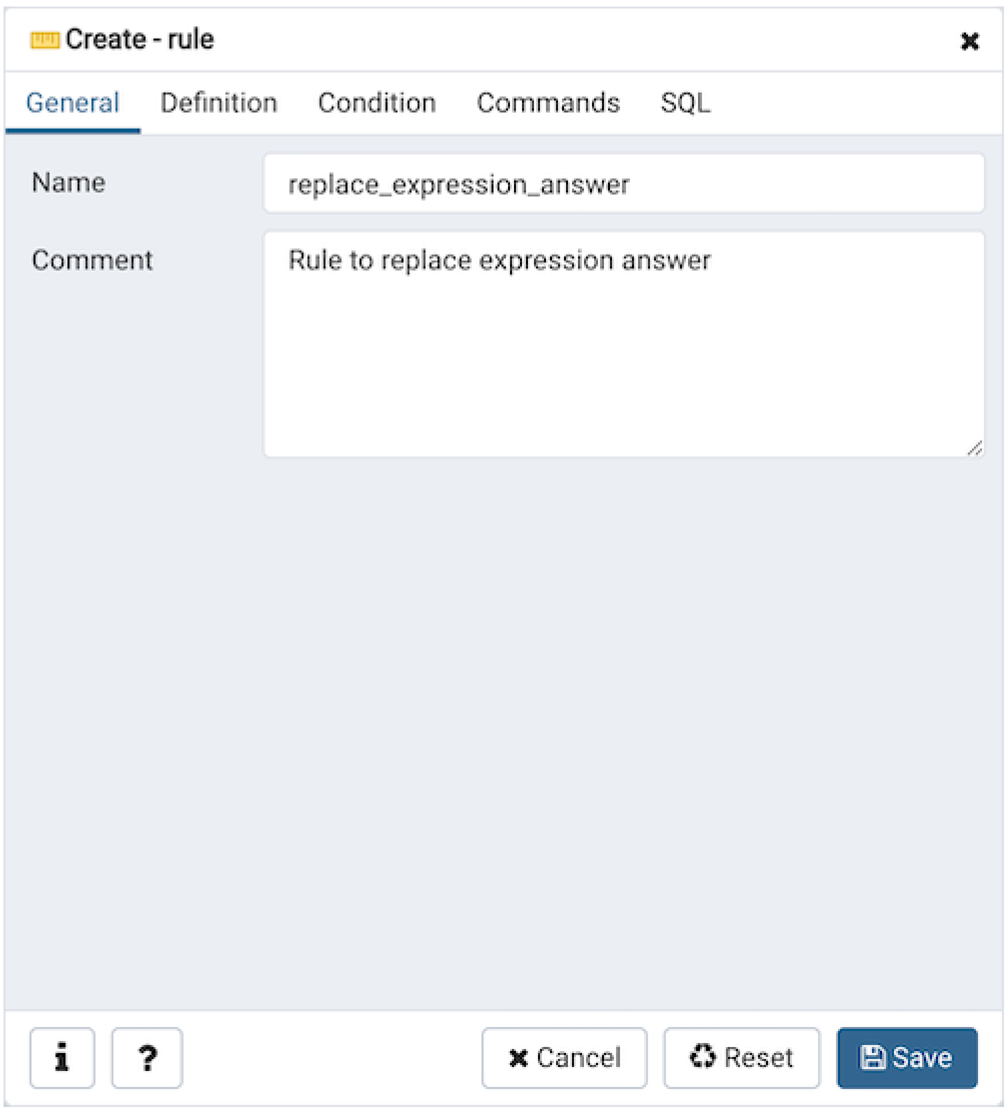
Warning: This documentation is for a pre-release version of pgAdmin 4

Contents

Rule Dialog ¶

Use the *Rule* dialog to define or modify a rule for a specified table or view. A PostgreSQL rule allows you to define an additional action that will be performed when a SELECT, INSERT, UPDATE, or DELETE is performed against a table.

The *Rule* dialog organizes the development of a rule through the *General*, *Definition*, *Condition*, *Commands* tabs. The *SQL* tab displays the SQL code generated by dialog selections.

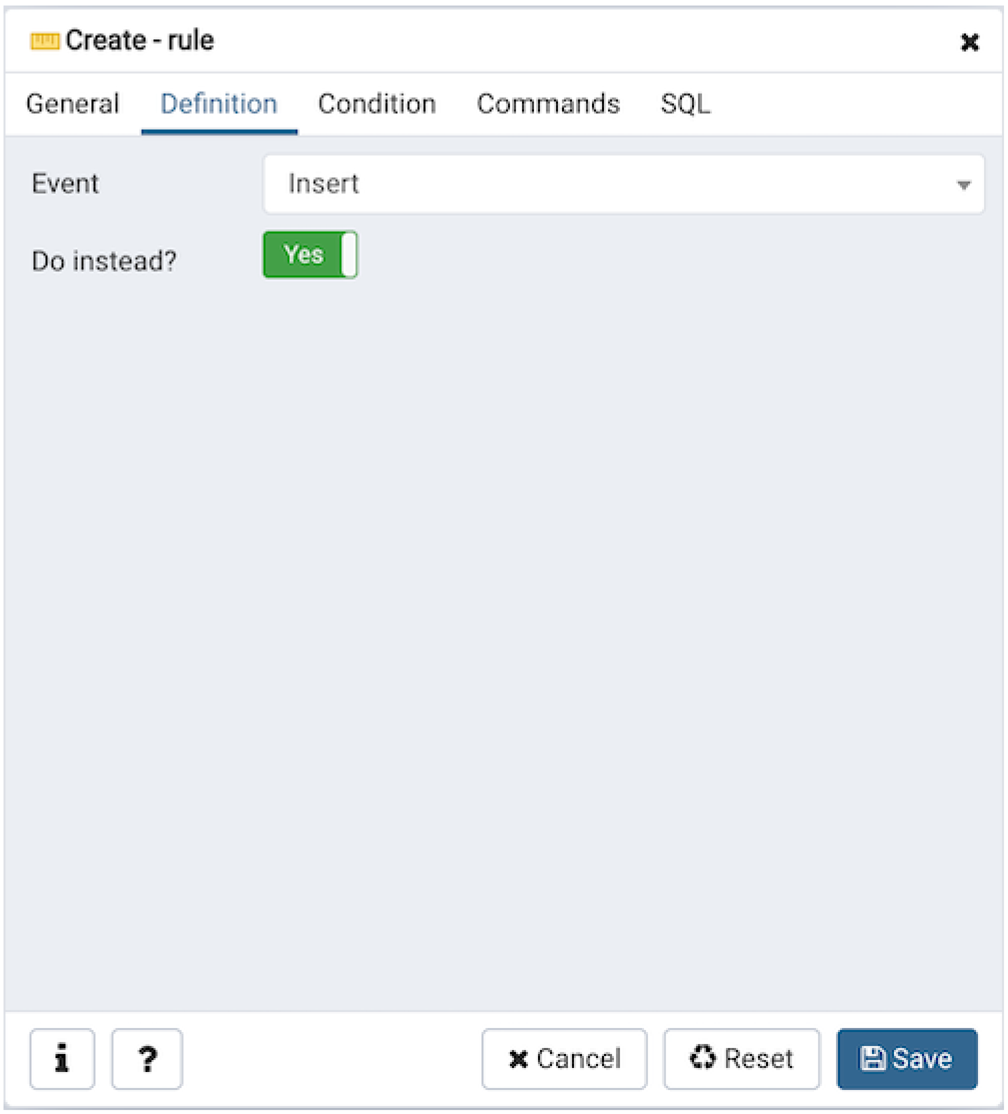


Name: replace_expression_answer
Comment: Rule to replace expression answer

Use the fields in the *General* tab to identify the rule:

- Use the *Name* field to add a descriptive name for the rule. The name will be displayed in the *pgAdmin* tree control. Multiple rules on the same table are applied in alphabetical name order.
- Store notes about the rule in the *Comment* field.

Click the *Definition* tab to continue.



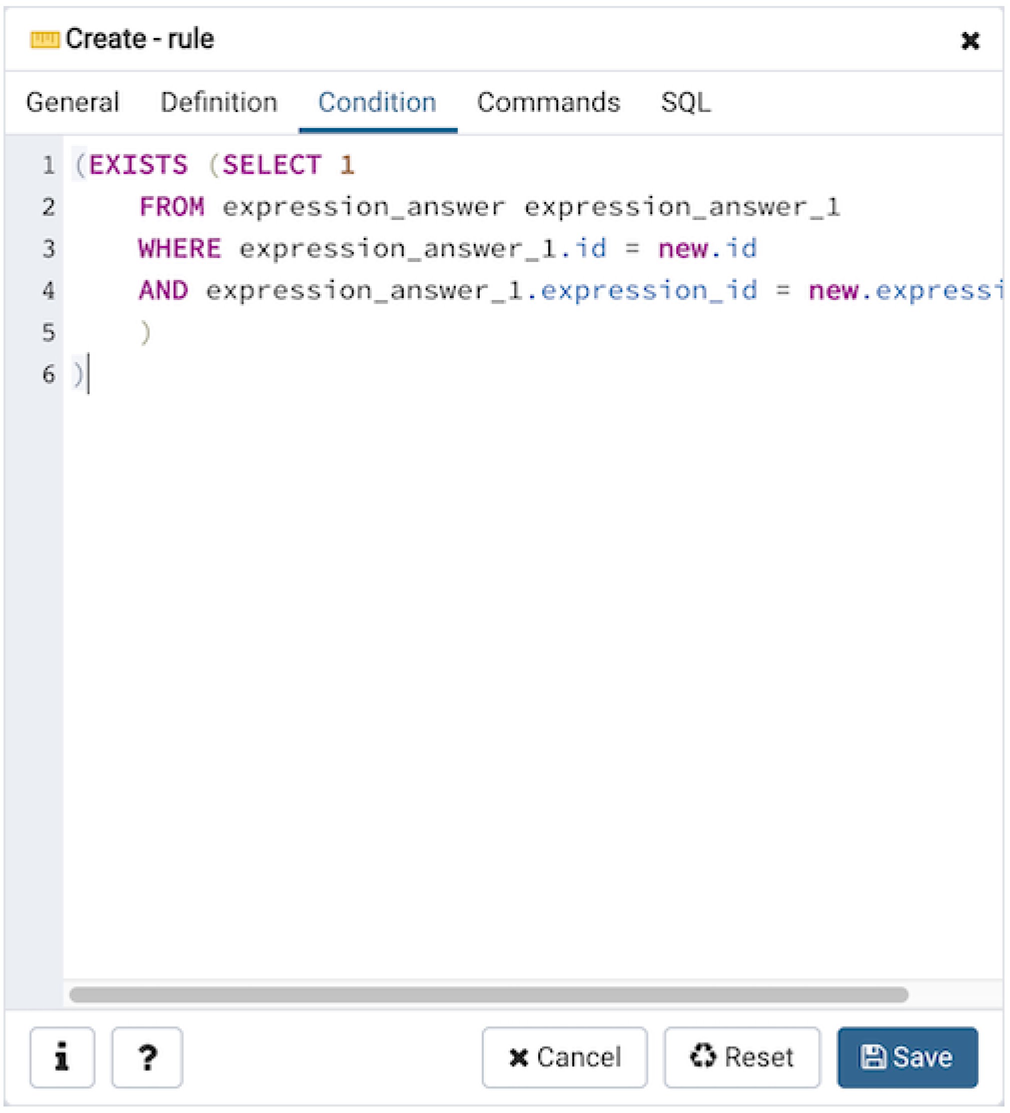
Event: Insert
Do instead?: Yes

Use the fields in the *Definition* tab to write parameters:

- Click inside the *Event* field to select the type of event that will invoke the rule; event may be *Select*, *Insert*, *Update*, or *Delete*.

- Move the *Do Instead* switch to *Yes* indicate that the commands should be executed instead of the original command; if Do Instead specifies *No*, the rule will be invoked in addition to the original command.

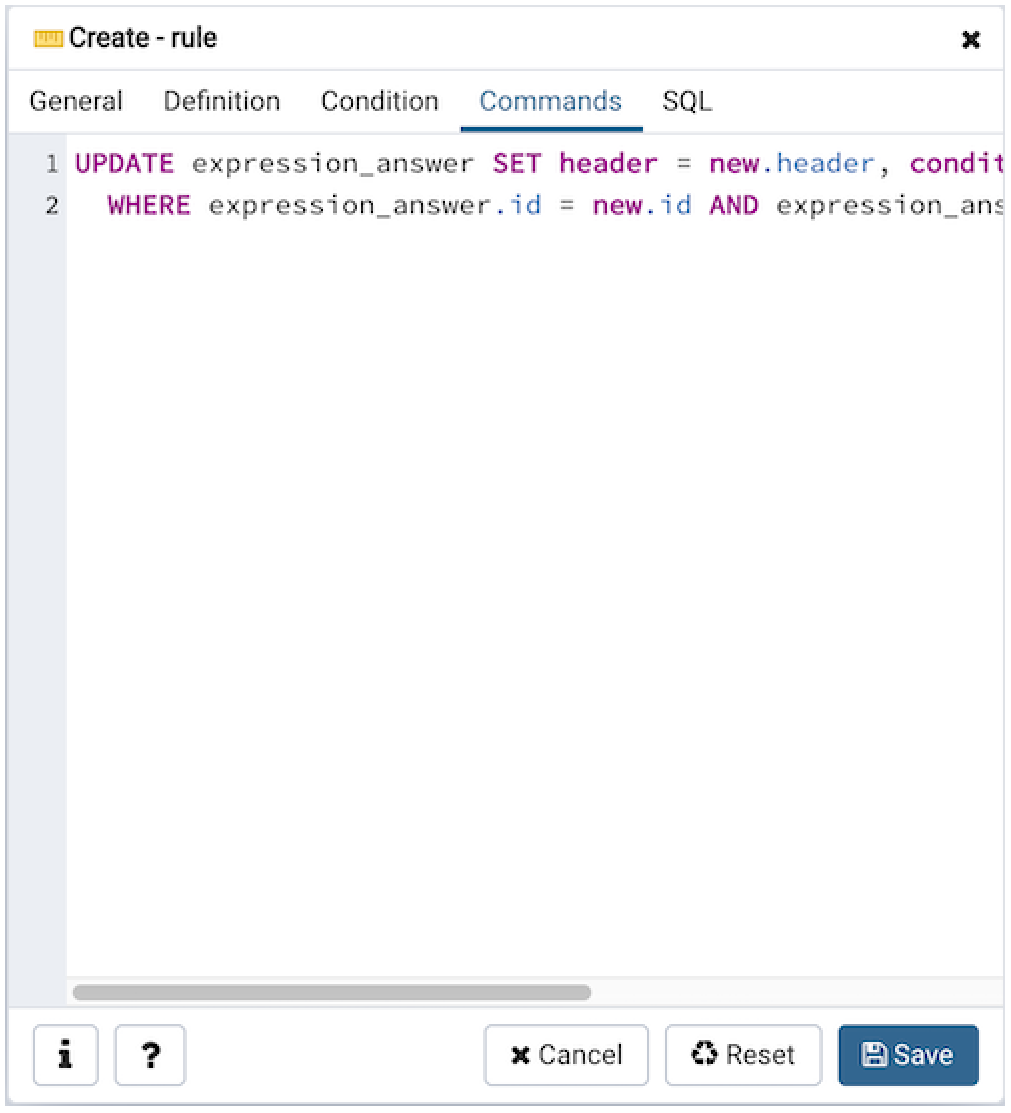
Click the *Condition* tab to continue.



```
(EXISTS (SELECT 1
    FROM expression_answer expression_answer_1
    WHERE expression_answer_1.id = new.id
    AND expression_answer_1.expression_id = new.expressi
    )
)
```

Specify a SQL conditional expression that returns a boolean value in the editor.

Click the *Commands* tab to continue.



```
UPDATE expression_answer SET header = new.header, condit
  WHERE expression_answer.id = new.id AND expression_ans
```

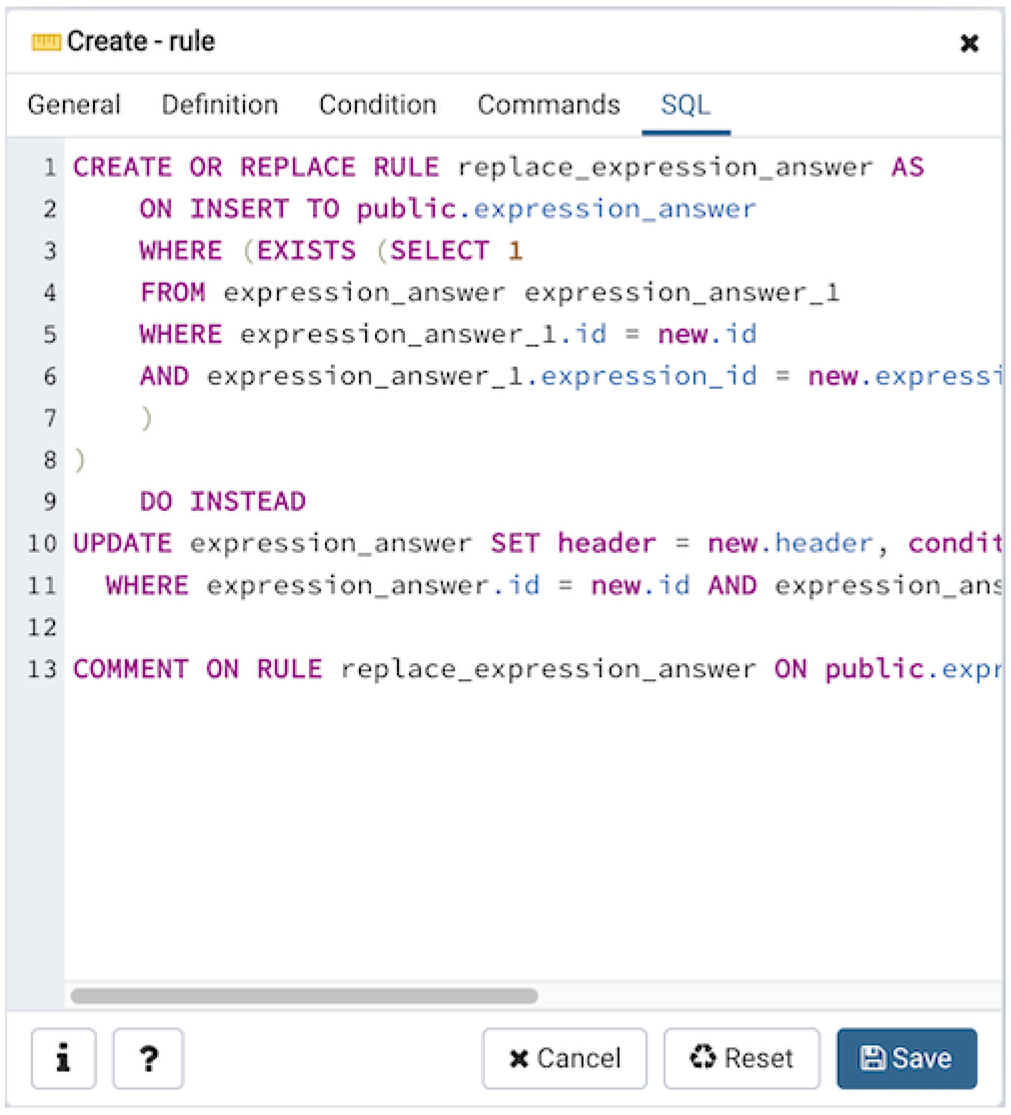
Provide a command in the editor that defines the action performed by the rule.

Click the *SQL* tab to continue.

Your entries in the *Rule* dialog generate a SQL command (see an example below). Use the *SQL* tab for review; revisit or switch tabs to make any changes to the SQL command.

Example ¶

The following is an example of the sql command generated by user selections in the *Rule* dialog:



```
CREATE OR REPLACE RULE replace_expression_answer AS
    ON INSERT TO public.expression_answer
    WHERE (EXISTS (SELECT 1
    FROM expression_answer expression_answer_1
    WHERE expression_answer_1.id = new.id
    AND expression_answer_1.expression_id = new.expressi
    )
)
    DO INSTEAD
UPDATE expression_answer SET header = new.header, condit
  WHERE expression_answer.id = new.id AND expression_ans

COMMENT ON RULE replace_expression_answer ON public.expr
```

The example sends a notification when an UPDATE executes against a table.

- Click the *Info* button (i) to access online help.
- Click the *Save* button to save work.
- Click the *Cancel* button to exit without saving work.
- Click the *Reset* button to restore configuration parameters.