

# Entity and Entity Selection Functions

Please send corrections and suggestions to [cannon@synergfarmsolutions.com](mailto:cannon@synergfarmsolutions.com). (v3)

Entity and entity selections are both declared as C.OBJECT. Many entity selection functions return an entity selection, but you can assign it to itself.

\$esExample:=\$esExample.query(...). Entities in an entity selection are lazy loaded for optimization (speed and memory) purposes. Note that when we say an entity or entity selection is returned, it is the reference that is returned.

The chart below attempts to group related function across data classes, entity selections, and entities into themes.

Even though several functions may be grouped in the same theme, keep in mind that each function must operate on the appropriate type of object. The colored bars are an attempt to visually identify function "pairs".

## Terminology:

Datastore = database  
 Dataclass = table  
 Attribute = field  
 Relation Attribute = relation  
 Entity Selection = selection  
 Entity = record

Don't forget about the new For each syntax:

**For each ( Object ; Collection { ; begin { ; end } } ) { Until | While } ( Boolean\_Expression )**

Can iterate through collections, entity selections, and object properties.

Query String Syntax (from `dataClass.query()` documentation)

**attributePath comparator value {logicalOperator attributePath comparator value}**

## Comparator

Comparison	Symbol(s)	Comment
Equal to	=, ==	Gets matching data, supports the wildcard (@), neither case-sensitive nor diacritic.
	===, IS	Gets matching data, considers the @ as a standard character, neither case-sensitive nor diacritic
Not equal to	#, !=	Supports the wildcard (@)
	!==, IS NOT	Considers the @ as a standard character
Less than	<	
Greater than	>	
Less than or equal to	<=	
Greater than or equal to	>=	
Included in	IN	Gets data equal to at least one of the values in a collection or in a set of values
Not condition applied on a statement	NOT	Parenthesis are mandatory when NOT is used before a statement containing several operators
Contains keyword	%	Keywords can be used in attributes of string or picture type

## Logical Operator

Conjunction	Symbol(s)
AND	&, &&, and
OR	,   , or

## Attribute Path

### Examples

```
"country = :1" ; "Canada"
"country.name = :1" ; "Canada"
"countries[].name = :1" ; "Canada"
```

## Meta Information

	Result Type	Similar To
<code>dataClass.getDataStore () -&gt; Datastore</code> Returns the datastore the dataclass belongs to.	Datastore (Object)	
<code>entitySelection.getDataClass () -&gt; Dataclass</code> Returns the dataclass the entity selection belongs to.	Dataclass (Object)	Table name
<code>dataClass.getInfo () -&gt; Object</code> Returns an object with information about the dataclass. The object will have the following keys: name, primaryKey, tableNumber.	Object	

## Working With Entity Selections

### Creating and Searching Entity Selections

	Result Type	Similar To
<code>dataClass.newSelection ( {dk_keep_ordered or dk_non_ordered} ) -&gt; Entity Selection</code> Returns an empty entity selection.	Entity Selection	
<code>dataClass.all ( {settings} ) -&gt; Entity Selection</code> Returns an entity selection with all the entities in the dataclass.	Entity Selection	ALL RECORDS
<code>dataClass.query ( queryString { ; value{ ; value2 ; ... ; valueN}{ ; querySettings} ) -&gt; Entity Selection</code> Returns an entity selection with entities from the dataclass that match the query string.	Entity Selection	QUERY
<code>entitySelection.query ( queryString { ; value{ ; value2 ; ... ; valueN}{ ; querySettings} ) -&gt; Entity Selection</code> Same as dataclass.query, except that it searches within the original entity selection.	Entity Selection	QUERY SELECTION
<code>entitySelection.slice ( startFrom { ; end} ) -&gt; Entity Selection</code> Returns a range of entities. Note the entity at index end is excluded.	Entity Selection	

### Sorting

<code>entitySelection.orderBy ( criteria ) -&gt; Entity Selection</code> Returns a sorted entity selection. Criteria may be a text string or a collection. Text string syntax: "attributePath1 {desc or asc}, attributePath2 {desc or asc},..." Collection syntax: [{"propertyPath": "...", "descending": false}, {...}]	Entity Selection	ORDER BY
<code>entitySelection.isOrdered () -&gt; Boolean</code> Returns true if the entity selection is ordered.	Boolean	

### Set Math

<code>entitySelection.and ( entity   entitySelection ) -&gt; Entity Selection</code> Returns a new, unordered entity selection that contains only entities that are in both entity selections.	Entity Selection	INTERSECTION
<code>entitySelection.or ( entity   entitySelection ) -&gt; Entity Selection</code> Returns a new, unordered entity selection which contains all the entities from both entity selections.	Entity Selection	UNION
<code>entitySelection.minus ( entity   entitySelection ) -&gt; Entity Selection</code> Returns a new, unordered entity selection which contains all the entities from the original entity selection minus the entities in the second entity selection.	Entity Selection	DIFFERENCE

### Math

<code>entitySelection.count ( attributePath ) -&gt; Real</code> Returns the number of non-null, non-object, and non-collection entities at the attribute path.	Real	Records in selection
<code>entitySelection.sum ( attributePath ) -&gt; Real</code> Returns the sum of all numeric values found at the attribute path. Attribute path cannot be a related attribute.	Real	Sum
<code>entitySelection.average ( attributePath ) -&gt; Real</code> Returns the average of all numeric values found at the attribute path. Attribute path cannot be a related attribute.	Real	Average
<code>entitySelection.min ( attributePath ) -&gt; Mixed</code> Returns the lowest of all values found at the attribute path. Attribute path cannot be a related attribute.	Mixed. Depends on the type at attributePath.	Min
<code>entitySelection.max ( attributePath ) -&gt; Mixed</code> Returns the highest of all values found at the attribute path. Attribute path cannot be a related attribute.	Mixed. Depends on the type at attributePath.	Max

Conversion With Collections		
entitySelection.toCollection ( {filter ;} { options {; begin {; howMany}} } ) -> Collection Creates a collection of objects from the entity selection.	Collection	
dataClass.fromCollection ( objectCol {; settings} ) -> Entity Selection Updates entities directly in the dataclass based on objects in the collection. New entities can be created in the dataclass.	Entity Selection	
entitySelection.distinct ( attributePath {; dk diacritical} ) -> Collection Creates a collection which contains the distinct non-null values at the attribute path. The collection will be sorted.	Collection	DISTINCT VALUES
Entities and Entity Selections		
	Result Type	Similar To
Inspection		
entity.getSelection () -> Entity Selection Returns the entity selection the entity belongs to.	Entity Selection	
entitySelection.contains ( entity ) -> Boolean Returns true if the entity exists in the entity selection.	Boolean	
entity.indexOf ( {entitySelection} ) -> Long Returns the index of the entity in the entity selection. -1 is returned if it does not exist in the entity selection.	Long	Selected record number
Moving Between Entities		
entitySelection.first () -> Entity Returns the first entity in the entity selection. Returns Null if the entity selection is empty.	Entity	FIRST RECORD
entitySelection.last () -> Entity Returns the first entity in the entity selection. Returns Null if the entity selection is empty.	Entity	LAST RECORD
entity.first () -> Entity Returns the first entity in the entity selection the original entity belongs to. Returns null if the entity does not belong to an entity selection.	Entity	FIRST RECORD
entity.previous () -> Entity Returns the previous entity in the entity selection the original entity belongs to. Returns null if the entity does not belong to an entity selection or if there isn't a previous entity (dropped entities are automatically skipped).	Entity	PREVIOUS RECORD
entity.next () -> Entity Returns the next entity in the entity selection the original entity belongs to. Returns null if the entity does not belong to an entity selection or if there isn't a next entity (dropped entities are automatically skipped).	Entity	NEXT RECORD
entity.last () -> Entity Returns the last entity in the entity selection the original entity belongs to. Returns null if the entity does not belong to an entity selection.	Entity	LAST RECORD
Adding and Removing		
entitySelection.add ( entity ) Adds an entity to the entity selection. Where it is added is not defined for unordered entity selections and at the end for ordered entity selections. Note that this doesn't mean the entity is saved into the dataclass!	Mutated	
entitySelection.drop ( {dk stop dropping on first error} ) -> Entity Selection Deletes the entities in the entity selection from the dataclass. If any entities could not be deleted, they will be returned in the new entity selection. If there were no errors, the returned entity selection will be empty. Note the original entity selection still exists in memory.	Entity Selection	DELETE SELECTION
Working With Entities		
	Result Type	Similar To
Basic Operations		
dataClass.new () -> Entity Creates a new entity with blank values in memory.	Entity	CREATE RECORD
entity.clone () -> Entity Creates a new entity referencing the same record as the original record. The two entities do not have the same reference and can independently change their values. However, they both point back to the same record in the database. This is not for duplicating a record, but for getting an independent reference to the same record.	Entity	
entity.save ( {dk auto merge} ) -> Object Saves the entity to the dataclass. Nothing happens if no attributes have been touched. See return object for success or errors.	Object	SAVE RECORD
entity.drop ( {dk force drop if stamp changed} ) -> Object Delete's the entity from the dataclass. The entity remains in memory. See return object for success or errors.	Object	DELETE RECORD
entity.lock ( {dk reload if stamp changed} ) -> Object Locks the entity, including all references to the entity, in the process. See return object for success or errors.	Object	READ WRITE
entity.unlock () -> Object Unlocks the entity, including all references to the entity. See return object for success or errors.	Object	READ ONLY
entity.reload () -> Object Reloads the entity from the dataclass. See return object for success or errors.	Object	LOAD RECORD
Searching		
dataClass.get ( primaryKey {; dk key as string} ) -> Entity Returns the entity with the primary key from the dataclass. Null is returned if it doesn't exist.	Entity	QUERY
Detecting Changes		
entity.touched () -> Boolean Returns true if any values in the entity have been set (even if the same value was assigned). False is returned for new entities that have never been saved.	Boolean	Modified record
entity.touchedAttributes () -> Collection Returns a collection of the names of the attributes which have been touched.	Collection	

<b>entity.diff</b> ( <i>entityToCompare</i> {; <i>attributesToCompare</i> } ) -> <i>Collection</i> Creates a collection which contains an object for each attribute that has a different value between the two entities. The object has the following properties: <i>attributeName</i> , <i>value</i> , <i>otherValue</i> . You can optionally narrow the initial comparison to certain attributes.	Collection	OLD
<b>Conversion With Objects</b>		
<b>entity.toObject</b> ( <i>filter</i> {; <i>dk with primary key</i> and/or <i>dk with stamp</i> } ) -> <i>Object</i> Returns an object that is built from the entity. Optionally you can filter the conversion to specific attributes.	Object	
<b>entity.fromObject</b> ( <i>object</i> ) Fills the entity from the object for attributes that match by name.	Mutated	
<b>Inspection</b>		
<b>entity.isNew</b> () -> <i>Boolean</i> Returns true if the entity is a new one that has never been saved to the dataclass.	Boolean	Is new record
<b>entity.getKey</b> ( { <i>dk key as string</i> } ) -> <i>Primary Key</i> Returns the primary key of the entity.	Primary Key (Text or Long)	
<b>entity.getDataClass</b> () -> <i>Dataclass</i> Returns the dataclass the entity belongs to.	Dataclass (Object)	Table name
<b>entity.getStamp</b> () -> <i>Long</i> Returns the internal stamp of the entity. It is automatically incremented by 4D each time it is saved.	Long	