



# INTRODUCCIÓN A LOS TIPOS DE DATOS LOB (Large Objects)

---

ORACLE

Diseño de Bases de Datos Multimedia



Universitat d'Alacant  
Universidad de Alicante



Departamento de  
Lenguajes y Sistemas Informáticos



# TIPOS DE DATOS LOB

- ¿Qué es LOB? *es un tipo de datos binario que puede contener una cantidad de datos variables. LOB significa "large objects" (en castellano, objetos grandes).*

CLOB	Un tipo de datos LOB que puede contener grandes cantidades de caracteres. Son compatibles con el carácter de base de datos establecida, tanto de caracteres de ancho fijo (tipo CHAR) y conjuntos de caracteres de ancho variable (VARCHAR),. El tamaño máximo es $(4 \text{ gigabytes} - 1) * (\text{tamaño del bloque de la base de datos})$ , es decir de 8 Terabytes para bloques de 2K, hasta 128 Terabytes para bloques de 7 K.
NCLOB	Lo mismo que el tipo de datos LOB, pero para caracteres UNICODE. Se usan cuando el juego de caracteres a almacenar no es el mismo que el de la base de datos. Por ello, se usará parte del campo para identificar el juego de caracteres.
BLOB	Un tipo de datos LOB binario. El tamaño máximo es $(4 \text{ gigabytes} - 1) * (\text{tamaño del bloque de la base de datos})$ . El contenido binario está almacenado en la BD.
BFILE	Contiene un localizador (puntero) a un archivo binario almacenado en una carpeta fuera de la base de datos. El tamaño máximo que puede contener una columna o variable de este tipo es de 4 gigabytes.



# TIPOS DE DATOS LOB

---

- Los LOB se pueden almacenar en las tablas de ORACLE de dos formas:
  1. El propio contenido del archivo como una columna más de una tabla.
  2. Una referencia a dónde se encuentra el archivo en el sistema operativo.



# TIPOS DE DATOS LOB

---

- **DBMS\_LOB** es un paquete que proporciona ORACLE para trabajar con datos LOB.
- Su uso en PL/SQL es DBMS\_LOB.función.
- Algunas de las funciones que podemos utilizar son:
  - **DBMS\_LOB.fileopen:** Abre un archivo.
  - **DBMS\_LOB.loadfromfile:** Lee un determinado número de bytes desde un archivo.
  - **DBMS\_LOB.getlength:** Devuelve el tamaño LOB en bytes.
  - **DBMS\_LOB.fileclose:** Cierra el archivo.

# TIPOS DE DATOS LOB

## ■ Todas las funciones de DBMS\_LOB son:

Summary of DBMS\_LOB Subprograms

Table B2-9 DBMS\_LOB Package Subprograms

Subprogram	Description
<a href="#">APPEND Procedures</a>	Appends the contents of the source <code>LOB</code> to the destination <code>LOB</code> .
<a href="#">CLOSE Procedure</a>	Closes a previously opened internal or external <code>LOB</code> .
<a href="#">COMPARE Functions</a>	Compares two entire <code>LOBs</code> or parts of two <code>LOBs</code> .
<a href="#">CONVERTTOLOB Procedure</a>	Reads character data from a source <code>CLOB</code> or <code>NCLOB</code> instance, converts the character data to the specified character, writes the converted data to a destination <code>LOB</code> instance in binary format, and returns the new offsets.
<a href="#">CONVERTTOCLOB Procedure</a>	Takes a source <code>LOB</code> instance, converts the binary data in the source instance to character data using the specified character, writes the character data to a destination <code>CLOB</code> or <code>NCLOB</code> instance, and returns the new offsets.
<a href="#">COPY Procedures</a>	Copies all, or part, of the source <code>LOB</code> to the destination <code>LOB</code> .
<a href="#">COPY_DBFS_LINK Procedures</a>	Copies the DBFS link in the source <code>LOB</code> to the destination <code>LOB</code> .
<a href="#">COPY_FROM_DBFS_LINK</a>	Retrieves the data for the LOB from the DBFS store.
<a href="#">CREATETEMPORARY Procedures</a>	Creates a temporary <code>LOB</code> or <code>CLOB</code> and its corresponding index in the user's default temporary tablespace.
<a href="#">DBFS_LINK_GENERATE_PATH Functions</a>	Returns a unique file path name for use in creating a DBFS Link.
<a href="#">ERASE Procedures</a>	Erases all or part of a <code>LOB</code> .
<a href="#">FILECLOSE Procedure</a>	Closes the file.
<a href="#">FILECLOSEALL Procedure</a>	Closes all previously opened files.
<a href="#">FILEEXISTS Function</a>	Checks if the file exists on the server.
<a href="#">FILEGETNAME Procedure</a>	Gets the directory object name and file name.
<a href="#">FILEOPEN Function</a>	Checks if the file was opened using the input <code>OFFILE</code> locators.
<a href="#">FILEOPEN Procedure</a>	Opens a file.
<a href="#">FRAGMENT_DELETE Procedure</a>	Deletes the data at the specified offset for the specified length from the <code>LOB</code> .
<a href="#">FRAGMENT_INSERT Procedures</a>	Inserts the specified data (limited to 32K) into the <code>LOB</code> at the specified offset.
<a href="#">FRAGMENT_MOVE Procedure</a>	Moves the amount of bytes ( <code>SLONG</code> ) or characters ( <code>CLOB/NCLOB</code> ) from the specified offset to the new offset specified.
<a href="#">FRAGMENT_REPLACE Procedures</a>	Replaces the data at the specified offset with the specified data (not to exceed 32K).
<a href="#">FREETEMPORARY Procedures</a>	Frees the temporary <code>LOB</code> or <code>CLOB</code> in the default temporary tablespace.
<a href="#">GET_DBFS_LINK Functions</a>	Returns the DBFS Link path associated with the specified SecureFile.
<a href="#">GET_DBFS_LINK_STATE Procedures</a>	Retrieves the current DBFS Link state of the specified SecureFile.
<a href="#">GETLINKSIZE Functions</a>	Returns the amount of space used in the <code>LOB</code> chunk to store the <code>LOB</code> value.
<a href="#">GETCONTENTTYPE Functions</a>	Returns the content ID string previously set by means of the <a href="#">SETCONTENTTYPE Procedure</a> .
<a href="#">GETLENGTH Functions</a>	Gets the length of the LOB value.
<a href="#">GETOPTIONS Functions</a>	Obtains settings corresponding to the <code>opt_lob_type</code> field for a particular <code>LOB</code> .
<a href="#">GET_STORAGE_LIMIT Function</a>	Returns the storage limit for LOBs in your database configuration.
<a href="#">INSTR Functions</a>	Returns the matching position of the <i>n</i> th occurrence of the pattern in the <code>LOB</code> .
<a href="#">ISOPEN Functions</a>	Checks to see if the <code>LOB</code> was already opened using the input locator.
<a href="#">ISTEMPORARY Functions</a>	Checks if the locator is pointing to a temporary <code>LOB</code> .
<a href="#">LOADLOBFROMFILE Procedure</a>	Loads <code>OFFILE</code> data into an internal <code>LOB</code> .
<a href="#">LOADCLOBFROMFILE Procedure</a>	Loads <code>OFFILE</code> data into an internal <code>CLOB</code> .
<a href="#">LOADFROMFILE Procedure</a>	Loads <code>OFFILE</code> data into an internal <code>LOB</code> .
<a href="#">MOVE_TO_DBFS_LINK Procedures</a>	Writes the specified SecureFile data to the DBFS store.
<a href="#">OPEN Procedures</a>	Opens a <code>LOB</code> (internal, external, or temporary) in the indicated mode.
<a href="#">READ Procedures</a>	Reads data from the <code>LOB</code> starting at the specified offset.
<a href="#">SET_DBFS_LINK Procedures</a>	Links the specified SecureFile to the specified path name. It does not copy the data to the path.
<a href="#">SETCONTENTTYPE Procedure</a>	Sets the content type string for the data in the LOB.
<a href="#">SETOPTIONS Procedures</a>	Enables CSCE features on a per-LOB basis, overriding the default LOB column settings.
<a href="#">SUBSTR Functions</a>	Returns part of the <code>LOB</code> value starting at the specified offset.
<a href="#">TRIM Procedures</a>	Trims the <code>LOB</code> value to the specified shorter length.
<a href="#">WRITE Procedures</a>	Writes data to the <code>LOB</code> from a specified offset.
<a href="#">WRITEAPPEND Procedures</a>	Writes a buffer to the end of a <code>LOB</code> .

[Acceso a documentación sobre  
DBMS\\_LOB](#)



# TIPOS DE DATOS LOB: Insertar datos en una columna de una tabla

---

- La función **EMPTY\_BLOB()** o **EMPTY\_CLOB()** nos devuelve un puntero vacío que nos permitirá insertar un valor en un campo BLOB, CLOB, y NCLOB. La inserción se hace en dos pasos.

## Ejemplo 1:

- Supongamos la tabla T con las columnas (id Number, Texto CLOB)
- El código ORACLE para insertar un valor en esa tabla sería:

```
Declare MiPuntero CLOB;  
Begin  
  INSERT into T (id, Texto)  
  VALUES (9244, empty_clob()) returning Texto into MiPuntero; -- Paso 1  
  dbms_lob.write( MiPuntero, longitud texto a añadir, posición desde la que se inserta , texto a insertar); -- Paso 2  
End
```



# TIPOS DE DATOS LOB: Insertar datos como referencia a un archivo externo.

---

- La función **BFILENAME()** crea un puntero que asocia un archivo existente en el sistema de ficheros del servidor con el ORACLE DIRECTORY (carpeta) que lo contiene.

## Ejemplo 2:

- Supongamos tabla T con campos (id Number, foto BFILE)
- Supongamos que la foto pepe.jpg está en la carpeta /img/fotos del sistema operativo.
- Además, hemos creado el siguiente objeto directorio de ORACLE con la sentencia:

```
CREATE DIRECTORY MIDIR AS '/img/fotos'
```

Entonces, para insertar (asociar ya que en la BD sólo se almacena el puntero) bastará con:

```
insert into tabla (id, Texto) VALUES(9244, BFILENAME('MIDIR','pepe.jpg');
```



## Usos prácticos de los tipos de dato LOB

---

- Cuando almacenemos información multimedia en nuestras bases de datos que tengan tamaño grande, usaremos los tipos de datos LOB.
- Recordad que la información se puede almacenar directamente en la base de datos, o incluir en ella una referencia a dónde se encuentra la información (en una carpeta del sistema operativo, o en una URL).