

ANYTYPE TYPE

An `ANYTYPE` `TYPE` can contain a type description of any persistent SQL type, named or unnamed, including object types and collection types. It can also be used to construct new transient type descriptions.

New persistent types can only be created using the `CREATE TYPE` statement. Only new transient types can be constructed using the `ANYTYPE` interfaces.

Starting with Oracle Database 19c, the `GETANYTYPEFROMPERSISTENT` function replaces the `GETPERSISTENT` static function.

This chapter discusses the following:

- [Summary of ANYTYPE Subprograms](#)

Summary of ANYTYPE Subprograms

This table lists the `ANYTYPE` subprograms in alphabetical order and briefly describes them.

Table 308-1 *ANYTYPE Subprograms*

Subprogram	Description
BEGINCREATE Static Procedure	Creates a new instance of <code>ANYTYPE</code> which can be used to create a transient type description.
SETINFO Member Procedure	Sets any additional information required for constructing a <code>COLLECTION</code> or builtin type.
ADDATTR Member Procedure	Adds an attribute to an <code>ANYTYPE</code> (of typecode <code>DBMS_TYPES.TYPECODE_OBJECT</code>).
ENDCREATE Member Procedure	Ends creation of a transient <code>ANYTYPE</code> . Other creation functions cannot be called after this call.
GETINFO Member Function	Gets the type information for the <code>ANYTYPE</code> .
GETATTRELEMINFO Member Function	Gets the type information for an attribute of the type (if it is of <code>TYPECODE_OBJECT</code>). Gets the type information for a collection's element type if the <i>self</i> parameter is of a collection type.
GETANYTYPEFROMPERSISTENT Function	This standalone function replaces the old static function <code>GETPERSISTENT</code> . This returns an <code>ANYTYPE</code> corresponding to a persistent type created earlier using the <code>CREATE TYPE SQL</code> statement.

BEGINCREATE Static Procedure

This procedure creates a new instance of `ANYTYPE` which can be used to create a transient type description.

Syntax

```
STATIC PROCEDURE BEGINCREATE(  
    typecode      IN      PLS_INTEGER,  
    atype         OUT NOCOPY ANYTYPE);
```

Parameters

Table 308-2 BEGINCREATE Procedure Parameters

Parameter	Description
typecode	Use a constant from <code>DBMS_TYPES</code> package. Typecodes for user-defined type: <ul style="list-style-type: none">• <code>DBMS_TYPES.TYPECODE_OBJECT</code>• <code>DBMS_TYPES.TYPECODE_VARRAY</code> or• <code>DBMS_TYPES.TYPECODE_TABLE</code> Typecodes for builtin types: <ul style="list-style-type: none">• <code>DBMS_TYPES.TYPECODE_NUMBER</code>, and similar types.
atype	<code>ANYTYPE</code> for a transient type

SETINFO Member Procedure

This procedure sets any additional information required for constructing a `COLLECTION` or builtin type.

Syntax

```
MEMBER PROCEDURE SETINFO(  
    self          IN OUT NOCOPY ANYTYPE,  
    prec          IN PLS_INTEGER,  
    scale         IN PLS_INTEGER,  
    len           IN PLS_INTEGER,  
    csid          IN PLS_INTEGER,  
    csfrm         IN PLS_INTEGER,  
    atype         IN ANYTYPE DEFAULT NULL,  
    elem_tc       IN PLS_INTEGER DEFAULT NULL,  
    elem_count    IN PLS_INTEGER DEFAULT 0);
```

Parameters

Table 308-3 SETINFO Procedure Parameters

Parameter	Description
self	The transient <code>ANYTYPE</code> that is being constructed.
prec	Optional. Required if typecode represents a <code>NUMBER</code> . Give precision and scale. Ignored otherwise.

Table 308-3 (Cont.) SETINFO Procedure Parameters

Parameter	Description
scale	Optional. Required if typecode represents a NUMBER. Give precision and scale. Ignored otherwise.
len	Optional. Required if typecode represents a RAW, CHAR, VARCHAR, or VARCHAR2 type. Gives length.
csid	Required if typecode represents types requiring character information such as CHAR, VARCHAR, or VARCHAR2.
csfrm	Required if typecode represents types requiring character information such as CHAR, VARCHAR, or VARCHAR2.
atype	Optional. Required if collection element typecode is a user-defined type such as TYPECODE_OBJECT, and similar others. It is also required for a built-in type that needs user-defined type information such as TYPECODE_REF. This parameter is not needed otherwise.

The Following Parameters Are Required For Collection Types

Table 308-4 SETINFO Procedure Parameters - Collection Types

Parameter	Description
elem_tc	Must be of the collection element's typecode (from DBMS_TYPES package).
elem_count	Pass 0 for elem_count if the self represents a nested table (TYPECODE_TABLE). Otherwise pass the collection count if self represents a VARRAY.

Exceptions

- DBMS_TYPES.INVALID_PARAMETER: Invalid Parameters (typecode, typeinfo)
- DBMS_TYPES.INCORRECT_USAGE: Incorrect usage (cannot call after calling ENDCREATE, and similar actions.)

Usage Notes

It is an error to call this function on an ANYTYPE that represents a persistent user defined type.

ADDATTR Member Procedure

This procedure adds an attribute to an ANYTYPE (of typecode DBMS_TYPES.TYPECODE_OBJECT).

Syntax

```
MEMBER PROCEDURE ADDATTR(  
    self          IN OUT NOCOPY ANYTYPE,  
    aname         IN VARCHAR2,  
    typecode      IN PLS_INTEGER,  
    prec          IN PLS_INTEGER,  
    scale         IN PLS_INTEGER,  
    len           IN PLS_INTEGER,  
    csid          IN PLS_INTEGER,
```

```
csfrm          IN PLS_INTEGER,  
attr_type      IN ANYTYPE DEFAULT NULL);
```

Parameters

Table 308-5 ADDATTR Procedure Parameters

Parameter	Description
self	The transient ANYTYPE that is being constructed. Must be of type DBMS_TYPES.TYPECODE_OBJECT.
aname	Optional. Attribute's name. Could be NULL.
typecode	Attribute's typecode. Can be built-in or user-defined typecode (from DBMS_TYPES package).
prec	Optional. Required if typecode represents a NUMBER. Give precision and scale. Ignored otherwise.
scale	Optional. Required if typecode represents a NUMBER. Give precision and scale. Ignored otherwise.
len	Optional. Required if typecode represents a RAW, CHAR, VARCHAR, or VARCHAR2 type. Give length.
csid	Optional. Required if typecode represents a type requiring character information, such as CHAR, VARCHAR, or VARCHAR2.
csfrm	Optional. Required if typecode represents a type requiring character information, such as CHAR, VARCHAR, or VARCHAR2.
attr_type	Optional. ANYTYPE corresponding to a user-defined type. This parameter is required if the attribute is a user defined type.

Exceptions

- DBMS_TYPES.INVALID_PARAMETERS: Invalid Parameters (typecode, typeinfo)
- DBMS_TYPES.INCORRECT_USAGE: Incorrect usage (cannot call after calling EndCreate, and similar actions.)

ENDCREATE Member Procedure

This procedure ends creation of a transient ANYTYPE. Other creation functions cannot be called after this call.

Syntax

```
MEMBER PROCEDURE ENDCREATE(  
    self          IN OUT NOCOPY ANYTYPE);
```

Parameter

Table 308-6 ENDCREATE Procedure Parameter

Parameter	Description
self	The transient ANYTYPE that is being constructed.

GETINFO Member Function

This function gets the type information for the `ANYTYPE`.

Syntax

```
MEMBER FUNCTION GETINFO (  
    self      IN ANYTYPE,  
    prec      OUT PLS_INTEGER,  
    scale     OUT PLS_INTEGER,  
    len       OUT PLS_INTEGER,  
    csid      OUT PLS_INTEGER,  
    csfrm     OUT PLS_INTEGER,  
    schema_name OUT VARCHAR2,  
    type_name  OUT VARCHAR2,  
    version   OUT varchar2,  
    numelems  OUT PLS_INTEGER)  
RETURN       PLS_INTEGER;
```

Parameters

Table 308-7 GETINFO Function Parameters

Parameter	Description
<code>self</code>	The <code>ANYTYPE</code> .
<code>prec</code>	If typecode represents a number. Gives precision and scale. Ignored otherwise.
<code>scale</code>	If typecode represents a number. Gives precision and scale. Ignored otherwise.
<code>len</code>	If typecode represents a <code>RAW</code> , <code>CHAR</code> , <code>VARCHAR</code> , or <code>VARCHAR2</code> type. Gives length.
<code>csid</code>	If typecode represents a type requiring character information such as: <code>CHAR</code> , <code>VARCHAR</code> , or <code>VARCHAR2</code> .
<code>csid</code>	If typecode represents a type requiring character information such as: <code>CHAR</code> , <code>VARCHAR</code> , or <code>VARCHAR2</code> .
<code>schema_name</code>	Type's schema (if persistent).
<code>type_name</code>	Type's typename.
<code>version</code>	Type's version.
<code>numelems</code>	If <i>self</i> is a <code>TYPECODE_VARRAY</code> , this gives the <code>VARRAY</code> count. If <i>self</i> is of <code>TYPECODE_OBJECT</code> , this gives the number of attributes.

Return Values

The typecode of *self*.

Exceptions

- `DBMS_TYPES.INVALID_PARAMETERS`: Invalid Parameters (position is beyond bounds or the `ANYTYPE` is not properly Constructed).

GETATTRELEMINFO Member Function

This function gets the type information for an attribute of the type (if it is of `TYPECODE_OBJECT`). Gets the type information for a collection's element type if the *self* parameter is of a collection type.

Syntax

```
MEMBER FUNCTION GETATTRELEMINFO (  
    self          IN ANYTYPE,  
    pos           IN PLS_INTEGER,  
    prec          OUT PLS_INTEGER,  
    scale         OUT PLS_INTEGER,  
    len           OUT PLS_INTEGER,  
    csid          OUT PLS_INTEGER,  
    csfrm         OUT PLS_INTEGER,  
    attr_elt_type OUT ANYTYPE  
    aname         OUT VARCHAR2)  
RETURN          PLS_INTEGER;
```

Parameters

Table 308-8 GETATTRELEMINFO Function Parameters

Parameter	Description
<i>self</i>	The ANYTYPE.
<i>pos</i>	If <i>self</i> is of <code>TYPECODE_OBJECT</code> , this gives the attribute position (starting at 1). It is ignored otherwise.
<i>prec</i>	If attribute/collection element typecode represents a <code>NUMBER</code> . Gives precision and scale. Ignored otherwise.
<i>scale</i>	If attribute/collection element typecode represents a <code>NUMBER</code> . Gives precision and scale. Ignored otherwise.
<i>len</i>	If typecode represents a <code>RAW</code> , <code>CHAR</code> , <code>VARCHAR</code> , or <code>VARCHAR2</code> type. Gives length.
<i>csid</i> , <i>csfrm</i>	If typecode represents a type requiring character information such as: <code>CHAR</code> , <code>VARCHAR</code> , or <code>VARCHAR2</code> . Gives character set ID, character set form.
<i>attr_elt_type</i>	If attribute/collection element typecode represents a user-defined type, this returns the ANYTYPE corresponding to it. User can subsequently describe the <i>attr_elt_type</i> .
<i>aname</i>	Attribute name (if it is an attribute of an object type, <code>NULL</code> otherwise).

Return Values

The typecode of the attribute or collection element.

Exceptions

`DBMS_TYPES.INVALID_PARAMETERS`: Invalid Parameters (position is beyond bounds or the ANYTYPE is not properly constructed).

GETANYTYPEFROMPERSISTENT Function

This standalone function returns an `ANYTYPE` corresponding to a persistent type created earlier using the `CREATE TYPE` SQL statement. Starting with Oracle Database 19c, the `GETANYTYPEFROMPERSISTENT` function replaces the `GETPERISTENT` static function.

Syntax

```
GETANYTYPEFROMPERSISTENT(  
    schema_name      IN VARCHAR2,  
    type_name        IN VARCHAR2)  
RETURN               ANYTYPE;
```

Parameters

Table 308-9 GETANYTYPEFROMPERSISTENT Function Parameters

Parameter	Description
<code>schema_name</code>	Schema name of the type.
<code>type_name</code>	Type name.

Return Values

An `ANYTYPE` corresponding to a persistent type created earlier using the `CREATE TYPE` SQL statement.