

# Queries



February 27, 2014

© Guidewire Software, Inc. 2001-2014. All rights reserved.  
Do not distribute without permission.

## Lesson objectives

- By the end of this lesson, you should be able to:
  - Use Gosu queries to retrieve sets of objects from the database that cannot be accessed through an existing array

This lesson uses the notes section for additional explanation and information.

To view the notes in PowerPoint, choose View→Normal or View→Notes Page.

If you choose to print the notes for the lesson, be sure to select "Print hidden slides."

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

2

G U I D E W I R E

## Lesson outline

- Gosu query basics
- Working with queries
- Working with result sets

## Gosu queries

- A **Gosu query** is an object associated with a specific entity that stores a database query and its results
  - Gosu queries are useful when code must work with a set of objects that does not exist as an array, such as "all contact notes created by this user"

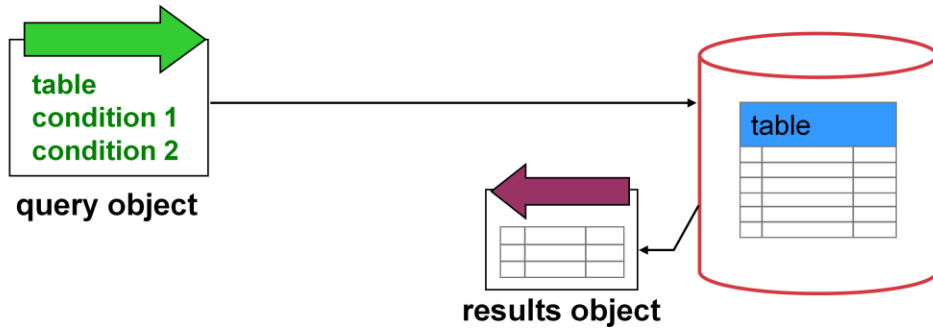
Gosu also supports find expressions, which can also query databases. Guidewire recommends that all new queries be written as Gosu queries, however, because Gosu queries have the following advantages:

They more readily support the construction of complex queries.

They use builder patterns, which simplify the creation of search screens.

They use a syntax more intuitive to people accustomed to writing queries in Structured Query Language (SQL).

# Objects used when querying database



- Query object stores criteria of query, such as:
  - Which entity to query
  - What restrictions (where clauses) to apply
- Results object stores results and related info:
  - How many results?
  - If necessary, how should they be ordered?

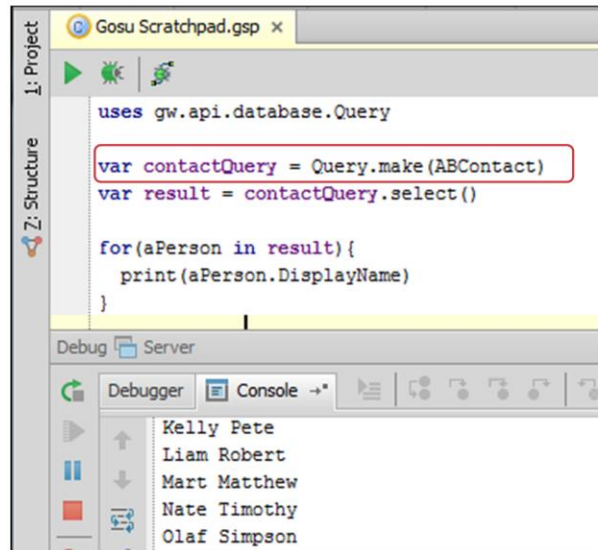
## Steps to execute a basic query

1. Create the query object using the make() method
2. Create the results object using the select() method
3. Process the results of the query as needed

## Step 1: Create the query object

- Syntax:

```
var queryObj =  
    gw.api.database.Query.make(EntityToQuery)
```



The screenshot shows the Gosu Scratchpad IDE. The main editor window displays the following code:

```
uses gw.api.database.Query  
  
var contactQuery = Query.make(ABContact)  
var result = contactQuery.select()  
  
for(aPerson in result){  
    print(aPerson.DisplayName)  
}
```

The line `var contactQuery = Query.make(ABContact)` is highlighted with a red box. Below the editor, the 'Console' tab is active, showing the output of the program:

```
Kelly Pete  
Liam Robert  
Mart Matthew  
Nate Timothy  
Olaf Simpson
```

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

7

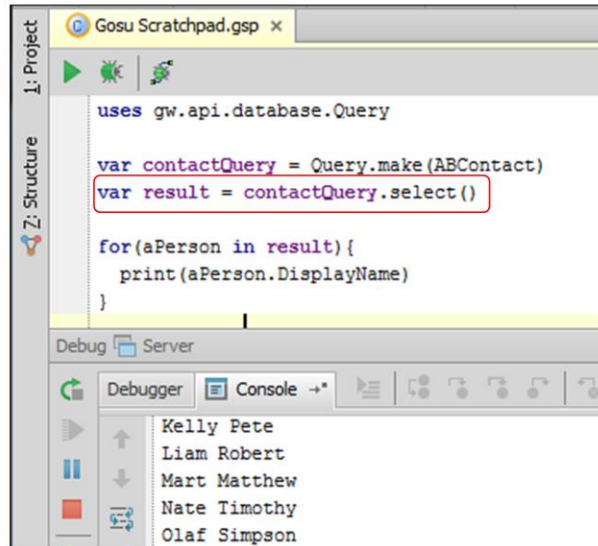
G U I D E W I R E

The make method is a method of the Query class. This class is declared in the gw.api.database package. A package is a collection of classes grouped together for convenience or because fields or methods in those classes should have access to one another while instances of classes outside of the package should not, or both.

## Step 2: Create the results object

- Syntax:

`var resultsObj = queryObj.select()`



You do not always need to create an explicit results object. The code shown in the screenshot above produces the same results if written like this:

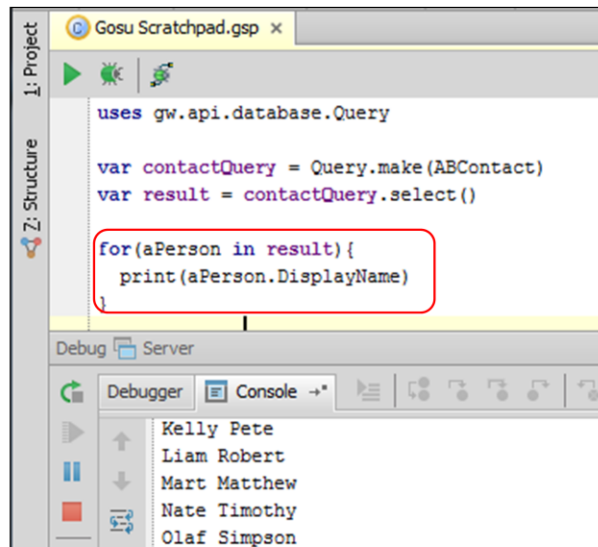
```
var policyPersonQuery = gw.api.database.Query.make(ABPolicyPerson)
for (aPolicyPerson in policyPersonQuery.select()) {
    print (i + ": " + aPolicyPerson.DisplayName)
}
```

The examples in this lesson typically show an explicit result set object being created to make the code in the examples more readable and to make it clear which object has the method in question.



## Step 3: Process results of query as needed

- You can use a for loop to iterate through the result set as if it were an array



The screenshot shows the Gosu Scratchpad IDE with a file named 'Gosu Scratchpad.gsp'. The code in the editor is as follows:

```
uses gw.api.database.Query

var contactQuery = Query.make(ABContact)
var result = contactQuery.select()

for(aPerson in result){
    print(aPerson.DisplayName)
}
```

The for loop is highlighted with a red box. Below the code editor, the 'Console' tab is active, displaying the output of the program:

```
Kelly Pete
Liam Robert
Mart Matthew
Nate Timothy
Olaf Simpson
```

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

9

G U I D E W I R E

Line 4 in the code above prints the display name of the current policy person. In production code, this would typically be achieved by writing "print (aPolicyPerson)". References to an object inherently return the object's display name. The ".DisplayName" is included in the code above solely to make the code more readable for instructional purposes.

Recall that the for loop can have an optional index variable that can be used to display the number of each result. For example, the following code...

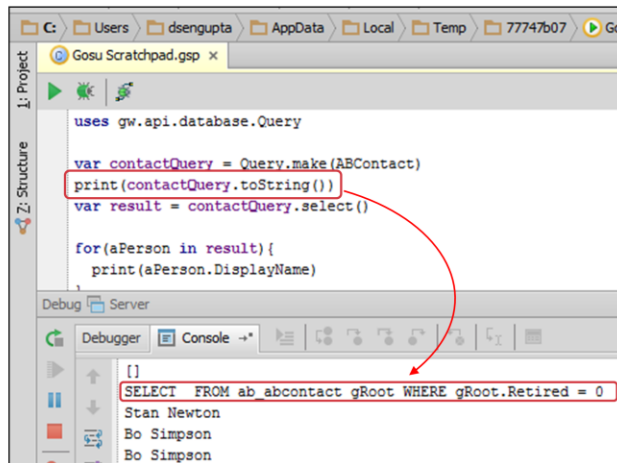
```
for (aPolicyPerson in resultSet index i) {
    print (i + ": " + aPolicyPerson.DisplayName)
}
```

...produces the following output...

- 1: Alex Newton
- 2: Bert Simpson
- 3: Charles Simpson

# Viewing approximation of SQL query

- Syntax: `queryObj.toString()`
  - Actual SQL may vary based on the RDBMS



© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

10

G U I D E W I R E

The query object's `toString` method provides an approximation of the SQL sent to the database to execute the query. These results are an approximation because the actual SQL may have vendor-dependent variations or vendor-dependent optimizations or both. A given query may have slight syntactic variations depending on whether it is used for an Oracle database or a SQL Server database.

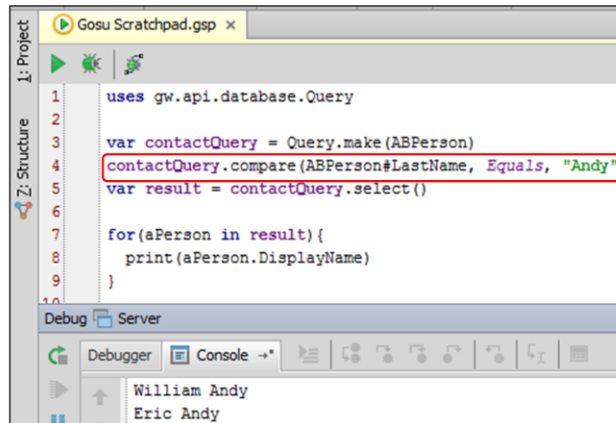
You can record the exact SQL query in the logs using the `withLogSQL` method on a query. The method takes a single boolean argument that, when set to true, logs the query in the system logs in the logging category `Server.Database`.

## Lesson outline

- Gosu query basics
- Working with queries
- Working with result sets

# Restricting queries

- Query objects have methods that add restrictions to query
  - Typically, they become SQL where clause



The screenshot shows the Gosu Scratchpad IDE with a file named 'Gosu Scratchpad.gsp'. The code is as follows:

```
1 uses gw.api.database.Query
2
3 var contactQuery = Query.make(ABPerson)
4 contactQuery.compare(ABPerson#LastName, Equals, "Andy")
5 var result = contactQuery.select()
6
7 for(aPerson in result){
8     print(aPerson.DisplayName)
9 }
10
```

The line `contactQuery.compare(ABPerson#LastName, Equals, "Andy")` is highlighted with a red box. Below the code editor, the 'Debug' tab is active, showing the 'Console' output with the following text:

```
William Andy
Eric Andy
```

## Restricting queries: compare method

- Syntax:

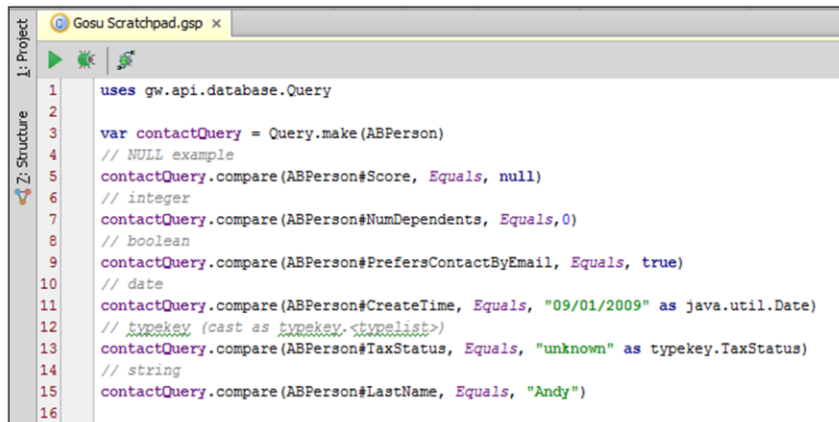
`queryObj.compare("field", operator, value)`

- Field and value must be strings
- Valid operators include:
  - Equals
  - NotEquals
  - LessThan
  - LessThanOrEquals
  - GreaterThan
  - GreaterThanOrEquals

```
4 contactQuery.compare(ABPerson#LastName, Equals, "Andy")
5 contactQuery.compare(ABPerson#FirstName, Equals, "William")
```

## Specifying values in compare method

- For strings, use quotes
- For datetime and typekey, use quotes and cast object



```
1 uses gw.api.database.Query
2
3 var contactQuery = Query.make(ABPerson)
4 // NULL example
5 contactQuery.compare(ABPerson#Score, Equals, null)
6 // integer
7 contactQuery.compare(ABPerson#NumDependents, Equals, 0)
8 // boolean
9 contactQuery.compare(ABPerson#PrefersContactByEmail, Equals, true)
10 // date
11 contactQuery.compare(ABPerson#CreateTime, Equals, "09/01/2009" as java.util.Date)
12 // typekey (cast as typekey.<typelist>)
13 contactQuery.compare(ABPerson#TaxStatus, Equals, "unknown" as typekey.TaxStatus)
14 // string
15 contactQuery.compare(ABPerson#LastName, Equals, "Andy")
16
```

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

14

G U I D E W I R E

When using the compare method with a date value, you must cast the value as a date value. Otherwise, Guidewire treats the value as a string and it throws a type mismatch exception. To cast a value as a date value, use the syntax: "<datevalue>" as java.util.Date.

When using the compare method with a typekey value, you must cast the value as a typecode from the appropriate typelist. Otherwise, Guidewire treats the value as a string and it throws a type mismatch exception. To cast a value as a typecode from the appropriate typelist, use the syntax: "<typecode>" as typekey.<typelist>.

## Null values for query restrictions

```
1 uses gw.api.database.Query
2 uses gw.api.database.Relop
3 var queryNull= Query.make(ABContact)
4 var queryNotNull= Query.make(ABContact)
5 // Contacts where Score IS NULL
6 queryNull.compare(ABContact#Score, Relop.Equals, null)
7 // Contact where Score IS NOT NULL
8 queryNotNull.compare(ABContactScore, Relop.NotEquals, null)
```

- Entities can have elements that allow for null values
  - If there is no value, the database leaves the field as null
- Restriction queries can compare for null or not null values

gw.api.database.Query	ANSI SQL
compare(entity.Property, Equals, null)	where table.columnName IS NULL
compare(entity.Property, NotEquals, null)	where table.columnName IS NOT NULL

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

15

GUIDEWIRE

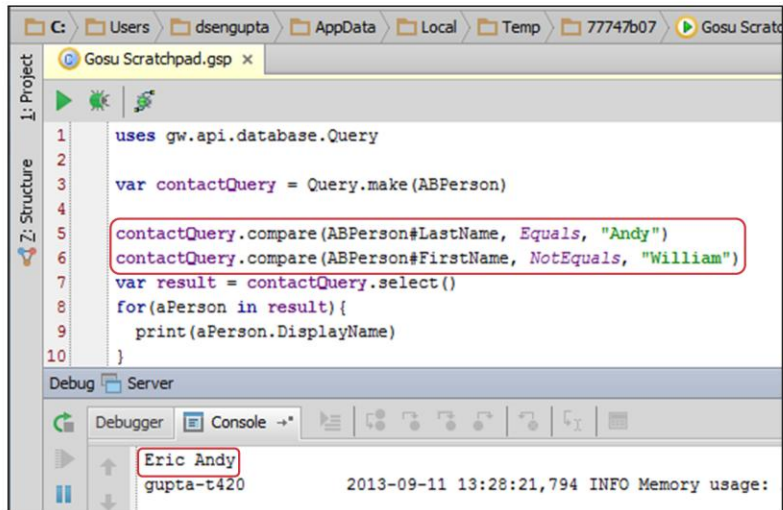
A null value signifies the absence of or void of a value.

In Guidewire applications, there are some data model entity elements that specify a nullok attribute. Column, foreignkey, onetoone, and typekey are elements that have the nullok attribute. When specified in the as an attribute of an element, you must define the nullok attribute (8.0.0).

Guidewire applications transform an entity element with a nullok=true attribute to a column definition in a physical database table. Relational databases create database tables using a form of Structure Query Language (SQL) known as Data Definition Language (DDL). With a nullok=false attribute, a Guidewire application executes a SQL / DDL statement to create a column constraint that specifies that the named column cannot contain a null value.

## Multiple restrictions ANDed together

- Query can have as many restrictions as needed
  - Restrictions are inherently ANDed together



The screenshot shows the Gosu IDE interface. The main editor displays a Gosu script file named 'Gosu Scratchpad.gsp'. The script contains the following code:

```
1 uses gw.api.database.Query
2
3 var contactQuery = Query.make(ABPerson)
4
5 contactQuery.compare(ABPerson#LastName, Equals, "Andy")
6 contactQuery.compare(ABPerson#FirstName, NotEquals, "William")
7 var result = contactQuery.select()
8 for(aPerson in result){
9     print(aPerson.DisplayName)
10 }
```

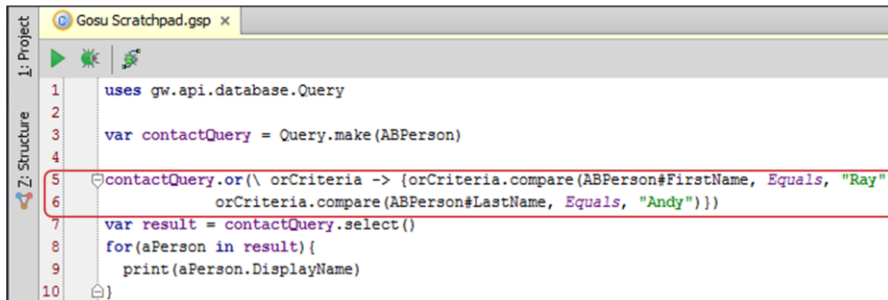
The two `compare` method calls on lines 5 and 6 are highlighted with a red rectangle, illustrating how multiple restrictions are ANDed together. Below the editor, the 'Debug' tab is active, showing the 'Console' output. The output displays the result of the query: 'Eric Andy' and 'gupta-t420', with the first line also highlighted by a red rectangle. The console also shows a timestamp and memory usage: '2013-09-11 13:28:21,794 INFO Memory usage:'.



## Multiple restrictions ORed together

- The or() method requires a block that specifies criteria to "OR" together:

```
queryObj.or( \ placeholder -> {  
    placeholder.criteria  
    placeholder.criteria // add as many criteria as needed  
})
```



© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

17

G U I D E W I R E

You can create a query that consists of multiple conditions that are ORed together. To do this, you must use the `or()` method. It takes a block as an input parameter. The block consists of a placeholder that represents the original query object. You can then list multiple criteria, all of which are ORed together. You can use any sort of criteria, including the comparison criteria discussed earlier in the lesson.

You can also combine conditions with AND and OR logic in complex ways. The following example finds all policy people whose first name is "John" or whose name is "Erica Hinds" (first name is Erica and last name is Hinds).

```
policyPersonQuery.or( \ orCriteria -> {  
    orCriteria.compare("FirstName", Equals, "John")  
    orCriteria.and( \ andCriteria -> {  
        andCriteria.compare("FirstName", Equals, "Erica")  
        andCriteria.compare("LastName", Equals, "Hinds")  
    })  
})
```

## Additional restriction options

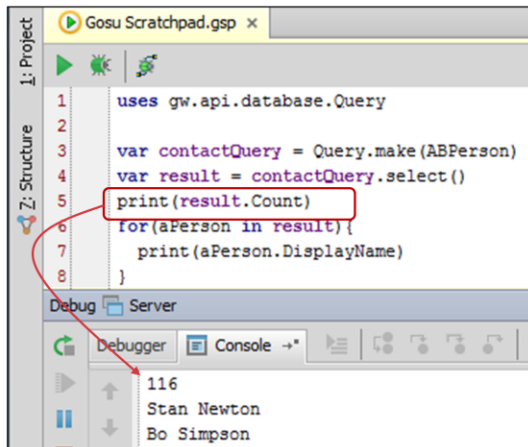
- Gosu Reference Guide contains complete list of all options for restricting queries, including restrictions that make use of:
  - between and Ranges
  - startsWith
  - contains

## Lesson outline

- Gosu query basics
- Working with queries
- Working with result sets

## Getting size of results

- Syntax: `resultsObj.Count`
  - Value will vary if number of rows in database that meet query's criteria changes



The screenshot shows the Gosu Scratchpad IDE with a project named 'Gosu Scratchpad.gsp'. The code editor displays the following Gosu code:

```
1 uses gw.api.database.Query
2
3 var contactQuery = Query.make(ABPerson)
4 var result = contactQuery.select()
5 print(result.Count)
6 for(aPerson in result){
7     print(aPerson.DisplayName)
8 }
```

The line `print(result.Count)` is highlighted with a red box. Below the code editor, the 'Server' tab is active, showing the output of the query:

```
116
Stan Newton
Bo Simpson
```

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

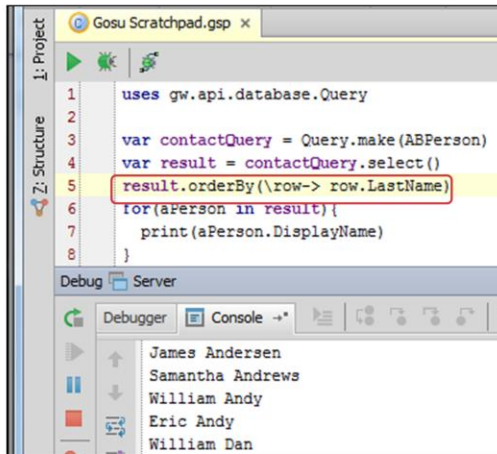
20

G U I D E W I R E

The Count attribute identifies the number of objects in the result set. Developers should keep in mind that Guidewire does not query the database for the complete result set, however, and does not store a snapshot of the results of a given query. Therefore, the value of Count can vary if the number of rows in the database that meet the query's criteria changes while the query results are being processed. The field is introduced here as it is useful for instructional and debugging purposes, but it can be problematic if used in implementation code as an absolutely accurate reflection of the number of rows in the database that meet the query's criteria. For a further discussion of this topic, refer to the *Gosu Reference Guide*.

## Sorting results

- Each sorting methods takes a block as an argument
  - Syntax:  
`resultsObj.orderBy( \ row -> row.FieldName)`  
`resultsObj.orderByDescending( \ row -> row.FieldName )`



The screenshot shows the Gosu Scratchpad.gsp editor with a project structure on the left. The main editor displays the following code:

```
1 uses gw.api.database.Query
2
3 var contactQuery = Query.make(ABPerson)
4 var result = contactQuery.select()
5 result.orderBy(\row-> row.LastName)
6 for(aPerson in result){
7     print(aPerson.DisplayName)
8 }
```

The console output at the bottom shows the results of the query, sorted by last name:

```
James Andersen
Samantha Andrews
William Andy
Eric Andy
William Dan
```

© Guidewire Software, Inc. 2001-2014. All rights reserved. Do not distribute without permission.

21

G U I D E W I R E

The two methods listed above sort the entire result set according to the field name in either ascending (`orderBy`) or descending (`orderByDescending`) order. There are two other methods, as well: `thenBy` and `thenByDescending`. These methods do not re-sort the entire result set. Instead, they preserve any existing ordering and sort rows that tied during the first sort. For example, if you wanted to sort by last name and then by first name, you would use the following code:

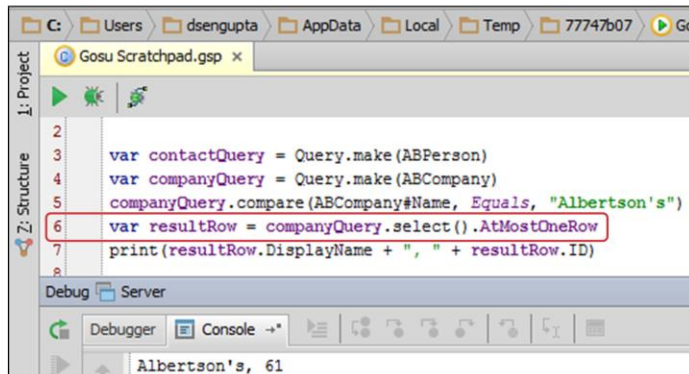
```
resultSet.orderBy(\row -> row.LastName)
resultSet.thenBy(\row -> row.FirstName)
```

# Querying when only one result is expected

- Syntax:

`queryObj.select().AtMostOneRow`

- If single item exists, method returns that single row
- If no item exists, method returns null
- If multiple items exist, method throws exception

A screenshot of a code editor window titled 'Gosu Scratchpad.gsp'. The code is as follows:

```
2  
3 var contactQuery = Query.make(ABPerson)  
4 var companyQuery = Query.make(ABCompany)  
5 companyQuery.compare(ABCompany#Name, Equals, "Albertson's")  
6 var resultRow = companyQuery.select().AtMostOneRow  
7 print(resultRow.DisplayName + ", " + resultRow.ID)  
8
```

The line `var resultRow = companyQuery.select().AtMostOneRow` is highlighted with a red box. Below the code editor, there is a 'Debug' section with a 'Server' tab. The 'Console' output shows the text 'Albertson's, 61'.

## Lesson objectives review

You should now be able to:

- Use Gosu queries to retrieve sets of objects from the database that cannot be accessed through an existing array

## Review questions

1. For each of the following, identify the reason you would use it and whether it can be found on the query object or on the results object.
  - a) The select method
  - b) The compare method
  - c) The Count field
  - d) The orderby method

### Answers

1.
  - a) This method produces the results of the query. It is on the query object.
  - b) This method creates a condition that restricts the results of the query. It is on the query object.
  - c) This field identifies the number of results in the query. It is on the results object.
  - d) This method sorts the results. It is on the results object.



# Notices

**Copyright © 2001-2014 Guidewire Software, Inc. All rights reserved.**

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Product Designer, Guidewire Live, Guidewire DataHub, Guidewire InfoCenter, Guidewire Standard Reporting, Guidewire ExampleCenter, Guidewire Account Manager Portal, Guidewire Claim Portal, Guidewire Policyholder Portal, ClaimCenter, BillingCenter, PolicyCenter, InsuranceSuite, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries.

All other trademarks are the property of their respective owners.

**This material is confidential and proprietary to Guidewire and subject to the confidentiality terms in the applicable license agreement and/or separate nondisclosure agreement.**

This file and the contents herein are the property of Guidewire Software, Inc. Use of this course material is restricted to students officially registered in this specific Guidewire-instructed course, or for other use expressly authorized by Guidewire. Replication or distribution of this course material in electronic, paper, or other format is prohibited without express permission from Guidewire.

Guidewire products are protected by one or more United States patents.