

Generar una consulta de selección de tipo de registro

Create Expression Rule

Create from scratch Duplicate existing expression rule

Name *

QYD_L3_E1

Description

Retornar lista de vehículos

Save In *

 QYD Rules & Constants 



[Create New Rule Folder](#)

CANCEL

CREATE

Query Editor

Select Data Source

Record Type Data Store Entity

Record Type

 Vehicle 

Data Preview

Record Type Relationships

vehicleid	make	model	color	mileage	year	vin	lastMaintenanceDate	nextMaintenanceDate	image	addedBy	dateAdd
1	Ford	F150	Red	500	2019	2F2DE48C8N4309374	6/8/2020	8/10/2020	3544	regina.registrar	5/4/20
2	Lexus	E5350	Pearl	7000	2019	2L3ED45V3D4030403	5/16/2020	8/24/2020	3573	regina.registrar	5/13/20

CANCEL

CONTINUE

Query Editor - Vehicle

 RULE INPUTS

category

condition

maintenance

- # requestId
- # vehicleId
- # requestDate
- # scheduled
- # issue
- # status
- # startDate
- # estimatedCompleteDate
- # actualCompleteDate
- # duration
- # cost
- # requester
- # mechanic
- # reviewStatus
- # reviewer

AGGREGATE FILTER SORT AND LIMIT

Performing Aggregations
A list of groupings can be provided to group fields by common values. A list of measures produces calculated results on the specified fields. [See the documentation for more details.](#) AGGREGATE RECORDS

Query Results Preview TEST QUERY

Querying related record fields from a one-to-many relationship. Querying data from a one-to-many relationship will return a maximum of 10 related records for each base record. To filter, sort, or limit the related records, use `relatedRecordData()` after generating the query. [See the documentation for more details.](#)

# vehicle	A make	A model	A color	# mileage	# year	A vin	# lastMaintenanceDate	# nextMaintenanceD
Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle
1	Ford	F150	Red	500	2019	2F2DE48C8N4309374	6/8/2020	8/10
2	Lexus	E5350	Pearl	7000	2019	2L3ED45V3D4030403	5/16/2020	8/24

BACK CANCEL GENERATE QUERY



The screenshot shows the GraphQL Studio interface for a query named 'QYD_L3_E1'. The query is:

```

1. alqueryRecordType(
2.   recordType: recordType!Vehicle,
3.   fields: {
4.     vehicle.maintenance.cost
5.   },
6.   pagingInfo: alpagingInfo(
7.     startIndex: 1,
8.     batchSize: 100
9.   )
10. ).data

```

The right-hand pane shows the 'Test Output' for an 'Ad Hoc Test'. The output is a list of 32 items, each representing a vehicle with its maintenance records. The first few items are:

- vehicleId 1 (Number (Integer))
 - maintenance List of Maintenance - 6 items
 - Maintenance
 - requestId 1 (Number (Integer))
 - cost 50 (Number (Integer))
 - Maintenance
 - requestId 42 (Number (Integer))
 - cost 50 (Number (Integer))
 - Maintenance
 - requestId 43 (Number (Integer))
 - cost 200 (Number (Integer))
 - Maintenance
 - requestId 59 (Number (Integer))
 - cost 900 (Number (Integer))
 - Maintenance
 - requestId 60 (Number (Integer))
 - cost 70 (Number (Integer))
 - Maintenance
 - requestId 114 (Number (Integer))
 - cost 120 (Number (Integer))

Crear una consulta de agregación de tipo de registro

Create Expression Rule

Create from scratch Duplicate existing expression rule

Name *

QYD_L3_E2

Description

Consulta de agregación

Save In *

QYD Rules & Constants

Create New Rule Folder

CANCEL

CREATE

The screenshot shows the GraphQL Studio interface for a query named 'QYD_L3_E2'. The query is partially visible:

```

1. alqueryRecordType(
2.   recordType: recordType!Vehicle
3. )

```

</> QYD_L3_E2
SAVE

```

1  alqueryRecordType(
2    recordType: recordType!Vehicle!,
3    fields: aggregationFields(
4      groupings: aggregation(
5        field: Vehicle.make!,
6        alias: "Vehiculomake"
7      ),
8      measures: almeasure(
9        field: Vehicle.make!,
10       function: "COUNT",
11       alias: "count"
12     )
13   ),
14   pagingInfo: alpagingInfo(
15     startIndex: 1,
16     batchSize: 100
17   )
18 )

```

alpagingInfo startIndex, batchSize, sort
Creates a Paging Info configuration for use with grids, queries, and todatasubset().

startIndex (Number (Integer)): Index where the current page begins. It must be greater than zero.

batchSize (Number (Integer)): The number of items to return. When used with records, valid values are 0 to 5,000. When used with an entity, valid values include those greater than or equal to -1. See the [documentation](#) for more information.

sort (List of SortInfo): Array of Sort Info configurations created with alsortInfo().

Ad Hoc Test | Test Cases (0)

Test Inputs

TEST RULE

Test Output

Time 87 ms (View Performance) Type Map

Value Formatted Raw Expression

- Map
 - success true (Boolean)
 - startIndex 1 (Number (Integer))
 - batchSize 100 (Number (Integer))
 - totalCount 8 (Number (Integer))
 - sort List of SortInfo - 1 item
 - SortInfo
 - field "Vehiculomake" (Text)
 - ascending true (Boolean)
 - data List of Map - 8 items
 - Map
 - Vehiculomake "Chevy" (Text)
 - count 1 (Number (Integer))
 - Map
 - Vehiculomake "Ford" (Text)
 - count 9 (Number (Integer))
 - Map
 - Vehiculomake "Honda" (Text)
 - count 8 (Number (Integer))
 - Map
 - Vehiculomake "Honda" (Text)
 - count 1 (Number (Integer))

Generar una consulta de selección de entidad de almacén de datos

Create Expression Rule

Create from scratch Duplicate existing expression rule

Name *

QYD_L3_E3

Description

Consulta por almacenamiento de dato

Save In *

QYD Rules & Constants

Create New Rule Folder

CANCEL

CREATE

Query Editor

Select Data Source

Record Type Data Store Entity

Data Store Entity

[AA_VEHICLES_DSE_POINTER](#) X

Data Preview

vehicleID	vehicleMake	vehicleModel	vehicleColor	vehicleCondition	vehicleStatus	vehicleCategory	vehicleMileage	vehicleYear	vehicleVIN
1	Ford	F150	Red	10	9	3	500	2019	2F2DE48C8N4309374
2	Lexus	ES350	Pearl	11	9	6	7000	2019	2L3ED45V3D4030403

CANCEL

CONTINUE

Query Editor - AA_VEHICLES_DSE_POINTER

RULE INPUTS

Fields

ADD ALL REMOVE ALL

- vehicleID - Number (Int...) +
- vehicleMake - Text +
- vehicleModel - Text +
- vehicleColor - Text +
- vehicleCondition - Numb... +
- vehicleStatus - Number ... +
- vehicleCategory - Numb... +
- vehicleMileage - Number... +
- vehicleYear - Number (I... +
- vehicleVIN - Text +
- vehicleLastMaintenanceD... +

Query Results Preview

Fetch total row count:

GROUP DATA TEST QUERY

vehicleID	vehicleMake	vehicleModel	vehicleMileage	vehicleYear
1	Ford	F150	500	2019
2	Lexus	ES350	7000	2019
3	VW	Corrado	25000	2015
4	Honda	Pilot	1000	2020
5	Honda	Odyssey	25	2020
6	Toyota	Camry	2000	2018
7	VW	Golf	120000	2010
8	Ford	F150	7000	2019

CHANGE DATA STORE ENTITY CANCEL

GENERATE QUERY

QYD_L3_E3

```
6 alqueryColumn(  
7   field: "vehicleID"  
8 ),  
9 alqueryColumn(  
10  field: "vehicleMake"  
11 ),  
12 alqueryColumn(  
13  field: "vehicleModel"  
14 ),  
15 alqueryColumn(  
16  field: "vehicleMileage"  
17 ),  
18 alqueryColumn(  
19  field: "vehicleYear"  
20 )  
21 }  
22 ),  
23 pagingInfo: alpagingInfo(  
24   startIndex: 1,  
25   batchSize: 50  
26 ),  
27 fetchTotalCount: false  
28 ),  
29 ).data
```

Place cursor on function, rule, or constant to display help

Test Inputs

TEST RULE

Test Output

Time 77 ms (View Performance) Type List of Variant

Value Formatted Raw Expression

- List of Dictionary - 32 items
 - Dictionary
 - vehicleID 1 (Number (Integer))
 - vehicleMake "Ford" (Text)
 - vehicleModel "F150" (Text)
 - vehicleMileage 500 (Number (Integer))
 - vehicleYear 2019 (Number (Integer))
 - Dictionary
 - vehicleID 2 (Number (Integer))
 - vehicleMake "Lexus" (Text)
 - vehicleModel "ES350" (Text)
 - vehicleMileage 7000 (Number (Integer))
 - vehicleYear 2019 (Number (Integer))
 - Dictionary
 - vehicleID 3 (Number (Integer))
 - vehicleMake "VW" (Text)
 - vehicleModel "Corrado" (Text)
 - vehicleMileage 25000 (Number (Integer))
 - vehicleYear 2015 (Number (Integer))
 - Dictionary
 - vehicleID 4 (Number (Integer))
 - vehicleMake "Honda" (Text)

Filtro de datos en tipo de registro de vehículo

Create Expression Rule

Create from scratch Duplicate existing expression rule

Name *

QYD_L4_E1

Description

Filtro de datos en tipo de registro de vehículo

Save In *

 QYD Rules & Constants 



[Create New Rule Folder](#)

CANCEL

CREATE

Query Editor

Select Data Source

Record Type Data Store Entity

Record Type

 Vehicle 

Data Preview

Record Type Relationships

vehicleid	make	model	color	mileage	year	vin	lastMaintenanceDate	nextMaintenanceDate	image	addedBy	dateAdd
1	Ford	F150	Red	500	2019	2F2DE48C8N4309374	6/8/2020	8/10/2020	3544	regina.registrar	5/4/20
2	Lexus	ES350	Pearl	7000	2019	2L3ED45V3D4030403	5/16/2020	8/24/2020	3573	regina.registrar	5/13/20

CANCEL

CONTINUE

Edit Value Expression

Enter an expression that returns a value of type Number (Integer) to define the filter value.

1

`cons!AA_VEHICLE_HIGH_MILEAGE`

cons!AA_VEHICLE_HIGH_MILEAGE

Cut off mileage for high mileage vehicles.

Returns: Number (Integer)

[Clear expression](#)

CANCEL

OK

Query Editor - Vehicle

[RULE INPUTS](#)

AGGREGATEFILTERIF SORT AND LIMIT

Filtering Records

A list of filter conditions can be provided to return a subset of record data. [See the documentation for more details.](#)

Guided Expression

Ignore filters with empty values

Only records that meet all filter conditions will be returned. [Learn more about creating filters.](#)

Field	Condition	Value	Apply Filter
mileage ✕	<	cons!AA_VEHICLE_HIGH_MILEAGE	Always ▶

[Add Filter](#) [Edit as Expression](#)

Query Results Preview TEST QUERY

i Querying related record fields from a one-to-many relationship. Querying data from a one-to-many relationship will return a maximum of 10 related records for ✕

BACK CANCEL

GENERATE QUERY

</> QYD_L4_E1
SAVE

```

1  a!queryRecordType(
2    recordType: recordType!Vehicle ,
3    fields: {
4      Vehicle.maintenance.cost
5    },
6    filters: a!queryLogicalExpression(
7      operator: "AND",
8      filters: {
9        a!queryFilter(
10         field: Vehicle.mileage,
11         operator: "<",
12         value: cons!AA_VEHICLE_HIGH_MILEAGE
13       )
14     },
15     ignoreFiltersWithEmptyValues: true
16   ),
17   pagingInfo: a!pagingInfo(
18     startIndex: 1,
19     batchSize: 100
20   )
21 ).data

```

Place cursor on function, rule, or constant to display help

Ad Hoc Test Test Cases (0)

Test Inputs

TEST RULE

Test Output

Time 63 ms [View Performance](#) Type List of Vehicle

Value Formatted Raw Expression

- ▲ List of Vehicle - 30 items
 - ▲ Vehicle
 - vehicleId 1 (Number (Integer))
 - ▲ maintenance List of Maintenance - 6 items
 - ▲ Maintenance
 - requestId 1 (Number (Integer))
 - cost 50 (Number (Integer))
 - ▲ Maintenance
 - requestId 42 (Number (Integer))
 - cost 50 (Number (Integer))
 - ▲ Maintenance
 - requestId 43 (Number (Integer))
 - cost 200 (Number (Integer))
 - ▲ Maintenance
 - requestId 59 (Number (Integer))
 - cost 900 (Number (Integer))
 - ▲ Maintenance
 - requestId 60 (Number (Integer))
 - cost 70 (Number (Integer))
 - ▲ Maintenance
 - requestId 114 (Number (Integer))
 - cost 120 (Number (Integer))

Filtrar datos de registros relacionados en el tipo de registro de mantenimiento

Duplicate QYD_L4_E1

Name *

QYD_L4_E2

Description

Filtro de datos en tipo de registro de vehículo

Save In *

QYD Rules & Constants

CANCEL

CREATE

```

relatedRecordData: a!relatedRecordData(
  relationship: Vehicle.maintenance ,
  filters: a!queryFilter(
    field: recordType!Vehicle .cost,
    operator: ">",
    value: 200
  )
)

```

Ordenar y limitar el tipo de registro base de su vehículo

Create Expression Rule

Create from scratch Duplicate existing expression rule



Name *

QYD_L5_E1

Description

Ordenar y limpiar datos, tipo de registro

Save In *

 QYD Rules & Constants 



Create New Rule Folder

CANCEL

CREATE

Query Editor - Vehicle

 RULE INPUTS

A MileageCategory
 # addedMonth
 # addedYear

> > category
> > condition
v < maintenance

- # requestId
- # vehicleId
- requestDate
- scheduled
- A issue
- A status
- startDate
- estimatedCompleteDate
- actualCompleteDate
- # duration
- # cost
- A requester

AGGREGATE FILTER **IF SORT AND LIMIT**

Sorting and Limiting Data

A list of sorting fields can be provided to determine the order in which query results are displayed. The data limit defines the max number of records to return. See the documentation for more details.

Guided Expression

Sort by **mileage** Ascending Order

Data Limit 10

Query Results Preview TEST QUERY

Querying related record fields from a one-to-many relationship. Querying data from a one-to-many relationship will return a maximum of 10 related records for each base record. To filter, sort, or limit the related records, use `alrelatedRecordData` after generating the query. See the documentation for more details.

# vehicleId	A make	A model	A color	# mileage	# year	A vin	lastMaintenanceDate	nextMaintenanceD
Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle

BACK CANCEL GENERATE QUERY

Query Editor - Vehicle

 RULE INPUTS

A MileageCategory
 # addedMonth
 # addedYear

> > category
> > condition
v < maintenance

- # requestId
- # vehicleId
- requestDate
- scheduled
- A issue
- A status
- startDate
- estimatedCompleteDate
- actualCompleteDate
- # duration
- # cost
- A requester

Query Results Preview TEST QUERY

Querying related record fields from a one-to-many relationship. Querying data from a one-to-many relationship will return a maximum of 10 related records for each base record. To filter, sort, or limit the related records, use `alrelatedRecordData` after generating the query. See the documentation for more details.

# vehicleId	A make	A model	A color	# mileage	# year	A vin	lastMaintenanceDate	nextMaintenanceD
Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle	Vehicle
5	Honda	Odyssey	Silver	25	2020	2H7XF64H8J6572134	3/2/2020	9/2
37	Ford	F150	Blue	300	2020	ABC12345678909876	11/8/2019	6/5
1	Ford	F150	Red	500	2019	2F2DE48C8N4309374	6/8/2020	8/10
8	Ford	F150	Red	700	2019	1F4CG3758P3628311	5/9/2021	5/10
24	Ford	F150	White	700	2019	5F3484833D5534544	4/7/2020	10/7

BACK CANCEL GENERATE QUERY

The screenshot shows a development environment with a code editor on the left and a test runner on the right. The code editor contains a GraphQL query:

```

1  alqueryRecordType(
2    recordType: recordType!Vehicle ,
3    fields: {
4      Vehicle.maintenance.cost
5    },
6    pagingInfo: alpagingInfo(
7      startIndex: 1,
8      batchSize: 10,
9      sort: {
10     alsortInfo(
11       field: Vehicle.mileage,
12       ascending: true
13     )
14   }
15 )
16 ).data

```

The test runner on the right shows the following output:

- Test Inputs:** A button labeled "TEST RULE".
- Test Output:**
 - Time: 76 ms (View Performance) Type: List of Vehicle
 - Value: Formatted (selected), Raw, Expression
 - List of Vehicle - 10 items
 - Vehicle
 - vehicleId 5 (Number (Integer))
 - maintenance List of Maintenance - 6 items
 - Maintenance
 - requestId 17 (Number (Integer))
 - cost 0 (Number (Integer))
 - Maintenance
 - requestId 19 (Number (Integer))
 - cost 0 (Number (Integer))
 - Maintenance
 - requestId 48 (Number (Integer))
 - cost 70 (Number (Integer))
 - Maintenance
 - requestId 85 (Number (Integer))
 - cost 0 (Number (Integer))
 - Maintenance
 - requestId 97 (Number (Integer))
 - cost 0 (Number (Integer))
 - Maintenance
 - requestId 101 (Number (Integer))
 - cost 0 (Number (Integer))

Below the code editor, there is a note: "Place cursor on function, rule, or constant to display help".

Ordenar y limitar el campo Costo de mantenimiento

Duplicate QYD_L5_E1

Name *

QYD_L5_E2

Description

Ordenar y limpiar campo Costo de mantenimiento

Save In *

QYD Rules & Constants

CANCEL

CREATE

```

relatedRecordData: alrelatedRecordData(
  relationship: Vehicle.maintenance ,
  limit: 1,
  sort: alsortInfo(
    field: recordType!Vehicle.requestDate
  )
)

```

Usar una consulta para rellenar una lista desplegable

Create Expression Rule

Create from scratch Duplicate existing expression rule


Name *

QYD_GetVehicleMakes

Description

Para lista desplegable

Save In *



 QYD Rules & Constants 



[Create New Rule Folder](#)

CANCEL

CREATE

 QYD_GetVehicleMakes  SAVE

```
1 • allLocalVariables(  
2 •   localldata:alqueryRecordType(  
3 •     recordType: recordTypeVehicle,  
4 •     fields: {  
5 •       Vehicle.make  
6 •     },  
7 •     pagingInfo: alpagingInfo(  
8 •       startIndex: 1,  
9 •       batchSize: 100  
10 •    )  
11 •   ).data[ Vehicle.make ],  
12 •   union(localldata,localldata)  
13 • )  
14
```

Test Inputs

Local Variables

Test Output

Time 48 ms [View Performance](#) Type List of Text String

Value Formatted Raw Expression

▲ List of Text String - 8 items

- "Ford" (Text)
- "Lexus" (Text)
- "VW" (Text)
- "Honda" (Text)
- "Honda" (Text)
- "Toyota" (Text)
- "MB" (Text)
- "Chevy" (Text)

allLocalVariables localVar1, localVarN, expression

Lets you define one or more local variables for use within an expression. When used within an interface, the value of each variable can be refreshed under a variety of conditions, configured using `alrefreshVariable()`. When used outside of an interface, all refresh properties configured using `alrefreshVariable()` are ignored.

localVar1 (Any Type): The local variable to use when evaluating the given expression. Use the 'local' domain to define and reference individual variables. By default, a local variable will automatically update when any variables it references are changed. To change the way variables are updated, use the `alrefreshVariable()` function. Variables can be refreshed under the following conditions: after each reevaluation, periodically on an interval, or when other variables change.

localVarN (Any Type): An unlimited number of local variables.

expression (Any Type): The expression to evaluate using the local variables' values.

Create Interface

Create from scratch Duplicate existing interface

Interface to Duplicate *

AA_AddUpdateVehicle x

Name *

QYD_L6_E1

Description

Interface to add and update the vehicle.

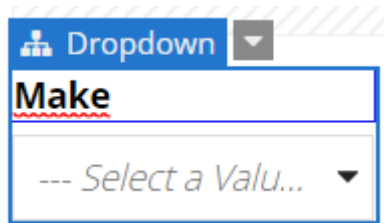
Save In *

AA Interfaces x

Create New Rule Folder

CANCEL

CREATE



QYD_L6_E1 DESIGN MODE EXPRESSION MODE

PALETTE COMPONENTS INTERIORS

Search Components...

INPUTS

MAX TEXT

PARAGRAPH

123 INTEGER

4.56 DECIMAL

DATE

DATE & TIME

ENCRYPTED TEXT

FILE UPLOAD

SIGNATURE

BARCODE

SELECTION

DROPDOWN

MULTIPLE DROPDOWN

RADIO BUTTONS

CHECKBOXES

CARD CHOICES

DISPLAY

IMAGE

BULleted LIST

TAG

VEHICLE INFORMATION

Vehicle information

Make

--- Select a Valu...

VIN *

Category *

--- Select category ---

Condition *

--- Select condition ---

CANCEL

Choice Labels (List of Text String)

Array of options for the user to select.

```
1 rule@getVehicleMakes()
```

Place cursor on function, rule, or constant to display help

Clear expression and reset value

CANCEL OK

RULE INPUTS

Name	Value
vehicle	[[vehicleId=, vehic...
cancel	null
comments	null

LOCAL VARIABLES

Name	Value
conditions (6 it...	[[Condition id=12, ...
category (6 re...	[[Category id=5, la...
status (3 items)	[[Status id=7, type...

COMPONENT CONFIGURATION

Items

Dropdown

Placeholder

--- Select a Value ---

Choice Labels

List of Text

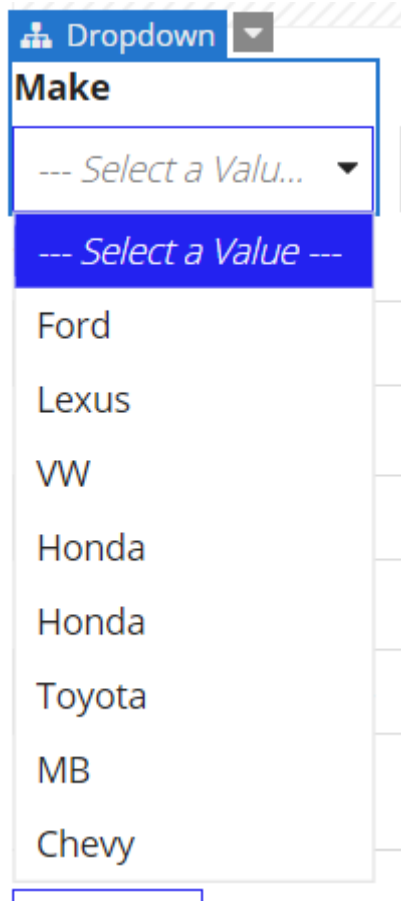
Choice Values

List of Any Type

Selected Value

Select a variable

Save Selection To



Usar una consulta para agregar un KPI a un informe

Create Expression Rule

Create from scratch Duplicate existing expression rule

Name*

QYD_GetVehiclesAddedThisMonth

Description

Para KPI

Save In*

 QYD Rules & Constants 



[Create New Rule Folder](#)

CANCEL

CREATE

</> QYD_GetVehiclesAddedThisMonth SAVE

```

1. aqueryRecordType(
2.   recordType: recordType!Vehicle,
3.   filters: aqueryFilter(
4.     field: Vehicle.addedMonth,
5.     operator: "=",
6.     value: month(today())
7.   ),
8.   fields: aaggregationFields(
9.     groupings: agrouping(
10.      field: Vehicle.addedMonth,
11.      alias: "month"
12.    ),
13.     measures: ameasure(
14.      field: Vehicle.addedMonth,
15.      function: "COUNT",
16.      alias: "count"
17.    )
18.   ),
19.   pagingInfo: apagingInfo(
20.     startIndex: 1,
21.     batchSize: 5000
22.   )
23. ).data.count

```

Ad Hoc Test | Test Cases (0)

Test Inputs

[TEST RULE](#)

Test Output

Time 57 ms ([View Performance](#)) | Type List of Number (Integer)

Value Formatted Raw Expression

- ▲ List of Number (Integer) - 1 item
 - └ 2 (Number (Integer))

Create Interface

Create from scratch Duplicate existing interface

Interface to Duplicate *

AA_RegistrarReport ×

Name *

Description

Save In *

AA Interfaces ×

[Create New Rule Folder](#)

[CANCEL](#)

[CREATE](#)

Section Layout

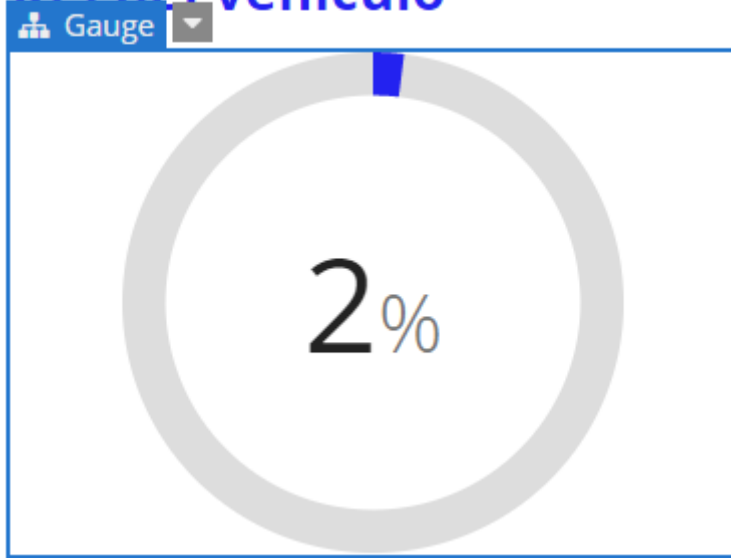
KPI del vehículo

Drop component here

Fill Percentage

rule!QYD_GetVehiclesAddedThisMonth()

KPI del vehículo



RESULTADO DE LAS ACTIVIDADES

<input type="checkbox"/>	Name	Description	Last Modified
<input type="checkbox"/>	QYD_L6_E2	Interface to pull in the reporting data for AA Registrars.	7/31/2022 3:05 AM by José Alberto Castill...
<input type="checkbox"/>	QYD_GetVehiclesAddedThisMonth	Para KPI	7/31/2022 3:04 AM by José Alberto Castill...
<input type="checkbox"/>	QYD_L6_E1	Interface to add and update the vehicle.	7/31/2022 2:58 AM by José Alberto Castill...
<input type="checkbox"/>	QYD_GetVehicleMakes	Para lista desplegable	7/31/2022 2:52 AM by José Alberto Castill...
<input type="checkbox"/>	QYD_L5_E2	Ordenar y limpiar campo Costo de mantenimiento	7/31/2022 2:39 AM by José Alberto Castill...
<input type="checkbox"/>	QYD_L5_E1	Ordenar y limpiar datos, tipo de registro	7/31/2022 2:30 AM by José Alberto Castill...
<input type="checkbox"/>	QYD_L4_E2	Filtro de datos en tipo de registro de vehículo	7/31/2022 1:59 AM by José Alberto Castill...
<input type="checkbox"/>	QYD_L4_E1	Filtro de datos en tipo de registro de vehículo	7/31/2022 1:54 AM by José Alberto Castill...
<input type="checkbox"/>	QYD_L3_E3	Consulta por almacenamiento de dato	7/31/2022 1:26 AM by José Alberto Castill...
<input type="checkbox"/>	QYD_L3_E2	Consulta de agregación	7/31/2022 1:12 AM by José Alberto Castill...
<input type="checkbox"/>	QYD_L3_E1	Retornar lista de vehículos	7/31/2022 1:04 AM by José Alberto Castill...