conquerors

Team 1, called “(no team)”, is a special “team” which is used for all competitors who are not members of a team.

Part 2 – Section 1

The normalised data model for recording the triathlon results has been produced, as follows. (Primary keys are shown in **bold and underlined**, while asterisks (\*) indicate foreign keys.)

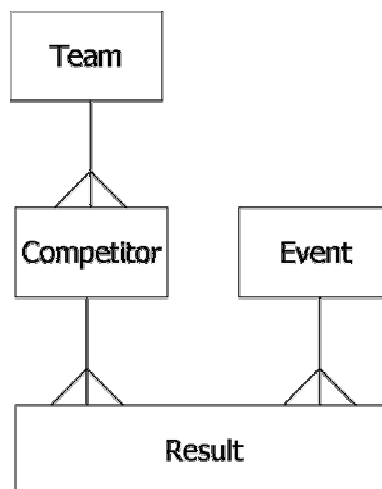
Team(**Team No**, Team Name)

Competitor(**Competitor No**, Forename, Surname, Town, Contact No, Sex, \*Team No)

Event(**Event No**, Event Name, Date)

Result(\***Event No**, \***Competitor No**, Time Taken, Position)

The entity-relationship diagram on the right represents the relationships between the entities.



A data dictionary has been produced prior to implementation in a relational database. Part of this is shown below.

Entity	Attribute	PK	FK	Data Type	Unique	Required	Validation	Format
Competitor	Competitor No	Yes	No	Text(3)	Yes	Yes		AAA
	Forename	No	No	Text(15)	No	Yes		
	Surname	No	No	Text(20)	No	Yes		
	Town	No	No	Text(20)	No	No		
	Contact No	No	No	Text(14)	No	Yes		(99999) 999999
	Sex	No	No	Text(1)	No	Yes	M/F	
	Team No	No	Yes	Integer	No	Yes	Lookup value from Team table	
Team	Team No	Yes	No	Integer	Yes	Yes		
	Team Name	No	No	Text(30)	Yes	Yes		
Event	Event No	Yes	No	Integer	Yes	Yes		
	Event Name	No	No	Text(30)	Yes	Yes		
	Event Date	No	No	Date	Yes	Yes		dd/mm/yyyy
Result	Event No							
	Competitor No							
	Time Taken							
	Position							

For text attributes, the number in brackets indicates the size of the attribute (number of characters). In the Format column, the symbols -

- ‘9’ and ‘A’ represent a single digit and letter in a text field
- ‘d’, ‘m’ and ‘y’ indicate digits used to represent the day, month and year in a date field

All times are recorded as follows:

0:43:21 represents 43 minutes and 21 seconds

1:01:08 represents 1 hour, 1 minute and 8 seconds.

<b>Task</b>	<b>Evidence Required</b>
6.	Complete the Data Dictionary for the Result entity.
	Completed Data Dictionary

**You must now submit your answer to Task 6 to your teacher/lecturer.**

Part 2 – Section 2

Before attempting the tasks in this section ask your teacher/ lecturer for a completed Data Dictionary. This Data Dictionary should be used to complete tasks 7–11. You will also need access to pre-prepared database tables as described in task 7.

<b>Task</b>		<b>Evidence Required</b>																										
7.	A database has been set up containing the Event, Competitor and	Result tables.																										
	(a) Set up the table structure for the Team table. Ensure that your database structure corresponds to the completed Data Dictionary as supplied.																											
	(b) Implement all the validation checks shown in the completed Data Dictionary.	Hardcopy (printout or screenshots) of table structure including field types, validation checks etc																										
	(c) Set up links between the tables, as required.	Hardcopy (printout or screenshots) of table structure showing relationships																										
	(d) Populate the Team table with the data shown below and produce a hard copy.	Printout of Team table																										
	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Team No</th> <th>Team Name</th> </tr> </thead> <tbody> <tr><td>1</td><td>(no team)</td></tr> <tr><td>2</td><td>Lothian Flyers</td></tr> <tr><td>3</td><td>Glasgow Kestrels</td></tr> <tr><td>4</td><td>West Highland Terriers</td></tr> <tr><td>5</td><td>Border Collies</td></tr> <tr><td>6</td><td>Fife Finches</td></tr> <tr><td>7</td><td>Tayside Tempests</td></tr> <tr><td>8</td><td>Viking Marauders</td></tr> <tr><td>9</td><td>Lanarkshire Lynchers</td></tr> <tr><td>10</td><td>Grampian Grouchos</td></tr> <tr><td>11</td><td>Forth Flitters</td></tr> <tr><td>12</td><td>Campsie Conquerors</td></tr> </tbody> </table>	Team No	Team Name	1	(no team)	2	Lothian Flyers	3	Glasgow Kestrels	4	West Highland Terriers	5	Border Collies	6	Fife Finches	7	Tayside Tempests	8	Viking Marauders	9	Lanarkshire Lynchers	10	Grampian Grouchos	11	Forth Flitters	12	Campsie Conquerors	
Team No	Team Name																											
1	(no team)																											
2	Lothian Flyers																											
3	Glasgow Kestrels																											
4	West Highland Terriers																											
5	Border Collies																											
6	Fife Finches																											
7	Tayside Tempests																											
8	Viking Marauders																											
9	Lanarkshire Lynchers																											
10	Grampian Grouchos																											
11	Forth Flitters																											
12	Campsie Conquerors																											
	(e) Explain how validation checks set in the Competitor table reduce the scope for error when entering data.	Explanation of validation checks																										
8.	(a) Create a layout/form to display competitor information, one competitor at a time.	Printout of layout/form and script/macro																										
	(b) Add your own navigational features to your layout/form in (a), that make use of macros and scripts, to move to the next or previous competitor.																											
9.	(a) By querying your database, locate and print all the triathlon Results for competitors who are not part of a team. These should be listed in chronological order of event, with the most recent listed last. Include the Competitor No, Forename, Surname, Name of Event, Time Taken, Position and Date of Event.	Create a report that shows the results of the query as described in part (a). The report should also show relevant data for parts (a) and (b).																										
	(b) Modify your report/layout in part (a) to show the total number of competitors who are not part of a team.																											

10.	<p>To find the winning team, the positions for each member of a team are totalled and the team with the lowest total is the winner.</p> <p>Display the total Position for each <b>team</b>, arranged with the lowest total first. You must exclude the result for “no team”.</p>	<p>Printout of total position for each team (excluding “no team”)</p>
-----	--	---

In the TriScot monthly newsletter, the results for a team are displayed as shown below:

### *Forth Flitters*

<i>Lothian Triathlon</i>		<i>14/05/2005</i>	<i>Position</i>
Louise	Brown		3
Keith	Farreau		9
George	Duffy		15
Gail	Brown		16
<i>Skye Triathlon</i>		<i>28/05/2005</i>	<i>Position</i>
Louise	Brown		1
George	Duffy		2
Gail	Brown		4
Keith	Farreau		19
<i>Granite City Triathlon</i>		<i>12/06/2005</i>	<i>Position</i>
Keith	Farreau		5
Gail	Brown		14
George	Duffy		14
Louise	Brown		24
<i>Strathspey Triathlon</i>		<i>25/06/2005</i>	<i>Position</i>
George	Duffy		2
Gail	Brown		5
Keith	Farreau		5
Louise	Brown		16

The information is arranged in chronological order of Event, with the most recent last, and with competitors listed for each event in order of Position, with the highest first.

<b>Task</b>		<b>Evidence Required</b>
11.	Use your database to create and print a layout or report showing the information for the Border Collies team.	Printout of Team information

**Total marks for Part 2: 32**

# **Appendix 1**

## **Completed Data Dictionary**

**(N.B. Only to be handed out after completion of Task 6)**

Entity	Attribute	PK	FK	Data Type	Unique	Required	Validation	Format
Competitor	Competitor No	Yes	No	Text(3)	Yes	Yes		AAA
	Forename	No	No	Text(15)	No	Yes		
	Surname	No	No	Text(20)	No	Yes		
	Town	No	No	Text(20)	No	No		
	Contact No	No	No	Text(14)	No	Yes		(99999) 999999
	Sex	No	No	Text(1)	No	Yes	M/F	
	Team No	No	Yes	Integer	No	Yes	Lookup value from Team table	
Team	Team No	Yes	No	Integer	Yes	Yes		
	Team Name	No	No	Text(30)	Yes	Yes		
Event	Event No	Yes	No	Integer	Yes	Yes		
	Event Name	No	No	Text(30)	Yes	Yes		
	Event Date	No	No	Date	Yes	Yes		dd/mm/yyyy

Result	Event No	Yes	Yes	Integer	No	Yes	Lookup value from Event table	
	Competitor No	Yes	Yes	Text(3)	No	Yes	Lookup value from Competitor table	
	Time Taken	No	No	Time	No	No		h:mm:ss
	Position	No	No	Integer	No	No		

Notes: Format of 'Time Taken' should be indicated but may be slightly different from that stated. Equivalent formats should be accepted.

**Coursework Task**

**Higher Information Systems**

*Valid for session 2006/2007 only*

# Coursework Task: Online Shopping

## Higher Information Systems Coursework Task 2006-2007

### Background

Articulate Clothing Limited is a wholesale supplier of mens and womens clothing to business customers. The company operates using e-commerce with a web-site on which customers can view products, and place and track orders.

A relational database is used to store details of customers, items in stock, and customer orders.

### Part I—Database Systems

The normalised data model for recording customer orders has been produced, as follows. (Primary keys are shown in **bold and underlined**, while asterisks (\*) indicate foreign keys.)

Customer(**Customer No**, Forename, Surname, Address1, Address2, Address3, Postcode, Telephone No)

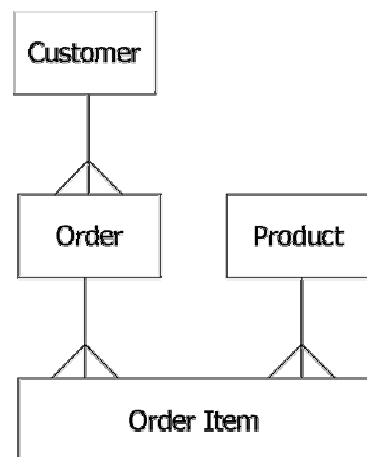
Order(**Order No**, Order Date, Dispatch Date, \*Customer No)

Product(**Product Code**, Product Name, Unit Cost)

Order Item(\***Order No**, \***Product Code**, **Size**, Quantity, )

The entity-relationship diagram on the right represents the relationships between the entities.

A data dictionary has been produced prior to implementation in a relational database. Part of this is shown overleaf.



Entity	Attribute	PK/ FK	Data Type	Uni que	Requ ired	Validation	Format
Customer	Customer No	PK	Integer	Yes	Yes	>=1000, <=9999	
	Forename		Text(15)	No	Yes		
	Surname		Text(20)	No	Yes		
	Address1		Text(20)	No	Yes		
	Address2		Text(20)	No	Yes		
	Address3		Text(20)	No	No		
	Postcode		Text(8)	No	Yes		
	Telephone No		Text(11)	No	Yes		(99999) 999999
Order	Order No	PK	Integer	Yes	Yes	>=10000, <=99999	
	Order Date		Date	No	Yes		dd/mm/yyyy
	Dispatch Date		Date	No	No		dd/mm/yyyy
	Customer No	FK	Integer	No	Yes	Lookup from Customer Table	
Product	Product Code	PK	Text(5)	Yes	Yes		AA999
	Product Name		Text(40)	No	Yes		
	Unit Cost		Integer	No	Yes	>=0, <=79	

Order Item	Order No	PK/ FK					
	Product Code	PK/ FK					
	Size	PK					
	Quantity						

For text attributes, the number in brackets indicates the size of the attribute (number of characters). In the Format column, the symbols

- 9 and A represent a single digit and letter in a text field
- d, m and y indicate digits used to represent the day, month and year in a date field

Each item for sale is given an alphanumeric reference code of the form AB123. Customers are restricted to 20 of any single product per order. All prices are in whole pounds, with the most expensive product in the current range priced at £79. Some items are offered for free during the season if there is excess stock to clear. Product names are no larger than 40 alphanumeric characters. Garment sizes are one of the following, depending on the type of garment:

XS, S, M, L, XL, XXL  
6, 8, 10, 12, 14, 16, 18  
26, 28, 32, 34, 36, 38, 40, 42, 44, 46, 48

**Part 1 – Section 1**

<b>Task</b>	<b>Evidence Required</b>
<b>1</b> Complete the data dictionary for the <b>Order Item</b> entity.	Completed data dictionary

**You must now submit your answer to Task 1 to your teacher/lecturer.**

## Part 1 – Section 2

Before attempting the tasks in this section ask your teacher/lecturer for a completed data dictionary. This data dictionary should be used to complete tasks 2 – 5. You will also need access to pre-prepared database tables as described in task 2.

Task	Evidence Required
<b>2</b> A database has been set up containing the Customer, Product and	Order Item tables.
(a) Set up the table structure for the <b>Order</b> table. Ensure that your database structure corresponds to the completed data dictionary that you have been issued with.	Hardcopy (printout or screenshots) of table structure including field types, validation checks etc.
(b) Implement all the validation checks shown in the data dictionary for <b>all</b> tables.	Printout of Order table
(c) Populate the Order table with the data shown overleaf and produce a hardcopy.	Hardcopy (printout or screenshots) of relationships between tables.
(d) Set up links between the tables, as required.	Written/typed explanation
(e) Explain how the database has been configured to eliminate the scope for error when entering the Customer No data into the Order table.	

### Data for Task 2(c)

There are 26 orders:

Order No	Order Date	Dispatch Date	Customer No
21142	13/04/2006	16/04/2006	2789
21143	20/04/2006	21/04/2006	3618
21144	20/04/2006	25/04/2006	5048
21145	22/04/2006	27/04/2006	4800
21146	23/04/2006	23/04/2006	9463
21147	23/04/2006	29/04/2006	1084
21148	25/04/2006	29/04/2006	6639
21149	27/04/2006	03/05/2006	2789
21150	28/04/2006	01/05/2006	5048
21151	29/04/2006	30/04/2006	6707
21152	04/07/2006	04/07/2006	5048
21153	04/07/2006	06/07/2006	9463
21154	15/07/2006	16/07/2006	2789
21155	15/07/2006	20/07/2006	5048
21156	21/07/2006	21/07/2006	8446
21157	22/07/2006	24/07/2006	2283
21158	23/07/2006	27/07/2006	9463
21159	26/07/2006	31/07/2006	7785
21160	28/07/2006	29/07/2006	6730
21161	30/07/2006	04/08/2006	9541
21162	11/10/2006	17/10/2006	2325
21163	14/10/2006	18/10/2006	9463
21164	14/10/2006	18/10/2006	9811
21165	15/10/2006	21/10/2006	5048
21166	18/10/2006	21/10/2006	8309
21167	28/10/2006	31/10/2006	2789

The Sales Director for Articulate Clothing requires summary information about quarterly sales (sales for 3 months of the year). The summary sales figures for the 3<sup>rd</sup> quarter of 2006 are shown below:

<b>Articulate Clothing</b>			
<b>Quarterly Sales Summary: 3rd Quarter 2006</b>			
<b>Product Code</b>	<b>Product Name</b>	<b>Total Quantity Sold</b>	<b>Total Orders Placed</b>
AW353	Short Sleeve Active Wear V-neck T-shirt	7	2
BC252	Bias Check No-Iron	20	2
BP536	Broadcloth Patterned	9	1
CH277	Relaxed Custom Hemmed	6	1
CJ716	Corduroy Jean	16	1
CL827	Cotton Liberty Print	5	1
.....	.....	...	...
.....	.....	...	...

You are required to produce summary sales figures for the Second Quarter in 2006 (1st April – 30th June inclusive).

<b>Task</b>		<b>Evidence Required</b>
<b>3</b>	By using summary fields or otherwise, calculate the total quantity of each product sold and the number of orders placed for each product during the Second Quarter of 2006 (1st April – 30th June inclusive). Display the details in a report/layout similar to the one shown above.	Printout of report / layout

The Marketing Director of the company would like to be able to browse through the product details one product at a time rather than viewing the details in a single table. She suggests a layout similar to the one shown below.

<b>Browse Product Details</b>	
▶ Product Code	<input type="text" value="PT635"/>
Product Name	<input type="text" value="Performance Super-T"/>
Unit Cost	<input type="text" value="12"/>
<input type="button" value="Cheapest Product"/> <input type="button" value="Dearest Product"/>	
Record: 2	

You are required to produce an appropriate screen layout for the Marketing Director.

<b>Task</b>		<b>Evidence Required</b>
<b>4</b>	(a) Produce a layout/form to display details of each product, one product at a time. Details of the cheapest product should be shown first (products of equal price should be shown in order of product code).	Printout of form / layout
	(b) Add your own navigational features to the layout/form produced in (a). These navigational features should make use of macros/scripts to move to the cheapest and dearest product.	Printout of macro / script used to control navigational features

When goods are delivered to customers, they are sent with an invoice that lists the goods ordered and total cost as shown in the example below:

<h1>Invoice</h1>			
<b>Customer No</b>	5284		
<b>Order No:</b>	21168		
<b>Items Ordered:</b>			
<b>Qty</b>	<b>Description</b>	<b>Unit Cost</b>	<b>Sub-Total</b>
2	Short Sleeve Active Wear V-neck T-shirt	£65.00	£130.00
1	Long Sleeve Linen Pattern	£59.00	£59.00
		<b>Total:</b>	<b>£189.00</b>

<b>Task</b>	<b>Evidence Required</b>
<b>5</b> Use the database to perform the following tasks.	
(a) Calculate and display the Sub-Total for each item in order <b>21153</b> . (N.B Not the order shown above)	Printout of report / layout
(b) By using a summary field with your answer to (a), or otherwise, calculate and display the Total for order 21153.	
(c) Using your answer to (b), or otherwise, create an invoice for order 21153 similar to the one shown above.	

**Total marks for Part I: 32**

# **Appendix 1**

## **Completed Data Dictionary**

**(N.B. Only to be handed out after completion of Task 1)**

Entity	Attribute	PK/FK	Data Type	Unique	Required	Validation	Format
Customer	Customer No	PK	Integer	Yes	Yes	>=1000, <=9999	
	Forename		Text(15)	No	Yes		
	Surname		Text(20)	No	Yes		
	Address 1		Text(20)	No	Yes		
	Address 2		Text(20)	No	Yes		
	Address 3		Text(20)	No	No		
	Postcode		Text (8)	No	Yes		
	Telephone No		Text (11)	No	Yes		(99999) 999999
Order	Order No	PK	Integer	Yes	Yes	>=10000, <=99999	
	Order Date		Date	No	Yes		dd/mm/yyyy
	Dispatch Date		Date	No	No		dd/mm/yyyy
	Customer No	FK	Integer	No	Yes	Lookup from Customer Table	
Product	Product Code	PK	Text (5)	Yes	Yes		AA999
	Product Name		Text(40)	No	Yes		
	Unit Cost		Integer	No	Yes	>=0, <=79	

Order Item	Order No	PK/ FK	Integer	No	Yes	Lookup from Order table	
	Product Code	PK/ FK	Text(5)	No	Yes	Lookup from Product table	
	Size	PK	Text(3)	No	Yes	Restricted choice	
	Quantity		Integer	No	Yes	>=1, <=20	

**Coursework Task**

**Higher Information Systems**

*Valid for session 2007/2008 only*

**Please read Section I carefully before issuing materials to candidates.**

Publication code: BB3418

# Coursework Task: Camp Ossipey Summer Camp

## Higher Information Systems Coursework Task 2007-2008

### Background

Camp Ossipey is a summer activities camp for children aged 6–16 years old. The camp operates for 7 weeks a year during July and August and is split into 3 sessions as shown.

Session	Start Date	End Date	Cost
Session 1	22/06/08	05/07/08	£850
Session 2	06/07/08	19/07/08	£850
Session 3	20/07/08	09/08/08	£1150

During each session the camp operates a two-day timetable which repeats for Monday through to Saturday. Sunday is a rest day. The timetable is divided into ten periods, periods 1 to 5 on day 1 and periods 6 to 10 on day 2 of the alternating system.

A sample timetable is shown below.

<b>Camper ID</b>	0022	<b>Name</b>	Aileen Tait
<b>Session</b>	1		
<b>Period</b>	<b>Activity</b>	<b>Period</b>	<b>Activity</b>
1	PowerBoating	6	In-Line Hockey
2	Lifeguard Course	7	Ropes Course
3	Martial Arts	8	Mountain Biking
4	Golf Clinic	9	Go Karts
5	Windsurfing	10	Riflery

Campers choose different activities for each of the ten periods.

Campers can attend more than one Session but must create a new timetable for each session.

The manager of the Camp has decided to implement an online booking system. Using the booking system campers can create a timetable for the session(s) they wish to attend. A different timetable can be created for each session. To create their timetable the campers enter their unique ID, the session number, and their choice of activities for each of the 10 periods of the two-day cycle.

Some activities are covered in the standard cost of their booking while others require an additional payment.

A relational database has been set up to store details of campers, activities, timetables and instructors.

## Part I—Database Systems

The normalised data model for the on-line booking system has been produced, as follows. (Primary keys are shown in **bold and underlined**, while asterisks (\*) indicate foreign keys.)

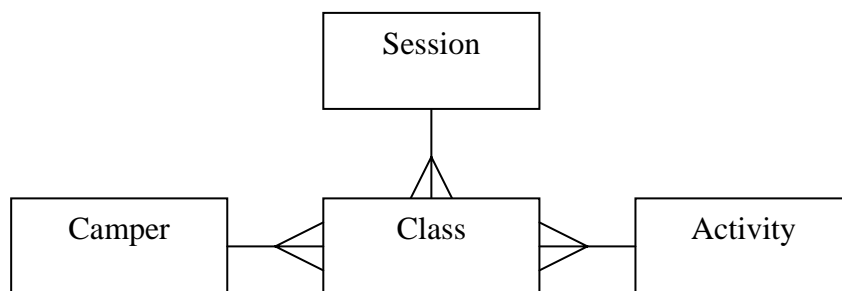
Camper (**Camper ID**, Forename, Surname, Address1, Address2, Address3, Postcode, DOB)

Activity (**Activity**, Location, Additional Cost)

Session (**Session No.**, Start Date, End Date, Session Cost)

Class (**Camper ID\***, **Session No.\***, **Period**, Activity\*, )

The entity-relationship diagram for the system is shown below.



A data dictionary has been produced prior to implementation in a relational database. Part of this is shown overleaf.

Entity	Attribute	PK/ FK	Data Type/ size	Uni que	Req uire d	Validation	Format
Camper	Camper ID	PK	Number	Yes	Yes	>=0001, <=9999	0000
	Forename		Text(15)	No	Yes		
	Surname		Text(30)	No	Yes		
	Address1		Text(30)	No	Yes		
	Address2		Text(30)	No	No		
	Address3		Text(30)	No	Yes		
	Postcode		Text(8)	No	Yes		
	DOB		Date	No	Yes		dd/mm/yyyy
Activity	Activity	PK	Text(20)	Yes	Yes		
	Location		Text(20)	No	Yes		
	Additional Cost		Number	No	No	<100	currency
Session	Session No	PK	Number	Yes	Yes	Restricted 1, 2, 3	
	Start Date		Date	Yes	Yes		dd/mm/yyyy
	End Date		Date	Yes	Yes		dd/mm/yyyy
	Session Cost		Number	No	Yes	Restricted 850, 1150	currency
Class	Camper ID						
	Session No						
	Period						
	Activity						

For text attributes, the number in brackets indicates the size of the attribute (number of characters).

In the Format column, the symbols

- 0 represents a single digit
- d, m and y indicate digits used to represent the day, month and year in a date field

Each camper is given a unique ID number of the form 0000.

Campers must choose an activity for each of the 10 periods.

Periods are restricted to values 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

**Part I – Section 1**

<b>Task</b>	<b>Evidence Required</b>
<b>1</b> Complete the data dictionary for the <b>Class</b> entity.	Completed data dictionary

**You must now submit your answer to Task 1 to your teacher/lecturer.**

## Part I – Section 2

Before attempting the tasks in this section ask your teacher/lecturer for a completed data dictionary.

You will also need access to pre-prepared database tables.

You **MUST** complete tasks in the sequence indicated.

Task	Evidence Required
<b>2</b> A database has been set up containing the Camper, Activity, Session and Class tables. (a) Edit the structure of each table to ensure it matches the data dictionary provided. You should pay particular attention to; <ul style="list-style-type: none"><li>• data type</li><li>• unique</li><li>• required</li><li>• validation</li><li>• format</li><li>• primary and foreign keys</li></ul>	Hardcopy (printout or screenshots) of table structure including field types, validation checks etc.
(b) Set up/edit the relationships between all the tables as indicated by the entity-relationship diagram.	Hardcopy (printout or screenshots) of relationships between tables.

**You must now submit all hardcopies for Task 2 to your teacher/lecturer.**

### Part I – Section 3

Ask your teacher/lecturer for the complete database. This database must be used to carry out Tasks 3-5.

The manager of the Camp wants to monitor how often the premium activities (ie those activities that carry an additional cost) are selected by campers. The breakdown of premium activities for Session 1 are shown below.

---

## Session 1 Premium Activities

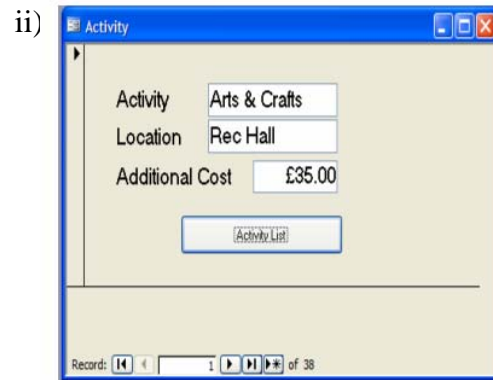
Activity	Number of Selections	Activity Revenue
Kayaking	16	£560.00
Nature Hiking	13	£156.00
Golf Clinic	10	£220.00
Horseback Riding	10	£950.00
Power Boating	10	£350.00
Go Karts	9	£450.00
Lifeguard Course	7	£266.00
Fishing	6	£300.00
Arts & Crafts	5	£175.00
Water Skiing	5	£250.00
Riflery	4	£180.00
Sailing	4	£60.00
<b>Session 1 Premium Activities Revenue</b>		<b>£3,917.00</b>

You are required to produce a breakdown of the premium activities for **Session 3**.

Task	Evidence Required
<b>3</b> (a) By using summary fields or otherwise, calculate:  i. the total number of times each premium activity has been chosen for Session 3, ii. the total revenue each premium activity has generated for Session 3, and iii. the overall revenue for premium activities for Session 3.  The activities should be listed with the most popular first. Where activities have been chosen the same number of times, they should be sorted alphabetically.	Printout of report/layout
(b) Repeat task 3(a), this time generating a report that details the totals for premium activities across <b>all three sessions</b> .	Printout of report/layout

Campers have requested that they would like to be able to browse through the Activity details one activity at a time as well as seeing the entire list of activities.

You are required to produce appropriate screen layouts that will enable the camper to browse the activities in this way. Two sample forms are shown below.



Task		Evidence Required
4	(a) Produce two layouts/forms to display: <ul style="list-style-type: none"> <li>i. a complete list of all activities</li> <li>ii. details of each activity, one activity at a time</li> </ul>	Printout of form/layout
	(b) Add to each form a navigational feature which will close the existing view and open the other view.	Printout of each macro/script used to control navigational features.



# **Appendix 1**

## **Completed Data Dictionary**

**(N.B. Only to be handed out after completion of Task 1)**

Entity	Attribute	PK/ FK	Data Type	Uni que	Requ ired	Validation	Format
Camper	Camper ID	PK	Number	Yes	Yes	>=0001, <=9999	0000
	Forename		Text(15)	No	Yes		
	Surname		Text(30)	No	Yes		
	Address1		Text(30)	No	Yes		
	Address2		Text(30)	No	No		
	Address3		Text(30)	No	Yes		
	Postcode		Text(8)	No	Yes		
	DOB		Date	No	Yes		dd/mm/yyyy
Activity	Activity	PK	Text(20)	Yes	Yes		
	Location		Text(20)	No	Yes		
	Additional Cost		Number	No	No	<100	currency
Session	Session No	PK	Number	Yes	Yes	Restricted 1, 2, 3	
	Start Date	PK	Date	Yes	Yes		dd/mm/yyyy
	End Date		Date	Yes	Yes		dd/mm/yyyy
	Session Cost		Number	No	Yes	Restricted 850, 1150	currency

Class	Camper ID	PK/ FK	Number	No	Yes	Lookup from Camper	0000
	Session No	PK/ FK	Number	No	Yes	Lookup from Session	
	Period	PK	Number	No	Yes	Restricted 1-10 OR >=1 and <=10	
	Activity	FK	Text(20)	No	Yes	Lookup from Activity	

**Coursework Task**

**Higher Information Systems**

*Valid for session 2008/2009 only*

**Please read Section I carefully before issuing materials to candidates.**

Publication code: BB3418

## Coursework Task: GoGoGadgets.com

### Higher Information Systems Coursework Task 2008-2009

#### Background

GoGoGadgets.com is a website that specialises in quirky and unusual gadgets that are available for purchase through its on-line catalogue. Before customers can make a purchase from the website, they must first register with GoGoGadgets. Customer details are stored by GoGoGadgets in a **Customer** table. On registering, customers are notified by e-mail of their unique customer ID which must be used in all future purchases.

GoGoGadgets' product range is stored in a table called **Item** and these items can be browsed through an on-line catalogue. Each item is categorised as one of the following: Boys Toys, Girls Gizmos, Office Distractions, Personal Grooming and Computer Accessories. All items cost less than £50.

When placing an order, customers must first log-in and create a new order – this order is added to an **Order** table. Customers can then add items from the catalogue to an order. More than one item can be added to a given order. Each individual item ordered is stored in a separate table called **OrderLine**.

A relational database has been set up to store details of customers, items, orders and order lines.

The normalised data model for GoGoGadgets has been produced, as follows. (Primary keys are shown in **bold and underlined**, while asterisks (\*) indicate foreign keys.)

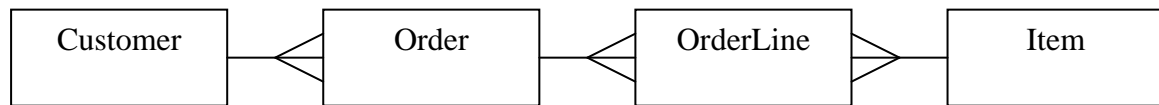
Customer (**Customer ID**, Forename, Surname, Address1, Address2, Address3, Postcode, Customer\_email)

Item (**ItemID**, Description, Category, Price)

Order (**Order No**, Customer ID\*, OrderDate, OrderDispatched)

Orderline (**Order No\***, **Item ID\***, Quantity)

The entity-relationship diagram for the system is shown below.



A data dictionary has been produced prior to implementation in a relational database. Part of this is shown overleaf.

Entity	Attribute	PK/ FK	Data Type/Size	Unique	Required	Validation	Format
Customer	CustomerID	PK	Number	Y	Y	>=0001 and <=9999	0000
	Forename		Text (10)	N	Y		
	Surname		Text (15)	N	Y		
	Address1		Text (20)	N	Y		
	Address2		Text (20)	N	N		
	Address3		Text (20)	N	Y		
	Postcode		Text (8)	N	Y		
	CustomerEmail		Text (25)	Y	<b>A</b>		
Item	ItemID	PK	Number	<b>B</b>	Y		0000
	Description		Text (50)	N	Y		
	Category		Text (20)	N	Y	<b>C</b>	
	Price		Number	N	Y	<50	Currency
Order	OrderNo	PK	Number	Y	Y		0000
	CustomerID	FK	Number	N	Y	<b>D</b>	0000
	OrderDate		Date	N	Y		dd/mm/yyyy
	OrderDispatched		Date	N	N		dd/mm/yyyy
OrderLine	OrderNo	PK/FK	Number	<b>E</b>	Y	Lookup from Order	0000
	ItemID	PK/FK	<b>F</b>	N	Y	Lookup from Item	0000
	Quantity		Number	N	Y		

For text attributes, the number in brackets indicates the size of the attribute (number of characters).

In the Format column, the symbols

- 0 represents a single digit
- d, m and y indicate digits used to represent the day, month and year in a date field

Task		Evidence Required
1	Using the background information provided, complete the data dictionary by making appropriate entries in the Task 1 Proforma.	Task 1 Proforma (Appendix 1)

**You must now submit your answer to Task 1 to your teacher/lecturer.**

Before attempting the tasks in this section ask your teacher/lecturer for a completed data dictionary.

You will also need access to pre-prepared database tables.

You **MUST** complete tasks in the sequence indicated.

Task	Evidence Required
2	A database has been set up containing the Customer, Item, Order and OrderLine tables.
(a) Edit the structure of each table to ensure it matches the data dictionary provided. You should pay particular attention to; <ul style="list-style-type: none"> <li>• primary and foreign keys</li> <li>• data type/size</li> <li>• unique</li> <li>• required</li> <li>• validation</li> <li>• format</li> </ul>	Hardcopy (printout or screenshots) of table structure including field types, validation checks etc.
(b) Set up/edit the relationships between all the tables as indicated by the entity-relationship diagram.	Hardcopy (printout or screenshots) of relationships between tables.

**You must now submit all hardcopies for Task 2 to your teacher/lecturer.**

Ask your teacher/lecturer for the complete database. This database must be used to carry out Tasks 3-5.

You are an employee of GoGoGadgets. One of your tasks is to produce invoices for each customer which details all of their orders for the finance department. These invoices need to show all items ordered, the individual item totals, the order total and the overall total for all that customer's order.

A sample invoice for customer Karen Springer is shown below.

<b>Customer Invoice</b>		<b>CustomerID 0023</b>			
		<b>Karen</b>	<b>Springer</b>		
<b>Order No</b>	4582	<b>Order Date</b>	21/09/2008		
<b>ItemID</b>	<b>Description</b>	<b>Category</b>	<b>Price</b>	<b>Quantity</b>	<b>Total Price</b>
1701	Foot Massager	Personal Grooming	£23.00	1	£23.00
<b>Order Total</b>					<b>£23.00</b>
<b>Order No</b>	2051	<b>Order Date</b>	14/06/2008		
<b>ItemID</b>	<b>Description</b>	<b>Category</b>	<b>Price</b>	<b>Quantity</b>	<b>Total Price</b>
6526	Miracle Eye Cream	Personal Grooming	£18.00	2	£36.00
3188	Mini Shaver	Boys Toys	£7.00	3	£21.00
<b>Order Total</b>					<b>£57.00</b>
<b>Order No</b>	4734	<b>Order Date</b>	22/12/2007		
<b>ItemID</b>	<b>Description</b>	<b>Category</b>	<b>Price</b>	<b>Quantity</b>	<b>Total Price</b>
9323	Electronic Parking Aid	Girls Gizmos	£32.00	1	£32.00
<b>Order Total</b>					<b>£32.00</b>
<b>Order No</b>	6212	<b>Order Date</b>	21/12/2007		
<b>ItemID</b>	<b>Description</b>	<b>Category</b>	<b>Price</b>	<b>Quantity</b>	<b>Total Price</b>
6584	Alpine Army Watch	Boys Toys	£34.00	2	£68.00
4908	Bunion Corrector	Personal Grooming	£11.00	2	£22.00
5472	USB Foot Warmer	Computer Accessories	£13.00	2	£26.00
<b>Order Total</b>					<b>£116.00</b>
<b>Invoice Total</b>					<b>£228.00</b>

<b>Task</b>	<b>Evidence Required</b>
<b>3</b> Using appropriate database functions create an invoice for customer Luke Muirhead, customer ID 7853. The layout should be similar to the one shown above. The data should be sorted with the most recent order first.	Printout of report/layout

GoGoGadgets ran a promotion in the run up to Christmas by offering a 5% discount to the total of any order placed in December 2007. A report was generated so that Head Office could analyse the impact of that promotion. This report is shown below.

Orders Placed December 2007				
OrderNo	OrderDate	OrderTotal	5 % Discount	Total After Discount
2030	04/12/2007	£103.00	£5.15	£97.85
9302	05/12/2007	£12.00	£0.60	£11.40
5482	20/12/2007	£42.00	£2.10	£39.90
3246	21/12/2007	£148.00	£7.40	£140.60
5435	21/12/2007	£54.00	£2.70	£51.30
6212	21/12/2007	£116.00	£5.80	£110.20
6867	21/12/2007	£17.00	£0.85	£16.15
4734	22/12/2007	£32.00	£1.60	£30.40
4708	24/12/2007	£10.00	£0.50	£9.50
<b>Totals For Month</b>		<b>£534.00</b>	<b>£26.70</b>	<b>£507.30</b>
<b>Number of orders placed</b>				<b>9</b>

The management of GoGoGadgets then decided to try and boost sales in the quiet period after Christmas by offering a 20% discount to the total order price of any order placed in January 2008.

Task	Evidence Required
<p><b>4</b> Head Office want to analyse how effective their January promotion has been and have asked you to prepare a report similar to the one shown above. This report must show:</p> <ul style="list-style-type: none"> <li>• all orders placed in January 2008 showing the order date and order total rather than individual items on the order</li> <li>• the value of each order's 20% discount</li> <li>• the value of each order after discount</li> <li>• a summary field showing the overall monthly total before discount</li> <li>• a summary field showing the overall total discount for the month</li> <li>• a summary field showing the overall monthly total after discount</li> <li>• the number of orders placed in January 2008</li> </ul> <p>The report should be sorted by order date as shown. Where two or more orders are placed on the same day, they should be sorted by OrderNo in ascending order.</p>	<p>Printout of form/layout</p>

A customer satisfaction survey has revealed that customers would like to be able to browse catalogue pages organised by category.

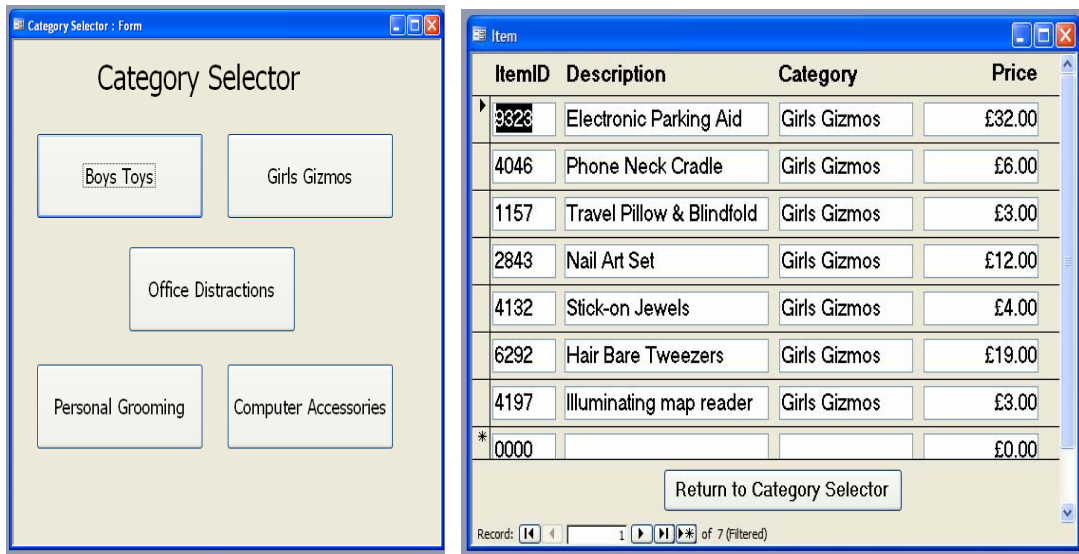


Fig (a): Category Selector Form      Fig (b): Single Category Form (Girls Gizmos)

Task		Evidence Required
5	Use the database to perform the following tasks.	Screen shots of forms/layouts  Printout of macros/scripts used to control navigational features
	<p>(a) Create a Category Selector form similar to the one shown in Fig (a) above. Choose any of the five categories and create a navigational link to a separate form that shows only items from the chosen category (eg Fig (b)).</p> <p>(b) Amend the single category item form to include a link back to the Category Selection form.</p> <p><b>Each navigational link should close the current view/form and open the selected form.</b></p>	

## **Section 3**

## **Marking Grid**

## Marking Grid

Name \_\_\_\_\_

Date \_\_\_\_\_

### Part I – Relational Database Systems

Task	Topic	Possible Marks	Marks Gained	Evidence	Comment
1.	A	1,0		Proforma	
	B	1,0			
	C	1,0			
	D	1,0			
	E	1,0			
	F	1,0			
2.	Edit Customer, Item, Order and OrderLine tables	4,3,2,1,0		Hardcopy of Database solution files (eg Documenter in Access)	
	Links between tables	2,1,0		Screenshot of relationships	
3.	Computed & Summary fields	3,2,1,0		Hardcopy of report	
	Report layout	2,1,0			
4.	Order details	3,2,1,0		Hardcopy of report	
	Summary fields	3,2,1,0			
	Report layout	2,1,0			
5.	Forms created	2,1,0		Screenshots of forms	
	Navigation features	3,2,1,0		Printout of macros/scripts used to control navigation	

## Appendix 1

### Task 1 Proforma

<b>Name</b>	
<b>Data Dictionary Entry</b>	
<b>A</b>	
<b>B</b>	
<b>C</b>	
<b>D</b>	
<b>E</b>	
<b>F</b>	

## Appendix 2

### Completed Data Dictionary

Only to be issued after completion and submission of Task 1

Entity	Attribute	PK/ FK	Data Type/Size	Unique	Required	Validation	Format
Customer	CustomerID	PK	Number	Y	Y	>=0001 and <=9999	0000
	Forename		Text (10)	N	Y		
	Surname		Text (15)	N	Y		
	Address1		Text (20)	N	Y		
	Address2		Text (20)	N	N		
	Address3		Text (20)	N	Y		
	Postcode		Text (8)	N	Y		
	CustomerEmail		Text (25)	Y	Y		
Item	ItemID	PK	Number	Y	Y		0000
	Description		Text (50)	N	Y		
	Category		Text (20)	N	Y	Restricted Choice (Boys Toys, Girls Gizmos, Personal Grooming, Office Distractions, Computer Accessories)	
	Price		Number	N	Y	<50	Currency
Order	OrderNo	PK	Number	Y	Y		0000
	CustomerID	FK	Number	N	Y	Lookup from Customer	0000
	OrderDate		Date	N	Y		dd/mm/yyyy
	OrderDispatched		Date	N	N		dd/mm/yyyy
OrderLine	OrderNo	PK/FK	Number	N	Y	Lookup from Order	0000
	ItemID	PK/FK	Number	N	Y	Lookup from Item	0000
	Quantity		Number	N	Y		

**Coursework Task**

**Higher Information Systems**

*Valid for session 2009/2010 only*

**Please read Section I carefully before issuing materials to candidates.**

Publication code: BB3418

# Coursework Task: OneTwoSki

## Higher Information Systems Coursework Task 2009-2010

### Background

OneTwoSki is a company that owns and manages a number of ski resorts in Canada. All resort details are stored as part of a relational database in a table called **Resort**. Resort names are unique and some of the resorts have spa and/or crèche facilities. Each resort has a resort manager who stores customer bookings in the centralised relational database.

When a customer first makes a booking they must provide their ski ability level (Beginner, Intermediate or Expert) in addition to their contact details. This information is stored by OneTwoSki in a **Customer** table.

Customer bookings are made by a party leader and each party booking is made for skiers of the same ability. Details of customer bookings are stored in a **Booking** table. Each booking is allocated a unique 6-digit number. Other details stored include information on the number of people staying, the duration of the stay and the number of ski days booked. The minimum stay is three days and no booking is allowed for more than two weeks. Ski passes for ski days are available as 3-day, 5-day, 7-day, 10-day or 12-day tickets. Arrival dates are stored in the form dd/mm/yyyy.

The company pride themselves on the information they can offer to their customers about the ski trails available in each of their resorts. All ski trails are allocated a colour which indicates its level of difficulty as follows:

Green	Beginner
Blue	Intermediate
Red	Advanced
Black	Expert

Each ski trail is assigned a unique ID number along with information about its name, colour and length (km). These details are stored in the **Trail** table.

The normalised data model for OneTwoSki was produced and a surrogate key, **Booking Ref**, was introduced to the Booking table to simplify the implementation.

The resulting data model is shown below where primary keys are shown in **bold and underlined**, while asterisks (\*) indicate foreign keys.)

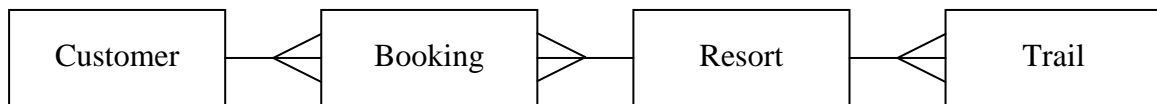
Customer (**Customer ID**, Firstname, Surname, Street, Town, Postcode, Ski Level)

Resort (**Resort**, Resort Manager, Resort Address, Spa, Crèche)

Booking (**Booking Ref**, Customer ID\*, Resort\*, Arrival Date, Persons, Nights, Ski Days)

Trail (**Trail ID**, Resort\*, Trail Name, Colour, Length (km))

The entity-relationship diagram for the system is shown below.



The data dictionary for the Customer, Resort and Trail tables is shown below, followed by a partially completed data dictionary for the Booking table.

Entity	Attribute	PK/ FK	Data Type/Size	Unique	Required	Validation	Format
Customer	CustomerID	PK	Number	Y	Y	>=1 and <=9999	0000
	Firstname		Text (15)	N	Y		
	Surname		Text (20)	N	Y		
	Street		Text (30)	N	Y		
	Town		Text (20)	N	Y		
	Postcode		Text (8)	N	Y		
	Ski Level		Text (12)	N	Y	Restricted (Beginner, Intermediate, Expert)	
Resort	Resort	PK	Text (20)	Y	Y		
	Resort Manager		Text (35)	N	Y		
	Resort Address		Text (50)	Y	Y		
	Spa		Boolean	N	Y		
	Crèche		Boolean	N	Y		
Trail	Trail ID	PK	Number	Y	Y	>=1 and <=999	000
	Resort	FK	Text (20)	N	Y	Lookup from Resort	
	Trail Name		Text (30)	N	Y		
	Colour		Text (5)	N	Y	Restricted (Green, Blue, Red, Black)	
	Length (km)		Number	N	Y		

Booking						
Attribute	PK/FK	Data Type/Size	Unique	Required	Validation	Format
Booking Ref	PK	Number	Y	Y	A	000000
Customer ID	FK	Number	N	Y	B	0000
Resort	FK	Text (20)	N	Y	C	
Arrival Date		Date	N	Y		dd/mm/yyyy
Persons		Number	N	Y		
Nights		Number	N	Y	D	
Ski Days		Number	N	Y	E	

Task		Evidence Required
1	Using the background information provided in the introduction to the task, complete the data dictionary entries for the Booking table.	Completed data dictionary

**You must now submit your answer to Task 1 to your teacher/lecturer.**

Before attempting the tasks in this section ask your teacher/lecturer for a completed data dictionary.

You will also need access to pre-prepared database tables.

You **MUST** complete tasks in the sequence indicated.

<b>Task</b>	<b>Evidence Required</b>
<b>2</b>	Database tables have been created for the Customer, Resort and Trail entities.
(a) Using the completed data dictionary issued by your teacher/lecturer, create the Booking table. You should pay particular attention to: <ul style="list-style-type: none"> <li>• data type</li> <li>• unique</li> <li>• required</li> <li>• validation</li> <li>• primary and foreign keys</li> </ul>	Hardcopy (printout or screenshots) of <b>Booking table structure</b> including field types, validation checks etc.
(b) Set up/edit the relationships between all the tables as indicated by the entity-relationship diagram.	Hardcopy (printout or screenshots) of relationships between tables – <b>evidence must show correct cardinality</b>

**You must now submit all hardcopies for Task 2 to your teacher/lecturer.**

Ask your teacher/lecturer for the complete database. This database must be used to carry out Tasks 3-5.

You work in the head office of OneTwoSki. Part of your job is to deal with customer queries. Use the database provided to produce reports for the following requests made by customers.

Task	Evidence Required																
<p><b>3</b> An expert skier wanted to know which resorts were booked most often by people with his ski ability, so the following report was produced.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <p style="text-align: center;"><b>Bookings by expert party leaders</b></p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; border-bottom: 1px solid black;">Resort</th> <th style="text-align: right; border-bottom: 1px solid black;">Number of bookings</th> </tr> </thead> <tbody> <tr> <td>CloudyBay</td> <td style="text-align: right;">5</td> </tr> <tr> <td>BlueRidge</td> <td style="text-align: right;">4</td> </tr> <tr> <td>SkyHaven</td> <td style="text-align: right;">4</td> </tr> <tr> <td>BlackBear</td> <td style="text-align: right;">1</td> </tr> <tr> <td>FreeSpirit</td> <td style="text-align: right;">1</td> </tr> <tr> <td>TranquilPeaks</td> <td style="text-align: right;">1</td> </tr> <tr> <td><b>Total bookings:</b></td> <td style="text-align: right;"><b>16</b></td> </tr> </tbody> </table> </div> <p>Another request has come in asking you to provide a similar report that shows the most popular resorts with Intermediate party leaders.</p> <p>The list should be sorted with the most popular resort first. Where two or more resorts have the same number of bookings, they should be sorted alphabetically.</p> <p>The total number of bookings should be displayed at the bottom of the report.</p>	Resort	Number of bookings	CloudyBay	5	BlueRidge	4	SkyHaven	4	BlackBear	1	FreeSpirit	1	TranquilPeaks	1	<b>Total bookings:</b>	<b>16</b>	<p>Printout of report/layout</p>
Resort	Number of bookings																
CloudyBay	5																
BlueRidge	4																
SkyHaven	4																
BlackBear	1																
FreeSpirit	1																
TranquilPeaks	1																
<b>Total bookings:</b>	<b>16</b>																

Task	Evidence Required																					
<p><b>4</b> A customer wanted to know the total lengths of the different trail colours at the BlackBear resort, as well as the total trail length for the resort.</p> <p>OneTwoSki provided the following report.</p> <table border="1" data-bbox="252 465 932 882"> <thead> <tr> <th colspan="3">Trail lengths - BlackBear Resort</th> </tr> <tr> <th>BlackBear</th> <th>Colour</th> <th>Trail length (km)</th> </tr> </thead> <tbody> <tr> <td></td> <td>Blue</td> <td>28.7</td> </tr> <tr> <td></td> <td>Red</td> <td>24.8</td> </tr> <tr> <td></td> <td>Green</td> <td>14.8</td> </tr> <tr> <td></td> <td>Black</td> <td>12.7</td> </tr> <tr> <td colspan="2"><b>Total length of runs (km)</b></td> <td><b>81.0</b></td> </tr> </tbody> </table> <p>Another customer has asked you for the same information about trail lengths but wants this for all resorts that have crèche facilities. The information should be sorted by resort and the trail details for each resort should be displayed with the colour of trail with the longest length first. Trail lengths should be displayed to one decimal place.</p>	Trail lengths - BlackBear Resort			BlackBear	Colour	Trail length (km)		Blue	28.7		Red	24.8		Green	14.8		Black	12.7	<b>Total length of runs (km)</b>		<b>81.0</b>	<p>Printout of report/layout</p>
Trail lengths - BlackBear Resort																						
BlackBear	Colour	Trail length (km)																				
	Blue	28.7																				
	Red	24.8																				
	Green	14.8																				
	Black	12.7																				
<b>Total length of runs (km)</b>		<b>81.0</b>																				

Task	Evidence Required
<p><b>5</b> A customer has requested a quicker way of navigating through the list of ski trails available.</p> <p>You have been asked to design a new form/layout for the Trail table that includes navigation buttons to skip forward and back 10 trails at a time. When you click on the forward button, for example, the pointer or cursor should skip to the 10<sup>th</sup> record then the 20<sup>th</sup> record, and so on. The new form/layout does not necessarily have to display only 10 trails at a time – it should simply allow this quicker method of navigation.</p> <p>These buttons should make use of a macro or script to control the navigation.</p>	<p>Screenshots of form/layout</p> <p>Printout of macros/scripts used to control navigational features</p>

## **Section 3**

## **Marking Grid**

# Marking Grid

Name \_\_\_\_\_ Date \_\_\_\_\_

Task	Possible Marks	Actual Marks	Evidence	Comment
1.	5		Completed data dictionary	
2.(a)	4		Printout of table structure	
2.(b)	3		Printout showing relationships	
3.	6		Printout of report	
4.	8		Printout of report	
5.	4		Printout of macros/scripts	
6.	12		Values printout Formulae printout	
7.	10		Printout of Gantt chart	
8	8		Printout of flyer	
	60			

## Appendix 1

### Task 1 Proforma

<b>Name</b>	
<b>Data Dictionary Entry</b>	
<b>A</b>	
<b>B</b>	
<b>C</b>	
<b>D</b>	
<b>E</b>	

## Appendix 2

### Completed Data Dictionary

Only to be issued after completion and submission of Task 1

Entity	Attribute	PK/ FK	Data Type/Size	Unique	Required	Validation	Format
Customer	CustomerID	PK	Number	Y	Y	>=1 and <=9999	0000
	Firstname		Text (15)	N	Y		
	Surname		Text (20)	N	Y		
	Street		Text (30)	N	Y		
	Town		Text (20)	N	Y		
	Postcode		Text (8)	N	Y		
	Ski Level		Text (12)	N	Y	Restricted (Beginner, Intermediate, Expert)	
Resort	Resort	PK	Text (20)	Y	Y		
	Resort Manager		Text (35)	N	Y		
	Resort Address		Text (50)	Y	Y		
	Spa		Boolean	N	Y		
	Crèche		Boolean	N	Y		
Trail	Trail ID	PK	Number	Y	Y	>=1 and <=999	000
	Resort	FK	Text (20)	N	Y	Lookup from Resort	
	Trail Name		Text (30)	N	Y		
	Colour		Text (5)	N	Y	Restricted (Green, Blue, Red, Black)	
	Length (km)		Number	N	Y		
Booking	Booking Ref	PK	Number	Y	Y	>=1 and <=999999	000000
	Customer ID	FK	Number	N	Y	Lookup from Customer	0000
	Resort	FK	Text (20)	N	Y	Lookup from Resort	
	Arrival Date		Date	N	Y		dd/mm/yyyy
	Persons		Number	N	Y		
	Nights		Number	N	Y	>=3 and <=14	
	Ski Days		Number	N	Y	Restricted (3,5,7,10,12)	

**Coursework Task**

**Higher Information Systems**

*Valid for session 2010/2011 only*

**Please read Section I carefully before issuing materials to candidates.**

Publication code: BB3418

# Coursework Task: GoToGigs

## Higher Information Systems Coursework Task 2010-2011

### Background

GoToGigs is an organisation that owns and manages a number of concert venues throughout Scotland. They have venues in Aberdeen, Dundee, Edinburgh and Glasgow. All venue details are stored as part of a relational database in a table called **Venue**. Venue names are unique. As well as the venue name and address, the following details are stored:

- Seating Capacity (between 1,000 and 12,000)
- Hire Cost (between £750 and £9,850)

Promoters of bands register with GoToGigs and their contact details are stored in a **Promoter** table. Promoters are allocated a unique ID. Promoters charge each band they represent a fixed commission fee for every gig they book, based on a percentage of ticket sales. No promoter charges more than a 10% fee.

Bands are represented and managed by promoters. A band can only be managed by one promoter, although the same promoter can be responsible for many bands. The details of the bands that each promoter represents are stored in a **Band** table. Band names are unique.

When a promoter wants to arrange a concert for one of the bands they represent they negotiate with GoToTickets to decide which venues will be played, which dates will be played, the price of tickets etc. Tickets are priced between £15 and £65. No venue can host more than one band on any given date. These details are stored in table called **Gig**.

Bands, promoters and customers attending concerts frequently make requests to GoToGigs for specific information.

GoToGigs use a relational database to manage this information.

The relational data model is shown on the next page.

## Entities

The resulting data model is shown below where primary keys are shown in **bold and underlined**, while asterisks (\*) indicate foreign keys.)

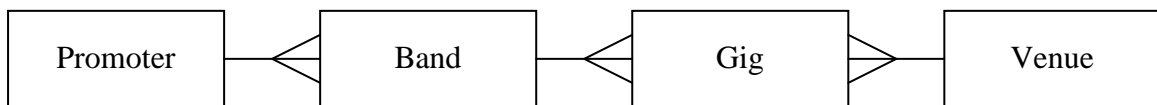
Promoter (**Promoter ID**, PromoterName, PromoterPhone, CommissionRate)

Band (**BandName**, NumberofMembers, YearFormed, PromoterID\*)

Venue (**VenueName**, VenueAddress, VenueCity, SeatingCapacity, HireCost)

Gig (**VenueName\***, **Date**, BandName\*, CostPerTicket, TicketsSold)

The entity-relationship diagram for the system is shown below.



A data dictionary has been produced prior to implementation in a relational database. Part of this is shown below.

### Data Dictionary

Entity	Attribute	PK/ FK	Data Type/Size	Unique	Required	Validation	Format
Promoter	PromoterID	PK	Number	Y	Y		
	PromoterName		Text (20)	N	Y		
	PromoterPhone		Text (15)	Y	Y		
	CommissionRate		Number	N	Y	<=0.10	
Band	BandName	PK	Text (30)	Y	Y		
	NumberOfMembers		Number	N	Y		
	YearFormed		Number	N	Y		
	PromoterID	FK	Number	N	Y	Lookup from Promoter	
Venue	VenueName	PK	Text(20)	Y	Y		
	VenueAddress		Text (40)	Y	Y		
	VenueCity		Text (15)	N	Y		
	SeatingCapacity		Number	N	Y	>=1000 and <=12000	
	HireCost		Number	N	Y	>=750 and <=9850	
Gig	VenueName			N	Y		
	Date		Date	N	Y		
	BandName			N	Y		
	CostPerTicket		Number	N	Y		
	TicketsSold		Number	N	Y		

Task		Evidence Required
1	<p>Your teacher/lecturer will provide you with a partially completed data dictionary for the Gig table.</p> <p>Using the background information provided, complete the data dictionary for the Gig table by making appropriate entries in the shaded areas as indicated.</p>	<p>Completed data dictionary (Proforma 1)</p>

**You must now submit your answer to Task 1 to your teacher/lecturer.**

Before attempting the tasks in this section ask your teacher/lecturer for a completed data dictionary.

You will also need access to pre-prepared database tables.

You **MUST** complete tasks in the sequence indicated.

Task	Evidence Required
<b>2</b>	Database tables have been created for the Venue, Promoter and Band tables.
(a) Using the completed data dictionary issued by your teacher/lecturer, create the Gig table. You should pay particular attention to: <ul style="list-style-type: none"> <li>• data type (text size)</li> <li>• required</li> <li>• validation</li> <li>• primary and foreign keys</li> </ul>	Hardcopy (printout or screenshots) of table structure including field types, validation checks etc.
(b) Set up/edit the relationships between all the tables as indicated by the entity-relationship diagram.	Hardcopy (printout or screenshots) of relationships between tables – <b>evidence must show correct cardinality</b>

**You must now submit all hardcopies for Task 2 to your teacher/lecturer.**

Ask your teacher/lecturer for the complete database. This database must be used to carry out Tasks 3-5.

A music magazine had made a request for information from GoToGigs concerning the popularity of venues for concert goers. They wanted to publish a list of the most popular cities in terms of total number of tickets sales. GoToGigs prepared the following report.

<i>Tickets sold by City</i>	
<i>VenueCity</i>	<i>Total number of tickets sold</i>
Edinburgh	162098
Glasgow	143575
Aberdeen	96879
Dundee	80880

The magazine now wants a similar list but this time it should show the **total revenue** generated by ticket sales for each city. The list should be ordered from the city with the largest revenue. The revenue is calculated as the sum of tickets sold multiplied by the cost per ticket.

<b>Task</b>	<b>Evidence Required</b>
<b>3</b> Using appropriate database functions, create a report that displays the total revenue generated by gigs played at each of the four cities. The report data should be ordered with the city with the biggest revenue first. Your report should include only the name of the city and the total amount of revenue generated. The revenue should be displayed in currency format.	Printout of report/layout

GoToGigs have been monitoring which of the promoters has been arranging the most concerts and created the report shown below.

<i>Total Gigs by Promoter</i>	
<u>PromoterName</u>	<u>Total gigs arranged</u>
Emma Domineche	8
Gordon Neil	15
Iain Allan	16
Liz Laird	9
Mary Hughes	22
Sean Stevens	10

They have decided that the report does not provide enough detail and have asked you to improve the information shown by displaying a breakdown of how often each promoter arranges gigs in each of the four cities. A sample layout has been provided below.

<i>Promoters gigs by City</i>			
<i>Promoter</i>	<i>Emma Domineche</i>	<i>City</i>	<i>Gigs</i>
		*****	**
		*****	**
		*****	**
		*****	**
		<b>Total Gigs</b>	**
<i>Promoter</i>	<i>Gordon Neil</i>	<i>City</i>	<i>Gigs</i>
		*****	**
		*****	**
		*****	**
		*****	**
		<b>Total Gigs</b>	**
<i>Promoter</i>	<i>Iain Allan</i>	<i>City</i>	<i>Gigs</i>
		*****	**
		*****	**
		*****	**
		*****	**
		<b>Total Gigs</b>	**

<b>Task</b>	<b>Evidence Required</b>
<p><b>4</b> Create a report as follows:</p> <ul style="list-style-type: none"> <li>• Show, for each promoter the number of gigs they have arranged in each of the four cities.</li> <li>• The data for each promoter should be arranged with the city booked most often first.</li> <li>• Where two or more cities have been used the same number of times they should be also sorted.</li> <li>• Include the promoter name, the name of each city, the number of times each city has been used for gigs and the total number of gigs for each promoter.</li> </ul>	<p>Printout of report/layout (no requirement for report to be sorted by Promoter)</p>

GoToGigs think it would be helpful to be able to view a list of gigs that are on at each venue. They have decided to create a Venue Selector form/layout where each venue is listed. When a specific venue is chosen the Venue select screen closes and is replaced by a list of all gigs at the chosen venue. The list of gigs also includes a link back to the Venue Selector screen, as is shown in the 2 sample screens below using “The Shack” venue as an example.

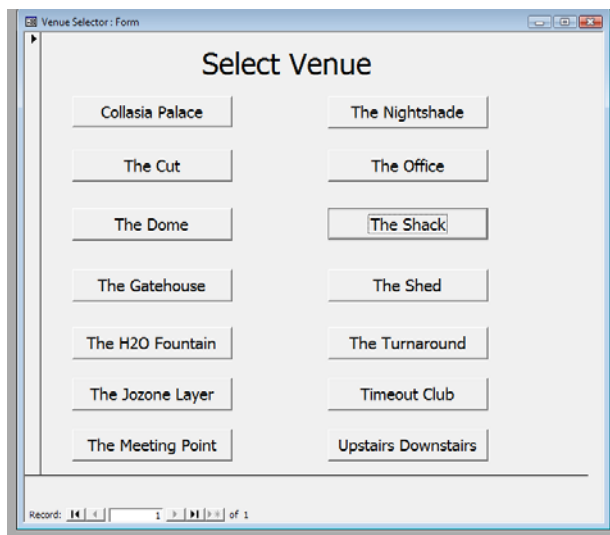


Fig (a) Venue Selector Form

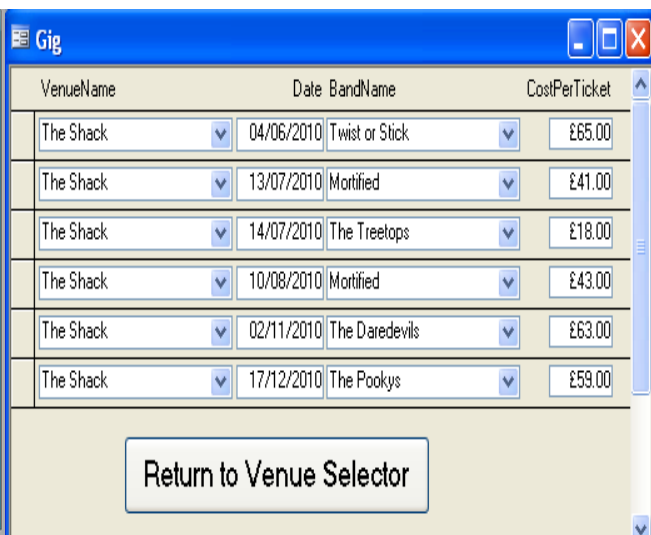


Fig (b) Gigs at chosen venue

Task	Evidence Required
<p><b>5</b> Your GoToGigs database already includes the Venue Selector form shown in Fig (a) above.</p> <p>Choose <b>one</b> venue and create, using a macro or script, a navigational link to a separate form/layout that shows only the gigs from that venue, as shown in Fig (b). <b>Do not include</b> the TicketsSold attribute. Include on this form/layout a link back to the Venue select screen, again through the use of a macro or script.</p> <p><b>Each navigational link should close the current form/layout and open the selected form/layout.</b></p>	<p>Printout of report/layout</p> <p>Printout of macros/scripts used to control navigational features</p>

## Appendix 1

### Task 1 Proforma

Name					
Data dictionary for Gig entity					
Attribute	PK/FK	Data Type/Size	Unique	Reqd	Validation
VenueName			N	Y	
Date		Date	N	Y	-
BandName			N	Y	
CostPerTicket	-	Number	N	Y	
TicketsSold	-	Number	N	Y	-

## Appendix 2

### Completed Data Dictionary

Only to be issued after completion and submission of Task 1

Entity	Attribute	PK/ FK	Data Type/Size	Unique	Required	Validation
Promoter	PromoterID	PK	Number	Y	Y	
	PromoterName		Text (20)	N	Y	
	PromoterPhone		Text (15)	Y	Y	
	CommissionRate		Number	N	Y	<=0-10
Band	BandName	PK	Text (30)	Y	Y	
	NumberofMembers		Number	N	Y	
	YearFormed		Number	N	Y	
	PromoterID	FK	Number	N	Y	Lookup from Promoter
Venue	VenueName	PK	Text(20)	Y	Y	
	VenueAddress		Text (40)	Y	Y	
	VenueCity		Text (15)	N	Y	
	SeatingCapacity		Number	N	Y	>=1000 and <=12000
	HireCost		Number	N	Y	>=750 and <=9850
Gig	VenueName	PK/FK	Text(20)	N	Y	Lookup from Venue
	Date	PK	Date	N	Y	
	BandName	FK	Text(30)	N	Y	Lookup from Bank
	CostPerTicket		Number	N	Y	>=15 and <=65
	TicketsSold		Number	N	Y	