

Se crea una nueva tabla

The screenshot shows the Appian Designer interface with the SQL editor open. The left sidebar displays a tree view of the database schema, including tables like 'address', 'bmaemployee', and 'bmaperson'. The main editor area contains the following SQL code:

```
1 Create table address (address_id int(11) not null,  
2 shipping_address varchar(100),  
3 unit_number int(5),  
4 city varchar(100),  
5 state_or_province varchar(2),  
6 postal_code int(5));  
7 ALTER TABLE `address`  
8 ADD PRIMARY KEY (`address_id`);  
9 ALTER TABLE `address`  
10 MODIFY `address_id` int(11) NOT NULL AUTO_INCREMENT;
```

Below the code, there are buttons for 'Limpiar', 'Formato', and 'Obtener consulta almacenada automáticamente'. There are also checkboxes for 'Enlazar parámetros' and 'Habilite la revisión de las claves foráneas'. A 'Continuar' button is located at the bottom right.

Se insertan registros en la tabla creada

The screenshot shows the Appian Designer interface with the SQL editor open, now showing the 'address' table selected in the left sidebar. The main editor area contains the following SQL code:

```
1 INSERT INTO `address` (`shipping_address`, `unit_number`,  
2 `state_or_province`, `postal_code`, `city`) VALUES ('47 New  
3 Saddle Ave.', NULL, 'SC', 29445, 'Goose Creek');  
4 INSERT INTO `address` (`shipping_address`, `unit_number`,  
5 `state_or_province`, `postal_code`, `city`) VALUES ('7208 Oakland  
6 Drive', 50, 'OH', 44266, 'Ravenna');  
7 INSERT INTO `address` (`shipping_address`, `unit_number`,  
8 `state_or_province`, `postal_code`, `city`) VALUES ('358 Playfair  
9 Dr.', 23, 'VA', 01902, 'Lynn');
```

Below the code, there are buttons for 'SELECT *', 'SELECT', 'INSERT', 'UPDATE', and 'DELETE', along with 'Limpiar', 'Formato', and 'Obtener consulta almacenada automáticamente'. There are also checkboxes for 'Enlazar parámetros' and 'Habilite la revisión de las claves foráneas'. A 'Continuar' button is located at the bottom right.

Se crea un nuevo grupo de ADV All Users

The screenshot shows the Appian 'Create Group' form. The 'Name' field is filled with 'ADV All Users'. The 'Description' field contains 'All users utilizing the ADV app'. The 'Parent Group' field is set to 'Select a group'. The 'Group Members' field is empty. The 'Group Type' is set to 'Custom'. The 'Visibility' is set to 'Restricted (only members and admins)'. The 'Privacy Policy' field is empty. On the right side, there is a 'Last Modified' table with the following data:

Last Modified
8/8/2022 5:59 PM by Jordi Brayan Vicen...
8/7/2022 11:58 PM by Jordi Brayan Vicen...
8/7/2022 11:48 PM by Jordi Brayan Vicen...
8/7/2022 11:45 PM by Jordi Brayan Vicen...
8/7/2022 11:44 PM by Jordi Brayan Vicen...
8/7/2022 11:41 PM by Jordi Brayan Vicen...
8/7/2022 11:37 PM by Jordi Brayan Vicen...
8/7/2022 11:32 PM by Jordi Brayan Vicen...
8/7/2022 11:32 PM by Jordi Brayan Vicen...
8/7/2022 11:32 PM by Jordi Brayan Vicen...
8/7/2022 11:32 PM by Jordi Brayan Vicen...

Se crea un nuevo grupo de ADV All Administrators

The screenshot shows the Appian 'Create Group' form. The 'Name' field is filled with 'ADV All Administrators'. The 'Description' field contains 'Contains all administrators'. The 'Parent Group' field is set to 'ADV All Users'. The 'Group Members' field is empty. The 'Group Type' is set to 'Custom'. The 'Visibility' is set to 'Restricted (only members and admins)'. The 'Privacy Policy' field is empty. On the right side, there is a 'Last Modified' table with the following data:

Last Modified
8/8/2022 10:27 PM by Jordi Brayan Vicen...
8/8/2022 5:59 PM by Jordi Brayan Vicen...
8/7/2022 11:58 PM by Jordi Brayan Vicen...
8/7/2022 11:48 PM by Jordi Brayan Vicen...
8/7/2022 11:45 PM by Jordi Brayan Vicen...
8/7/2022 11:44 PM by Jordi Brayan Vicen...
8/7/2022 11:41 PM by Jordi Brayan Vicen...
8/7/2022 11:37 PM by Jordi Brayan Vicen...
8/7/2022 11:32 PM by Jordi Brayan Vicen...
8/7/2022 11:32 PM by Jordi Brayan Vicen...
8/7/2022 11:32 PM by Jordi Brayan Vicen...

Se crea un nuevo Data Type

Create Data Type

Namespace *
urn:com:appiantypes:ADV
Formatted as a URI, for example 'urn:com:appiantypes:COB' for a client onboarding application

Name *
ADV_Address

Description
Data Type para address

Column Name	Field Name	Column Type	Field Type	Key	Nulls	Unique			
address_id	addressId	INT	Number (Integer)		<input type="checkbox"/>	<input checked="" type="checkbox"/>	↑	↓	×
shipping_address	shippingAddress	VARCHAR(100)	Text		<input checked="" type="checkbox"/>	<input type="checkbox"/>	↑	↓	×
unit_number	unitNumber	INT	Number (Integer)		<input checked="" type="checkbox"/>	<input type="checkbox"/>	↑	↓	×

[GO BACK](#) [CANCEL](#) [CREATE](#)

Se crea un Data Store

Create Data Store

Name *
ADV_DS

Description
Data Store for the ADV app

Data Source *
jdbc/Appian (Tomcat)

[CANCEL](#) [CREATE](#)

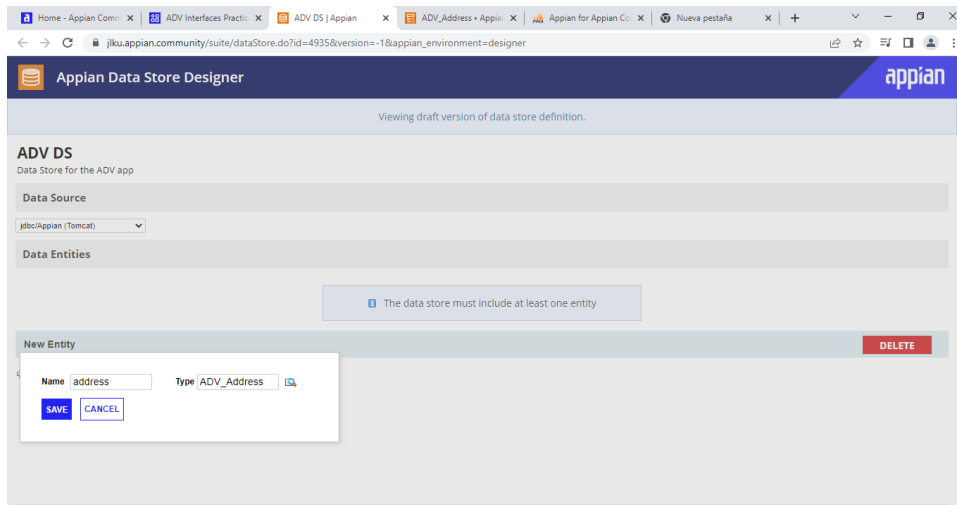
Object Type List:

- Connected System
- Constant
- Data Store
- Data Type
- Decision
- Document
- Expression Rule
- Feed
- Folder
- Group
- Integration
- Interface
- Process Model
- Query Rule
- Record Type
- Report
- Site

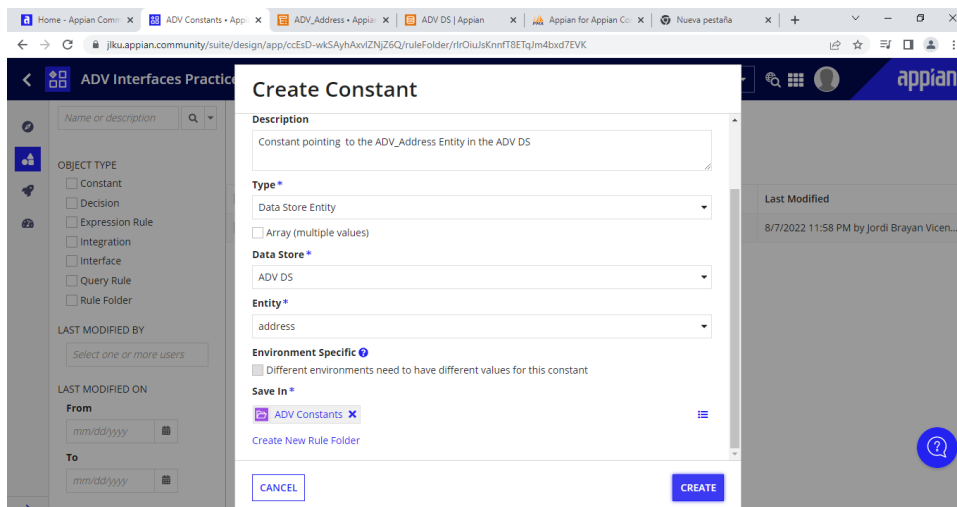
Application Objects Table:

Name or description	Last Modified
ADV Interfaces	8/9/2022 12:01 AM by Jordi Brayan Vicen...
ADV Documents and Files	8/8/2022 10:33 PM by Jordi Brayan Vicen...
ADV_ItemInventory (urn:com:appianty...	8/8/2022 10:32 PM by Jordi Brayan Vicen...
ADV Administrators	8/8/2022 5:59 PM by Jordi Brayan Vicen...
	8/7/2022 11:58 PM by Jordi Brayan Vicen...
	8/7/2022 11:48 PM by Jordi Brayan Vicen...
	8/7/2022 11:45 PM by Jordi Brayan Vicen...
	8/7/2022 11:44 PM by Jordi Brayan Vicen...
	8/7/2022 11:41 PM by Jordi Brayan Vicen...
	8/7/2022 11:37 PM by Jordi Brayan V...
	8/7/2022 11:32 PM by Jordi Brayan Vicen...

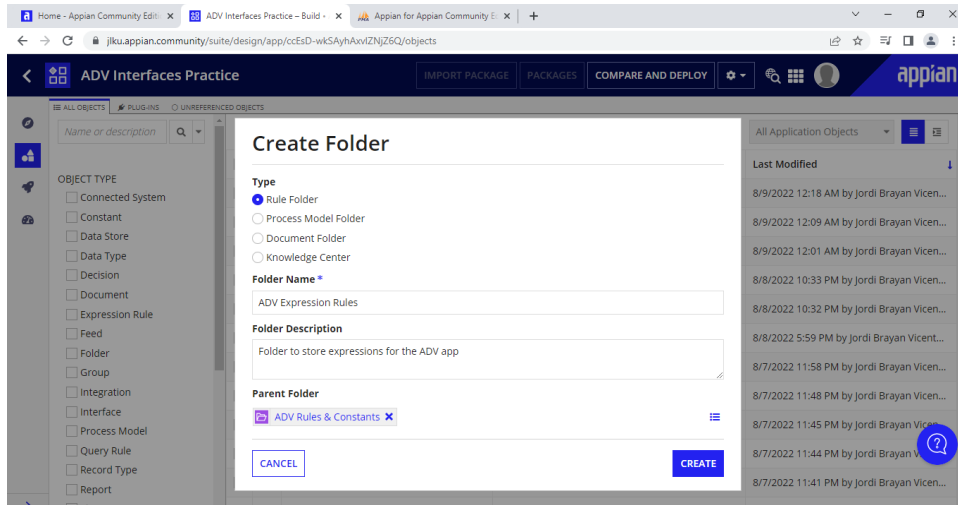
Se agrega el entity de address



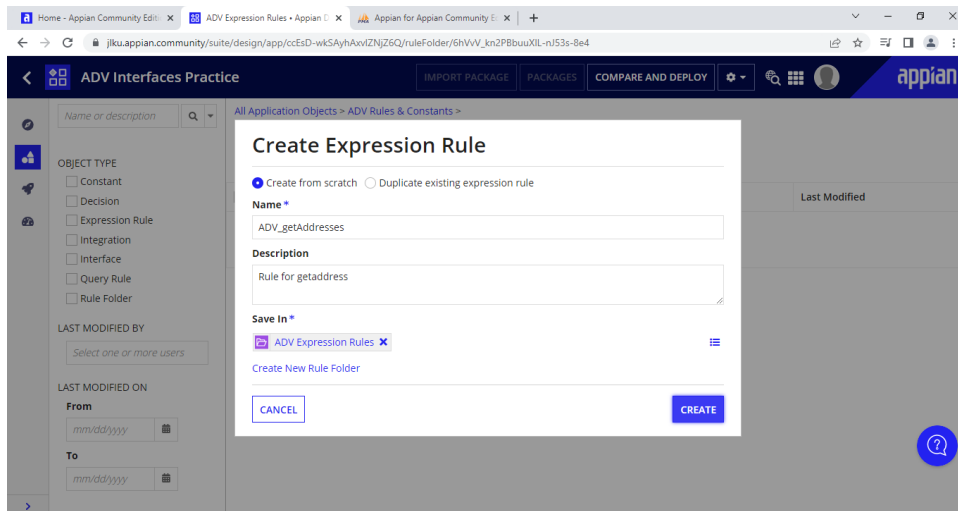
Se crea una nueva constante



Se crea un nuevo folder llamado ADV Expressions Rules



Se crea una nueva regla de Expresion



Se agregan nuevas configuraciones para después testear la regla

The screenshot shows the Appian IDE interface for editing a rule named 'ADV_getAddresses'. The left pane displays the rule's logic, which includes a 'cast' function and a 'localQueryResults' function. The right pane shows the 'Test Output' for the rule, which is a list of three 'ADV_Address' objects. The 'RULE INPUTS' table shows the input 'addressid' of type 'Number'.

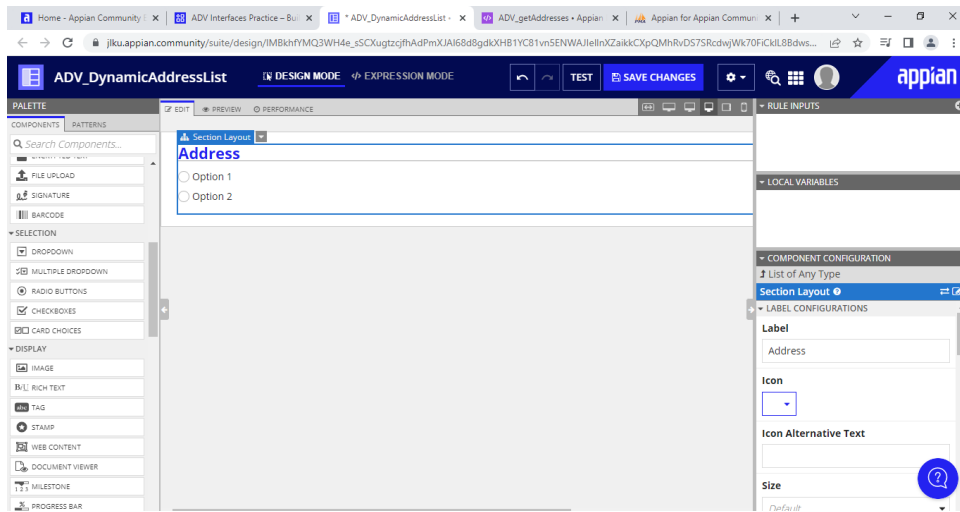
Name	Type	Array
addressid	Number	

Test Output
Time: 42 ms (View Performance) Type: List of ADV_Address
Value: Formatted (selected) Raw Expression
List of ADV_Address - 3 items
ADV_Address
- addressid 3 (Number (Integer))
- shippingAddress "368 Mayfair Dr." (Text)
- unitNumber 23 (Number (Integer))
- city "Lynn" (Text)
- stateOrProvince "MA" (Text)
- postalCode 1902 (Number (Integer))
ADV_Address
- addressid 1 (Number (Integer))
- shippingAddress "47 New Saddle Ave." (Text)
- unitNumber null (Number (Integer))
- city "Goose Creek" (Text)
- stateOrProvince "SC" (Text)
- postalCode 29445 (Number (Integer))
ADV_Address
- addressid 8 (Number (Integer))
- shippingAddress "7208 Oakland Drive" (Text)
- unitNumber 50 (Number (Integer))
- city "Ravenna" (Text)
- stateOrProvince "OH" (Text)
- postalCode 44266 (Number (Integer))

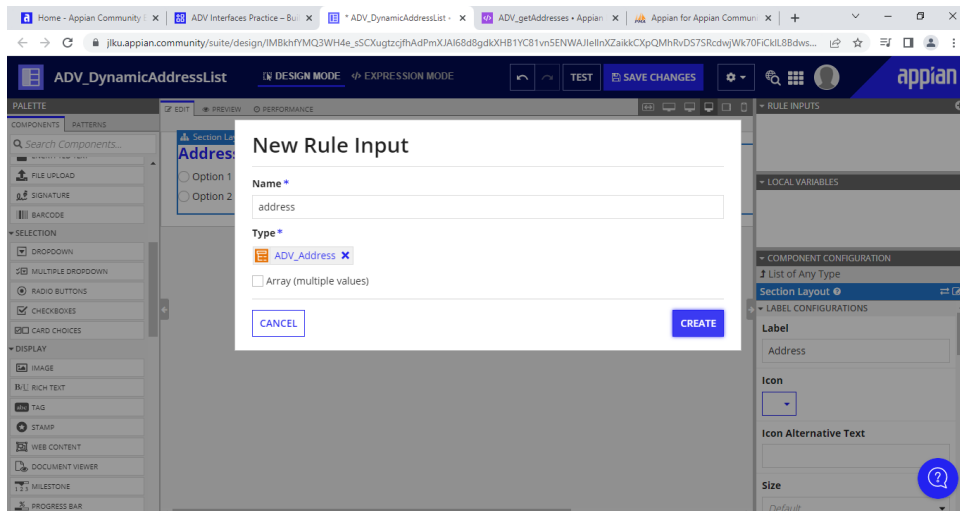
Se crea una nueva interface

The screenshot shows the 'Create Interface' dialog box in the Appian IDE. The dialog is titled 'Create Interface' and has the following fields:
- **Name ***: ADV_DynamicAddressList
- **Description**: Interface for DynamicAddressList
- **Save In ***: ADV Interfaces X
The 'CREATE' button is highlighted in blue. The background shows the 'ADV Interfaces Practice' workspace with a list of application objects.

Se agregan los componentes de seccion y radio buttons



Se crea una nueva regla



Se agregan configuraciones dentro del modo de expresión

The screenshot shows the Appian Expression Mode interface. On the left, the 'INTERFACE DEFINITION' pane contains the following code:

```
1. allLocalVariables(  
2. {  
3.   aselectionlayout(  
4.     label: "Address",  
5.     contents: {  
6.       aRadioButtonField(  
7.         label: "Radio Buttons",  
8.         labelPosition: "COLLAPSED",  
9.         choiceLabels: ("Option 1", "Option 2")  
10.        choiceValues: {  
11.          saveInto: {},  
12.          choiceLayout: "STACKED",  
13.          validations: {}  
14.        }  
15.      )  
16.    }  
17.  )  
18. }
```

Below the code, the 'allLocalVariables' section is expanded, showing a table with columns 'Name' and 'Value'. The 'address' variable is listed with a value of 'null'. A 'LOCAL VARIABLES' section is also visible, currently empty.

Below the code, there is a text block explaining local variables:

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

Se verifica que al seleccionar un botón se guardan los datos en nuestra variable local

The screenshot shows the Appian Expression Mode interface. On the left, the 'INTERFACE DEFINITION' pane contains the following code:

```
1. allLocalVariables(  
2. {  
3.   aselectionlayout(  
4.     label: "Address",  
5.     contents: {  
6.       aRadioButtonField(  
7.         label: "Radio Buttons",  
8.         labelPosition: "COLLAPSED",  
9.         choiceLabels: ("Option 1", "Option 2")  
10.        choiceValues: {  
11.          saveInto: {},  
12.          choiceLayout: "STACKED",  
13.          validations: {}  
14.        }  
15.      )  
16.    }  
17.  )  
18. }
```

Below the code, the 'allLocalVariables' section is expanded, showing a table with columns 'Name' and 'Value'. The 'address' variable is listed with a value of '[addressid=3, ship...'. A 'LOCAL VARIABLES' section is also visible, currently empty.

Below the code, there is a text block explaining local variables:

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

Home - Appian | ADV Interfaces | ADV_DynamicAd | ADV_getAddress | ADV_DS | Appian | ADV_address | Appian for Appi

jku.appian.community/suite/design/IMBkhfYMq3WH4e_sSCXugtztqfAdPmXIAI68d8gdXh81YCB1vn5ENWAJlelnXZaikkCXpQMhRvD575RcdwJWk70FCKl8Bdaws...

ADV_DynamicAddressList DESIGN MODE EXPRESSION MODE TEST SAVE CHANGES

EDIT PREVIEW PERFORMANCE

Address

- 358 Mayfair Dr. Unit #23 Lynn, MA 1902
- 47 New Saddle Ave. Goose Creek, SC 29445
- 7208 Oakland Drive Unit #50 Ravenna, OH 44266

