



MultiBase

Notas a la versión 3.6, rel. 13 para Windows

BASE100

BASE 100, S.A.
www.base100.com



Índice

1. IMPLEMENTACIONES	3
1.1 CTSQL.....	3
1.1.1 <i>Estadísticas CTSQL</i>	3
1.1.2 <i>Tablas derivadas de la cláusula FROM</i>	5
1.1.3 <i>Otras implementaciones</i>	5
1.2 SQL-INTERACTIVO (CSQL.EXE)	6
1.3 TTTOOLS	6
1.4 ODBC.....	7
2. CORRECCIONES	8
2.1 CTL	8
2.2 CTSQL.....	8

© 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.

[NTMB3613v1.0]



1. Implementaciones

1.1 CTSQL

En esta versión de MultiBase se ha incluido la versión de motor 3.6 0.47.

1.1.1 Estadísticas CTSQL

El gestor de base de datos genera opcionalmente un fichero de estadísticas donde se muestra información de la ejecución de instrucciones SQL en la conexión a la base de datos:

- Prepare, Open, Execute, Fetch de cada instrucción.
- Tiempo empleado en la ejecución de cada una de estas funciones para cada instrucción.

Para activar/desactivar las estadísticas se han implementado los siguientes mecanismos:

- Mediante una instrucción SQL:

```
set statistics to 1 (activa las estadísticas)
```

```
set statistics to 0 (desactiva las estadísticas)
```

El fichero se genera cuando se ejecuta la instrucción «set statistics to 0» o cuando el cliente se desconecta del servidor.

- Mediante la variable de entorno CTSQLSTATISTICS:

Los posibles valores son:

TRUE/YES Activa las estadísticas.

FALSE/NO Desactiva las estadísticas.

Esta variable se puede definir en el fichero de configuración del motor (ctsql.ini).

El fichero de estadísticas se genera cuando finaliza la conexión a la base de datos.

Si está definida la variable de entorno y se emplea la instrucción SQL, el fichero se generará cuando se ejecute la sentencia SQL (set statistics to 0).



ID	Statement	Nº de Prepare	Nº de Open	Nº de Execute	Nº de Fetch	Tiempo Prepare (ms)	Tiempo Open (ms)	Tiempo Ejecución (ms)	Tiempo Fetch (ms)	Instrucción	Índice Usado
0	FNDBCHANGE	1	0	1	0	0,04	0,00	2,57	0,00	database almafac;	<ninguno>
1	FNSELECT	1	1	0	50	0,18	0,23	0,00	0,16	select albaranes.albaran,albaranes.cliente,albaranes.cliente,albaranes.formpago,albaranes.formpago,albaranes.fecha_albaran,albaranes.fecha_envio,albaranes.fecha_pago,albaranes.estado from albaranes;	<ninguno>
2	FNSELECT	3	52	0	52	1,57	11,71	0,00	1,15	select empresa from clientes where cliente = ? ;	clientes:primary
3	FNSELECT	1	50	0	50	0,91	10,97	0,00	0,98	select descripcion from formpagos where formpago = ? ;	formpagos:primary
4	FNSELECT	1	0	0	0	1,25	0,00	0,00	0,00	select articulo,proveedor from lineas;	<ninguno>
5	FNSELECT	1	3	0	38	0,17	0,69	0,00	0,21	select lineas.linea,lineas.articulo,lineas.proveedor, lineas.cantidad,lineas.descuento,lineas.precio,lineas.articulo,lineas.proveedor from lineas where albaran = ? ;	lineas:primary
6	FNSELECT	1	21	0	21	1,25	4,82	0,00	0,55	select descripcion from articulos where articulo = ? and proveedor = ? ;	articulos:primary
7	FNSELECT	1	1	0	50	0,99	0,24	0,00	0,11	select provincia, descripcion from provincias;	<ninguno>
8	FNSELECT	1	50	0	100	0,09	10,74	0,00	1,38	select empresa from clientes where provincia = ?;	<ninguno>

Por cada frase SQL, los datos que muestra son:

ID	Identificador interno de la instrucción.
Statement	Código interno de la instrucción SQL.
Nº de Prepare:	Indica el número de veces que se ha preparado la frase SQL.
Nº de Open	Indica el número de veces que se ha abierto el cursor de la frase SQL, en el caso de que se trate de una query.
Nº de Execute	Indica el número veces que se ha ejecutado la frase SQL.
Nº de Fetch	Indica el número veces que se ha ejecutado la instrucción Fetch sobre la frase SQL en el caso de que se trate de una query.
Tiempo Prepare (ms)	Tiempo (en milisegundos) que ha tardado el gestor en ejecutar las 'n' instrucciones Prepare.
Tiempo Open(ms)	Tiempo (en milisegundos) que ha tardado el gestor en ejecutar las 'n' instrucciones Open.
Tiempo Ejecución (ms)	Tiempo (en milisegundos) que ha tardado el gestor en ejecutar las 'n' instrucciones Execute.
Tiempo Fetch (ms)	Tiempo (en milisegundos) que ha tardado el gestor en ejecutar las 'n' instrucciones Fetch.
Instrucción	Frase SQL.



Índice usado

Nombre del índice seleccionado por el motor.

Esta columna, además, se mostrará con un color de fondo. Estos colores serán los siguientes:

- Verde. La celda se mostrará con este color cuando:
 - La instrucción SQL precisa de un índice y el gestor ha seleccionado uno.
 - Si la instrucción no precisa índice para su optimización (p.e.: “database stock”).
- Naranja. La celda será de color naranja si la sentencia SQL ejecutada precisa de un índice para ser optimizada y no se ha utilizado ninguno (p.e.: “select * from articulos where existencias > 75”), y no existe un índice que tenga como primer campo “existencias”.

1.1.2 Tablas derivadas de la cláusula FROM

Una tabla derivada en la cláusula FROM es una subselect que se sitúa después de la sentencia FROM de la consulta.

La tabla derivada solo existe en la consulta en la que es ejecutada, es decir, no forma parte del esquema de la base de datos.

Ejemplo:

```
select fecha_año, count(distinct cliente) from  
(select year(fecha_albaran) as fecha_año , cliente from albaranes) group by  
fecha_año
```

Año	Clientes
1998	34
2015	1
2016	2

NOTAS:

- a) No están implementadas las tablas derivadas como parte de los campos de la instrucción SELECT.
- b) Las tablas derivadas se deben emplear cuando queramos anidar funciones agregadas, hallar el promedio de las cantidades compradas o filtrar las filas de la tabla principal antes de un JOIN.

1.1.3 Otras implementaciones

- Permitir “select count(campo)” para que muestre el numero de registros donde el campo no es nulo. Ahora solo permite “select count(distinct campo)”.
- Permitir el uso de comillas dobles en la sintaxis del SQL para referirnos a identificadores (tablas, columnas, etc.). Variable de entorno ALLOWQUOTEDIDENTIFIERS.
- Posibilidad de que, si el número de registros es 0 y existe un agregado, retorne al menos un registro con valor nulo (AGGRETURNNULLROW).



1.2 SQL-Interactivo (csql.exe)

Se podrá ejecutar una sola sentencia. Para ello se deberá seleccionar la instrucción y después pulsar el botón **Execute selection** o la combinación de teclas [Ctrl]+[Enter].

A partir de esta versión se podrá ejecutar un fichero SQL desde la línea de comando utilizando el comando **csql**.

Sintaxis:

```
csql [fichero] [-ini inifile] [-runscript -connection <conexion>
-database <base de datos>]
```

Nuevos parámetros:

-runscript	Ejecuta el script sql sin esperar a que el usuario pulse el botón de ejecutar.
-connection <conexion>	Abre la conexión indicada. Esta conexión debe existir en el fichero de configuración que se pasa en el parámetro “-ini”. Parámetro obligatorio.
-database <bdd>	Abre la base de datos indicada. Parámetro obligatorio.

Ejemplos:

Línea de comando:

```
c:\MultiWay\bin\csql.exe -ini c:\MultiWay\etc\cosmos.ini c:\ctl\mbdemo\almacen.sql -
connection almacen -database almacen -runscript
```

Definir conexión en el fichero de configuración cosmos.ini:

```
[Connection]
almacen_local=local
[Environment almacen_local]
DBPATH=c:\ctl\mbdemo
DBNAME=alamcen
```

NOTAS:

- Se podrán borrar y crear bases de datos.
- En caso de error la ejecución se detendrá, se mostrará por pantalla el error al usuario, se abrirá el Sql-Interactivo y el cursor quedará posicionado en la línea del fichero que ha provocado el error.
- Abre la base de datos que se indica en el parámetro database independientemente del valor que se indique en la variable de entorno DBNAME.

1.3 TTTOOLS

Se han añadido dos nuevos parámetros a la herramienta de chequeo y reparación de índices TTTools.

Sintaxis:

```
TTTools [-ndl | dl] [-all] [path] [tabla]
```



Parámetros:

[-ndl dl]	Parámetros opcionales y excluyentes.
Ndl	Los índices se regeneran sin límite de duplicados.
dl	Los índices se regeneran con límite de duplicados (32.767). Si no se indica ningún valor los índices se regeneran sin límite de duplicados.
[-all]	Parámetro opcional. Se repararán todas las tablas de la base de datos. No pide confirmación.
[path]	Parámetro opcional. Directorio de la base de datos sin la extensión (dbs).
[tabla]	Parámetro opcional. Nombre de la tabla que se quiere reparar.

NOTA: Los parámetros [-all] y [tabla] son excluyentes.

Ejemplos:

Reparar todas las tablas de la base de datos sin que se muestre la parte gráfica de la herramienta:

```
tttools.exe -all c:\ctl\mbdemo\almacen
```

Reparar una tabla de la base de datos sin mostrar la parte gráfica:

```
Tttools.exe c:\ctl\mbdemo\almacen clientes
```

1.4 ODBC

A partir de esta release se distribuye con MultiBase la versión 3.00.7.17 del ODBC en arquitecturas de 32 y 64 bits.



2. Correcciones

2.1 CTL

- La función run no admitía que se pasasen más del 256 caracteres como parámetro.

2.2 CTSQL

- Al ejecutar una select de una view con agregados retornaba valores nulos.
- El optimizador del motor seleccionaba erróneamente una columna para índice cuando ésta se encontraba como parámetro de una función.
- La función to_char no funcionaba correctamente para DAY, DY y DL cuando el día de la semana era sábado. Retornaba Domingo.
- La función round siempre redondeaba a 2 decimales.
- Al crear muchas conexiones a una base de datos (1000) a veces se producía un mensaje de error: “Imposible abrir el catálogo del sistema (systables)”.