159
DBMS_REDACT

The DBMS_REDACT package provides an interface to Oracle Data Redaction, which enables you to redact data that is returned from queries issued by low-privileged users or an application.

This chapter contains the following topics:

> **See Also:**
>
> - *Oracle Database Data Redaction Guide* regarding using Data Redaction to protect sensitive data

DBMS_REDACT Overview

Data redaction provides a way to define redaction policies for an application. Oracle Data Redaction provides functionality to redact data that is returned from user SELECT queries in an application. The redaction takes place in real time.

The Data Redaction policy applies to the querying user, depending on this user's SYS_CONTEXT or XS_SYS_CONTEXT values. This redaction process does not require that the queried data be static or unchanging, or for the entire data set to be redacted at one time in an off-line manner. Oracle Database redacts the data only for the rows specified by the user's query, not the data for the entire column. The redaction takes place immediately before the data is returned to the querying user or application.

DBMS_REDACT Security Model

If the querying user has the EXEMPT REDACTION POLICY privilege, redaction will not be performed. If the user does not have the EXEMPT REDACTION POLICY privilege, the policy expression will be evaluated in the current user's environment. If the policy expression evaluates to TRUE, then redaction will be performed, otherwise no redaction will be performed.

You must have the EXECUTE privilege on the DBMS_REDACT package, as well as the ADMINISTER REDACTION POLICY system or schema privilege, to execute its subprograms. Procedures in the interface are executed with privileges of the current user.

DBMS_REDACT Constants

The DBMS_REDACT package defines several constants for specifying parameter values.

Table 159-1 Values for function_type Parameter of DBMS_REDACT.ADD_POLICY

Constant	Value	Type	Description
FULL	1	BINARY_INTEGER	Redact to fixed values
PARTIAL	2	BINARY_INTEGER	Partial redaction, redact a portion of the column data
RANDOM	4	BINARY_INTEGER	Random redaction, each query results in a different random value
REGEXP	5	BINARY_INTEGER	Regular expression based redaction
REGEXP_WIDTH	7	BINARY_INTEGER	Regular expression based redaction that preserves the width of a column that uses a regular expression; designed for applications that use the OCI_ATTR_CHAR_SIZE attribute of the Oracle OLE DB Provider interface
NULLIFY	6	BINARY_INTEGER	Returns a null value as a redacted value
NONE	0	BINARY_INTEGER	No redaction

Table 159-2 Values for action Parameter of DBMS_REDACT.ALTER_POLICY

Constant	Value	Type	Description
ADD_COLUMN	1	BINARY_INTEGER	Add a column to the redaction policy
DROP_COLUMN	2	BINARY_INTEGER	Drop a column from the redaction policy
MODIFY_EXPRESSION	3	BINARY_INTEGER	Modify the expression of a redaction policy (the expression evaluates to a BOOLEAN value: if TRUE then redaction is applied, otherwise not)
MODIFY_COLUMN	4	BINARY_INTEGER	Modify a column in the redaction policy to change the redaction function_type or the function_parameters
SET_POLICY_DESCRIPTION	5	BINARY_INTEGER	Set a description for the redaction policy
SET_COLUMN_DESCRIPTION	6	BINARY_INTEGER	Set a description for the redaction performed on the column

DBMS_REDACT Operating Procedures

The following table presents the relationship between the type of redaction function and its parameters, based on the datatype of the column being redacted. Examples of the various format strings are provided, showing how to perform some commonplace redaction for a string datatype (in this case, a Social Security Number (SSN)), a DATE datatype, and various examples of redaction for the number datatype.

Table 159-3 Data Redaction Function Types

function_type	function_parameters	Examples
DBMS_REDACT.NONE	-	-
DBMS_REDACT.FULL	-	-
DBMS_REDACT.NULLIFY	-	-
DBMS_REDACT.PARTIAL (for character types)	A comma-separated list, containing the following five fields (with no spaces after the commas delimiting the fields): • REDACT_PARTIAL_INPUT_FORMAT • REDACT_PARTIAL_OUTPUT_FORMAT • REDACT_PARTIAL_MASKCHAR • REDACT_PARTIAL_MASKFROM • REDACT_PARTIAL_MASKTO See Table 159-4.	'VVVFVVFVVVV,VVV-VV-VVVV,X,1,5' for redacting the first 5 digits of SSN strings like 123-45-6789, adding dashes back to format it, resulting in strings like XXX-XX-6789 'VVVFVVFVVVV,VVV VV VVVV,X,1,5' for redacting the first 5 digits of SSN strings like 123-45-6789, adding spaces to format it, resulting in strings like XXX XX 6789
DBMS_REDACT.PARTIAL (for character types), continued	REDACT_PARTIAL_INPUT_FORMAT - the input format: V for value to be possibly redacted, F for formatting character to be ignored	The REDACT_PARTIAL_INPUT_FORMAT field value VVVFVVFVVVV for matching SSN strings like 123-45-6789
DBMS_REDACT.PARTIAL (for character types), continued	REDACT_PARTIAL_OUTPUT_FORMAT - the output format: V for output of redaction. Any other character will be treated as a formatting character and output literally.	The REDACT_PARTIAL_OUTPUT_FORMAT field value VVV-VV-VVVV can be used to redact SSN strings into XXX-XX-6789 (X comes from REDACT_PARTIAL_MASKCHAR field).
DBMS_REDACT.PARTIAL (for character types), continued	REDACT_PARTIAL_MASKCHAR - the character used to redact the input	The value X for redacting SSN strings into XXX-XX-6789.
DBMS_REDACT.PARTIAL (for character types), continued	REDACT_PARTIAL_MASKFROM - specifies which V within the REDACT_PARTIAL_INPUT_FORMAT from which to start the redaction (see explanation following the next entry, REDACT_PARTIAL_MASKTO)	The value 1 for redacting SSN strings starting at the first V of REDACT_PARTIAL_INPUT_FORMAT of VVVFVVFVVVV into strings like XXX-XX-6789

Table 159-3 (Cont.) Data Redaction Function Types

function_type	function_parameters	Examples
DBMS_REDACT.PARTIAL (for character types), continued	REDACT_PARTIAL_MASKTO - specifies which V within the REDACT_PARTIAL_INPUT_FORMAT at which to end the redaction	The value 5 for redacting SSN strings up to and including the fifth V within REDACT_PARTIAL_INPUT_FORMAT of VVVFVVFVVVV into strings like XXX-XX-6789. Note how the format character '-' (corresponding to the first F within REDACT_PARTIAL_INPUT_FORMAT) is ignored as far as redaction is concerned, so the value here is 5 as opposed to 6.
DBMS_REDACT.PARTIAL (for character types), continued	The REDACT_PARTIAL_MASKFROM and REDACT_PARTIAL_MASKTO field values are specified as counts of the number of V characters in the REDACT_PARTIAL_INPUT_FORMAT field, up to and including the intended position, starting from the leftmost V. This way, REDACT_PARTIAL_MASKFROM and REDACT_PARTIAL_MASKTO are independent of the specific formatting of the data. For example, in the common use case of redacting an SSN to show only the last four digits, data like 123456789 (with REDACT_PARTIAL_INPUT_FORMAT of VVVVVVVVV) and data like 123-45-6789 (with REDACT_PARTIAL_INPUT_FORMAT of VVVFVVFVVVV), would both use REDACT_PARTIAL_MASKFROM of 1 and REDACT_PARTIAL_MASKTO of 5.	-
DBMS_REDACT.PARTIAL (for number types)	A comma-separated list, containing the following three fields (with no spaces after the commas delimiting the fields): • REDACT_PARTIAL_MASKCHAR • REDACT_PARTIAL_MASKFROM • REDACT_PARTIAL_MASKTO See Table 159-4.	'9,1,5' for redacting the first 5 digits of an SSN number 123456789 into 999996789; or '0,1,2' for redacting a number 1.23 to 0.03.

Table 159-3 (Cont.) Data Redaction Function Types

function_type	function_parameters	Examples
DBMS_REDACT.PARTIAL (for number types), continued	REDACT_PARTIAL_MASKCHAR - the character used to redact the input, in the range between 0 and 9 REDACT_PARTIAL_MASKFROM - the position, starting from 1, from which to start the redaction. The position does not include the decimal point if it is present. REDACT_PARTIAL_MASKTO - the position at which to end the redaction	-
DBMS_REDACT.PARTIAL (for datetime datatypes)	A list, containing the following five fields (concatenated so that there is no space between the fields): • REDACT_PARTIAL_DATE_MONTH • REDACT_PARTIAL_DATE_DAY • REDACT_PARTIAL_DATE_YEAR • REDACT_PARTIAL_DATE_HOUR • REDACT_PARTIAL_DATE_MINUTE • REDACT_PARTIAL_DATE_SECOND See Table 159-4.	'm12DYHMS', which changes 01-May-01 01:01:01 to 01-Dec-01 01:01:01.

Table 159-3 (Cont.) Data Redaction Function Types

function_type	function_parameters	Examples
DBMS_REDACT.PARTIAL (for datetime datatypes), continued	REDACT_PARTIAL_DATE_MONTH: 'M' (no redacting of month) or 'm#' (redact month to a specific month, if possible), where # (the month specified by its number) is between 1 and 12 REDACT_PARTIAL_DATE_DAY: 'D' (no redacting of date) or 'd#' (redact day to #, if possible), # between 1 and 31 REDACT_PARTIAL_DATE_YEAR: 'Y' (no redacting of year) or 'y#' (redact year to #, if possible), # between 1 and 9999 REDACT_PARTIAL_DATE_HOUR: 'H' (no redacting of hour) or 'h#' (redact hour to #, if possible), # between 0 and 23 REDACT_PARTIAL_DATE_MINUTE: 'M' (no redacting of minute) or 'm#' (redact minute to #, if possible), # between 0 and 59 REDACT_PARTIAL_DATE_SECOND: 'S' (no redacting of second) or 's#' (redact second to #, if possible), # between 0 and 59	-
DBMS_REDACT.REGEXP	-	-
DBMS_REDACT.REGEXP_WIDTH	-	-
DBMS_REDACT.RANDOM	-	-

Table 159-4 Format Descriptors with Component Field Names and Delimiters

Datatype	Format Descriptor for Partial redaction
Character	REDACT_PARTIAL_INPUT_FORMAT || ',' || REDACT_PARTIAL_OUTPUT_FORMAT || ',' || REDACT_PARTIAL_MASKCHAR || ',' || REDACT_PARTIAL_MASKFROM || ',' || REDACT_PARTIAL_MASKTO
Number	REDACT_PARTIAL_MASKCHAR || ',' || REDACT_PARTIAL_MASKFROM || ',' || REDACT_PARTIAL_MASKTO
Datetime	REDACT_PARTIAL_DATE_MONTH || REDACT_PARTIAL_DATE_DAY || REDACT_PARTIAL_DATE_YEAR || REDACT_PARTIAL_DATE_HOUR || REDACT_PARTIAL_DATE_MINUTE || REDACT_PARTIAL_DATE_SECOND

Summary of DBMS_REDACT Subprograms

This table lists and briefly describes the `DBMS_REDACT` package subprograms.

Table 159-5 DBMS_REDACT Package Subprograms

Subprogram	Description
ADD_POLICY Procedure	Defines a Data Redaction policy for a table or view
ALTER_POLICY Procedure	Alters a Data Redaction policy for a table or view
APPLY_POLICY_EXPR_TO_COL Procedure	Applies a Data Redaction named policy expression to a redacted column
CREATE_POLICY_EXPRESSION Procedure	Creates a Data Redaction named policy expression
DISABLE_POLICY Procedure	Disables a Data Redaction policy
DROP_POLICY Procedure	Drops a Data Redaction policy
DROP_POLICY_EXPRESSION Procedure	Drops a Data Redaction named policy expression
ENABLE_POLICY Procedure	Enables a Data Redaction policy
UPDATE_FULL_REDACTION_VALUES Procedure	Modifies the default displayed values for a Data Redaction policy for full redaction
UPDATE_POLICY_EXPRESSION Procedure	Updates a Data Redaction named policy expression

ADD_POLICY Procedure

This procedure defines a Data Redaction policy for a table or view.

Syntax

```
DBMS_REDACT.ADD_POLICY (
   object_schema               IN    VARCHAR2 := NULL,
   object_name                 IN    VARCHAR2,
   policy_name                 IN    VARCHAR2,
   column_name                 IN    VARCHAR2 := NULL,
   function_type               IN    BINARY_INTEGER := DBMS_REDACT.FULL,
   function_parameters         IN    VARCHAR2 := NULL,
   expression                  IN    VARCHAR2,
   enable                      IN    BOOLEAN := TRUE,
   regexp_pattern              IN    VARCHAR2 := NULL,
   regexp_replace_string       IN    VARCHAR2 := NULL,
   regexp_position             IN    BINARY_INTEGER := 1,
   regexp_occurrence           IN    BINARY_INTEGER := 0,
   regexp_match_parameter      IN    VARCHAR2 := NULL,
   policy_description          IN    VARCHAR2 := NULL,
   column_description          IN    VARCHAR2 := NULL);
```

Parameters

Table 159-6 *ADD_POLICY Procedure Parameters*

Parameter	Description
`object_schema`	Schema owning the table or view, current user if `NULL`
`object_name`	Name of table or view on which to add a Data Redaction policy
`policy_name`	Name of policy
`column_name`	[Optional] Name of one column to which the redaction policy applies. If you must redact more than one column, use the ALTER_POLICY Procedure to add the additional columns.
`function_type`	Type of redaction function to use. Possible values are: - `DBMS_REDACT.NONE` - `DBMS_REDACT.FULL` (default) - `DBMS_REDACT.NULLIFY` - `DBMS_REDACT.PARTIAL` - `DBMS_REDACT.RANDOM` - `DBMS_REDACT.REGEXP` - `DBMS_REDACT.REGEXP_WIDTH` See Table 159-1 for an overview of the meanings of these values, and for some examples of their use.

Table 159-6 (Cont.) *ADD_POLICY Procedure Parameters*

Parameter	Description
function_parameters	Parameters to the redaction function. The possible values depend on the value of the function_type provided. Use function_parameters parameter only for DBMS_REDACT.PARTIAL. Use regexp_* parameters only for DBMS_REDACT.REGEXP or DBMS_REDACT.REGEXP_WIDTH to define the Data Redaction policy. If the function_type is DBMS_REDACT.REGEXP or DBMS_REDACT.REGEXP_WIDTH, then you must omit the function_parameters parameter, and use the regexp_* parameters to define the Data Redaction policy. - DBMS_REDACT.NONE: Can be omitted entirely and defaults to NULL - DBMS_REDACT.FULL: Can be omitted entirely and defaults to NULL - DBMS_REDACT.NULLIFY: Can be omitted entirely and defaults to NULL - DBMS_REDACT.RANDOM: Can be omitted entirely and defaults to NULL - Redacting parameters for partial character redacting. For character datatypes, a comma-separated list containing these fields: • Input format: 'V' for value to be possibly redacted, 'F' for formatting character to be ignored • Output format: 'V' for output of redacting, any other characters will be treated as formatting characters. • Redact character: a character that will be used to replace the actual values. Examples are '*' and 'x'. • Starting digit position: specifies the starting (character) position to begin replacing actual values with the redacting character. The beginning of the string is position 1. Positions do not include formatting characters. • Ending digit position: specifies the ending (character) position to end redacting. An example is 'VVVFVVFVVVV,VVV-VV-VVVV,X,1,5' for redacting the first 5 digits of SSN string 123-45-6789, and adding dashes back to format it like an SSN, resulting in XXX-XX-6789. For number datatypes, a comma-separated list containing these fields: • Redact character: this is a character between '0' to '9' that will be used to replace the actual values. • Starting digit position: specifies the starting (digit) position to begin replacing actual values with the redacting character. The beginning of the string is position 1. Positions do not include the decimal point. • Ending digit position: this specifies the ending digit position to end redacting. An example is '9,1,5' for redacting the first 5 digits of Social Security Number 123456789, resulting in 999996789. For datetime datatypes, the format is a packed string (no spaces or commas) containing the following sequence of fields. Please note that each field can consist of one or more characters, and the field length depends on whether redacting is required. The one-character fields are used to specify that no redaction of that component of the datetime value is to take place. The longer fields indicate a specific

Table 159-6 (Cont.) *ADD_POLICY Procedure Parameters*

Parameter	Description
	time or date to use as the redacted value of that component of the datetime value. • Month: `'M'` (no redacting of month) or `'m#'` (redact month to a specific month, if possible), where `#` (the month specified by its number) is between `1` and `12` • Day: `'D'` (no redacting of day) or `'d#'` (redact day to `#`, if possible), `#` between `1` and `31` • Year: `'Y'` (no redacting of year) or `'y#'` (redact year to `#`, if possible), `#` between `1` and `9999` • Hour: `'H'` (no redacting of hour) or `'h#'` (redact hour to `#`, if possible), `#` between `0` and `23` • Minute: `'M'` (no redacting of minute) or `'m#'` (redact minute to `#`, if possible), `#` between `0` and `59` • Second: `'S'` (no redacting of second) or `'s#'` (redact second to `#`, if possible), `#` between `0` and `59` An example is `'m12d1y2001h1m1s1'`, which changes `02-May-13 12:30:23` to `01-Dec-01 01:01:01`. For partial character and number-redacting shortcuts, see *Oracle Database Data Redaction Guide*.
`expression`	Default boolean expression for the table or view. If this expression is used, then redaction takes place only if this policy expression evaluates to `TRUE`. The following functions are supported: • `SYS_CONTEXT` • `XS_SYS_CONTEXT` • `SUBSTR` functions (includes `SUBSTRB`, `SUBSTRC`, `SUBSTR2`, `SUBSTR4`) • `LENGTH` functions (includes `LENGTHB`, `LENGTHC`, `LENGTH2`, `LENGTH4`) • `V` (`APEX_UTIL.GET_SESSION_STATE`) • `NV` (`APEX_UTIL.GET_NUMERIC_SESSION_STATE`) • `OLS_LABEL_DOMINATES` • `DOMINATES` • `OLS_DOMINATES` • `OLS_DOM` • `DOM` • `OLS_STRICTLY_DOMINATES` • `STRICTLY_DOMINATES` • `S_DOM` • `SA_UTL.DOMINATES` • `SA_UTL.CHECK_READ` • `SA_UTL.NUMERIC_LABEL` • `CHAR_TO_LABEL` • `SA_SESSION.LABEL` See *Oracle Database Data Redaction Guide* for more information about these supported functions

Table 159-6 (Cont.) *ADD_POLICY Procedure Parameters*

Parameter	Description
`enable`	Boolean value that determines whether the Data Redaction policy is enabled on creation. The default value is `TRUE`, which means that the policy is automatically enabled upon creation. If the `enable` parameter is set to `FALSE`, the policy takes effect only when it is subsequently enabled by calling the `DBMS_REDACT.ENABLE_POLICY` procedure.
`regexp_pattern`	Regular expression pattern up to 512 bytes. Use only if the `function_type` parameter is `DBMS_REDACT.REGEXP` or `DBMS_REDACT.REGEXP_WIDTH`. Also, do not specify the `function_parameters` parameter when `function_type` is `DBMS_REDACT.REGEXP` or `DBMS_REDACT.REGEXP_WIDTH`. See *Oracle Database SQL Language Reference* for more information and examples on using regular expression patterns.
`regexp_replace_string`	Replacement string (up to 4000 characters in length) with up to 500 back-references to subexpressions in the form `\n`, where `n` is a number between 1 and 9. Use only if the `function_type` parameter is `DBMS_REDACT.REGEXP`
`regexp_position`	Integer counting from 1, specifies the position where the search must begin. The default is 1. Use only if the `function_type` parameter is `DBMS_REDACT.REGEXP` or `DBMS_REDACT.REGEXP_WIDTH`.
`regexp_occurrence`	• Use 0 to replace all occurrences of the match. The default is 0. • Use positive integer *n* to replace the *n*-th occurence of the match. Use only if the `function_type` parameter is `DBMS_REDACT.REGEXP` or `DBMS_REDACT.REGEXP_WIDTH`.
`regexp_match_parameter`	Changes the default matching behavior, possible values are a combination of 'i', 'c', 'n', 'm', 'x' Use only if the `function_type` parameter is `DBMS_REDACT.REGEXP` or `DBMS_REDACT.REGEXP_WIDTH`. See *Oracle Database SQL Language Reference* for more information and examples on using regular expression match parameters.
`policy_description`	Description of redaction policy
`column_description`	Description of the column being redacted

Exceptions

- `ORA-28060` - A Data Redaction policy already exists on this column.
- `ORA-28061` - This object cannot have a Data Redaction policy defined on it.
- `ORA-28062` - The policy expression is too long.
- `ORA-28063` - The policy expression is empty.
- `ORA-28064` - The redaction function is not valid.
- `ORA-28066` - Invalid column *`column`*.
- `ORA-28067` - Missing or invalid column name.

- ORA-28069 - A Data Redaction policy already exists on this object.
- ORA-28073 - The column *column_name* has an unsupported datatype or attribute.
- ORA-28074 - The *field_name* field of the redaction parameters is not valid.

  The field can be any of the following:

  - REDACT_PARTIAL_INPUT_FORMAT
  - REDACT_PARTIAL_OUTPUT_FORMAT
  - REDACT_PARTIAL_MASKCHAR
  - REDACT_PARTIAL_MASKFROM
  - REDACT_PARTIAL_MASKTO
  - REDACT_PARTIAL_DATE_MONTH
  - REDACT_PARTIAL_DATE_DAY
  - REDACT_PARTIAL_DATE_YEAR
  - REDACT_PARTIAL_DATE_HOUR
  - REDACT_PARTIAL_DATE_MINUTE
  - REDACT_PARTIAL_DATE_SECOND

  See Table 159-3 and Table 159-4 for examples of the field contents and field ordering.
- ORA-28075 - The data redaction policy expression had an error.
- ORA-28076 - Empty role name specified as SYS_SESSION_ROLES attribute.
- ORA-28077 - The specified role name *role_name* for SYS_SESSION_ROLES exceeds the maximum length.
- ORA-28078 - A regular expression parameter is missing or invalid.
- ORA-28082 - The parameter *parameter* is invalid (where the possible values are function_parameters, column_description, and policy_description).
- ORA-28085 - The input and output lengths of the redaction do not match.
- ORA-28086 - The data redaction policy expression had an error
- ORA-28087 - The policy expression has an unsupported (use of) operator <operator>.
- ORA-28088 - The policy expression has an unsupported PL/SQL function function_name.
- ORA-28089 - Invalid SYS_CONTEXT namespace <namespace> in policy expression.
- ORA-28090 - Unsupported nesting of function function_name within function function_name in policy expression.
- ORA-28091 - Unsupported use of PL/SQL function function_name in policy expression.
- ORA-28092 - The parameter parameter_name with value value has an error.
- ORA-28096 - The parameter parameter_name is unexpected for ADD_POLICY.
- ORA-28097 - A data redaction policy cannot be applied to an object owned by SYS.
- ORA-28104 - Input value for *parameter* is not valid.

Usage Notes

If you specify `1=1` for the expression parameter, then it is always `TRUE`, which causes the redaction to be performed as long as the querying user is not exempt from the Oracle Data Redaction policy. Specifying `1=1` for the expression parameter will improve Data Redaction performance because Oracle Database will not evaluate the policy expression.

See DBMS_REDACT Security Model for a list of the required privileges for this procedure.

See Operating Procedures for more information regarding function types and function parameters with related examples.

The first time you add a full redaction policy to a boolean column after the `COMPATIBLE` database initialization parameter is set to 23 or higher, you will see the `BOOLEAN_VALUE` column appear in the catalog view for `REDACTION_VALUES_FOR_TYPE_FULL` and the `BOOLCOL` column appear in the `SYS.RADM_FPTM$` data dictionary table.

A Data Redaction named policy expression that has been applied to a redacted column takes precedence over the expression defined in the `expression` parameter. To find redacted columns that are affected by named policy expressions, query the `REDACTION_EXPRESSIONS` data dictionary view.

Example

Partial redaction policy:

```
BEGIN
   DBMS_REDACT.ADD_POLICY(
     object_schema        => 'hr',
     object_name          => 'employees',
     column_name          => 'employee_id',
     policy_name          => 'redact_emp_id_nums',
     function_type        => DBMS_REDACT.PARTIAL,
     function_parameters  => '7,1,5',
     expression           => '1=1');
END;
```

Full redaction policy:

```
BEGIN
   DBMS_REDACT.ADD_POLICY(
     object_schema        => 'hr',
     object_name          => 'employees',
     column_name          => 'employee_id',
     policy_name          => 'redact_emp_ids',
     function_type        => DBMS_REDACT.FULL,
     expression           => 'SYS_CONTEXT(''SYS_SESSION_ROLES'',''CLERK'')
                                                                    = ''FALSE''');
END;
```

ALTER_POLICY Procedure

This procedure alters an existing Data Redaction policy for a table or view.

It alters the policy in one or more of the following ways:

- By changing the policy expression
- By changing the type of redaction for a specified column
- By changing the parameters to the redaction function for a specified column

- By adding a column to the redaction policy (the redaction type and any parameters must be specified).
- By removing a column from the redaction policy
- By changing the description of the policy
- By changing the description of the column

Syntax

```
DBMS_REDACT.ALTER_POLICY (
   object_schema                 IN    VARCHAR2 := NULL,
   object_name                   IN    VARCHAR2,
   policy_name                   IN    VARCHAR2,
   action                        IN    BINARY_INTEGER := DBMS_REDACT.ADD_COLUMN,
   column_name                   IN    VARCHAR2 := NULL,
   function_type                 IN    BINARY_INTEGER := DBMS_REDACT.FULL,
   function_parameters           IN    VARCHAR2 := NULL,
   expression                    IN    VARCHAR2,
   regexp_pattern                IN    VARCHAR2 := NULL,
   regexp_replace_string         IN    VARCHAR2 := NULL,
   regexp_position               IN    BINARY_INTEGER := 1,
   regexp_occurrence             IN    BINARY_INTEGER := 0,
   regexp_match_parameter        IN    VARCHAR2 := NULL,
   policy_description            IN    VARCHAR2 := NULL,
   column_description            IN    VARCHAR2 := NULL);
```

Parameters

Table 159-7 ***ALTER_POLICY Procedure Parameters***

Parameter	Description
object_schema	Schema owning the table or view, current user if NULL
object_name	Name of table or view on which to alter a Data Redaction policy
policy_name	Name of policy limited to 30 bytes
action	Action to take. For more information see Table 159-2.
column_name	(Optional) Name of one column to which the redaction policy applies. Required for the following actions: • add_column • drop_column • modify_column • set_column_description

Table 159-7 (Cont.) *ALTER_POLICY Procedure Parameters*

Parameter	Description
function_type	Type of redaction function to use. Possible values are: - DBMS_REDACT.NONE - DBMS_REDACT.FULL (default) - DBMS_REDACT.NULLIFY - DBMS_REDACT.PARTIAL - DBMS_REDACT.RANDOM - DBMS_REDACT.REGEXP - DBMS_REDACT.REGEXP_WIDTH If the function_type is not DBMS_REDACT.PARTIAL, then you must omit the function_parameters parameter. If the function_type is not DBMS_REDACT.REGEXP or DBMS_REDACT.REGEXP_WIDTH, then you must omit the regexp_* parameters. If the function_type is DBMS_REDACT.REGEXP or DBMS_REDACT.REGEXP_WIDTH, then use the regexp_pattern, regexp_replace_string, regexp_position, regexp_occurrence, and regexp_match_parameter to define the Data Redaction policy. See Table 159-1 for an overview of the meanings of these values, and for some examples of their use.

Table 159-7 (Cont.) *ALTER_POLICY Procedure Parameters*

Parameter	Description
function_parameters	Parameters to the redaction function. Use only if function_type is DBMS_REDACT.PARTIAL. If the function_type is DBMS_REDACT.REGEXP or DBMS_REDACT.REGEXP_WIDTH, then you must omit the function_parameters parameter, and use the regexp_pattern, regexp_replace_string, regexp_position, regexp_occurrence, and regexp_match_parameter to define the Data Redaction policy. - If the function_type is DBMS_REDACT.NONE or DBMS_REDACT.NULLIFY, the function_parameters parameter must be omitted. - If the function_type is DBMS_REDACT.FULL or DBMS_REDACT.RANDOM, the function_parameters parameter must be omitted. - If the function_type is DBMS_REDACT.PARTIAL, the function_parameters parameter represents the redacting parameters for partial redacting. • Input format: 'V' for value to be possibly redacted, 'F' for formatting character to be ignored • Output format: 'V' for output of redaction, any other characters will be treated as formatting characters • Redact character: a character that will be used to replace the actual values. Examples are '*' and 'x' • Starting digit position: specifies the starting (character) position to begin replacing actual values with the redacting character. The beginning of the string is position 1. Positions do not include formatting characters. • Ending digit position: specifies the ending (character) position to end redacting. An example is 'VVVFVVFVVVV,VVV-VV-VVVV,X,1,5' for redacting the first 5 digits of SSN string 123-45-6789, and adding dashes back to format it like an SSN, resulting in XXX-XX-6789. For number datatypes, a comma-separated list containing these fields: • Redact character: this is a character between '0' to '9' that will be used to replace the actual values. • Starting digit position: specifies the starting (digit) position to begin replacing actual values with the redacting character. The beginning of the string is position 1. Positions do not include the decimal point. • Ending digit position: this specifies the ending digit position to end redacting. An example is '9,1,5' for redacting the first 5 digits of Social Security Number 123456789, resulting in 999996789. For datetime datatypes, the format is a packed string (no spaces or commas) containing the following sequence of fields. Please note that each field can consist of one or more characters, and the field length depends on whether redacting is required. The one-character fields are used to specify that no redaction of that component of the datetime value is to take place. The longer fields indicate a specific time or date to use as the redacted value of that component of the datetime value.

Table 159-7 (Cont.) *ALTER_POLICY Procedure Parameters*

Parameter	Description
	• Month: `'M'` (no redacting of month) or `'m#'` (redact month to a specific month, if possible), where `#` (the month specified by its number) is between `1` and `12`. • Day: `'D'` (no redacting of date) or `'d#'` (redact day to `#`, if possible), `#` between `1` and `31`. • Year: `'Y'` (no redacting of year) or `'y#'` (redact year to `#`, if possible), `#` between `1` and `9999`. • Hour: `'H'` (no redacting of hour) or `'h#'` (redact hour to `#`, if possible), `#` between `0` and `23`. • Minute: `'M'` (no redacting of minute) or `'m#'` (redact minute to `#`, if possible), `#` between `0` and `59`. • Second: `'S'` (no redacting of second) or `'s#'` (redact second to `#`, if possible), `#` between `0` and `59`. An example is `'m12DYHMS'`, which changes `01-May-01 01:01:01` to `01-Dec-01 01:01:01` For partial character and number-redacting shortcuts, see *Oracle Database Data Redaction Guide*.
`expression`	Default boolean expression for the table or view. Redaction takes place only if this policy expression evaluates to `TRUE`. The following functions are supported: • `SYS_CONTEXT` • `XS_SYS_CONTEXT` (See *Oracle Database Data Redaction Guide* • `NV` (`APEX_UTIL.GET_NUMERIC_SESSION_STATE`) • `V` (`APEX_UTIL.GET_SESSION_STATE`) • `OLS_LABEL_DOMINATES` • `DOMINATES` • `OLS_DOMINATES` • `OLS_DOM` • `DOM` • `OLS_STRICTLY_DOMINATES` • `STRICTLY_DOMINATES` • `S_DOM` • `SA_UTL.DOMINATES` • `SA_UTL.CHECK_READ` • `SA_UTL.NUMERIC_LABEL` • `CHAR_TO_LABEL` • `SA_SESSION.LABEL`
`regexp_pattern`	Regular expression pattern up to 512 bytes. Use only if the `function_type` parameter is `DBMS_REDACT.REGEXP` or `DBMS_REDACT.REGEXP_WIDTH`. Also, do not specify the `function_parameters` parameter when `function_type` is `DBMS_REDACT.REGEXP` or `DBMS_REDACT.REGEXP_WIDTH`. See *Oracle Database SQL Language Reference* for more information and examples on using regular expression patterns.

Table 159-7 (Cont.) *ALTER_POLICY Procedure Parameters*

Parameter	Description
`regexp_replace_string`	Replacement string (up to 4000 characters in length) with up to 500 back-references to subexpressions in the form `\n`, where `n` is a number between 1 and 9. Use only if the `function_type` parameter is `DBMS_REDACT.REGEXP` or `DBMS_REDACT.REGEXP_WIDTH`.
`regexp_position`	Integer counting from 1, specifies the position where the search must begin. Use only if the `function_type` parameter is `DBMS_REDACT.REGEXP` or `DBMS_REDACT.REGEXP_WIDTH`. The default is 1 meaning that Oracle Database begins the search at the first character of the `column_name` data.
`regexp_occurrence`	• Use 0 to replace all occurrences of the match. 0 is the default. • Use positive integer *`n`* to replace the *`n`*-th occurence of the match. Use only if the `function_type` parameter is `DBMS_REDACT.REGEXP` or `DBMS_REDACT.REGEXP_WIDTH`.
`regexp_match_parameter`	Changes the default matching behavior, possible values are a combination of 'i', 'c', 'ic', 'iq', 'n', 'm', 'x'. The behavior of this parameter is the same for this function as for the `REGEXP_REPLACE` SQL function. Use only if the `function_type` parameter is `DBMS_REDACT.REGEXP` or `DBMS_REDACT.REGEXP_WIDTH`. See *Oracle Database SQL Language Reference* for more information and examples on using regular expression match parameters.
`policy_description`	Description of redaction policy
`column_description`	Description of the column being redacted

Exceptions

- `ORA-28060` - A data redaction policy already exists on this column.
- `ORA-28061` - This object cannot have a data redaction policy defined on it.
- `ORA-28062` - The policy expression is too long.
- `ORA-28063` - The policy expression is empty.
- `ORA-28064` - The redaction function is not valid.
- `ORA-28066` - Invalid column *`column`*.
- `ORA-28067` - Missing or invalid column name.
- `ORA-28068` - The object *`object`* does not have a Data Redaction policy.
- `ORA-28069` - A Data Redaction policy already exists on this object.
- `ORA-28070` - The column *`column`* does not have a Data Redaction policy.
- `ORA-28071` - The action is not valid.
- `ORA-28072` - The specified policy name is incorrect.
- `ORA-28073` - The column *`column_name`* has an unsupported datatype or attribute.
- `ORA-28074` - The *`field_name`* field of the redaction parameters is not valid.

The field can be any of the following:

- REDACT_PARTIAL_INPUT_FORMAT
- REDACT_PARTIAL_OUTPUT_FORMAT
- REDACT_PARTIAL_MASKCHAR
- REDACT_PARTIAL_MASKFROM
- REDACT_PARTIAL_MASKTO
- REDACT_PARTIAL_DATE_MONTH
- REDACT_PARTIAL_DATE_DAY
- REDACT_PARTIAL_DATE_YEAR
- REDACT_PARTIAL_DATE_HOUR
- REDACT_PARTIAL_DATE_MINUTE
- REDACT_PARTIAL_DATE_SECOND

See Table 159-3 and Table 159-4 for examples of the field contents and field ordering.

- ORA-28075 - The data redaction policy expression had an error.
- ORA-28076 - Empty role name specified as SYS_SESSION_ROLES attribute.
- ORA-28077 - The specified role name *role_name* for SYS_SESSION_ROLES exceeds the maximum length.
- ORA-28078 - A regular expression parameter is missing or invalid.
- ORA-28082 - The parameter *parameter* is invalid (where the possible values are function_parameters, column_description, and policy_description).
- ORA-28085 - The input and output lengths of the redaction do not match.
- ORA-28086 - The data redaction policy expression had an error
- ORA-28087 - The policy expression has an unsupported (use of) operator <operator>.
- ORA-28088 - The policy expression has an unsupported PL/SQL function function_name.
- ORA-28089 - Invalid SYS_CONTEXT namespace <namespace> in policy expression.
- ORA-28090 - Unsupported nesting of function function_name within function function_name in policy expression.
- ORA-28091 - Unsupported use of PL/SQL function function_name in policy expression.
- ORA-28092 - The parameter parameter_name with value value has an error.
- ORA-28095 - The parameter parameter_name is unexpected for ACTION <action> for ALTER_POLICTY.
- ORA-28097 - A data redaction policy cannot be applied to an object owned by SYS.
- ORA-28104 - Input value for *parameter* is not valid.

Usage Notes

See DBMS_REDACT Security Model for a list of the required privileges for this procedure.

See Operating Procedures for more information regarding Function Types and Function Parameters with related examples.

The first time you alter a Data Redaction policy on a column of boolean datatype to turn it into a full redaction policy, which is done by specifying the value DBMS_REDACT.FULL for the function_type parameter, after the COMPATIBLE database initialization parameter is set to 23 or higher, you will see the BOOLEAN_VALUE column appear in the catalog view for REDACTION_VALUES_FOR_TYPE_FULL and the BOOLCOL column appear in the SYS.RADM_FPTM$ data dictionary table.

A Data Redaction named policy expression that has been applied to a redacted column takes precedence over the expression defined in the expression parameter. To find redacted columns that are affected by named policy expressions, query the REDACTION_EXPRESSIONS data dictionary view.

Examples

```
BEGIN
   DBMS_REDACT.ALTER_POLICY(
     object_schema           => 'HR',
     object_name             => 'EMPLOYEES',
     policy_name             => 'redact_emp_id_nums',
     action                  => DBMS_REDACT.DROP_COLUMN,
     column_name             => 'EMAIL');
END;
```

APPLY_POLICY_EXPR_TO_COL Procedure

This procedure associates an Oracle Data Redaction named policy expression with a redacted column from a table or view.

Syntax

```
DBMS_REDACT.APPLY_POLICY_EXPR_TO_COL (
   object_schema                 IN VARCHAR2 := NULL,
   object_name                   IN VARCHAR2,
   column_name                   IN VARCHAR2,
   policy_expression_name        IN VARCHAR2 := NULL);
```

Parameters

Table 159-8 *APPLY_POLICY_EXPR_TO_COL Procedure Parameters*

Parameter	Description
object_schema	Name of the schema that contains the redacted column. If omitted, then the current schema is used.
object_name	Name of the object (table or view) that contains the redacted column
column_name	Name of the redacted column to which the policy expression is applied
policy_expression_name	If NULL, then the named policy expression associated with the redacted column column_name is removed.

Exceptions

- ORA-28061 - This object cannot have a data redaction policy defined on it.
- ORA-28066 - Invalid column column_name.
- ORA-28067 - Missing or invalid column name.

- ORA-28068 - The object *object* does not have a Data Redaction policy.
- ORA-28070 - The column column_name does not have a Data Redaction policy.
- ORA-28082 - The parameter *parameter* is invalid.
- ORA-28092 - The parameter *parameter* with value *value* has an error.

Usage Notes

See DBMS_REDACT Security Model for a list of the required privileges for this procedure.

You can find existing Data Redaction policy expressions by querying the REDACTION_EXPRESSIONS data dictionary view. To find columns that have been redacted, query the REDACTION_COLUMNS data dictionary view.

Example

```
BEGIN
   DBMS_REDACT.APPLY_POLICY_EXPR_TO_COL(
      object_schema           => 'OE',
      object_name             => 'CUSTOMERS',
      column_name             => 'INCOME_LEVEL',
      policy_expression_name  => 'oe_redact_pol');
END;
```

CREATE_POLICY_EXPRESSION Procedure

This procedure creates an Oracle Data Redaction named policy expression.

Syntax

```
DBMS_REDACT.CREATE_POLICY_EXPRESSION (
   policy_expression_name          IN    VARCHAR2,
   expression                      IN    VARCHAR2,
   policy_expression_description   IN    VARCHAR2 := NULL);
```

Parameters

Table 159-9   *CREATE_POLICY_EXPRESSION Procedure Parameters*

Parameter	Description
policy_expression_name	Name of the policy expression
expression	Definition of the policy expression.
policy_expression_description	Description of the policy expression

Exceptions

- ORA-28082 - The parameter *parameter* is invalid.
- ORA-28092 - The parameter *parameter* with value *value* has an error.

Usage Notes

If the expression parameter is 1=1, then it is always TRUE, which causes the redaction to be performed as long as the querying user is not exempt from the Oracle Data Redaction policy. Specifying this parameter as 1=1 will improve Data Redaction performance because Oracle Database will not evaluate the policy expression.

See DBMS_REDACT Security Model for a list of the required privileges for this procedure.

See APPLY_POLICY_EXPR_TO_COL Procedure for how to apply policy expressions to table columns.

After you create a policy expression, you can associate it with a redacted table or view column by running the `DBMS_REDACT.APPLY_POLICY_EXPR_TO_COL` procedure. To find existing redacted columns, query the `REDACTION_COLUMNS` data dictionary view.

Example

```
BEGIN
   DBMS_REDACT.CREATE_POLICY_EXPRESSION(
     policy_expression_name          => 'oe_redact_pol',
     expression                      => 'SYS_CONTEXT(''USERENV'',''SESSION_USER'') =
''OE'''),
     policy_expression_description   => 'Enables policy for user OE ');
END;
```

DISABLE_POLICY Procedure

This procedure disables a Data Redaction policy.

Syntax

```
DBMS_REDACT.DISABLE_POLICY (
   object_schema                 IN     VARCHAR2 := NULL,
   object_name                   IN     VARCHAR2,
   policy_name                   IN     VARCHAR2);
```

Parameters

Table 159-10 ***DISABLE_POLICY Procedure Parameters***

Parameter	Description
`object_schema`	Schema owning the table or view, current user if `NULL`
`object_name`	Name of table or view for which to disable a Data Redaction policy
`policy_name`	Name of policy to be disabled

Exceptions

- `ORA_28061` - This object cannot have a Data Redaction policy defined on it.
- `ORA-28068` - The object *`object`* does not have a Data Redaction policy.
- `ORA-28072` - The specified policy name is incorrect.
- `ORA-28080` - The policy was already disabled.
- `ORA_28104` - Input value for `policy-name` is not valid.

Usage Notes

See DBMS_REDACT Security Model for a list of the required privileges for this procedure.

Examples

```
BEGIN
   DBMS_REDACT.DISABLE_POLICY (
     object_schema    => 'hr',
     object_name      => 'employees',
     policy_name      => 'redact_emp_ids');
END;
```

DROP_POLICY Procedure

This procedure drops a Data Redaction policy by removing a redaction policy from the table or view.

Syntax

```
DBMS_REDACT.DROP_POLICY (
   object_schema                 IN     VARCHAR2 := NULL,
   object_name                   IN     VARCHAR2,
   policy_name                   IN     VARCHAR2);
```

Parameters

Table 159-11 *DROP_POLICY Procedure Parameters*

Parameter	Description
object_schema	Schema owning the table or view, current user if NULL
object_name	Name of table or view from which to drop a Data Redaction policy
policy_name	Name of policy to be dropped

Exceptions

- ORA_28061 - This object cannot have a Data Redaction policy defined on it.
- ORA-28068 - The object *object* does not have a Data Redaction policy.
- ORA-28072 - The specified policy name is incorrect.
- ORA_28104 - Input value for policy-name is not valid.

Usage Notes

See DBMS_REDACT Security Model for a list of the required privileges for this procedure.

Examples

```
BEGIN
   DBMS_REDACT.DROP_POLICY (
      object_schema    =>  'hr',
      object_name      =>  'employees',
      policy_name      =>  'redact_emp_ids');
END;
```

DROP_POLICY_EXPRESSION Procedure

This procedure drops a named policy expression.

Syntax

```
DBMS_REDACT.DROP_POLICY_EXPRESSION (
   policy_expression_name        IN     VARCHAR2);
```

Parameters

Table 159-12 *DROP_POLICY_EXPRESSION Procedure Parameters*

Parameter	Description
policy_expression_name	Name of the policy expression

Exceptions

- ORA-28082 - The parameter *parameter* is invalid.
- ORA-28092 - The parameter *parameter* with value *value* has an error.

Usage Notes

See DBMS_REDACT Security Model for a list of the required privileges for this procedure.

You can find existing Data Redaction policy expressions by querying the REDACTION_EXPRESSIONS data dictionary view.

Example

```
BEGIN
   DBMS_REDACT.DROP_POLICY_EXPRESSION(
     policy_expression_name  => 'oe_redact_pol');
END;
```

ENABLE_POLICY Procedure

This procedure re-enables a Data Redaction policy.

Syntax

```
DBMS_REDACT.ENABLE_POLICY (
   object_schema                 IN    VARCHAR2 := NULL,
   object_name                   IN    VARCHAR2,
   policy_name                   IN    VARCHAR2);
```

Parameters

Table 159-13 *ENABLE_POLICY Procedure Parameters*

Parameter	Description
object_schema	Schema owning the table or view, current user if NULL
object_name	Name of table or view on which to enable a Data Redaction policy
policy_name	Name of policy to be enabled

Exceptions

- ORA-28068 - The object *object* does not have a Data Redaction policy.
- ORA-28071 - The action is not valid.
- ORA-28072 - The specified policy name is incorrect.
- ORA-28079 - The policy was already enabled.

Usage Notes

See DBMS_REDACT Security Model for a list of the required privileges for this procedure.

Examples

```
BEGIN
   DBMS_REDACT.ENABLE_POLICY (
      object_schema    =>  'hr',
      object_name      =>  'employees',
      policy_name      =>  'redact_emp_ids');
END;
```

UPDATE_FULL_REDACTION_VALUES Procedure

This procedure modifies the redacted output that will be displayed when querying a column having a Data Redaction policy where the `function_type` is `DBMS_REDACT.FULL`. Restart the database instance to apply these changes.

Syntax

```
DBMS_REDACT.UPDATE_FULL_REDACTION_VALUES (
   number_val        IN NUMBER                        := NULL,
   binfloat_val      IN BINARY_FLOAT                  := NULL,
   bindouble_val     IN BINARY_DOUBLE                 := NULL,
   char_val          IN CHAR                          := NULL,
   varchar_val       IN VARCHAR2                      := NULL,
   nchar_val         IN NCHAR                         := NULL,
   nvarchar_val      IN NVARCHAR2                     := NULL,
   date_val          IN DATE                          := NULL,
   ts_val            IN TIMESTAMP                     := NULL,
   tswtz_val         IN TIMESTAMP WITH TIME ZONE      := NULL,
   blob_val          IN BLOB                          := NULL,
   clob_val          IN CLOB                          := NULL,
   nclob_val         IN NCLOB                         := NULL,
   boolean_val       IN BOOLEAN                       := NULL);
```

Parameters

Table 159-14 ***UPDATE_FULL_REDACTION_VALUES Procedure Parameters***

Parameter	Description
`number_val`	Modifies the default value for columns of the `NUMBER` datatype
`binfloat_val`	Modifies the default value for columns of the `BINARY_FLOAT` datatype
`bindouble_val`	Modifies the default value for columns of the `BINARY_DOUBLE` datatype
`char_val`	Modifies the default value for columns of the `CHAR` datatype
`varchar_val`	Modifies the default value for columns of the `VARCHAR2` datatype
`nchar_val`	Modifies the default value for columns of the `NCHAR` datatype
`nvarchar_val`	Modifies the default value for columns of the `NVARCHAR2` datatype
`date`	Modifies the default value for columns of the `DATE` datatype
`ts_val`	Modifies the default value for columns of the `TIMESTAMP` datatype
`tswtz_val`	Modifies the default value for columns of the `TIMESTAMP WITH TIME ZONE` datatype

Table 159-14 (Cont.) *UPDATE_FULL_REDACTION_VALUES Procedure Parameters*

Parameter	Description
`blob_val`	Modifies the default value for columns of the `BLOB` datatype
`clob_val`	Modifies the default value for columns of the `CLOB` datatype
`nclob_val`	Modifies the default value for columns of the `NCLOB` datatype
`boolean_val`	Modifies the default value for columns of the `BOOLEAN` datatype

Example 159-1 Change the Current Value for Columns That Use the `NUMBER` Data Type

1.
```
EXEC DBMS_REDACT.UPDATE_FULL_REDACTION_VALUES (number_val => 7);
```
2. Close the PDB:
```
ALTER PLUGGABLE DATABASE <pdb_name> CLOSE IMMEDIATE;
```
3. Re-open the PDB:
```
ALTER PLUGGABLE DATABASE <pdb_name> OPEN;
```

Usage Notes

See DBMS_REDACT Security Model for a list of the required privileges for this procedure.

The first time `DBMS_REDACT.UPDATE_FULL_REDACTION_VALUES` is invoked with a non-`NULL` value for its `boolean_val` parameter after the `COMPATIBLE` database initialization parameter is set to 23 or higher, you will see the `BOOLEAN_VALUE` column appear in the catalog view for `REDACTION_VALUES_FOR_TYPE_FULL` and the `BOOLCOL` column appear in the `SYS.RADM_FPTM$` data dictionary table.

UPDATE_POLICY_EXPRESSION Procedure

This procedure updates an Oracle Data Redaction named policy expression.

Syntax

```
DBMS_REDACT.UPDATE_POLICY_EXPRESSION (
   policy_expression_name          IN    VARCHAR2,
   expression                      IN    VARCHAR2,
   policy_expression_description   IN    VARCHAR2 := NULL);
```

Parameters

Table 159-15 *UPDATE_POLICY_EXPRESSION Procedure Parameters*

Parameter	Description
`policy_expression_name`	Name of the policy expression
`expression`	Definition of the policy expression
`policy_expression_description`	Description of the policy expression

Exceptions

- ORA-28082 - The parameter *parameter* is invalid.
- ORA-28092 - The parameter *parameter* with value *value* has an error.

Usage Notes

See DBMS_REDACT Security Model for a list of the required privileges for this procedure.

You can find existing policy expressions by querying the REDACTION_EXPRESSIONS data dictionary view.

Example

```
BEGIN
   DBMS_REDACT.UPDATE_POLICY_EXPRESSION(
     policy_expression_name         => 'oe_redact_pol',
     expression                     => 'SYS_CONTEXT(''USERENV'',''SESSION_USER'') !=
''OE'''),
     policy_expression_description  => 'Updates policy expression for oe_redact_pol');
END;
```