

LAB 5: More queries and forms

Introduction

In this lab you will practice some more queries and create some web pages to view data in a database.

Task 1: Queries

In this task you will practice using SQL to query the GCUTours database.

YOU will use the same database as you used in lab 3. As before, download the file **gcutourswm.zip** from your module web site and extract the contents, which should consist of a folder **gcutourswm**. Start up Microsoft WebMatrix and select the **Site From Folder** option, and navigate to this folder. Switch to the **Databases** workspace in WebMatrix. Select *gcutourswm.sdf* in the content pane and click the **New Query** icon on the Home ribbon to open a new query window. You are now ready to type an SQL query and execute it.

Create a new Word document called **INTDD_lab5.doc**. You will use this document to record all the SQL statements you use in this lab.

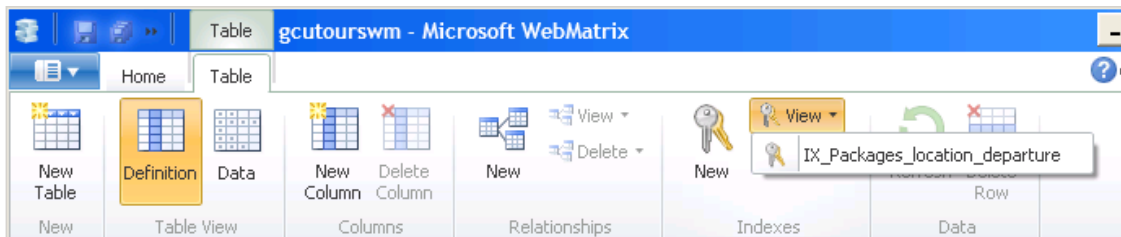
In each of the following questions you are given a description of the query you are require to create, and where appropriate, a listing of the result which your query should return.

For each question, type the **SQL statement** in the query window and **run the query** to test it. You will probably sometimes need some trial-and-error before you get the correct result.

Write the heading *Task 1* in your Word document, and enter the question number in each case and copy and paste your SQL statement.

1. There may be a frequent need to look up packages by location and departure. Create (using SQL) a compound index called *IX_Packages_location_departure* on these two fields.

Check that the index has been created. Open the *Packages* table in Definition view using the Indexes -> View option on the Table ribbon, and check that the correct fields are included in the index. Close the table



It is decided that there is no need for this index. Using SQL, drop the index *IX_Packages_location_departure*.

Open the *Packages* table in Definition view and check that there are no indexes on the table.

2. In Lab 3 you found the package name and location for the tour with tourID = 3

| name | location |
|-------------------|----------|
| Western Adventure | USA |

using a subquery, like this:

```
SELECT packagename, location
FROM Packages
WHERE packageID IN
    (SELECT packageID FROM Tours WHERE tourID = 3)
```

Write this query using a join instead of a subquery.

3. List for every booking the first name and last name of the user and the number of adults and children booked.

| firstname | lastname | adults | children |
|-----------|----------|--------|----------|
| Ferdinand | Magellan | 2 | 3 |
| Marco | Polo | 2 | 2 |
| Marco | Polo | 1 | 0 |
| Marco | Polo | 2 | 2 |
| Vasco | daGama | 4 | 0 |
| Vasco | daGama | 2 | 0 |

4. List for every booking for which tickets have not been sent the first name and last name of the user and the number of adults and children booked

| firstname | lastname | adults | children |
|-----------|----------|--------|----------|
| Marco | Polo | 1 | 0 |
| Marco | Polo | 2 | 2 |
| Vasco | daGama | 2 | 0 |
| Ferdinand | Magellan | 2 | 3 |

5. List for every booking the tour departure date and the first name and last name of the user and the number of adults and children booked

| departureDate | firstname | lastname | adults | children |
|---------------|-----------|----------|--------|----------|
| 01/03/2008 | Marco | Polo | 2 | 2 |
| 05/06/2008 | Marco | Polo | 1 | 0 |
| 01/08/2008 | Marco | Polo | 2 | 2 |
| 01/03/2008 | Vasco | daGama | 4 | 0 |
| 02/09/2008 | Vasco | daGama | 2 | 0 |
| 01/06/2008 | Ferdinand | Magellan | 2 | 3 |

6. List for every booking the package name, the tour departure date and the first name and last name of the user and the number of adults and children booked

| name | departureDate | firstname | lastname | adults | children |
|----------------------------|---------------|-----------|----------|--------|----------|
| Western Adventure | 01/03/2008 | Marco | Polo | 2 | 2 |
| Roof of the World Explorer | 05/06/2008 | Marco | Polo | 1 | 0 |
| Amazon & Inca Adventure | 01/08/2008 | Marco | Polo | 2 | 2 |
| Roof of the World Explorer | 01/03/2008 | Vasco | daGama | 4 | 0 |
| Reef and Outback Adventure | 02/09/2008 | Vasco | daGama | 2 | 0 |
| Trans-Siberian Express | 01/06/2008 | Ferdinand | Magellan | 2 | 3 |

7. You try to list the name, location and departure date for every tour using the following query. Run this query. What happens? Why?

```
SELECT packageame, location, departuredate
FROM Packages, Tours
```

Modify the query to give the result below.

| name | location | departuredate |
|----------------------------|---------------|---------------------|
| Western Adventure | USA | 01/03/2011 00:00:00 |
| Western Adventure | USA | 05/06/2011 00:00:00 |
| Western Adventure | USA | 02/09/2011 00:00:00 |
| Roof of the World Explorer | Asia | 02/09/2011 00:00:00 |
| Roof of the World Explorer | Asia | 01/03/2011 00:00:00 |
| Alpine Action | Europe | 05/06/2011 00:00:00 |
| Reef and Outback Adventure | Australia | 01/03/2011 00:00:00 |
| Reef and Outback Adventure | Australia | 01/03/2011 00:00:00 |
| Trans-Siberian Express | Asia | 01/03/2011 00:00:00 |
| Trans-Siberian Express | Asia | 01/03/2011 00:00:00 |
| Trans-Siberian Express | Asia | 01/03/2011 00:00:00 |
| Borneo Explorer | Asia | 02/09/2011 00:00:00 |
| Amazon & Inca Adventure | South America | 01/03/2011 00:00:00 |
| Amazon & Inca Adventure | South America | 01/06/2011 00:00:00 |
| Patagonia Trek | South America | 01/08/2011 00:00:00 |
| Colorado Winter Adventure | USA | 08/03/2011 00:00:00 |
| Colorado Winter Adventure | USA | 01/02/2011 00:00:00 |
| Raft the Grand Canyon | USA | 01/08/2011 00:00:00 |
| Raft the Grand Canyon | USA | 01/05/2011 00:00:00 |
| Raft the Grand Canyon | USA | 01/02/2011 00:00:00 |
| Raft the Grand Canyon | USA | 01/03/2011 00:00:00 |
| Rising Sun Explorer | Asia | 01/05/2011 00:00:00 |

8. List the tourID and departure date for every booking with at least one child. Write and run two versions of this query, one joining tables with WHERE and one using INNER JOIN.

| tourID | departuredate |
|--------|---------------------|
| 1 | 01/03/2011 00:00:00 |
| 14 | 01/06/2011 00:00:00 |
| 10 | 01/03/2011 00:00:00 |

9. List the booking ID, package name and total cost for every booking. *Hint: the total cost will be the (number of adults x adult price) + (number of children x child price)*

| bookingID | name | TotalCost |
|-----------|----------------------------|-----------|
| 1 | Western Adventure | 4996 |
| 2 | Roof of the World Explorer | 1599 |
| 3 | Amazon & Inca Adventure | 3996 |
| 4 | Roof of the World Explorer | 5996 |
| 5 | Reef and Outback Adventure | 1798 |
| 8 | Trans-Siberian Express | 9645 |

10. List the total cost of each user's bookings. *Hint: you will need to group by username*

| username | SumTotalCost |
|----------|--------------|
| ferdy | 9645 |
| mpolo | 10591 |
| vdagama | 7794 |

11. Find the number of bookings made for each location

| location | NoOfBookings |
|---------------|--------------|
| Asia | 3 |
| Australia | 1 |
| South America | 1 |
| USA | 1 |

12. Increase the prices of all packages by 10%.

13. Change the password for mpolo to Pa\$\$w0rd

14. Delete the tour with tourID 5.

This query shouldn't work! Why not?

Use ALTER TABLE statements to:

- Drop the foreign key relationship between Tours and Bookings
- Add a new foreign key relationship between Tours and Bookings so that when you delete a tour you delete related bookings

Run your query again and note changes to both tables.

Task 2: Creating web pages to view data

In this lab you will create web pages to view data in the GCUCars database. These will be a pair of master-detail pages. The first page will display a list of **car types** (e.g. SUV). There will be a hyperlink for each type which will lead to a page which displays all the **car models** which are of that type. You may need to refer back to lab 1 for help on creating web page files in WebMatrix.

CarTypes page

1. Download the file **gcucars.zip** from your module web site and extract the contents, which should consist of a folder **gcucars**. Start up Microsoft WebMatrix and select the **Site From Folder** option, and navigate to this folder. Switch to the **Files** workspace in WebMatrix.
2. Create a new CSHTML file called *cartypes.cshtml*. Enter the following code:

```
@{
    var db = Database.Open("gcucars");
    var selectQueryString = "SELECT * FROM Cartypes";
}
<!DOCTYPE html>
<html>
    <head>
        <title>GCUCars Car Types</title>
    </head>
    <body>
        <h1>GCUCars Car Types</h1>
        <table>
            <thead>
                <tr>
                    <th>Description</th>
                    <th>Daily cost</th>
                    <th>CDW daily cost</th>
                    <th>Models</th>
                </tr>
            </thead>
            <tbody>
                @foreach (var row in db.Query(selectQueryString)){
                    <tr>
                        <td>@row.description</td>
                        <td>£@row.dailyCost</td>
                        <td>£@row.CDWDailyCost</td>
                        <td><a href="carmodels.cshtml?description=@row.description">models</a></td>
                    </tr>
                }
            </tbody>
        </table>
        <p>The SQL query which retrieved the data for this page was:<br/>
        @selectQueryString</p>
    </body>
</html>
```

3. Save the file and run it. Your web browser should open and the page content should look like this. Note that the links will not work yet as you have not created the "models" page.

GCUCars Car Types

| Description | Daily cost | CDW daily cost | Models |
|--------------|------------|----------------|------------------------|
| Economy | £38 | £12 | models |
| Compact | £45 | £14 | models |
| Intermediate | £49 | £15 | models |
| SUV | £58 | £19 | models |

The SQL query which retrieved the data for this page was:
 SELECT * FROM CarTypes

CarModels page

1. Create a new CSHTML file called *carmodels.cshtml*. You should base the code for this page on the code you entered for the first page. You will need to change the SQL query, some text and the names of the database fields you display in the page, You don't need to have a hyperlink for every row displayed, so the HTML should be slightly simpler than the first page.

Your query should be defined like this:

```
@{
    var type = Request["description"];
    var db = Database.Open("gcucars");
    var selectQueryString =
        "SELECT * FROM CarModels WHERE carType='" + type + "'";
}
```

2. Add the following HTML after the table to provide a link back to the first page:

```
<p><a href="cartypes.cshtml">All types</a></p>
```

3. Save your file. Go to *cartypes.cshtml* in WebMatrix and run it. You should now see a page like the following when you click on one of the 'models' links.

GCUCars Car Models: Compact

| Make | Model | Engine Size | Engine Type |
|----------|-------|-------------|-------------|
| Vauxhall | Astra | 1.8 | petrol |
| Ford | Focus | 1.8 | petrol |

[All types](#)

The SQL query which retrieved the data for this page was:
 SELECT * FROM CarModels WHERE carType='Compact'

CarModels data entry page

As a final challenge, create a data entry form web page which allows new car types to be added to the database. You can base this on the example in lab 1.