



Getting Started with QueryCell

The following guide is designed to help new users get up and running with QueryCell.

This Guide is also available as a [PDF Document](#).

QueryCell is an add-in for Microsoft Excel that allows you to query Excel data using SQL.

Please contact us at support@querycell.com if you have any questions or suggestions about this guide.

Contents

1. [What is QueryCell in 2 minutes](#) - A brief Overview of QueryCell
2. [Installing QueryCell](#) - Getting QueryCell running on your computer
3. [The QueryCell Tutorial](#) - Your first SQL Select using QueryCell
4. [SQL Basics](#) - A short introduction to SQL Select Statements
5. [Getting Help](#) - Where to Next ?

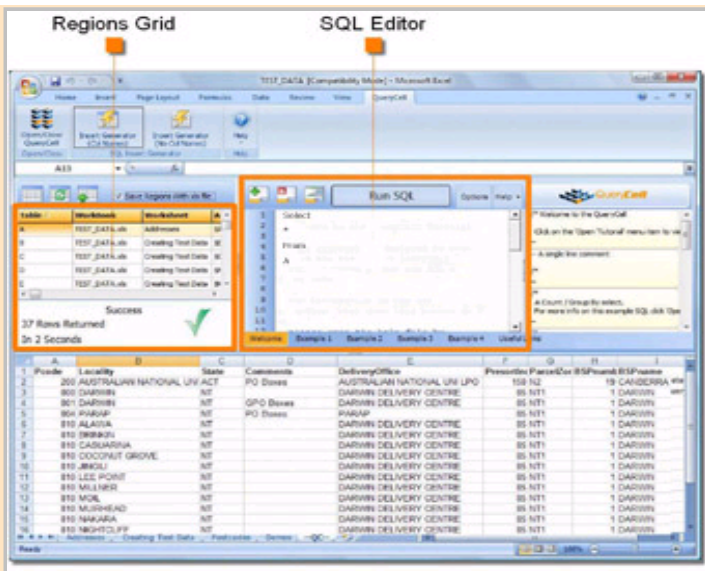
1. What is QueryCell in 2 minutes - A brief Overview of QueryCell

QueryCell is an Add-In for Microsoft Excel that allows you to query regions of data within Excel as though they were tables in a relational database.

QueryCell also allows you to generate SQL Insert Statements for Excel data in Seconds. [more information on SQL Insert Generation](#).

The Two most important parts of QueryCel are the Regions Grid and the SQL Editor.

(Click Image To Enlarge)



1.1 Regions Grid

This is where all the available data regions that you may query are listed.

The column titled 'Table' gives each region a table name, it is these table names you will use in your SQL queries. In the screenshot the regions have all been given single letter names : A,B,C,D etc.

1.2 SQL Editor

The SQL Editor is where you type in your SQL queries.

Simply enter your SQL query into the edit box and click the 'Run SQL' button above the edit box to execute the SQL.

1.3 A Couple of Example Queries

Below the edit box are tabs that allow you to switch between different SQL queries. In the example screenshot the tabs are labeled 'Welcome', 'Example 1', 'Example 2' etc.

You could select all the data from the first region by running the SQL statement

```

/*
this simple SQL Select retrieves all the rows in table A
*/

SELECT
*
FROM
A

```

for example.

You could run the SQL below to join the data in table a to table h based on the suburb/locality

```

/*

```

this SQL Select retrieves the rows in table a joined to table h where the suburb column in table a is the same as the locality column in table h

(using the Upper() function to convert the values to upper case before comparing)

```
*/  
SELECT  
  *  
FROM  
  a,  
  h  
WHERE  
  Upper(a.suburb) = Upper(h.locality)  
AND  
  h.comments is null
```

for example.

1.4 Generating SQL Inserts

QueryCell also provides you the ability to generate SQL insert statements for your data in seconds, for more information see the section [Generating SQL Inserts](#) and the FAQ Question [SQL Inserts](#).

[back to top](#)

2. Installing QueryCell - Getting QueryCell running on your computer

In order to get QueryCell running on your computer read the instructions below.

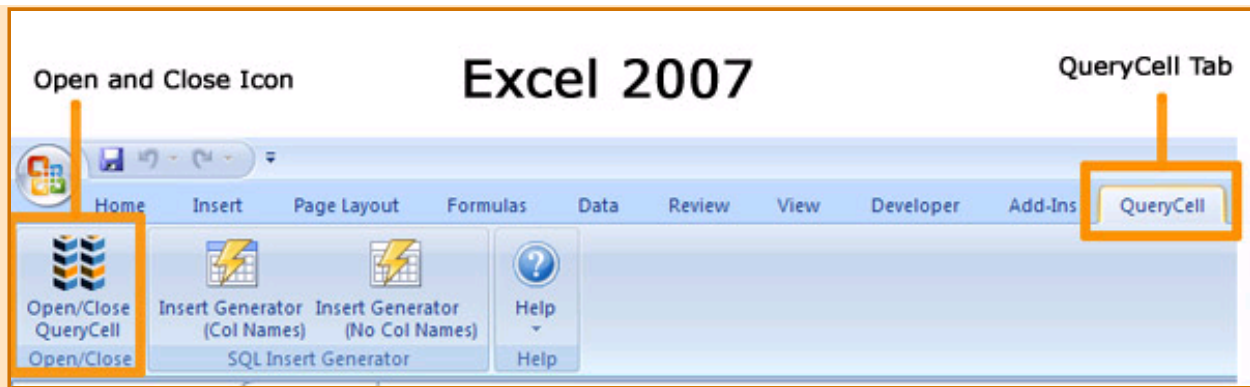
1 - Requirements. QueryCell requires Windows XP or Vista and Microsoft Excel 2003 or Microsoft Excel 2007.

2 - Download the QueryCell Installer. You can download QueryCell at the [QueryCell Try Now Page](#)

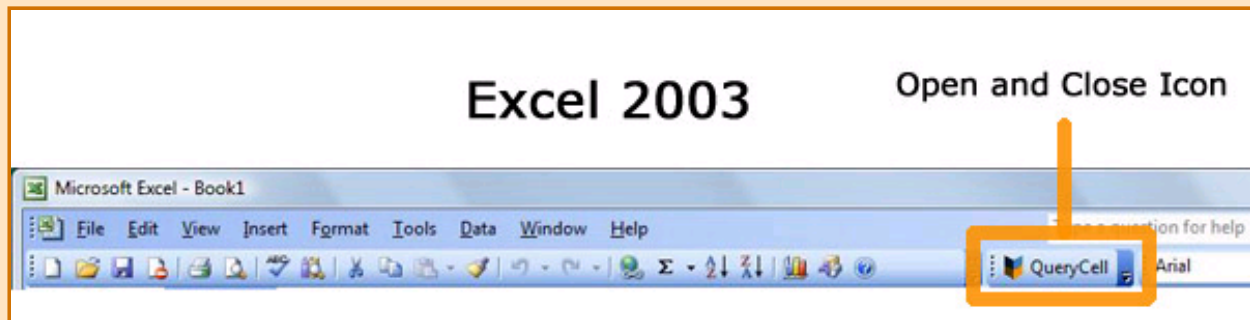
3 - Run the Installer. Close all running Microsoft Office Applications (word, excel, access, etc) and then run the QueryCell.exe file you have downloaded.

4 - Run QueryCell. Once the installation is complete open Microsoft Excel and you should see the QueryCell Icon on the Excel Menu. Clicking the QueryCell Icon will open QueryCell.

To open QueryCell in Microsoft Excel 2007 Click the QueryCell Tab and then the Open and Close Icon.



To open QueryCell in Microsoft Excel 2003 Click the QueryCell Tab and then the Open and Close Icon.



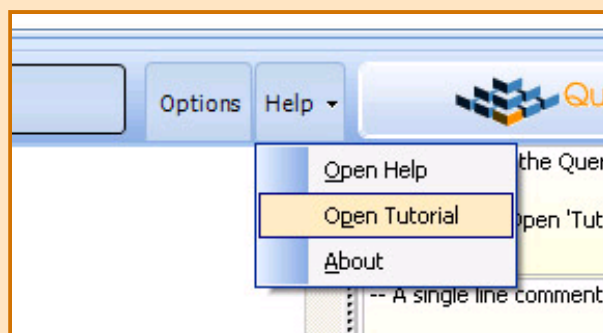
[back to top](#)

3. The QueryCell Tutorial - Your first SQL Select using QueryCell

QueryCell comes with a built in tutorial that demonstrates some simple SQL Select Statements.

When you open the QueryCell tutorial it will load an Excel workbook containing example data and load the SQL editor with SQL Select statements that can be run on that data.

To open QueryCell Tutorial Click the Help Tab within QueryCell and then click the 'Open Tutorial' Menu option.



Tabs

Welcome - Text Welcoming you to the tutorial

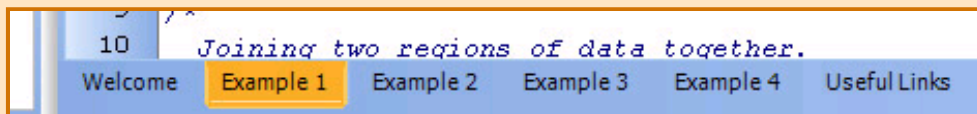
Example 1 - Joining two regions of data together.

Example 2 - A Count / Group By select.

Example 3 - Generating SQL Inserts From Excel Data.

Example 4- Generating Test Data.

Useful Links - Links to useful websites.



To get full value from this tutorial you can step through the SQL Editor tabs, reading the text. When you have selected tabs Example 1 - 4 you can click the 'Run SQL' button above the SQL editor to see the results of running the SQL.

[back to top](#)

4. SQL Basics - A short introduction to SQL Select Statements

While we don't intend to take on the job of teaching SQL, this brief introduction to SQL Select statements may be helpful in getting people new to SQL going.

SQL (Structured Query Language) is a powerful tool for manipulating data. In QueryCell you are able to use the SQL 'SELECT' statement to return data that meets a specific criteria, summarize data and join different regions of data together.

The SELECT statement is the mechanism in SQL to retrieve data (rather than add/insert or modify/update).

4.1 A Simple Select Statement

```
/*
this simple SQL Select retrieves the FIRST_NAME, LAST_NAME and
DATE_OF_BIRTH from the rows in table PERSON
*/

SELECT
  FIRST_NAME, LAST_NAME, DATE_OF_BIRTH
FROM
  PERSON
```

If table PERSON contained the data listed below the result-set returned from

```
/*
Contents of the PERSON table
*/
```

PERSON_ID	COUNTRY	FIRST_NAME	MIDDLE_NAME	LAST_NAME	CAPTAIN	DATE_OF_BIRTH
9001	Australia	Ricky	Thomas	Ponting	Y	12-19-1974
8997	England	Nasser		Hussain	Y	03-28-1968
8967	England	Darren		Gough	N	09-18-1970
8345	Sri Lanka	Ajantha	Winslo	Mendis	N	03-11-1985
45392	New Zealand	Nathan	Leslie	McCullum	N	09-01-1980
9006	West Indies	Vivian	Alexander	Richards	Y	03/07/1952

running the Select statement would be :

```
/*  
the result-set from running the SELECT statement above  
*/
```

Ricky	Ponting	12-19-1974
Nasser	Hussain	03-28-1968
Darren	Gough	09-18-1970
Ajantha	Mendis	03-11-1985
Nathan	McCullum	09-01-1980
Vivian	Richards	03/07/1952

Note that if the table PERSON or any of the columns selected (FIRST_NAME, LAST_NAME, DATE_OF_BIRTH) did not exist running the SELECT statements would result in an error message being displayed and no data returned.

Note If we wanted to select all the columns from the table we could either list them all or use the symbol * to denote 'all columns'.
eg. SELECT * FROM PERSON

4.2 Adding a WHERE clause

```
/*  
this simple SQL Select retrieves all the columns from the rows in table PERSON  
*/
```

```
SELECT  
*  
FROM  
PERSON  
WHERE  
CAPTAIN = 'Y'
```

the result from running the Select statement above would be :

```
/*  
the result-set from running the SELECT statement above  
*/
```

9001	Australia	Ricky	Thomas	Ponting	Y	12-19-1974
8997	England	Nasser		Hussain	Y	03-28-1968
9006	West Indies	Vivian	Alexander	Richards	Y	03/07/1952

Note Since we SELECTED * all columns from the table are returned.

Note Only Rows where the CAPTAIN columns contains Y are returned.

4.3 Joining Data

SQL provides ways to join data from different tables. This is very useful as it supports data normalization principle the idea of storing information in a single

place.

Above we gave an example PERSON table, suppose we also wanted to store information about the Country the people in the table played sport for (they are all international sports people).

If we wished to add the capital city and population of the country associated with each person we could add additional columns to the PERSON table but that would not obey the principle that we should store data information in a single place. This is because we can have more than one person from the same country. So for example the population and capital city of England would appear in both the row for Nasser Hussain and the row for Darren Gough.

The correct way to store this new data is in a separate table and then when we wish to list people with their country information to join the two tables together.

```
/*  
Contents of the COUNTRY_INFO table  
*/
```

COUNTRY	CAPITAL	POPULATION	TEAM_SHIRT_COLOR
Australia	Canberra	21,000,000	Gold
New Zealand	Wellington	4,000,000	Black
West Indies	No Capital (Federation)	3,000,000	Red
England	London	61,000,000	Blue
Sri Lanka	Colombo	20,000,000	Purple

```
/*  
this simple SQL Select retrieves rows in table PERSON joined to the  
COUNTRY_INFO table  
*/
```

```
SELECT  
PERSON.FIRST_NAME, PERSON.LAST_NAME, PERSON.DATE_OF_BIRTH  
COUNTRY_INFO.COUNTRY, COUNTRY_INFO.CAPITAL  
FROM  
PERSON, COUNTRY_INFO WHERE  
PERSON.COUNTRY = COUNTRY_INFO.COUNTRY
```

```
/*  
the result-set from running the SELECT statement joining PERSON and  
COUNTRY_INFO above  
*/
```

Ricky	Ponting	12-19-1974	Australia	Sydney
Nasser	Hussain	03-28-1968	England	London
Darren	Gough	09-18-1970	England	London
Ajantha	Mendis	03-11-1985	Sri Lanka	Colombo
Nathan	McCullum	09-01-1980	New Zealand	Wellington
Vivian	Richards	03/07/1952	West Indies	No Capital (Federation)

Note There are different types of joins in SQL. This is a simple 'Full Inner Join', it will only return rows where the COUNTRY value exists both in the PERSON and the COUNTRY_INFO table. To get started researching the other types of joining in SQL do a web search with the following keywords :
SQL Join Inner Outer Full

4.4 Using a Group By

Using a 'Group By' in SQL allows us to summarize data rather than retrieve individual rows.

The SQL below will give us a count of how many people are associated with each country.

```
/*  
this simple SQL Select counts records grouped by Country  
*/  
  
SELECT  
    PERSON.COUNTRY, COUNT(*)  
FROM  
    PERSON  
GROUP BY PERSON.COUNTRY
```

```
/*  
the result-set from running the SELECT statement above  
*/
```

Australia	1
England	2
West Indies	1
Sri Lanka	1
New Zealand	1

4.5 Using an Order By

In SQL an 'Order By' clause allows you to sort the order in which data is returned.

The SQL below will return records from the PERSON table ordered by the DATE_OF_BIRTH of the person in descending order. By descending order we mean the person with the greatest DATE_OF_BIRTH (most recent) will be listed first.

```
/*  
this simple SQL Select returns all rows from the PERSON table from youngest to  
oldest  
*/  
  
SELECT
```

```
*  
FROM  
PERSON  
ORDER BY PERSON.DATE_OF_BIRTH DESC
```

```
/*  
Results of the Order By Query  
*/
```

PERSON_ID	COUNTRY	FIRST_NAME	MIDDLE_NAME	LAST_NAME	CAPTAIN	DATE_OF_BIRTH
8345	Sri Lanka	Ajantha	Winslo	Mendis	N	03-11-1985
45392	New Zealand	Nathan	Leslie	McCullum	N	09-01-1980
9001	Australia	Ricky	Thomas	Ponting	Y	12-19-1974
8967	England	Darren		Gough	N	09-18-1970
8997	England	Nasser		Hussain	Y	03-28-1968
9006	West Indies	Vivian	Alexander	Richards	Y	03/07/1952

Note The rows are ordered based on the DATE_OF_BIRTH column from greatest to least.

Note If we wanted to change the ordering, to list the oldest person first and youngest person last we could simply change the 'DESC' in the SQL query (Descending) to 'ASC' (Ascending).

[back to top](#)

5. Getting Help - Where to Next ?

5.1 More information about QueryCell and how it works

To learn more about QueryCell and how it works check out :

The [QueryCell Website](#)

[Selectable - The QueryCell Blog](#)

The QueryCell help file. Once QueryCell is open clicking the Help tab and then clicking 'Open Help' will launch the Help File for QueryCell. This help file is the place to look when you want to learn the specifics of how QueryCell works such as what particular icons and buttons do.

Feel free to email us as support@querycell.com with your questions.

5.2 Help With SQL

The Internet is probably the best place to look for help with SQL. There are a large number of freely available articles and tutorials out there. It's worth noting that QueryCell uses the Firebird database engine. While Much of SQL is standardised and the same everywhere there are specific extensions for different

implementations. If you're wondering if QueryCell supports a SQL extension you should first research if Firebird supports the extension.

Rather than give specific website recommendations without knowing what area of SQL you need help with here are some phrases that you might like to put into a search engine (such as google), they should yeild helpful results

"SQL help" "SQL tutorial" "SQL Introduction" "SQL SELECT" "SELECT STATMENTS"
"Inner Join" "Outer Join" "Group By" "Order By" "Count(*)" "SQL Query" "Relational Theory" "Normalisation"

The [StackOverflow Website](#) has a large collection of SQL question and answers.

5.3 Help With Excel

As with SQL there are a large number of freely available resources on the web to help with your Excel Questions.

I would recommend the forums at www.excelforum.com/ and [MrExcel](#) as great places to look for help with Microsoft Excel.

[back to top](#)

[Home](#) [Learn More](#) [Buy Now!](#) [Try Now](#) [Support](#) [About](#)

Copyright Oak Focus Software / QueryCell 2009 (All Rights Reserved).

Excel is a trademark of Microsoft Corporation.

The QueryCell Product and it's website are not associated with Microsoft Corporation