Unit 33 Data Analysis & Design

User Documentation

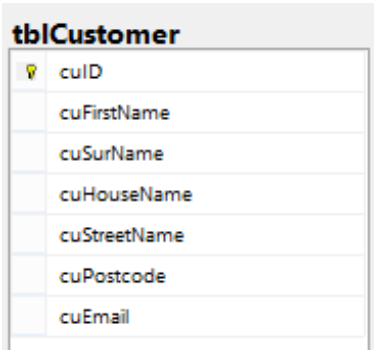
Aulayna MacLeod

Table of Contents

How Each Table is Used 2
tblCustomer 2
tblPostcode 2
tblBusiness 2
tblSale 2
tblPhone 2
How Queries are Used 3
Creating a Query 3
Customer Information Query 3
Customer Information from a Specific Town/City Query 4
Total Users Subscribed to a Specific Category Query 4

How Each Table is Used

tblCustomer



TblCustomer is used to store the data which directly related to the customer. This includes the customer ID, first name, surname, house name, street name, postcode, email address and which category the customer has chosen to follow.

The cuPostcode entity is linked to the tblPostcode this means that for a postcode to be allowed in this field it must first already exist within the tblPostcode table.

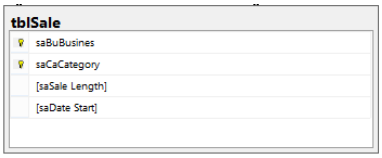
tblPostcode

TblPostcode is used to store the town and postcode. In order for tblCustomer to be able to use a postcode the postcode first must be entered into this table.

tblBusiness

This table is used to store the different businesses within the database. Before a business can appear in another table it must first appear within this table otherwise an error message will appear.

tblSale

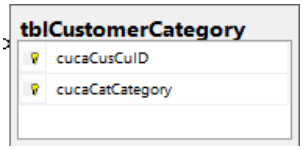


This tables details the information about any sale events. This includes the business, category, length of the sale and the start date of the sale. Before a business can appear in another table it must first appear within this table otherwise an error message will appear. For a category to be able to exist in the tblSale table it must first already exist within the tblCategory table.

tblPhone

This table stores the information related to the customers' phone numbers. coCuID is the customer ID coType is the type of phone number (landline or mobile) and coNumber is the phone number. coCuID is linked to the cuID field in the tblCustomer table if the customer ID does not already exist in the tblCustomer table then it will not be valid within this table.

tblCustomerCategory



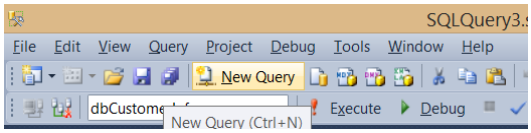
This table is used to link the customer table with the category table and is needed to make the database function effectively. For a customer ID and a category to exist in this table they must first be present in tblCustomer, for the customer ID, and in tblCategory for the category.

tblCategory

This table is used to store the different categories within the database. Before a category can appear in another table it must first appear within this table otherwise an error message will appear.

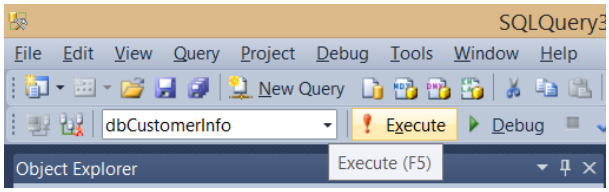
How Queries are Used

Creating a Query

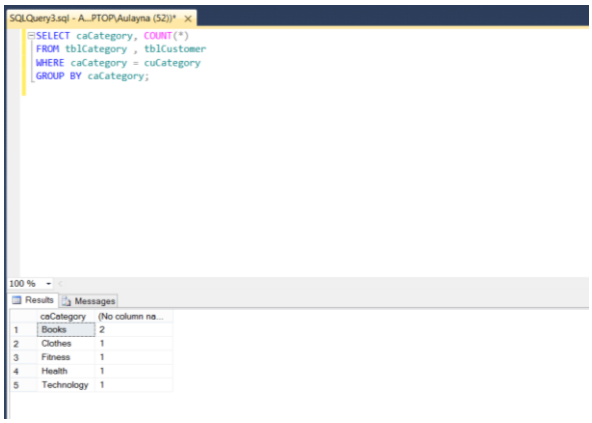


First a new query must be selected. This can be found on the menu bar or by using the shortcut Crtl+N.

A new query window will appear in which the query can be written. The basic language used to write a query in this application is called Transact-SQL.



Once the query has been written it can be executed by clicking the execute button or by using the shortcut F5.

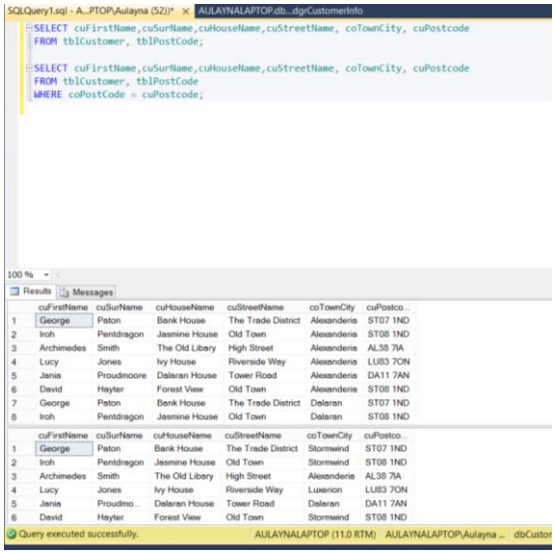


After a query has been executed successfully the results will appear underneath the query.

Customer Information Query

```
SELECT cuFirstName,cuSurName,cuHouseName,cuStreetName, coTownCity, cuPostcode
FROM tblCustomer, tblPostCode
WHERE coPostCode = cuPostcode;
```

This query will show a list of the information linked to a customer. It will take the information stored on the database which matches the fields used in the SELECT statement. The tables which store these fields must be used in the FROM statement. WHERE is used to define the clause of the query.



As the postcode present in both of the tables it will produce redundant data unless the command WHERE coPostCode = cuPostcode is used as this command will omit the redundant data from the result as it ensures that only postcodes which are supposed to be shown.

This screenshot shows two queries which have been executed. The first one omits the WHERE coPostCode = cuPostcode command and as a result postcodes which do not match the address are also shown (the redundant data) However the second query shows the WHERE coPostCode = cuPostcode command and only the correct postcodes are shown with the address.

Customer Information from a Specific Town/City Query

```
SELECT cuID, cuFirstName,cuSurName,cuHouseName,cuStreetName, coTownCity, cuPostcode,
cuEmail, coType ,coNumber
FROM tblCustomer, tblPostCode, tblPhone
WHERE coPostCode = cuPostcode AND coCuID =cuID AND coTownCity = 'Stormwind' ;
```

This query uses the same commands as the first query however it adds more elements to the query to make the results more precise. In addition to that more column names are added to the SELECT command. As some of the extra data needed is in another table the third table has to be added to the FROM section.

As before the WHERE command is needed to reduce redundant data however this time it is needed twice as there are two columns which are dependant on other columns for data. In addition to this the `coTownCity = 'Stormwind'` command is used to ensure that the results only display town/cities which match Stormwind.

Total Users Subscribed to a Specific Category Query

```
SELECT caCategory, COUNT(*)
FROM tblCategory , tblCustomer
WHERE caCategory = cuCategory
GROUP BY caCategory;
```

As mentioned above the SELECT command is used to select a column(s) from a table.

The COUNT command is used to find the total amount of times a value appears within the column of the table.

WHERE is used to define the clause of the query. In this case the value of both of the columns from different tables must be the same.

The GROUP BY command is used to sort the results in ascending order to make the results of the query look tidier.