190

DBMS_SQL_TRANSLATOR

The `DBMS_SQL_TRANSLATOR` package provides an interface for creating, configuring, and using SQL translation profiles.

This chapter contains the following topics:

> **See Also:**
>
> SQL Translation Framework Architecture and Overview in*Oracle Database Migration Guide*

DBMS_SQL_TRANSLATOR Security Model

`DBMS_SQL_TRANSLATOR` is a invoker's rights package.

When translating a SQL statement or error, the translator package procedure will be invoked with the same current user and current schema as those in which the SQL statement being parsed. The owner of the translator package must be granted the `TRANSLATE SQL` user privilege on the current user.

Additionally, the current user must be granted the `EXECUTE` privilege on the translator package.

DBMS_SQL_TRANSLATOR Constants

`DBMS_SQL_TRANSLATOR` defines several constants to use when specifying parameter values.

These are shown in the following table.

Table 190-1 DBMS_SQL_TRANSLATOR Constants

Constant	Value	Type	Description
ATTR_EDITIONABLE	'EDITIONABLE'	VARCHAR2(30)	Name of the SQL translation profile attribute that specifies whether the SQL translation profile becomes an editioned or noneditioned object if editioning is later enabled for the schema object type SQL translation profile in the owner's schema (see Operational Notes)
ATTR_FOREIGN_SQL_SYNTAX	'FOREIGN_SQL_SYNTAX'	VARCHAR2(30)	Name of the SQL translation profile attribute that indicates if the profile is for translation of foreign SQL syntax (see Operational Notes)
ATTR_LOG_TRANSLATION_ERROR	'TRANSLATION_ERROR'	VARCHAR2(30)	Name of the SQL translation profile attribute that controls if the profile should log translation error in the database alert log (see Operational Notes)
ATTR_RAISE_TRANSLATION_ERROR	'TRANSLATION_ERROR '	VARCHAR2(30)	Name of the SQL translation profile attribute that controls if the profile should raise translation error if a SQL statement or error fails to be translated (see Operational Notes)
ATTR_TRANSLATE_NEW_SQL	'TRANSLATE_NEW_SQL'	VARCHAR2(30)	Name of the SQL translation profile attribute that controls if the profile should translate new SQL statements and errors (see Operational Notes)
ATTR_TRACE_TRANSLATION	'TRACE_TRANSLATION'	VARCHAR2(30)	Name of the SQL translation profile attribute that controls tracing (see Operational Notes)
ATTR_TRANSLATOR	'TRANSLATOR'	VARCHAR2(30)	Name of the SQL translation profile attribute that specifies the translator package (see Operational Notes)
ATTR_VALUE_TRUE	'TRUE'	VARCHAR2(30)	Value to set a SQL translation profile attribute to true (see Operational Notes)
ATTR_VALUE_FALSE	'FALSE'	VARCHAR2(30)	Value to set a SQL translation profile attribute to false (see Operational Notes)

DBMS_SQL_TRANSLATOR Operational Notes

The subprograms that modify a profile have DDL transaction semantics and when invoked will commit any open transaction in the session.

ATTR_EDITIONABLE Constant

Editionable is true by default.

ATTR_FOREIGN_SQL_SYNTAX Constant

Foreign SQL syntax is true by default.

ATTR_LOG_TRANSLATION_ERROR Constant

- If log translation is enabled in a SQL translation profile, an alert log is written to the database alert log if no custom translation is found for a SQL statement or error. This allows the user to catch any error in the custom translation in a profile.
- Log translation error is false by default.

ATTR_RAISE_TRANSLATION_ERROR Constant

Raise translation error is false by default.

ATTR_TRANSLATE_NEW_SQL Constant

- The name of the SQL translation profile attribute that controls if the profile should translate new SQL statements and errors. If so, the translator package, if registered, will translate a new SQL statement or error not already translated in custom translations, and also register the new translation as custom translation. If not, any new SQL statement or error encountered will result in a translation error
- Translate new SQL statements and errors is true by default.

ATTR_TRACE_TRANSLATION Constant

- If tracing is enabled in a SQL translation profile, any SQL statement or error translated by the profile in a database session and its translation is written to the database session's trace file.
- Tracing is disabled by default.

ATTR_TRANSLATOR Constant

- The translator package must be a PL/SQL package with the following three procedures. The TRANSLATE_SQL Procedure and the TRANSLATE_ERROR Procedure are called to translate SQL statements and errors. The names of the parameters of the translate procedures must be followed.

```
PROCEDURE TRANSLATE_SQL(
   sql_text        IN  CLOB,
   translated_text OUT CLOB);

PROCEDURE TRANSLATE_ERROR(
   error_code          IN  BINARY_INTEGER,
   translated_code     OUT BINARY_INTEGER,
   translated_sqlstate OUT VARCHAR2);

Parameters:
  profile_name        - profile name
  sql_text            - SQL statement to be translated
  translated_text     - translated SQL statement
  error_code          - Oracle error code
  translated_code     - translated error code
  translated_sqlstate - translated SQLSTATE
```

- When `NULL` is returned in `translated_text`, `translated_code`, or `translated_sqlstate`, it means that no translation is required and the original SQL statement, error code, or `SQLSTATE` is used instead.
- The name of the translator package follows the naming rules for database packages of the form `[schema.]package_name`. When the schema and package names are used, they are set to uppercase by default unless surrounded by double quotation marks. For example, setting a translator package, `translator => 'dbms_tsql_translator'` is the same as `translator => 'Dbms_Tsql_Translator'` and `translator => 'DBMS_TSQL_TRANSLATOR'`, but not the same as `translator => '"dbms_tsql_translator"'`. The default schema name is the profile owner.
- The translator attribute is not set by default.

ATTR_VALUE_TRUE Constant

The value to set a SQL translation profile attribute to true.

ATTR_VALUE_FALSE Constant

The value to set a SQL translation profile attribute to false.

DBMS_SQL_TRANSLATOR Exceptions

This table lists the exceptions raised by the `DBMS_SQL_TRANSLATOR` package.

Table 190-2 Exceptions Raised by DBMS_SQL_TRANSLATOR

Exception	Error Code	Description
BAD_ARGUMENT	29261	Bad argument is passed to the PL/SQL interface
INSUFFICIENT_PRIVILEGE	1031	User has insufficient privilege for the operation
NO_SUCH_PROFILE	24252	Profile does not exist
NO_SUCH_USER	1918	Profile owner does not exist
NO_TRANSLATION_FOUND	24253	No translation of the SQL statement or error code found
PROFILE_EXISTS	955	Profile already exists

DBMS_SQL_TRANSLATOR Examples

This is an example of basic SQL translation using `DBMS_SQL_TRANSLATOR`.

Basic SQL Translation

```
BEGIN
  DBMS_SQL_TRANSLATOR.CREATE_PROFILE(
    profile_name     => ' tsql_application');
  DBMS_SQL_TRANSLATOR.SET_ATTRIBUTE(
    profile_name     =>  'tsql_application',
    attribute_name   =>  DBMS_SQL_TRANSLATOR.ATTR_TRANSLATOR,
    attribute_value  =>  'migration_repo.sybase_tsql_translator');
END;
```

Summary of DBMS_SQL_TRANSLATOR Subprograms

This table lists the `DBMS_SQL_TRANSLATOR` subprograms and briefly describes them.

Table 190-3 DBMS_SQL_TRANSLATOR Package Subprograms

Subprogram	Description
CREATE_PROFILE Procedure	Creates a SQL translation profile
DEREGISTER_SQL_TRANSLATION Procedure	Deregisters the custom translation of a SQL statement in a SQL translation profile

Table 190-3 (Cont.) DBMS_SQL_TRANSLATOR Package Subprograms

Subprogram	Description
DEREGISTER_ERROR_TRANSLATION Procedure	Deregisters the translation of an Oracle error code and SQLSTATE in a SQL translation profile
DROP_PROFILE Procedure	Drops a SQL translation profile and its contents
ENABLE_ERROR_TRANSLATION Procedure	Enables or disables a custom translation of an Oracle error code in a SQL translation profile
ENABLE_SQL_TRANSLATION Procedure	Enables or disables a custom translation of a SQL statement in a SQL translation profile
EXPORT_PROFILE Procedure	Exports the content of a SQL translation profile
IMPORT_PROFILE Procedure	Imports the content of a SQL translation profile
REGISTER_ERROR_TRANSLATION Procedure	Registers a custom translation of an Oracle error code and SQLSTATE in a SQL translation profile
REGISTER_SQL_TRANSLATION Procedure	Registers a custom translation of a SQL statement in a SQL translation profile
SET_ATTRIBUTE Procedure	Sets an attribute of a SQL translation profile
SQL_HASH Function	Computes the hash value of a SQL statement in a SQL translation profile
SQL_ID Function	Computes the SQL identifier of a SQL statement in a SQL translation profile
TRANSLATE_ERROR Procedure	Translates an Oracle error code and an ANSI SQLSTATE using a SQL translation profile
TRANSLATE_SQL Procedure	Translates a SQL statement using a SQL translation profile

CREATE_PROFILE Procedure

This procedure creates a SQL translation profile.

Syntax

```
DBMS_SQL_TRANSLATOR.CREATE_PROFILE (
   profile_name    IN  VARCHAR2);
```

Parameters

Table 190-4 *CREATE_PROFILE Procedure Parameters*

Parameter	Description
profile_name	Name of profile

Exceptions

Table 190-5 CREATE_PROFILE Procedure Exceptions

Exception	Description
BAD_ARGUMENT	Bad argument is passed to the PL/SQL interface
INSUFFICIENT_PRIVILEGE	User has insufficient privilege for the operation
NO_SUCH_USER	Profile owner does not exist
PROFILE_EXISTS	Profile already exists

Usage Notes

- A SQL translation profile is a database schema object that resides in SQL translation profile namespace. Its name follows the naming rules for database objects of the form `[schema.]name`. When the schema and profile names are used in the `DBMS_SQL_TRANSLATOR` package, they are uppercased unless surrounded by double quotation marks. For example, the translation profile `profile_name => 'tsql_application'` is the same as `profile_name => 'Tsql_Application'` and `profile_name => 'TSQL_APPLICATION'`, but not the same as `profile_name => '"tsql_application"'`.
- A SQL translation profile is an editionable object type.
- A SQL translation profile cannot be created as a common object in a multitenant container database (CDB).
- To destroy a SQL translation profile, use the DROP_PROFILE Procedure.

Examples

```
BEGIN
  DBMS_SQL_TRANSLATOR.CREATE_PROFILE(profile_name => 'tsql_application');
END;
```

DEREGISTER_SQL_TRANSLATION Procedure

This procedure deregisters the custom translation of a SQL statement in a SQL translation profile.

Syntax

```
DBMS_SQL_TRANSLATOR.DEREGISTER_SQL_TRANSLATION (
   profile_name      IN  VARCHAR2,
   sql_text          IN  CLOB);
```

Parameters

Table 190-6 *DEREGISTER_SQL_TRANSLATION Procedure Parameters*

Parameter	Description
profile_name	Name of profile
sql_text	SQL statement

Exceptions

Table 190-7 DEREGISTER_SQL_TRANSLATION Procedure Exceptions

Exception	Description
BAD_ARGUMENT	Bad argument is passed to the PL/SQL interface
INSUFFICIENT_PRIVILEGE	User has insufficient privilege for the operation
NO_SUCH_USER	Profile owner does not exist
PROFILE_EXISTS	Profile already exists

Examples

```
BEGIN
    DBMS_SQL_TRANSLATOR.DEREGISTER_SQL_TRANSLATION(
      profile_name => 'tsql_application',
      sql_text     => 'select top 5 * from emp');
END;
```

DEREGISTER_ERROR_TRANSLATION Procedure

This procedure deregisters the translation of an Oracle error code and SQLSTATE in a SQL translation profile.

Syntax

```
DBMS_SQL_TRANSLATOR.DEREGISTER_ERROR_TRANSLATION (
   profile_name          IN   VARCHAR2,
   error_code            IN   PLS_INTEGER);
```

Parameters

Table 190-8 *DEREGISTER_ERROR_TRANSLATION Procedure Parameters*

Parameter	Description
profile_name	Name of profile
error_code	Oracle error code

Exceptions

Table 190-9 DEREGISTER_ERROR_TRANSLATION Procedure Exceptions

Exception	Description
BAD_ARGUMENT	Bad argument is passed to the PL/SQL interface
INSUFFICIENT_PRIVILEGE	User has insufficient privilege for the operation
NO_SUCH_USER	Profile owner does not exist
NO_SUCH_PROFILE	Profile does not exist

Examples

```
BEGIN
   DBMS_SQL_TRANSLATOR.DEREGISTER_ERROR_TRANSLATION(
      profile_name    => 'tsql_application',
      error_code      => 1);
END;
```

DROP_PROFILE Procedure

This procedure drops a SQL translation profile and its contents.

Syntax

```
DBMS_SQL_TRANSLATOR.DROP_PROFILE (
   profile_name     IN    VARCHAR2);
```

Parameters

Table 190-10 *DROP_PROFILE Procedure Parameters*

Parameter	Description
profile_name	Name of profile

Exceptions

Table 190-11 DROP_PROFILE Procedure Exceptions

Exception	Description
BAD_ARGUMENT	Bad argument is passed to the PL/SQL interface
INSUFFICIENT_PRIVILEGE	User has insufficient privilege for the operation
NO_SUCH_USER	Profile owner does not exist
NO_SUCH_PROFILE	Profile does not exist

Examples

```
BEGIN
    DBMS_SQL_TRANSLATOR.DROP_PROFILE(
      profile_name    =>  'tsql_application');
END;
```

ENABLE_ERROR_TRANSLATION Procedure

This procedure enables or disables a custom translation of an Oracle error code in a SQL translation profile.

Syntax

```
DBMS_SQL_TRANSLATOR.ENABLE_ERROR_TRANSLATION (
   profile_name      IN  VARCHAR2,
   sql_text          IN CLOB,
   enable            IN BOOLEAN DEFAULT TRUE);
```

Parameters

Table 190-12 *ENABLE_ERROR_TRANSLATION Procedure Parameters*

Parameter	Description
profile_name	Name of profile
sql_text	SQL statement
enable	Enable or disable the translation

Exceptions

Table 190-13 ENABLE_ERROR_TRANSLATION Procedure Exceptions

Exception	Description
BAD_ARGUMENT	Bad argument is passed to the PL/SQL interface
INSUFFICIENT_PRIVILEGE	User has insufficient privilege for the operation
NO_SUCH_USER	Profile owner does not exist
NO_SUCH_PROFILE	Profile does not exist

Examples

```
BEGIN
  DBMS_SQL_TRANSLATOR.ENABLE_ERROR_TRANSLATION(
      profile_name => 'tsql_application',
      sql_text     => 'SELECT TOP 5 * FROM emp'
      enable       => TRUE);
END;
```

ENABLE_SQL_TRANSLATION Procedure

This procedure enables or disables a custom translation of a SQL statement in a SQL translation profile.

Syntax

```
DBMS_SQL_TRANSLATOR.ENABLE_SQL_TRANSLATION (
   profile_name      IN    VARCHAR2,
   sql_text          IN    CLOB,
   enable            IN BOOLEAN DEFAULT TRUE);
```

Parameters

Table 190-14 *ENABLE_SQL_TRANSLATION Procedure Parameters*

Parameter	Description
profile_name	Name of profile
sql_text	SQL statement
enable	Enable or disable the translation

Exceptions

Table 190-15 ENABLE_SQL_TRANSLATION Procedure Exceptions

Exception	Description
BAD_ARGUMENT	Bad argument is passed to the PL/SQL interface
INSUFFICIENT_PRIVILEGE	User has insufficient privilege for the operation
NO_SUCH_USER	Profile owner does not exist
NO_SUCH_PROFILE	Profile does not exist

Examples

```
BEGIN
  DBMS_SQL_TRANSLATOR.ENABLE_SQL_TRANSLATION(
      profile_name => 'tsql_application',
      sql_text     => 'select top 5 * from emp',
      enable       => TRUE);
END;
```

EXPORT_PROFILE Procedure

This procedure exports the content of a SQL translation profile.

Syntax

```
DBMS_SQL_TRANSLATOR.EXPORT_PROFILE (
   profile_name     IN          VARCHAR2,
   content          OUT NOCOPY  CLOB);
```

Parameters

Table 190-16 *EXPORT_PROFILE Procedure Parameters*

Parameter	Description
profile_name	Name of profile
content	Content of profile

Exceptions

Table 190-17 EXPORT_PROFILE Procedure Exceptions

Exception	Description
BAD_ARGUMENT	Bad argument is passed to the PL/SQL interface
INSUFFICIENT_PRIVILEGE	User has insufficient privilege for the operation
NO_SUCH_USER	Profile owner does not exist
NO_SUCH_PROFILE	Profile does not exist

Usage Notes

- The content of the SQL translation profile is exported in XML format as follows. Note that the profile name will not be exported.

```
SQLTranslationProfile Translator="translator package name"
                      ForeignSQLSyntax="TRUE|FALSE"
                      TranslateNewSQL="TRUE|FALSE"
                      RaiseTranslationError="TRUE|FALSE"
                      LogTranslationError="TRUE|FALSE"
                      TraceTranslation="TRUE|FALSE"
                      Editionable="TRUE|FALSE">
  <SQLTranslations>
    <SQLTranslation Enabled="TRUE|FALSE">
      <SQLText>original SQL text</SQLText>
      <TranslatedText>translated SQL text</TranslatedText>
    </SQLTranslation>
    ...
  </SQLTranslations>
  <ErrorTranslations>
    <ErrorTranslation Enabled="TRUE|FALSE">
      <ErrorCode>Oracle error code</ErrorCode>
      <TranslatedCode>translated error code</TranslatedCode>
      <TranslatedSQLSTATE>translated SQLSTATE</TranslatedSQLSTATE>
    </ErrorTranslation>
    ...
  </ErrorTranslations>
</SQLTranslationProfile>
```

- To import the content to a SQL translation profile, use the IMPORT_PROFILE Procedure.

Examples

```
DECLARE
  content CLOB;
BEGIN
    DBMS_SQL_TRANSLATOR.EXPORT_PROFILE(
      profile_name    =>  'tsql_application',
      content         =>  content);
END;
```

IMPORT_PROFILE Procedure

This procedure imports the content of a SQL translation profile.

Syntax

```
DBMS_SQL_TRANSLATOR.IMPORT_PROFILE (
   profile_name       IN   VARCHAR2,
   content            IN   CLOB);
```

Parameters

Table 190-18 ***IMPORT_PROFILE Procedure Parameters***

Parameter	Description
profile_name	Name of profile
content	Content of profile

Exceptions

Table 190-19 IMPORT_PROFILE Procedure Exceptions

Exception	Description
BAD_ARGUMENT	Bad argument is passed to the PL/SQL interface
INSUFFICIENT_PRIVILEGE	User has insufficient privilege for the operation
NO_SUCH_USER	Profile owner does not exist

Usage Notes

- The content of the SQL translation profile must be in XML format as used by the EXPORT_PROFILE Procedure. All elements and attributes are optional.
- If the profile does not exist, it is created. If it exists, the content overrides any existing attribute, translator package, SQL or error translation registration.
- To export the content to a SQL translation profile, use the EXPORT_PROFILE Procedure.

Examples

```
DECLARE
  content CLOB;
BEGIN
    DBMS_SQL_TRANSLATOR.IMPORT_PROFILE(
      profile_name    =>  'tsql_application',
      content         =>  content);
END;
```

REGISTER_ERROR_TRANSLATION Procedure

This procedure registers a custom translation of an Oracle error code and SQLSTATE in a SQL translation profile.

Syntax

```
DBMS_SQL_TRANSLATOR.REGISTER_ERROR_TRANSLATION (
   profile_name          IN   VARCHAR2,
   error_code            IN   PLS_INTEGER,
   translated_code       IN   PLS_INTEGER DEFAULT NULL,
   translated_sqlstate   IN   VARCHAR2 DEFAULT NULL,
   enable                IN   BOOLEAN DEFAULT TRUE);
```

Parameters

Table 190-20 *REGISTER_ERROR_TRANSLATION Procedure Parameters*

Parameter	Description
profile_name	Name of profile
error_code	Oracle error code
translated_code	Translated error code
translated_sqlstate	Translated SQLSTATE
enable	Enable or disable the translation

Exceptions

Table 190-21 REGISTER_ERROR_TRANSLATION Procedure Exceptions

Exception	Description
BAD_ARGUMENT	Bad argument is passed to the PL/SQL interface
INSUFFICIENT_PRIVILEGE	User has insufficient privilege for the operation
NO_SUCH_USER	Profile owner does not exist
NO_SUCH_PROFILE	Profile does not exist

Usage Notes

- When the Oracle Database translates an Oracle error code using a translation profile, it searches for the registered custom translation first, and only invokes the translator package if no match is found.
- When a translation is registered in a profile, it may be disabled. Oracle Database does not search for disabled translations.
- The old translation of the error code and SQLSTATE, if present, is replaced with the new translation.
- To deregister a translation, use the DEREGISTER_ERROR_TRANSLATION Procedure.

Examples

```
BEGIN
      DBMS_SQL_TRANSLATOR.REGISTER_ERROR_TRANSLATION(
      profile_name     => 'tsql_application',
      error_code       => 1,
      translated_code => 2601);
END;
```

REGISTER_SQL_TRANSLATION Procedure

This procedure registers a custom translation of a SQL statement in a SQL translation profile.

Syntax

```
DBMS_SQL_TRANSLATOR.REGISTER_SQL_TRANSLATION (
   profile_name      IN VARCHAR2,
   sql_text          IN CLOB,
   translated_text   IN CLOB DEFAULT NULL,
   enable            IN BOOLEAN DEFAULT TRUE);
```

Parameters

Table 190-22 *REGISTER_SQL_TRANSLATION Procedure Parameters*

Parameter	Description
profile_name	Name of profile
sql_text	SQL statement
translated_text	Translated SQL statement

Table 190-22 (Cont.) *REGISTER_SQL_TRANSLATION Procedure Parameters*

Parameter	Description
enable	Enable or disable the translation

Exceptions

Table 190-23 REGISTER_SQL_TRANSLATION Procedure Exceptions

Exception	Description
BAD_ARGUMENT	Bad argument is passed to the PL/SQL interface
INSUFFICIENT_PRIVILEGE	User has insufficient privilege for the operation
NO_SUCH_USER	Profile owner does not exist
NO_SUCH_PROFILE	Profile does not exist

Usage Notes

- When the Oracle Database translates a statement using a translation profile, it searches for the registered custom translation first, and only invokes the translator package if no match is found.
- When a translation is registered in a profile, it may be disabled. Oracle Database does not search for disabled translations.
- When `translated_text` is `NULL`, no translation is required and the original statement is used.
- The old translation of the SQL statement, if present, is replaced with the new translation.
- To deregister a translation, use the DEREGISTER_SQL_TRANSLATION Procedure.

Examples

```
BEGIN
    DBMS_SQL_TRANSLATOR.REGISTER_SQL_TRANSLATION(
      profile_name    => 'tsql_application',
      sql_text        => 'select top 5 * from emp',
      translated_text => 'SELECT * FROM emp WHERE rownum <= :SYS_N_001');
END;
```

SET_ATTRIBUTE Procedure

This procedure sets an attribute of a SQL translation profile.

Syntax

```
DBMS_SQL_TRANSLATOR.SET_ATTRIBUTE (
   profile_name      IN  VARCHAR2,
   attribute_name    IN  VARCHAR2,
   attribute_value   IN  VARCHAR2;)
```

Parameters

Table 190-24 *SET_ATTRIBUTE Procedure Parameters*

Parameter	Description
profile_name	Name of profile
attribute_name	Name of attribute
attribute_value	Value of attribute

Exceptions

Table 190-25 SET_ATTRIBUTE Procedure Exceptions

Exception	Description
BAD_ARGUMENT	Bad argument is passed to the PL/SQL interface
INSUFFICIENT_PRIVILEGE	User has insufficient privilege for the operation
NO_SUCH_USER	Profile owner does not exist
NO_SUCH_PROFILE	Profile does not exist

Usage Notes

See Constants

SQL_HASH Function

This procedure computes the hash value of a SQL statement in the session's SQL translation profile.

Syntax

```
DBMS_SQL_TRANSLATOR.SQL_HASH (
   sql_text         IN    CLOB)
RETURN NUMBER DETERMINISTIC;
```

Parameters

Table 190-26 *SQL_HASH Function Parameters*

Parameter	Description
sql_text	SQL statement

Return Values

Returns hash value of the SQL statement in the SQL translation profile

Exceptions

Table 190-27 SQL_HASH Function Exceptions

Exception	Description
BAD_ARGUMENT	Bad argument is passed to the PL/SQL interface

Examples

```
DECLARE
  sqltext CLOB;
  txltext CLOB;
  sqlhash NUMBER;
BEGIN
  sqltext  :=  'SELECT TOP 1 * FROM emp';
  sqlhash  :=  DBMS_SQL_TRANSLATOR.SQL_HASH (sqltext);
  SELECT translated_text INTO txltext
    FROM user_sql_translations
    WHERE sql_hash = sqlhash
    AND DBMS_LOB.COMPARE (sql_text, sqltext) = 0;
END;
```

SQL_ID Function

This procedure computes the SQL identifier of a SQL statement in a SQL translation profile.

Syntax

```
DBMS_SQL_TRANSLATOR.SQL_ID (
   sql_text         IN    CLOB)
RETURN VARCHAR2 DETERMINISTIC;
```

Parameters

Table 190-28 *SQL_ID Function Parameters*

Parameter	Description
sql_text	SQL statement

Return Values

Returns the SQL ID of the SQL statement in the SQL translation profile

Exceptions

Table 190-29 SQL_ID Function Exceptions

Exception	Description
BAD_ARGUMENT	Bad argument is passed to the PL/SQL interface

Examples

```
DECLARE
  sqltext CLOB;
```

```
  sqlid   VARCHAR2(13);
BEGIN
  sqltext :=  'SELECT TOP 1 * FROM emp';
  sqlid   :=  DBMS_SQL_TRANSLATOR.SQL_ID (sqltext);
END;
```

TRANSLATE_ERROR Procedure

This procedure translates an Oracle error code and an ANSI SQLSTATE using the session's SQL translation profile

Syntax

```
DBMS_SQL_TRANSLATOR.TRANSLATE_ERROR (
   error_code           IN           PLS_INTEGER,
   translated_code      OUT          PLS_INTEGER,
   translated_sqlstate  OUT NOCOPY   VARCHAR2);
```

Parameters

Table 190-30 *TRANSLATE_ERROR Procedure Parameters*

Parameter	Description
error_code	Oracle error code
translated_code	Translated error code
translated_sqlstate	Translated SQLSTATE

Exceptions

Table 190-31 TRANSLATE_ERROR Procedure Exceptions

Exception	Description
BAD_ARGUMENT	Bad argument is passed to the PL/SQL interface
INSUFFICIENT_PRIVILEGE	User has insufficient privilege for the operation
NO_SUCH_USER	Profile owner does not exist
NO_SUCH_PROFILE	Profile does not exist
NO_TRANSLATION_FOUND	No translation of the SQL statement or error code is found

Examples

```
DECLARE
  translated_code      BINARY_INTEGER;
  translated_sqlstate  VARCHAR2(5);
BEGIN
    DBMS_SQL_TRANSLATOR.TRANSLATE_ERROR(
      error_code           => 1,
      translated_code      => translated_code,
      translated_sqlstate  => translated_sqlstate);
END;
```

TRANSLATE_SQL Procedure

This procedure translates a SQL statement using a SQL translation profile.

Syntax

```
DBMS_SQL_TRANSLATOR.TRANSLATE_SQL (
   sql_text          IN           CLOB,
   translated_text   OUT  NOCOPY  CLOB);
```

Parameters

Table 190-32 *TRANSLATE_SQL Procedure Parameters*

Parameter	Description
sql_text	SQL statement
translated_text	Translated SQL statement

Exceptions

Table 190-33 TRANSLATE_SQL Procedure Exceptions

Exception	Description
BAD_ARGUMENT	Bad argument is passed to the PL/SQL interface
INSUFFICIENT_PRIVILEGE	User has insufficient privilege for the operation
NO_SUCH_USER	Profile owner does not exist
NO_SUCH_PROFILE	Profile does not exist

Examples

```
ALTER SESSION SET SQL_TRANSLATION_PROFILE =  tsql_application;

DECLARE
  translated_text CLOB;
BEGIN
    DBMS_SQL_TRANSLATOR.TRANSLATE_SQL(
      sql_text         => 'select top 5 * from emp',
      translated_text  => translated_text);
END;
```