

# MultiWay

Acceso desde MultiBase y Cosmos a los gestores de base de datos de Oracle e Informix

**BASE100**

BASE 100, S.A.  
[www.base100.com](http://www.base100.com)

## Índice

<b>1. LA TECNOLOGÍA MULTIWAY .....</b>	<b>3</b>
<b>2. CONFIGURACIÓN EN RED LOCAL Y EN INSTALACIONES CLIENTE-SERVIDOR .....</b>	<b>5</b>
<b>3. CONFIGURACIÓN DE MULTIBASE EN RED LOCAL.....</b>	<b>6</b>
<b>4. CONFIGURACIÓN EN CLIENTE-SERVIDOR. MUTIWAY .....</b>	<b>7</b>
<b>5. FICHEROS DE CONFIGURACIÓN EN MULTIWAY .....</b>	<b>8</b>
5.1    DEFINICIÓN DE VARIABLES DE ENTORNO Y CONFIGURACIÓN.....	8
5.2    “MAPPING” DE CARACTERES: SECCIÓN “.TRANSLATIONS” .....	8
5.3    FUNCIONES DEL SERVIDOR: SECCIÓN “.FUNCTIONS”:	9
<b>6. INSTRUCCIONES ESPECÍFICAS DE MULTIWAY .....</b>	<b>10</b>
<b>7. CONFIGURACIÓN EN CLIENTE-SERVIDOR CONTRA EL GESTOR CTSQL.....</b>	<b>11</b>
7.1    CONFIGURACIÓN DE LAS MÁQUINAS CLIENTES .....	11
7.2    CONFIGURACIÓN DEL SERVIDOR.....	11
<b>8. CONFIGURACIÓN EN CLIENTE-SERVIDOR CONTRA EL GESTOR DE ORACLE .....</b>	<b>12</b>
8.1    CONFIGURACIÓN DE LAS MÁQUINAS CLIENTES .....	12
8.2    CONFIGURACIÓN DEL SERVIDOR.....	12
8.3    PORTABILIDAD MULTIBASE-ORACLE .....	15
8.3.1    Tipos de datos .....	15
8.3.2    Funciones .....	16
8.3.3    Instrucciones .....	16
8.3.4    Otros.....	20
<b>9. CONFIGURACIÓN EN CLIENTE-SERVIDOR CONTRA EL GESTOR DE INFORMIX .....</b>	<b>22</b>
9.1    CONFIGURACIÓN DE LAS MÁQUINAS CLIENTES .....	22
9.2    CONFIGURACIÓN DEL SERVIDOR.....	22
9.3    PORTABILIDAD MULTIBASE-INFORMIX .....	23
9.3.1    Tipos de datos .....	24
9.3.2    Funciones .....	24
9.3.3    Instrucciones .....	25
9.3.4    Otros.....	27
<b>10. CASOS PRÁCTICOS.....</b>	<b>28</b>
10.1    ORACLE.....	28
10.2    INFORMIX .....	29
<b>11. MENSAJES DE ERROR .....</b>	<b>31</b>

© Copyright BASE 100, S.A. Todos los derechos reservados. Ninguna parte de este documento puede ser reproducida ni transmitida por medio alguno sin permiso previo por escrito del titular del copyright. Todos los productos citados en este documento son marcas registradas o marcas comerciales registradas de sus respectivos propietarios.

[NT\_multiway\_v6]

## 1. La Tecnología MultiWay

---

El gestor de base de datos CTSQL puede funcionar tanto en modo local como en cliente-servidor, disponiendo en ambos casos de capacidad "multiservidor". Es decir, una misma instancia del lenguaje de cuarta generación CTL puede mantener abiertas de manera simultánea varias bases de datos a través de una única instancia del CTSQL.

MultiBase dispone de dos mecanismos diferentes de conectividad. El primero de ellos, a través del estándar ODBC de Microsoft, asegura un intercambio de datos con otras aplicaciones en los entornos gráficos más actuales. El segundo, MultiWay, permite la conexión en cliente-servidor tanto con bases de datos propias de MultiBase como con las bases de datos con interfaz SQL más extendidas del mercado. Este tipo de conexión a través de MultiWay presenta la ventaja de no necesitar modificación alguna en los programas fuentes.

MultiWay es una tecnología que dota a MultiBase y a Cosmos de unas extraordinarias prestaciones en instalaciones con arquitectura cliente-servidor.

MultiWay es una capa de software que permite a los lenguajes de MultiBase y Cosmos, y por tanto a cualquier aplicación desarrollada con ambas herramientas, el acceso transparente a diferentes gestores de bases de datos. Dicho acceso puede hacerse en modo local (lenguaje de desarrollo y servidor de base de datos en una misma máquina) o en modo cliente-servidor (lenguaje de desarrollo y servidor de base de datos en máquinas distintas).

Para ello, MultiWay se apoya en diferentes estándares del mercado, el SQL como lenguaje de interfaz con bases de datos relacionales, el TCP/IP como protocolo de comunicaciones y la librería WINSOCK.DLL como estándar para TCP/IP en el caso del Windows (máquina cliente).

La arquitectura del sistema es sencilla y muy ligera, no exigiendo, entre otras cosas, el soporte de red proporcionado por el correspondiente fabricante del gestor de base de datos.

Esta capa de software, que se sitúa al lado del servidor de la base de datos, hace a MultiBase independiente de otras capas de software de red de otros fabricantes de bases de datos, permitiendo la conexión remota y el enlace lógico entre el lenguaje de cuarta generación CTL y el SQL de distintos fabricantes.

MultiWay se encarga de proporcionar una visión homogénea de cualquier SQL que se conecte a los lenguajes de desarrollo de MultiBase y Cosmos, ofreciendo un único "set" de tipos de datos entre los diferentes "sets" de los distintos gestores de base de datos y una única gramática, al tiempo que se encarga de que cualquier conversión de tipo sea realizada de forma ortogonal.

Por último, MultiWay se encarga de la traducción de sentencias de SQL entre los distintos gestores de bases de datos soportados. Dichos gestores son los siguientes:

- CTSQL de MultiBase y Cosmos.
- Oracle (versiones 6, 7, 8, 9,10 y 11).
- Informix-OnLine e Informix-SE (versiones 4, 5, 7.9.40,10 y 11).

En este documento trataremos de las particularidades y diferencias existentes en MultiBase cuando se utilice cualquiera de los gestores de bases de datos anteriormente citados. En los epígrafes que siguen se indican las características propias en cada caso concreto.

NOTAS:

- Los límites de MultiBase no podrán ser superados por los gateways.
- Los gateways son aplicaciones de 32 bits. Esto implica que para que el gateway se pueda conectar a la base de datos de Oracle o Informix, es imprescindible que se instalen en el servidor las librerías de 32 bits de los gestores. El servidor donde se instalen debe permitir la ejecución de aplicaciones de 32 bits.
- Del mismo modo, CTSQL es también una aplicación de 32 bits, por lo que el servidor en el que se instale debe permitir la ejecución de este tipo de aplicaciones.

## 2. Configuración en red local y en instalaciones cliente-servidor

---

MultiBase y Cosmos poseen una serie de variables, denominadas variables de entorno, que se utilizan para personalizar el sistema. El lugar donde tendremos que especificar estas variables será el correspondiente fichero de personalización.

En este documento hacemos hincapié sobre aquellas variables de entorno y configuración necesarias para comenzar a trabajar con MultiBase y Cosmos en instalaciones de red local y en arquitecturas cliente-servidor.

La explicación de cada tipo de configuración se ha estructurado en los siguientes apartados:

- Configuración de MultiBase en red local.
- Instalaciones Cliente-Servidor. MultiWay.
  - Ficheros de configuración en cada gateway.
  - Instrucciones específicas de MultiWay.
- Configuración en instalaciones cliente-servidor con el gestor de base de datos CTSQL.
- Configuración en instalaciones cliente-servidor con gateways:
  - Gateway contra Oracle.
  - Gateway contra Informix.

### 3. Configuración de MultiBase en red local

---

La configuración de MultiBase en red local precisa la definición de las siguientes variables de entorno:

<b>DBPATH</b>	Lista de directorios del servidor de red donde existen o se pueden crear bases de datos. Si no se define esta variable se asumirá por defecto el directorio de trabajo.
<b>DBUSER</b>	Nombre de usuario de la base de datos que será utilizado para otorgar o revocar privilegios de acceso a la misma.
<b>STRANSDIR</b>	Directorio del servidor de la red donde está instalado MultiBase.

Ejemplo de entorno en las máquinas clientes:

```
DBTEMP=h:\tmp  
DBPATH=h:\apli1\almacen  
DBUSER=ctl  
STRANSDIR=h:\MultiBase
```

Estos ficheros indican lo siguiente:

- El nombre de usuario para acceder a la base de datos se denomina "ctl".
- MultiBase se encuentra instalado en el servidor en el directorio "MultiBase" de la partición "h:".

## 4. Configuración en cliente-servidor. MutiWay

El lenguaje de cuarta generación de MultiBase (CTL) dispone de toda la gramática del SQL, de forma que cuando encuentra una sentencia (estática o dinámica) de SQL la envía al gestor de la base de datos, que ejecutará la sentencia y devolverá los datos (el resultado) a CTL. En el caso de tener instalado MultiWay en cliente-servidor, CTL enviará la sentencia de SQL a través de la red al servidor que contenga la base de datos indicada. Dicha sentencia la recibirá MultiWay, quien localmente se pondrá en comunicación con el correspondiente gestor de base de datos, le pasará la sentencia SQL y, una vez ejecutada por el gestor, recibirá el resultado y lo enviará por la red al CTL que realizó la petición.

MultiWay está compuesto por una serie de gateways, uno para cada gestor de base de datos soportado. Estos gateways siempre ejecutan en máquinas Unix/Linux y deberán estar dados de alta (los que se vayan a utilizar) como servicios de red de la máquina correspondiente.

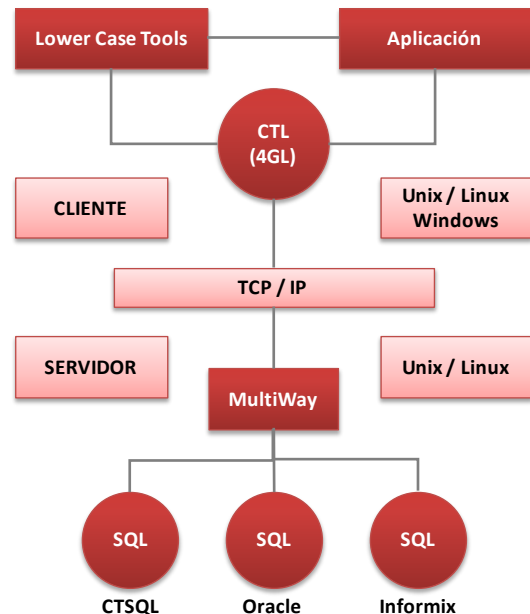
En el caso del gateway de Oracle existe también una versión para Windows.

Básicamente, las tareas que realiza MultiWay son las siguientes:

- 1) Traducción: Las gramáticas de los distintos SQL del mercado, si bien todas ellas conforman los estándares ANSI e ISO, suelen tener peculiaridades y extensiones que las hacen diferir unas de otras, en unos casos ligeramente y en otros de una forma considerable. CTL utiliza la gramática de SQL que le proporciona el CTSQL. Estas sentencias son traducidas por MultiWay antes de ser enviadas al correspondiente gestor de base de datos.
- 2) Conversión de tipos: Lo mismo que sucede con las gramáticas de los distintos SQL del mercado ocurre con los tipos de datos que soportan. Por ejemplo, mientras que CTSQL soporta un tipo DATE y un tipo TIME, otros SQL soportan un tipo DATETIME que engloba ambos. MultiWay se encargará de las conversiones de tipos adecuadas en ambas direcciones (CTL-gestor y gestor-CTL). Para ello, MultiWay utiliza un catálogo propio que se crea en la misma base de datos que se va a utilizar. Este catálogo será gestionado por MultiWay y está compuesto por una serie de tablas cuyos nombres comienzan por "mb" (mbtables, mbcolumns, mbindexes, etc.). Estas tablas guardan total simetría con las tablas de catálogo del CTSQL de MultiBase (systables, syscolumns, sysindexes, etc.).

Por lo tanto, y como resumen, puede decirse que MultiWay se encarga de que cualquier SQL del mercado sea visto por CTL como si fuese CTSQL.

De esta forma, MultiWay proporciona la absoluta portabilidad de cualquier aplicación desarrollada con MultiBase. El que una aplicación funcione en modo cliente-servidor o local y que funcione sobre un determinado servidor de SQL u otro ubicado en una máquina u otra dependerá únicamente de los valores que tomen determinadas variables de entorno.



## 5. Ficheros de configuración en MultiWay

---

Los ficheros de configuración de cada gateway definen las variables de entorno y de configuración específicas de cada gestor de base de datos (Informix u Oracle). Asimismo, estos ficheros pueden contener información relativa al “mapping” de caracteres a realizar entre MultiBase y el servidor de base de datos, así como las funciones propias del servidor que pueden utilizarse en instrucciones CTSQL desde MultiBase. Estos ficheros tienen la siguiente estructura:

### 5.1 Definición de variables de entorno y configuración

Al principio de este fichero, sin indicar ningún tipo de sección ni palabra reservada, se especifican los nombres de las variables de configuración de cada gestor de base de datos. Opcionalmente, también se podrán especificar aquellas variables de entorno propias del gestor. Las variables de entorno y de configuración de cada gestor se indican más adelante en este mismo documento.

Ejemplo:

```
ORACLE_HOME=/usr/oracle      #Oracle está instalado en /usr/oracle
ORACLE_SID=contabilidad      #La base de datos Oracle se
                              #denomina "contabilidad"
ORACLE_UID=jose/jose123      #Usuario y password Oracle.
DBSYN:conta=contabilidad     #Base de datos "conta" en MultiWay y
                              #"contabilidad" en Oracle.
```

### 5.2 “Mapping” de caracteres: Sección “.TRANSLATIONS”

Esta sección permite trabajar con una base de datos que no utilice el juego de caracteres de MultiBase (el GCS#2 de IBM). Para ello se define en el fichero de configuración del gateway correspondiente (Informix u Oracle) la sección “.TRANSLATIONS”, donde se especificarán las traducciones necesarias. En cada línea deben figurar dos parámetros: el carácter del juego de caracteres utilizado en la base de datos y su equivalente en el GCS#2 de IBM. Ambos parámetros pueden especificarse en cualquiera de los siguientes formatos: carácter entre comillas simples (‘a’), hexadecimal (0x61) u octal (\141). Cualquier texto escrito después de esta definición en la misma línea será considerado como un comentario.

Ejemplo:

```
.TRANSLATIONS
\341  \240  a acentuada
\351  \202  e acentuada
\355  \241  i acentuada
\363  \242  o acentuada
\372  \243  u acentuada
\361  \244  eñe minúscula
\321  \245  eñe mayúscula
\344  \204  a diéresis
\353  \211  e diéresis
\357  \213  i diéresis
\366  \224  o diéresis
\374  \201  u diéresis
\272  \247  `
\252  \246  &
```



### 5.3 Funciones del servidor: Sección “.FUNCTIONS”:

Existe también la posibilidad de utilizar funciones propias del servidor no reconocidas por MultiBase en frases SQL. Para ello habrá que definir en el fichero de configuración del gateway correspondiente (Informix u Oracle) una sección encabezada con “.FUNCTIONS” en la que se especifican las funciones del servidor que MultiBase ha de reconocer y el tipo de dato que devuelven. En cada línea debe figurar el nombre de la función y el tipo de dato CTSQL del valor devuelto, así como la longitud de éste en el caso del tipo de dato CHAR. Si el tipo de dato del valor devuelto coincide con el de algún parámetro enviado a dicha función se especificará el número de orden de ese parámetro precedido por el carácter “\$”.

Ejemplo:

```
.FUNCTIONS
decode      $3
func        integer
subs        char 20
```

## 6. Instrucciones específicas de MultiWay

---

Las instrucciones que se enumeran a continuación han de ejecutarse a través del método `SqlExec` de la clase `SqlServer` predefinida en MultiBase. Por ejemplo:

```
tsql "create multibase catalogs"
```

### CREATE MULTIBASE CATALOGS

Esta instrucción crea el catálogo de MultiWay para la base de datos en curso. Como ya se ha indicado anteriormente, MultiWay utiliza un catálogo propio, compuesto por una serie de tablas cuyos nombres comienzan por "mb" (mbtables, mbcolumns, mbindexes, etc.). Estas tablas guardan total simetría con las tablas de catálogo del CTSQL (systables, syscolumns, sysindexes, etc.).

### DROP MULTIBASE CATALOGS

Al contrario que la anterior, esta instrucción se encarga de borrar el catálogo de la base de datos en curso. Como ya se ha indicado, estas tablas en los gateways se identifican porque su nombre comienza por "mb".

### BEGIN MAINTENANCE MODE

A partir de la ejecución de esta directiva, las instrucciones SQL que se ejecuten a continuación no tendrán ningún efecto sobre la base de datos, salvo el que respecta a los catálogos de MultiWay.

Esta directiva es obligatoria para actualizar el catálogo de MultiWay cuando se han ejecutado instrucciones que afectan a la definición de la base de datos, utilizando el servidor directamente y no desde MultiBase.

### END MAINTENANCE MODE

Esta instrucción finaliza el modo de mantenimiento abierto mediante la instrucción `BEGIN MAINTENANCE MODE`.

NOTA: `BEGIN MAINTENANCE MODE` y `END MAINTENANCE MODE` se utilizarán para alimentar las tablas del catálogo de MultiWay (tablas "mb") con la información de las tablas creadas anteriormente con el propio servidor de la base de datos. Por ejemplo, si ya existe una base de datos en cualquiera de los gestores de base de datos y se instala el módulo MultiWay, habrá que alimentar las tablas del catálogo para que éste pueda manejar dicha base de datos. Más adelante se exponen unos ejemplos prácticos sobre esta operación.

### EXECSQL "instrucción\_sql\_servidor"

Esta instrucción ejecuta en modo transparente la frase SQL especificada en "instrucción\_sql\_servidor". Esto significa que dicha instrucción es enviada sin analizar ni traducir al servidor de la base de datos correspondiente. Es muy necesaria cuando se quiere ejecutar desde MultiBase alguna instrucción específica del servidor, y por lo tanto no perteneciente al CTSQL (la "instrucción\_sql\_servidor" nunca puede ser una sentencia `SELECT`).

## 7. Configuración en cliente-servidor contra el gestor CTSQL

---

Además de las variables de entorno generales, la instalación en cliente-servidor con el gestor CTSQL precisa, además, la configuración de las siguientes variables en las máquinas clientes:

<b>DBHOST</b>	Nombre del servidor donde se encuentra la base de datos.
<b>DBUSER</b>	Nombre de usuario para la base de datos que será utilizado para otorgar o revocar privilegios de acceso a la misma.
<b>XDBTEMP</b>	Directorio en el servidor para la creación de ficheros temporales.
<b>DBPATH</b>	Directorio en el servidor para la localización de la base de datos.
<b>DBPASSWORD</b>	Palabra clave (password) del usuario del servidor indicado en la variable de entorno DBUSER.
<b>DBSERVICE</b>	(Opcional). Nombre del gestor de base de datos en el servidor. Su valor por defecto es CTSQL.

### 7.1 Configuración de las máquinas clientes

Ejemplo de entorno:

```
DBSERVICE=ctsql
DBHOST=nombre_host
DBUSER=user1
DBPATH=/usr/almacen
XDBTEMP=/tmp
DBPASSWORD=clave
```

Este fichero indica las siguientes características:

- El gestor de la base de datos es CTSQL.
- El servidor donde se encuentran tanto la base de datos como el gestor se llama "nombre\_host".
- El usuario "user1" debe tener permiso de acceso a la base de datos que se encuentre en el directorio indicado por DBPATH.
- La variable de entorno DBPATH contiene un directorio de la máquina servidor denominado "/usr/almacen".
- El directorio donde se crearán los ficheros temporales es el "/tmp" de la máquina "servidor".
- La clave del usuario "user1" es "clave".

### 7.2 Configuración del servidor

Para trabajar con el CTSQL sólo será necesario configurar los ficheros de comunicaciones inetd.conf y "/etc/services" en servidores Unix/Linux. La forma de configurar estos ficheros se explica en el Anexo I del documento **Instalación del motor de base de datos en modo cliente-servidor**. En este documento se explican también los pasos a seguir para definir un servicio en un servidor Windows.

## 8. Configuración en cliente-servidor contra el gestor de Oracle

---

### 8.1 Configuración de las máquinas clientes

<b>DBSERVICE</b>	Nombre del gateway para el gestor de la base de datos Oracle. Por ejemplo: "gworacle".
<b>DBHOST</b>	Nombre de la máquina servidor. Por ejemplo: "unix".
<b>DBUSER</b>	Nombre de usuario de acceso al servidor. Por ejemplo: "user1".
<b>XDBTEMP</b>	Directorio en el servidor UNIX para la creación de ficheros temporales. Por ejemplo: "/tmp".
<b>DBPASSWD</b>	Palabra clave (password) del usuario UNIX incluido en la variable de entorno DBUSER.

Ejemplo de entorno en máquina cliente:

```
Environment oracle]
DBSERVICE=gworacle
DBHOST=nombre_host
DBUSER=user1
XDBTEMP=/tmp
DBPASSWD=clave
```

Este fichero indica las siguientes características:

- El gestor de la base de datos es Oracle.
- El servidor donde se encuentran tanto la base de datos como el gestor se llama "nombre\_host".
- El usuario "user1" debe tener permiso de acceso a la base de datos y al servidor.
- El directorio donde se crearán los ficheros temporales en el servidor es "/tmp".
- La clave del usuario UNIX/LINUX indicado en la variable DBUSER es "clave".

### 8.2 Configuración del servidor

En el servidor debe configurarse el fichero del entorno del gworacle.env. En este fichero podrán definirse tanto las variables de configuración como, opcionalmente, las variables de entorno que se indican a continuación, siendo en este caso prioritario el valor del entorno de usuario.

#### **IMPORTANTE**

*Todas las variables que se definan en el servidor tendrán que configurarse en el fichero gworacle.env. Asimismo, las variables ORACLE\_HOME, ORACLE\_UID y ORACLE\_SID se pueden configurar desde un programa CTL en la máquina cliente mediante el método putenv de la clase Module.*

Variables de configuración:

<b>ORACLE_PROC</b>	Indica al gateway si el gestor de Oracle permite o no la generación de procedimientos SQL. Sus posibles valores son ON y OFF, siendo el primero su valor por defecto.
<b>ORACLE_DECIMAL</b>	Indica el separador decimal utilizado por Oracle. Su valor por defecto es el punto decimal.
<b>DBLONGCHAR</b>	Esta variable indica el número máximo de caracteres que considerará MultiBase para el tipo de datos LONG de Oracle. El valor por defecto asignado a esta variable es de 2.000 caracteres.
<b>MBCOMMIT</b>	Indica si ha de hacerse o no un COMMIT después de cada instrucción SQL siempre que no haya cursores FOR UPDATE abiertos. Los únicos valores que puede tomar son ON y OFF (por defecto ON).
<b>MBLOADCOMMIT</b>	Indica que se producirá un COMMIT por cada una de las filas que se inserten con la instrucción de carga de datos LOAD. Sus posibles valores son ON, OFF y "n" (por defecto OFF). En el caso de asignar un número "n" a esta variable significará que se hará un COMMIT por cada "n" filas que se inserten con la instrucción LOAD.
<b>DBSYN</b>	Define sinónimos del nombre de la base de datos existente en Oracle. Establece la equivalencia entre el nombre de la base de datos de MultiWay con la del gestor de Oracle.
<b>DBSERVER</b>	Indica la versión del servidor a utilizar. Sus posibles valores son OR60, OR70 y OR80 (este último es válido para las versiones 8, 9,10 y 11 de Oracle).

Variables de entorno:

<b>ORACLE_UID</b>	Indica el nombre del usuario Oracle junto con su correspondiente clave de acceso (password).  Si el nombre del usuario Oracle y su clave de acceso coinciden con los del usuario del sistema operativo puede omitirse ORACLE_UID, ya que por defecto se tomará éste.
<b>ORACLE_SID</b>	Indica el nombre de la base de datos Oracle.  Para poder utilizar sinónimos de la base de datos habrá que configurar la variable DBSYN tantas veces como sinónimos existan. Si no se define ninguna de las dos variables, se considerará como valor de ORACLE_SID el especificado en la instrucción correspondiente de conexión a la base de datos del programa MultiBase que vaya a dialogar con Oracle.
<b>ORACLE_HOME</b>	Indica el directorio donde se encuentra instalado el servidor de la base de datos Oracle (obligatoria).

**ORACLE\_CONN** Indica el nombre del alias de la conexión existente en el fichero TNSNAMES.ORA. Esta variable de entorno es imprescindible a la hora de realizar conexiones del gateway en las que el servidor de base de datos Oracle y el gateway se encuentran en máquinas distintas o cuando la versión del gestor es de 64 bits y la del gateway es de 32 bits. En el caso de utilizar esta variable de entorno, el valor de ORACLE\_HOME deberá indicar la ruta donde se encuentra instalado el cliente Oracle que vamos a utilizar en la conexión (en el caso de gateways de 32 bits el cliente Oracle tendrá que ser el de 32 bits).

Ejemplo:

Contenido del fichero "gworacle.env" para funcionar en cliente-servidor:

```
ORACLE_HOME=/usr/oracle      # Oracle está instalado en /usr/oracle
ORACLE_SID=contabilidad      # La base de datos Oracle se
                              # denomina "contabilidad"
ORACLE_UID=jose/jose123      # Usuario y password Oracle
DBSERVER=OR80                # A partir de la versión 8.0 del gestor
                              # Oracle
DBSYN:conta=contabilidad     # Base de datos "conta" en MultiBase y
                              # "contabilidad" en Oracle
```

Si el servidor es Unix/Linux será necesario configurar los ficheros de comunicaciones inetd.conf o xinetd y "/etc/services".

Ejemplos:

- Fichero inetd.conf:

```
gworacle stream tcp nowait root /usr/bin/env env
LIBPATH=/u01/app/oracle/product/11.1.0.7/lib /home/ctl/lib/gworacle gwora-
cle system 3.0 /home/ctl/etc/gworacle.env NET
```

En la variable de entorno LIBPATH debe indicarse la ruta donde se encuentran instaladas las librerías de Oracle. Si la versión de Oracle es de 64 bits, el valor debe ser el que indique la ruta de las librerías de 32 bits.

- Sistemas operativos en los que el servicio ha de definirse en los ficheros de configuración del directorio "/etc/xinetd.d" (por ejemplo: RHEL, SLES):

```
service gworacle
{
    socket_type      = stream
    protocol         = tcp
    wait            = no
    user            = root
    env             = LD_LIBRARY_PATH=Librerías de oracle
    server          = /home/ctl/lib/gworacle
    server_args     = system 3.0 /home/ctl/etc/gworacle.env NET
    log_on_failure  += USERID
    port            = 20020
}
```

**LD\_LIBRARY\_PATH** Indicar la ruta donde están instaladas las librerías de Oracle. Éstas suelen estar en \$ORACLE\_HOME/lib si la instalación es de 32 bits y en \$ORACLE\_HOME/lib32 si es de 64 bits.

Si el servidor es Windows será necesario configurar el fichero de comunicaciones “services” e instalar y arrancar un servicio. En el documento **Instalación del motor de base de datos en modo cliente-servidor** se detallan las características que debe tener un usuario de Windows y los parámetros que deben pasarse al ejecutable mwlisten.exe para realizar este proceso.

Posibles mensajes de error:

- Usuario erróneo o sin permisos. Posible causa:
  - Valor incorrecto de la variable de entorno DBUSER.
- Imposible conectar con el servidor de SQL. Connection Failed (10061). Posibles causas:
  - Valor incorrecto de la variable de entorno DBSERVICE.
  - El servicio gworacle no está arrancado.
- Imposible conectar con el servidor de SQL. Connection Failed (10060). Posibles causas
  - Valor incorrecto de la variable de entorno DBHOST.
- Error -1034. Posibles causas:
  - La instancia de Oracle no está arrancada.
  - El nombre que se indica en la variable de entorno ORACLE\_SID no es correcto.
  - El nombre de la base de datos que se indica en DBNAME no es el mismo que el indicado en el primer parámetro de la variable DBSYN.
- Error -1017. Posibles causas:
  - Valores incorrectos en la variable de entorno ORACLE\_UID.

### 8.3 Portabilidad MultiBase-Oracle

Este apartado es una pequeña guía práctica de portabilidad de aplicaciones construidas con MultiBase en entornos con CTSQL a entornos con servidor Oracle versiones 6 a 10.

#### 8.3.1 Tipos de datos

Las conversiones entre los tipos de datos del CTSQL de MultiBase y los del servidor de base de datos de Oracle se realizan de forma automática según se indica en las siguientes tablas:

CTSQL	ORACLE
<b><i>Si es versión 6 y “size &lt; 256”</i></b>	
CHAR(size)	CHAR(size)
<b><i>Si es versión 7 y “size &lt; 2000”</i></b>	
CHAR(size)	VARCHAR2
<b><i>Si es versión 6 y “size &gt; 255” o versión 7 y “size &gt; 2000”</i></b>	
CHAR(size)	LONG
SMALLINT	SMALLINT
INTEGER	INTEGER
DECIMAL(p,s)	DECIMAL(p,s)

CTSQL	ORACLE
MONEY(p,s)	DECIMAL(p,s)
SERIAL	INTEGER
DATE	DATE
TIME	CHAR(11)
ROWID	ROWID

ORACLE	CTSQL
NUMBER(p)	INTEGER
NUMBER(p)	SMALLINT
NUMBER(p,s)	DECIMAL(p,s)
NUMBER(p,s)	MONEY(p,s)
CHAR(size)	CHAR(size)
VARCHAR2	CHAR(size)
LONG	CHAR(DBLONGCHAR)
DATE (sin TIME)	DATE

DBLONGCHAR es la variable de configuración que limita la longitud del LONG para su manejo en programas MultiBase.

El programador tendrá que tener en cuenta estas tablas si quiere actualizar el catálogo de MultiWay con una base de datos ya existente en Oracle, creando un esquema acorde con las conversiones automáticas que realiza el gateway (ver [“Casos prácticos”](#) al final de este documento).

### 8.3.2 Funciones

Restricciones de funcionalidad: Con la versión 6 del gestor de Oracle no se pueden utilizar las funciones contempladas por MultiBase en frases SQL (ver apartado “Funciones del CTSQL” en el capítulo 2 del Manual del SQL de MultiBase), al no existir equivalencia de las mismas en dicho gestor ni posibilidad de generarlas.

Por el contrario, la versión 7 del gestor de Oracle sí permite la creación de funciones SQL (previa instalación de la utilidad correspondiente de Oracle) que se almacenan en la misma base de datos. En este momento, y siguiendo este procedimiento, MultiWay sólo crea la función typelength de forma automática, siempre y cuando la variable ORACLE\_PROC se encuentre activada. El resto podrían ser simuladas sin dificultad siguiendo el mismo mecanismo.

Por lo que respecta a las funciones SQL propias de Oracle, éstas podrán utilizarse si se especifican adecuadamente en la sección “.FUNCTIONS” del fichero de configuración gworacle.env.

### 8.3.3 Instrucciones

#### Primera fase de traducción

Algunas de las diferencias sintácticas existentes en instrucciones del CTSQL respecto a las del servidor de Oracle se resuelven por el gateway de forma transparente al programador en una primera fase de traducción. Por ejemplo:

- Las comillas dobles se traducen por comillas simples.
- La cláusula MATCHES y los metacaracteres “\*?” por LIKE “%\_”.
- La cláusula UNIQUE por DISTINCT.



- La cláusula WHERE CURRENT OF 'cursor\_name' por WHERE ROWID = 'rowid'.
- DECLARE CURSOR FOR UPDATE por DECLARE CURSOR FOR UPDATE OF <columnas>.
- En subindicaciones: variable[n1,n2] por substr(varname,n1,n2-n1+1).
- Un literal vacío ("" ) por un blanco (" ").
- La palabra reservada TEXT por MBTX.
- Las expresiones TODAY y NOW por sysdate y sysdate con el formato HH24:MI:SS.

Las fases subsiguientes resuelven, de manera particularizada, más diferencias de este tipo, así como el resto de incompatibilidades existentes entre las instrucciones del CTSQL y las del servidor de Oracle.

### CREATE DATABASE, DROP DATABASE, START DATABASE y DATABASE

Estas instrucciones suponen la ejecución automática por parte del gateway de los siguientes pasos:

CREATE DATABASE	=>	CONNECT + CREATE MULTIBASE CATALOGS
DROP DATABASE	=>	DISCONNECT + DROP MULTIBASE CATALOGS
START DATABASE	=>	CONNECT
DATABASE	=>	CONNECT

Siendo CREATE MULTIBASE CATALOGS y DROP MULTIBASE CATALOGS instrucciones nuevas de MultiWay (disponibles para el programador), y CONNECT y DISCONNECT instrucciones internas para el gestor de Oracle.

Como puede observarse, la creación como tal de la base de datos ha de realizarse previamente a través del administrador de Oracle.

Cada una de las instrucciones anteriores asigna a la variable de entorno ORACLE\_SID el nombre de la base de datos de conexión (considerando los posibles valores de la variable de configuración DBSYN existente en gworacle.env). El usuario/password al que se conecta viene determinado por la variable de entorno ORACLE\_UID (definida en el entorno o en gworacle.env).

NOTA al programador: El cambio de base de datos de MultiBase tiene su equivalente frente a Oracle en las siguientes instrucciones dentro de un programa CTL:

```
close database <oldbname>
call putenv ("ORACLE_UID",<valor>,1)
database <newdbname>;
```

Siendo oldbname y newdbname los nombres de las bases de datos de Oracle (ORACLE\_SID) o sinónimos de las mismas, en cuyo caso deberán figurar como tal (DBSYN:dbsynonym=dbname) en el fichero de configuración gworacle.env. Por su parte, "valor" es el nuevo usuario/password (variable de entorno ORACLE\_UID del entorno de usuario o del fichero gworacle.env) al que se cambia.

### CREATE TABLE

Frente a la versión 6 del gestor de Oracle:

- El atributo NOT NULL se mantiene directamente por el SQL de Oracle.
- La cláusula PRIMARY KEY se simula de forma automática por el gateway a través de la creación de un índice único sobre las columnas especificadas y cuyo identificador es el nombre de la tabla precedido por PK.
- El atributo DEFAULT es simulado también por el gateway en las operaciones de INSERT cuyos valores en VALUES se introduzcan a través de constantes o variables hosts (esto se hace extensivo al FORM).

Restricciones de funcionalidad:

- Respecto al atributo DEFAULT, la simulación no funciona cuando los valores se introducen a través de una instrucción SELECT, ya que no puede controlarse su resultado (INSERT... SELECT...).
- El resto de atributos (entre ellos CHECK) y la clave referencial se aceptan gramaticalmente, pero no funcionan a nivel de SQL.
- No obstante, la definición de cualquier atributo queda almacenada en los catálogos de MultiBase, con lo que, al nivel de programa (FORM), muchos de ellos siguen teniendo validez (RIGHT, ZEROFILL, etc.), ya que se utiliza esta información como base.

Frente a la versión 7 del gestor de Oracle:

El SQL mantiene directamente los atributos CHECK y DEFAULT, además de NOT NULL. También las cláusulas PRIMARY KEY y FOREIGN KEY son manejadas automáticamente, con lo que el gateway ya no tiene que realizar ninguna labor de simulación para ellas.

Restricción de funcionalidad: El resto de atributos no funcionan al nivel de SQL.

#### CREATE TEMP TABLE

El empleo y manejo de tablas temporales desde programas CTL no varía a pesar de que internamente son tablas normales de usuario (y no temporales) identificadas con el número de proceso y el prefijo MBT. Dado que en Oracle no existe este tipo de tablas, éstas son simuladas automáticamente por el gateway, resultando invisible al programador esta carencia. Y lo mismo sucede con la tabla temporal de la instrucción "SELECT... INTO TEMP...".

NOTA al programador: Si a lo largo de la ejecución se ha cambiado sucesivamente de base de datos, puede que al finalizar el proceso se hayan creado tablas temporales en diferentes usuarios Oracle. En estos casos, el gateway las borrará naturalmente para simular esta temporalidad, pero haciendo dos supuestos a la hora de conectarse a los usuarios: Que no se haya cambiado de ORACLE\_SID y que la password de estos usuarios coincida con su nombre. En cualquier otro caso quedarían tablas sin borrar.

#### CREATE INDEX

Restricción de funcionalidad: El atributo DESC no es efectivo, ya que los índices son siempre ascendentes.

NOTAS al programador:

- Las tablas y los índices han de tener identificadores distintos, dado que así lo exige Oracle, y 18 caracteres como máximo (límite de CTSQL).
- Si la base de datos existía en Oracle con anterioridad a su manejo con MultiWay, puede suceder que haya identificadores que superen esa longitud de 18 caracteres. El problema se soluciona fácilmente creando sinónimos en Oracle para todos esos identificadores con la longitud admitida por CTSQL. Estos nombres serán los que se manejen desde programas CTL.
- Si al ejecutar una frase SQL que genera, modifica o borra un objeto de la base de datos se produce un error actualizando el catálogo de MultiBase, éste puede quedar inconsistente, ya que la acción sobre el objeto no se deshace. En este caso es necesario rectificarlo para que cualquier aplicación MultiBase funcione correctamente.

## ALTER TABLE

Como información, esta instrucción supone internamente la ejecución de los pasos que se indican a continuación:

```
CREATE TEMP TABLE <newtable>          (instrucción simulada a su vez)
INSERT INTO <newtable> SELECT FROM <oldtable>
DROP TABLE <oldtable>
RENAME TABLE <newtable> TO <oldtable>
CREATE INDEX ON...                      (los índices que tuviera la tabla)
```

## RENAME COLUMN

No funciona en esta versión.

## GRANT CONNECT

La sintaxis de esta instrucción en Oracle difiere de la del CTSQL. Para evitar esta incompatibilidad, el gateway añade al final de la misma la cláusula IDENTIFIED BY, y como password sigue el criterio de utilizar el mismo identificador empleado en el nombre de usuario.

## GRANT SELECT

No funciona la opción "GRANT SELECT (<columna1>... <columnan>)...", sino únicamente "GRANT SELECT...", sin especificación de columnas.

## INSERT

La inserción automática para el tipo de dato SERIAL (inexistente en Oracle) se simula en parte gracias a una tabla creada a tal efecto (MBSERIAL), de donde el gateway lee el último valor que asignó de esta forma (por cada tabla con una columna de este tipo, existe una fila con dicho valor).

## FETCH CURSOR

Restricciones de funcionalidad:

- En MultiBase, todos los cursores permiten el acceso a las filas siguiente, anterior, primera y última (NEXT, PREVIOUS, FIRST, LAST).
- Aunque Oracle no tiene cursores SCROLL, el gateway hace que esto se siga cumpliendo de forma transparente al programador, pero restringiéndolo a los cursores que no sean FOR UPDATE. Para ello emplea un fichero secuencial (SCRnnnpid.GW) donde realiza el scroll necesario de los registros seleccionados.

## LOCK TABLE

La instrucción LOCK TABLE será capaz de bloquear una tabla si la variable de configuración MBCOMMIT tiene el valor OFF (en el fichero gworacle.env), ya que este COMMIT automático después de cada instrucción SQL desbloquea las tablas cuando MultiBase funciona frente al SQL de Oracle. Lo normal es encerrar esta instrucción dentro de una transacción controlada por el programador, de manera que cuando se quiera desbloquear de verdad se finalice la transacción con COMMIT.

## OPEN CURSOR

- Los identificadores de CURSOR tendrán como máximo 15 caracteres en lugar de 18 (máximo permitido por CTSQL para cualquier identificador de la base de datos frente a cualquier SQL). Esta restricción permite distinguir cursores con igual nombre en diferentes módulos CTL, aspecto éste que sólo sabe manejar el CTSQL.
- Si el CURSOR tiene la cláusula FOR UPDATE, el comportamiento de esta instrucción es algo diferente frente al servidor de Oracle:

- Este SQL ejecuta la frase SELECT en el momento de abrir el CURSOR, bloqueando todas las filas obtenidas en la selección hasta el COMMIT de la transacción que la incluye.
- Si el usuario no controla las transacciones desde su programa, es la variable de configuración MBCOMMIT=ON (del fichero gworacle.env) la que puede provocar este COMMIT.
- Tanto en un caso como en otro, es necesario cerrar previamente el CURSOR, ya que el COMMIT sólo es enviado al SQL de Oracle por el gateway, si no existe ningún CURSOR abierto FOR UPDATE.
- Si al abrir el CURSOR Oracle encuentra problemas para bloquear todas las filas seleccionadas, el CURSOR permanecerá cerrado. La ejecución del programa CTL sigue su curso siempre y cuando el CURSOR haya sido definido "FOR UPDATE... NOWAIT", quedando registrado este incidente en la variable locked, que tomará el valor TRUE. Esto implica que para que un programa CTL se comporte de manera equivalente cuando funcione frente al servidor de Oracle es necesario controlar también en el OPEN los problemas de interbloqueo de cursores, y no sólo en el FETCH, como sucede en MultiBase con CTSQL.

### SELECT... FROM...

NOTA al programador: El gateway está simulando el comportamiento de CTSQL en un "SELECT {MIN,MAX,AVG,SUM}" sin GROUP BY (ya que Oracle devuelve una fila con valor NULL si el número de filas seleccionadas es cero, mientras que CTSQL no devuelve nada), con la salvedad de que el programador no puede distinguir el caso de una selección de filas con valores nulos (aunque esto no es habitual).

Restricciones de funcionalidad:

- La restricción DISTINCT/UNIQUE NO se puede incluir junto con la cláusula FOR UPDATE.
- NO se pueden utilizar alias en las cláusulas ORDER BY y GROUP BY. La solución para aquellos casos en que sea necesario su empleo (referencias a columnas compuestas: expresiones, agregados, etc.) es utilizar en su lugar los números de orden dentro de la lista de campos de la SELECT.

### ROLLFORWARD DATABASE y UPDATE STATISTICS

Al no tener equivalencia en Oracle, estas instrucciones no son enviadas a su SQL por el gateway, con lo que sus efectos son nulos aunque se utilicen en programas CTL, favoreciendo de esta manera la compatibilidad de fuentes frente a distintos servidores de bases de datos.

### UNLOCK TABLE

Esta instrucción es ignorada por el gateway, ya que Oracle no la contempla. Para conseguir el desbloqueo de una tabla ha de realizarse un COMMIT de la transacción dentro de la cual se hizo el bloqueo (para más información vea lo comentado anteriormente para la instrucción LOCK TABLE).

#### 8.3.4 Otros

- Tablas compartidas: Para compartir una tabla entre dos bases de datos (dos usuarios Oracle) se sigue un camino similar al realizado frente al gestor CTSQL.

El usuario A, creador de la tabla a compartir, da los permisos que considere oportunos al usuario B sobre la misma y sobre la tabla mbserial si existe algún campo cuyo tipo de dato es SERIAL.

El usuario B crea el sinónimo:

```
CREATE SYNONYM nombre_de_tabla
FOR usuario_A.nombre_de_tabla
```

para poder utilizar el mismo nombre de tabla. Además, crea la tabla en modo mantenimiento para que su catálogo la contenga y, con una instrucción UPDATE, modifica el campo dirpath de la fila que haya

aparecido en mbtables para esa tabla, cambiando el valor "usuarioB" por "usuarioA" (es en A donde realmente se halla la tabla).

Si en algún momento se altera la estructura de la tabla hay que repetir el procedimiento anterior (ALTER en modo mantenimiento y cambio de dirpath).

- Catálogo de MultiBase: Es posible romper la relación biunívoca existente entre bases de datos MultiBase y usuarios Oracle.

Si se quiere hacer corresponder una base de datos MultiBase con varios usuarios Oracle manteniendo un solo catálogo, éstos tendrían que compartir las tablas "mb\*" siguiendo el procedimiento anterior. Uno de los usuarios sería el creador real del catálogo (dador de permisos) y los demás compartirían las tablas (creando sinónimos).

La tabla mbserial no aparece en el catálogo, dada su naturaleza interna, pero sí han de poder manipularla todos, para lo cual, además de los permisos:

```
EXECSQL GRANT ALL PRIVILEGES ON mbserial TO PUBLIC
```

cada uno de los usuarios, excepto el creador del catálogo, necesita el siguiente sinónimo:

```
EXECSQL CREATE SYNONYM mbserial  
FOR usuario_creador.mbserial
```

Este procedimiento podría seguirse también con las tablas del entorno de programación (tablas "ep\*") si así se desea.

## 9. Configuración en cliente-servidor contra el gestor de Informix

---

### 9.1 Configuración de las máquinas clientes

<b>DBSERVICE</b>	Nombre del gateway para el gestor de la base de datos Informix. Por ejemplo: "gwinformix".
<b>DBHOST</b>	Nombre de la máquina "servidor". Por ejemplo: "unix".
<b>DBUSER</b>	Nombre de usuario para la base de datos que será utilizado para otorgar o revocar privilegios de acceso a la misma. Por ejemplo: "user1".
<b>XDBTEMP</b>	Directorio en el servidor UNIX para la creación de ficheros temporales. Por ejemplo: "/tmp".
<b>DBPATH</b>	Directorio en el servidor UNIX para la localización de la base de datos en el caso de emplear Informix SE. Por ejemplo: "/usr/apli1/almacen".
<b>DBPASSWD</b>	Palabra clave (password) del usuario UNIX indicado en la variable DBUSER.

Ejemplo de entorno en las máquinas clientes:

```
DBSERVICE=gwinformix
DBHOST=nombre_host
DBUSER=user1
DBPASSWD=clave
DBPATH=/usr/almacen          (sólo en Informix SE)
XDBTEMP=/tmp
```

Este fichero indica las siguientes características:

- El gestor de la base de datos es Informix.
- El servidor donde se encuentran tanto la base de datos como el gestor se llama "nombre\_host".
- El usuario "user1" debe tener permiso de acceso a la base de datos que se encuentre en el directorio indicado en DBPATH.
- La variable de entorno DBPATH contiene un directorio de la máquina servidor denominado "/usr/almacen".
- El directorio donde se crearán los ficheros temporales en el servidor es "/tmp".
- La clave del usuario UNIX indicado en la variable DBUSER es "clave".

### 9.2 Configuración del servidor

La configuración en el servidor se realiza a través del fichero de configuración de Informix "gwinformix.env". En este fichero podrán definirse tanto las variables de configuración como, opcionalmente, las variables de entorno que se indican a continuación, siendo en este caso prioritario el valor del entorno de usuario.

#### **IMPORTANTE**

*Todas las variables que se definan en el servidor tendrán que configurarse en el fichero gwinformix.env. Asimismo, la variable INFORMIXDIR se puede configurar desde un programa CTL en la máquina cliente mediante el método putenv de la clase Module.*

VARIABLES DE CONFIGURACIÓN:

**DBSERVER** Indica la versión del servidor a utilizar. Sus posibles valores son IX40, IX50 e IX70, siendo el primero el valor por defecto.

**DBEMBED** Indica la versión del embedded utilizado por el gateway de Informix. Sus posibles valores son: EIX40, EIX50 e EIX70. El valor por defecto dependerá del que tenga la variable de entorno DBSERVER. La relación entre sus valores es la siguiente:

DBSERVER	DBEMBED
IX40	EIX40
IX50	EIX50
IX70	EIX70

**DBSQL** Indica si el servidor de la base de datos es Informix On-Line o Informix-SE. Sus posibles valores son SE o ninguno, siendo este último el valor por defecto (Informix On-Line).

**DBSYN** Define sinónimos del nombre de la base de datos existente en Informix. Establece la equivalencia entre el nombre de la base de datos de MultiBase con la del gestor de Informix.

**INFORMIXSERVER** Indica el nombre del servidor Informix al que se conectará el gateway. Esta información debe estar en el fichero \$INFORMIXDIR/etc/sqlhosts del servidor en el que está instalado Informix. El protocolo de comunicación utilizado deber ser onsoctcp.

VARIABLE DE ENTORNO:

**INFORMIXDIR** Esta variable indica el path donde se encuentra instalado el servidor de la base de datos de Informix. Su definición es obligatoria.

Ejemplo:

Contenido del fichero "gwinformix.env" para funcionar en cliente servidor:

```
INFORMIXDIR=/usr/informix      # Informix está instalado en
                               # /usr/informix
DBSERVER=IX50                 # Versión 5.0 del gestor Informix
DBEMBED=EIX50                 # Versión 5.0 del Embedded
DBSYN:conta=contabilidad     # Base de datos "conta" en MultiBase y
                               # "contabilidad" en Informix
```

### 9.3 Portabilidad MultiBase-Informix

Este apartado es una pequeña guía práctica de portabilidad de aplicaciones construidas con MultiBase en entornos con CTSQL a entornos con servidor Informix-On Line (OL) o Informix-SE (SE).

### 9.3.1 Tipos de datos

Las conversiones entre los tipos de datos del CTSQL de MultiBase y los del servidor de base de datos de Informix se realizan de forma automática según las siguientes tablas:

CTSQL	INFORMIX
CHAR(size)	CHAR(size)
SMALLINT	SMALLINT
INTEGER	INTEGER
DECIMAL(p,s)	DECIMAL(p,s)
MONEY(p,s)	MONEY(p,s)
SERIAL	SERIAL
DATE	DATE
TIME	CHAR(11)

INFORMIX	CTSQL
CHAR(size)	CHAR(size)
SMALLINT	SMALLINT
INTEGER	INTEGER
DECIMAL(p,s)	DECIMAL(p,s)
MONEY(p,s)	MONEY(p,s)
SERIAL	SERIAL
DATE	DATE
DATETIME	CHAR(size)
INTERVAL	—
BYTE (versión 5)	—
TEXT (versión 5)	—
VARCHAR (size)	CHAR(size)
FLOAT	DECIMAL(p,s)
SMALLFLOAT	DECIMAL(p,s)

El programador tendrá que tener en cuenta estas tablas si quiere actualizar el catálogo de MultiBase con una base de datos ya existente en Informix, creando un esquema acorde con las conversiones automáticas que realiza el gateway (ver [“Casos prácticos”](#) al final de este documento).

### 9.3.2 Funciones

Restricciones de funcionalidad:

- Frente a la versión 4 del gestor de Informix:

No se pueden utilizar las funciones permitidas por el CTSQL de MultiBase (ver apartado “Funciones del CTSQL” en el capítulo 2 del Manual del SQL de MultiBase) en frases SQL que manejan o devuelven tipos TIME o expresiones de este tipo, al no existir equivalente de las mismas en el gestor de Informix ni posibilidad de generarlas. Esta clase de funciones son time(), hour(), minute(), second(), hms(), mt() y tomt(). Asimismo, tampoco puede emplearse la función typelength.



- Frente a la versión 5 del gestor de Informix:
  - Permite la creación de procedimientos SQL que se almacenan en la misma base de datos. En este momento, y siguiendo este método, MultiBase crea la función `typelength` de forma automática. El resto podrían ser simuladas siguiendo el mismo mecanismo.
  - Las funciones SQL propias de Informix pueden utilizarse si se especifican adecuadamente en la sección `“.FUNCTIONS”` del fichero de configuración.

### 9.3.3 Instrucciones

#### Primera fase de traducción

Algunas de las diferencias sintácticas existentes en instrucciones del CTSQL respecto a las del servidor de Informix se resuelven por el gateway de forma transparente al programador en una primera fase de traducción. Por ejemplo:

- `“DECLARE CURSOR FOR UPDATE”` por `“DECLARE CURSOR FOR UPDATE OF <columnas>”`.
- La palabra reservada `TEXT` por `MBTX`.
- La expresión `NOW` por un literal con la hora del sistema en formato `DBTIME` (variable de entorno de MultiBase).

Las fases subsiguientes resuelven, de manera particularizada, más diferencias de este tipo, así como el resto de incompatibilidades existentes entre las instrucciones del CTSQL y las del servidor de Informix.

#### CREATE DATABASE, DROP DATABASE, START DATABASE y DATABASE

Estas instrucciones suponen la ejecución automática por parte del gateway de los siguientes pasos:

```
CREATE DATABASE      => CREATE DATABASE + CREATE MULTIBASE CATALOGS
DROP DATABASE       => DROP DATABASE + DROP MULTIBASE CATALOGS
START DATABASE (SE) => START DATABASE + INSERT 'SYSLOG' EN 'MBTABLES'
START DATABASE (OL) => DATABASE + INSERT 'SYSLOG' EN 'MBTABLES'
DATABASE            => DATABASE
```

Siendo `CREATE MULTIBASE CATALOGS` y `DROP MULTIBASE CATALOGS` instrucciones nuevas de MultiBase disponibles para el programador.

Cada una de las instrucciones anteriores considera los posibles valores de la variable de configuración `DBSYN` existente en `gwinformix.env` para activar la base de datos correcta.

NOTAS al programador:

- El identificador `syslog` hace referencia a un registro especial de la tabla `mbtables` del catálogo cuya existencia indica que la base de datos es transaccional.
- En Informix On-Line (OL) el modo de funcionamiento transaccional se adquiere en el momento de la creación, si se especifica, o posteriormente vía administrador. Este último modo ha de reflejarse en el catálogo de MultiBase utilizando `START DATABASE WITH LOG` para mantener su integridad.

#### CREATE TABLE

Frente a la versión 4 del gestor de Informix:

- El atributo `NOT NULL` se mantiene directamente por el SQL de Informix.

- El atributo DEFAULT se simula en las operaciones de INSERT cuyos valores en VALUES se introduzcan a través de constantes o variables host. La simulación no funciona cuando los valores se introducen a través de una instrucción SELECT, ya que no es posible controlar su resultado (INSERT... SELECT...).
- La cláusula PRIMARY KEY se simula de forma automática por el gateway a través de la creación de un índice único sobre las columnas especificadas, y cuyo identificador es el nombre de la tabla precedido por PK.

Restricciones de funcionalidad:

- El resto de los atributos y la clave referencial se aceptan gramaticalmente, pero no funcionan al nivel de SQL.
- No obstante, la definición de cualquier atributo queda almacenada en los catálogos de MultiBase, con lo que, al nivel de programa (FORM), muchos de ellos siguen teniendo validez (RIGHT, ZEROFILL, etc.), ya que se utiliza esta información como base.

Frente a la versión 5 del gestor de Informix:

El SQL mantiene directamente los atributos CHECK y DEFAULT, además del NOT NULL. También las cláusulas PRIMARY KEY y FOREIGN KEY son manejadas automáticamente, con lo que el gateway ya no tiene que realizar ninguna labor de simulación para ellas.

Restricción de funcionalidad: El resto de atributos no funcionan a nivel de SQL.

Frente a la versión de Informix-Online:

El modo de bloqueo que se asigna por defecto al crear la tabla es por fila, es decir, "lock mode row", lo que permite la máxima concurrencia en el acceso a la misma.

NOTA al programador: Si al ejecutar una frase SQL que genera, modifica o borra un objeto de la base de datos se produce un error actualizando el catálogo de MultiBase, éste puede quedar inconsistente, ya que la acción sobre el objeto no se deshace. En este caso es necesario rectificarlo para que cualquier aplicación MultiBase funcione correctamente.

## **FETCH CURSOR**

Restricciones de funcionalidad:

- En MultiBase, todos los cursores permiten el acceso a las filas siguiente, anterior, primera y última (NEXT, PREVIOUS, FIRST, LAST).
- Frente a Informix, este tipo de operaciones se restringe a los cursores que no hayan sido definidos FOR UPDATE.

Frente a la versión de Informix-SE:

- Por razones de eficacia, en modo transaccional los cursores secuenciales (todos aquellos que no realicen operaciones de tipo scroll y que no sean FOR UPDATE) son definidos por el gateway como cursores scroll de Informix.
- En modo no transaccional, los cursores scroll (aquellos que realicen operaciones de tipo scroll y que no sean FOR UPDATE) son simulados por el gateway en base a cursores secuenciales de Informix. Para ello emplea un fichero secuencial (SCRnnnpid.GW) donde realiza el scroll necesario de los registros seleccionados.
- Las SELECT directas siempre se definen sobre cursores secuenciales de Informix.

Frente a la versión de Informix-Online.

- Por razones de eficacia, en modo transaccional los cursores secuenciales (todos aquellos que no realicen operaciones de tipo scroll y que no sean FOR UPDATE) son definidos por el gateway como cursores scroll de Informix.
- Las SELECT directas siempre se definen sobre cursores secuenciales de Informix.

#### **LOCK TABLE**

Si el servidor de Informix se encuentra funcionando en modo transaccional, la instrucción LOCK TABLE será capaz de bloquear una tabla si se encuentra dentro de una transacción. Bajo este modo será la instrucción COMMIT WORK la que produzca el mismo efecto que UNLOCK TABLE.

#### **OPEN CURSOR**

Los identificadores de CURSOR han de tener como máximo 15 caracteres en lugar de 18, que es el máximo permitido por CTSQL para cualquier identificador de la base de datos (frente a cualquier SQL). Esta pequeña restricción permite seguir diferenciando cursores con el mismo nombre en diferentes módulos CTL, aspecto éste que sólo sabe manejar el CTSQL.

#### **SELECT... FROM...**

El gateway está simulando el comportamiento de CTSQL en un "SELECT {MIN,MAX,AVG,SUM}" sin GROUP BY (ya que Informix devuelve una fila con valor NULL si el número de filas seleccionadas es cero, mientras que MultiBase no devuelve nada), con la salvedad de que el programador no puede distinguir el caso de una selección de filas con valores nulos (aunque esto no es habitual).

#### **UNLOCK TABLE**

En modo transaccional, para conseguir el desbloqueo de una tabla ha de realizarse el COMMIT de la transacción dentro de la cual se hizo el bloqueo de la misma (ver comentarios sobre la instrucción LOCK TABLE).

#### **9.3.4 Otros**

Tablas compartidas: Si las bases de datos son transaccionales es posible compartir tablas entre ellas. Para ello se deberá proceder como sigue: Desde el gestor de Informix ha de crearse el sinónimo "CREATE SYNONYM 'nombre' FOR 'base\_datos\_propietaria:nombre\_tabla'" en la base de datos desde la cual se desea acceder. Los permisos deberán otorgarse de acuerdo a la operación a realizar. A continuación, desde MultiBase, ha de crearse dicha tabla en modo mantenimiento para que la descripción figure en el catálogo de MultiWay.

## 10. Casos prácticos

---

Suponiendo que se haya desarrollado una aplicación en MultiBase frente al gestor de base de datos CTSQL, los pasos a seguir para que dicha aplicación funcione con el resto de gestores son los que se exponen más adelante para cada uno de ellos.

NOTA: Si se tuvieron en cuenta las restricciones en instrucciones CTL que existen para cada uno de los servidores de bases de datos a la hora de desarrollar la aplicación, eligiendo el mínimo común denominador de todas ellas no será necesario modificar los fuentes CTL de la misma ni recompilarla. Los mismos programas serán válidos con independencia del servidor, y los ejecutables CTL serán directamente utilizables.

### 10.1 Oracle

- 1) Generar previamente sobre el servidor CTSQL de MultiBase el esquema de la base de datos (fichero con extensión "sql" de igual nombre al de la base de datos. Por ejemplo: "almacen.sql"), para poder reproducir después la misma estructura frente al servidor de Oracle.
- 2) Confeccionar las variables del fichero de configuración del servidor de Oracle (gworacle.env).
- 3) En instalaciones cliente-servidor la variable DBSERVICE debe tener el valor gworacle. Junto a esta variable hay que configurar también las variables de entorno DBHOST, DBUSER y DBPASSWD.
- 4) Crear la base de datos a través del administrador de bases de datos de Oracle.
- 5) Crear un usuario de conexión a la base de datos a través del administrador de la base de datos Oracle.
- 6) A continuación existen dos posibilidades:
  - a) Mediante un programa CTL ejecutar, sobre la base de datos de Oracle, el esquema generado anteriormente sobre el fichero "almacen.sql". Por ejemplo:

```
main begin
    create database almacen
    tsql file "almacen"
end
```

La primera instrucción crea la base de datos "almacen" en MultiBase. Esta operación en MultiWay implica la generación del catálogo de la base de datos. Dicho catálogo está compuesto por las tablas "mb\*" que se crearán en la base de datos de Oracle.

La segunda instrucción crea las tablas, índices, claves primarias, etc. que estuviesen definidas en el fichero "almacen.sql". Estas tablas se crean en la base de datos de Oracle. Asimismo, el catálogo de la base de datos de MultiWay se carga con la información que genera dicha creación de elementos SQL.

- b) Ejecutar el esquema desde el entorno Oracle con instrucciones específicas de este servidor. Utilizando este procedimiento sería necesario crear después el catálogo de MultiWay y, entrando en modo mantenimiento, el resto del esquema actualizando dicho catálogo. Por ejemplo:

```
main begin
    create database almacen
    tsql "begin maintenance mode"
```

```
tsql file "almacen"  
tsql "end maintenance mode"  
  
end
```

Este ejemplo es similar al anterior, pero con la diferencia de que las tablas ya están creadas en la base de datos de Oracle. Por lo tanto, la única operación será actualizar el catálogo de MultiWay (tablas "mb\*"). Esta actualización se realiza al ejecutar el esquema "almacen.sql" pero en modo mantenimiento (BEGIN MAINTENANCE MODE y END MAINTENANCE MODE).

Ésta es la operación que hay que realizar en MultiBase cuando se desea importar desde el CTSQL una base de datos creada en Oracle.

- 7) Cargar los datos si procede.

## 10.2 Informix

- 1) Generar previamente sobre el servidor CTSQL de MultiBase el esquema de la base de datos (fichero con extensión "sql" cuyo nombre es igual que el de la base de datos. Por ejemplo, "almacen.sql"), para poder reproducir después la misma estructura frente al servidor de Informix.
- 2) Confeccionar las variables del fichero de configuración del servidor de Informix (gwinformix.env).
- 3) En instalaciones cliente-servidor la variable DBSERVICE debe tener el valor gwinformix. Junto a esta variable hay que configurar también las variables de entorno DBHOST, DBUSER y DBPASSWD.
- 4) A continuación existen dos posibilidades:
  - a) Mediante un programa CTL crear la base de datos y ejecutar el esquema generado anteriormente sobre el fichero "almacen.sql". Por ejemplo:

```
main begin  
    create database almacen  
    tsql file "almacen"  
end
```

La primera instrucción crea la base de datos "almacen" en Informix y el catálogo de la base de datos de MultiWay. Dicho catálogo está compuesto por las tablas "mb\*" que se crearán como si fuesen unas tablas más de la base de datos de Informix.

La segunda instrucción crea las tablas, índices, claves primarias, etc. que estuviesen definidas en el fichero "almacen.sql". Estas tablas se crean en la base de datos de Informix. Asimismo, el catálogo de la base de datos de MultiWay se carga con la información que genera dicha creación de elementos SQL.

- b) Crear la base de datos y ejecutar el esquema desde el entorno de Informix con instrucciones específicas de este servidor. Utilizando este procedimiento sería necesario crear después el catálogo de MultiWay y, entrando en modo mantenimiento, cargarlo con la información que genera la creación de los elementos SQL incluidos en el fichero "almacen.sql". Por ejemplo:

```
main begin  
    tsql "create multibase catalogs"  
    tsql "begin maintenance mode"  
    tsql file "almacen"
```

```
tsql "end maintenance mode"  
end
```

Este ejemplo se diferencia del anterior en que las tablas ya están creadas en la base de datos de Informix. Por lo tanto, la única operación será crear y actualizar el catálogo de MultiWay (tablas "mb\*"). Esta actualización se realiza al ejecutar el esquema "almacen.sql" en modo mantenimiento (BEGIN MAINTENANCE MODE y END MAINTENANCE MODE).

Ésta es la operación que hay que realizar en MultiBase cuando se desea importar desde CTSQL una base de datos creada en Informix.

- 5) Cargar los datos si procede.

## 11. Mensajes de error

Existe un cierto número de mensajes de error devueltos por el servidor que se esté utilizando (Informix u Oracle) que se traducen a mensajes del servidor CTSQL de MultiBase. El resto aparecerá bajo el texto genérico:

**ERROR número DEVUELTO POR EL SERVIDOR SQL**

Siendo “número” el número de mensaje de error devuelto por Informix u Oracle.

Los números de los errores traducidos en cada uno de los servidores de bases de datos son los que se indican a continuación.

Informix	CTSQL	Informix	CTSQL	Informix	CTSQL	Informix	CTSQL
-200	-1325	-239	-1299	-283	-1327	-326	-1205
-201	-1310	-240	-1115	-284	-1261	-327	-1175
-202	-1324	-241	-1171	-285	-1102	-328	-1359
-203	-1323	-242	-1375	-286	-1191	-329	-1397
-204	-1322	-243	-1118	-287	-1112	-331	-1396
-205	-1153	-244	-1119	-289	-1211	-337	-1149
-206	-1371	-245	-1118	-290	-1335	-346	-1113
-207	-1380	-246	-1119	-291	-1211	-347	-1210
-208	-1101	-247	-1170	-292	-1285	-348	-1214
-210	-1326	-250	-1117	-293	-1284	-349	-1393
-211	-1104	-251	-1271	-294	-1277	-350	-1290
-212	-1297	-253	-1325	-296	-1371	-352	-1363
-214	-1413	-254	-1328	-297	-1198	-355	-1376
-215	-1412	-255	-1173	-300	-1276	-356	-1354
-216	-1296	-257	-1244	-301	-1275	-357	-1148
-217	-1361	-258	-1102	-302	-1229	-358	-1392
-218	-1331	-259	-1243	-303	-1251	-359	-1391
-219	-1320	-261	-1374	-305	-1317	-360	-1264
-220	-1321	-262	-1240	-306	-1316	-362	-1356
-221	-1411	-263	-1117	-307	-1315	-364	-1357
-222	-1410	-264	-1410	-308	-1364	-365	-1334
-223	-1377	-265	-1173	-309	-1272	-367	-1253
-224	-1164	-266	-1116	-310	-1372	-370	-1355
-225	-1381	-269	-1283	-311	-1107	-371	-1291
-226	-1298	-270	-1110	-312	-1106	-373	-1390
-229	-1411	-271	-1114	-313	-1105	-374	-1273
-230	-1410	-272	-1225	-315	-1221	-375	-1166
-231	-1250	-273	-1224	-316	-1294	-376	-1163
-232	-1108	-274	-1223	-317	-1365	-377	-1176
-233	-1212	-275	-1222	-318	-1163	-378	-1214
-234	-1152	-276	-1243	-320	-1292	-379	-1227
-235	-1358	-277	-1241	-321	-1252	-380	-1162
-236	-1333	-279	-1229	-323	-1228	-381	-1226
-237	-1167	-280	-1319	-324	-1360	-382	-1147
-238	-1168	-282	-1318	-325	-1165	-383	-1146

<b>Informix</b>	<b>CTSQL</b>	<b>Informix</b>	<b>CTSQL</b>	<b>Informix</b>	<b>CTSQL</b>	<b>Oracle</b>	<b>CTSQL</b>
-384	-1145	-396	-1184	-1204	-1576	-1	-1299
-385	-1120	-397	-1109	-1205	-1575	-54	-1212
-386	-1282	-398	-1173	-1206	-1574	-904	-1363
-387	-1232	-399	-1161	-1210	-1571	-942	-1371
-388	-1231	-400	-1526	-1213	-1570	-1427	-1261
-389	-1230	-406	-1540	-1214	-1569		
-390	-1332	-691	-1202	-1215	-1568		
-391	-1281	-692	-1196	-1217	-1567		
-393	-1182	-1200	-1580	-1218	-1566		
-393	-1182	-1201	-1579	-1225	-1565		
-394	-1144	-1202	-1578	-1266	-1564		
-395	-1183	-1203	-1577				