



# Glosario técnico de SQL

# Sumario

Base de datos relacional	6
Base de datos plana	6
Motor de base de datos	6
SQL6	
No-SQL	6
Tabla	6
Campos	7
Registros	7
Transact SQL	7
SGBD	7
DDL	7
DML	7
DCL	7
TCL	8
SELECT FROM	8
WHERE	8
Clave primaria	8
Clave foránea	8
Clave candidata	8
Clave Concatenada	9
Relación uno a uno	9
Relación uno a muchos	9
Relación muchos a muchos	9
Tablas	9
Campos y columnas	9
Registros	9
Operadores de comparación	10
ORDER BY	10
ASC, DESC	10
LIMIT	10
JOIN	10
UNION	10
COUNT()	11
SUM()	11
MIN()	11
MAX()	11
AVG()	11
HAVING	11
GROUP BY	12
Funciones de agrupación	12
Tipos de datos	12
Operador LIKE	12
Caracteres comodín	12
Subconsultas SELECT	12
DDL CREATE	13
DDL ALTER	13
ALTER ADD	13
ALTER MODIFY	13
DDL DROP	13
DDL TRUNCATE	13

NOT NULL	14
AUTO_INCREMENT	14
USE "Schema"	14
Funciones Escalares	14
Funciones de transformación	14
Vistas	14
Stored Procedures	15
Funciones	15
Tablas de hecho	15
Tablas transaccionales	15
Tablas dimensionales	15
DESCRIBE	16
Índices	16
INSERT	16
UPDATE	16
DELETE	16
BEGIN	16
END	17
DECLARE	17
SET	17
CREATE FUNCTION	17
RETURN	17
Comentarios	17
CREATE PROCEDURE	18
SET @variable = 'valor';	18
CALL	18
IN	18
OUT	18
PREPARE FROM	18
EXECUTE	19
DEALLOCATE PREPARE	19
IF THEN	19
ELSE	19
TRIGGER	19
CREATE TRIGGER	20
FOR EACH ROW	20
NEW	20
Funciones de Fecha y Hora	20
Funciones de Usuario	20
Funciones de Plataforma	20
SYS	20
Mysql.user	21
CREATE USER	21
IDENTIFIED BY	21
ALTER USER	21
RENAME USER	21
DROP USER	21
GRANT	21
REVOKE	21
SHOW GRANT	22
GRANT ALL	22
REVOKE ALL	22
START TRANSACTION	22
COMMIT	22

ROLLBACK	22
@@AUTOCOMMIT	23
SAVEPOINT	23
ROLLBACK TO <savepoint>	23
RELEASE <savepoint>	23
BACKUP	23
RESTORE	23
mysqldump	23
*nix	<b>23</b>

## Base de datos relacional

Elemento que almacena información respetando una estructura determinada y normalizada de tablas, campos y registros, usualmente relacionados entre sí, de manera tal que se mantenga la consistencia de la información allí alojada.

## Base de datos plana

La base de datos plana, a diferencia de la relacional, puede almacenar información general y depender de una o dos tablas no relacionadas entre sí. Usualmente este tipo de base de datos se utiliza para almacenar LOGs de una base de datos relacional más potente.

## Motor de base de datos

Aplicación o conjunto de aplicaciones que le dan vida al software de base de datos en sí.

## SQL

Siglas provenientes del inglés: structured query language, que corresponden a la forma en la cual se denomina al lenguaje de programación SQL. El lenguaje de consulta estructurada hace referencia a la forma casi natural de la sintaxis SQL para trabajar con objetos y datos de una base de datos.

## No-SQL

Siglas provenientes del inglés: not only SQL, que hacen referencia a un tipo de base de datos no relacionales. Nacieron y se popularizaron fuertemente a partir de 2010, y conforman documentos al tipo de tablas donde se almacena la información. A diferencia de SQL, estas bases de datos pueden tener una estructura diferente dentro de un mismo documento, que difiere entre sus registros.

## Tabla

Objeto de una base de datos donde se almacena la información, en forma de filas y columnas. Las filas son denominadas registros mientras que las columnas corresponden a los campos descriptivos de la información allí almacenada.

## **Campos**

Son las columnas de una tabla de datos que identifican al tipo de información que allí se almacena. Los campos le dan identidad a cada dato almacenado como registro.

## **Registros**

Tipo de información que es almacenada en una tabla, respetando la estructura y tipos de datos definida por los campos.

## **Transact SQL**

Nombre del lenguaje de programación que le da vida a SQL, particularmente dentro de SQL Server, y que determina la forma de la sintaxis de éste.

## **SGBD**

Siglas correspondientes a Sistema de Gestión de Base de Datos. Hacen referencia a la aplicación de software que permite la gestión total de la información de las bases de datos. Usualmente estos sistemas son aplicaciones visuales que facilitan la interacción con los diferentes objetos de una bb.dd.

## **DDL**

Siglas correspondientes a Data Definition Language. Hacen referencia al lenguaje SQL utilizado para crear y definir la estructura de los objetos de una base de datos: tablas, vistas, procedimientos almacenados, funciones, etc.

## **DML**

Siglas correspondientes a Data Manipulation Language. Hacen referencia a las sentencias SQL que nos permiten manipular la información almacenada en una base de datos (select, insert, update, delete).

## **DCL**

Siglas correspondientes a Data Control Language. Hacen referencia a los comandos SQL que permiten al administrador de una bb.dd. gestionar los datos

almacenados en esta, a través de los comandos (grant, revoke) asociados a los usuarios de la bb.dd.

## TCL

Siglas correspondientes a Transaction Control Language. Hacen referencia a los comandos que permiten confirmar o cancelar operaciones con datos en bb.dd. sql, a través de los comandos (begin transaction, commit, rollback).

## SELECT FROM

Comando SQL que forma parte de DML, el cual permite acceder a la información almacenada en una o más tablas. SELECT se utiliza en combinación con FROM, además de tener que definir al menos un campo de la tabla, y el nombre de ésta.

## WHERE

Comando SQL que forma parte de DML, y permite definir una serie de condiciones sobre los datos o registros con los cuales se esta operando. Usualmente se define un campo y la condición que debe cumplir, a posterior de la sentencia WHERE.

## Clave primaria

Es la columna o grupo de columnas de una tabla que identifica de forma exclusiva cada fila de la tabla. Impiden que los datos de dicho registro sean del tipo Nulo y agilizan la búsqueda de la información utilizando mecanismos acordes.

## Clave foránea

Es una clave específica que permite referenciar a dos o más tablas de una base de datos. La misma debe estar conformada por una clave primaria o clave candidata en la tabla de referencia.

## Clave candidata

Son claves que conforman atributos que identifican de manera única cada tupla (fila – tabla). Permiten definir que sus valores no se repiten en ninguna otra fila de la tabla.

## Clave Concatenada

Es un tipo de clave que se puede definir en las tablas, que involucra a más de un campo. Se usa en situaciones donde una clave primaria no satisface la necesidad de identificar unívocamente a un registro.

## Relación uno a uno

Se denomina relación uno a uno al vínculo de información entre dos tablas. Cada registro de cada tabla aparecerá solo una vez en éstas.

## Relación uno a muchos

Se denomina relación de uno a muchos cuando un registro de una tabla se asocia con uno o varios registros de otra tabla. Por lo tanto, la clave primera de una tabla se enlazará a la clave foránea de otra tabla.

## Relación muchos a muchos

Se denomina relación de muchos a muchos cuando uno o varios registros de una tabla se asocian con uno o varios registros de otra tabla. Para que esto pueda resolverse en SQL se deberá optar por incluir una tercera tabla que se ocupe de dicha relación.

## Tablas

Las tablas son el objeto en una base de datos que permite almacenar registros, teniendo como premisa una estructura definida por campos y registros, estableciendo una forma normalizada para el almacenamiento de dicha información.

## Campos y columnas

Los campos y columnas en una tabla de datos guardan relación directa, siendo las columnas quien define o estructura la forma relacional en la cual se almacenan los datos en dicha tabla, mientras que los campos hacen referencia al tipo de dato que se almacenará en cada posición.



## Registros

Es el conjunto de información que se almacena en las tablas de una base de datos. Cada registro es un cúmulo de datos homogéneo que representa a una entidad definida.

## Operadores de comparación

Son los símbolos que permiten establecer una comparación entre un tipo de datos específico de una entidad. En SQL se denomina a los operadores (igual, mayor que, menor que, mayor o igual que, menor o igual que, distinto de, cómo, entre otros).

## ORDER BY

Es la cláusula SQL propia de DML que permite establecer el orden de la información a listar, a través de uno o más campos. ORDER BY permite además definir la forma ascendente o descendente por la cual se ordenará dicha información.

## ASC, DESC

sentencia SQL que se utiliza en conjunto con ORDER BY para definir el orden de la información. ASC = ascendente, DESC igual DESCENDENTE.

## LIMIT

Es una cláusula SQL utilizada en las variantes ORACLE, MYSQL y MariaDB, entre otras, que permite definir la cantidad de información que se mostrará. Puede aceptar uno o dos parámetros, que definan un máximo de registros a mostrar o un máximo de registros a mostrar comenzando desde cierto punto del total de registros.

Su equivalente en SQL Server es TOP, y solo permite definir una cantidad de información a mostrar: (*N registros*).

## JOIN

Es la sentencia SQL que se utiliza para recuperar datos de varias tablas al mismo tiempo, debiendo estar éstas previamente relacionadas. Existen cuatro tipos de JOIN diferentes (INNER JOIN, LEFT JOIN, RIGHT JOIN, OUTER JOIN), los cuales evitan valores nulos, traen datos de la tabla principal, de la tabla secundaria, o unifican los datos de ambas tablas, incluyendo nulos.

## UNION

Permite unificar en una consulta de resultado, los datos almacenados en dos o más tablas. Los datos a unificar deben coincidir en tipos de datos para que esto pueda llevarse a cabo.

## COUNT()

Función integrada en SQL que permite contabilizar numéricamente un conjunto de registros a partir de un campo o columna específico. Dicha sentencia se utiliza para devolver un total numérico, no el conjunto de registros a los que se la relaciona.

## SUM()

Función integrada en SQL que permite obtener un total numérico mediante la suma de dichos datos almacenados en una tabla. El campo que se utiliza para calcular la suma debe ser del tipo numérico, entre todas las variantes posibles de acuerdo a los tipos de datos numéricos que SQL maneja.

## MIN()

La sentencia MIN() permite obtener el valor mínimo de un conjunto de datos numéricos alojados en la columna de una tabla.

## MAX()

La sentencia MAX() permite obtener el valor máximo de un conjunto de datos numéricos alojados en la columna de una tabla.

## AVG()

La sentencia AVG() permite obtener el valor promedio de un conjunto de datos numéricos, alojados en la columna de una tabla. Suma todos los datos numéricos y los divide por el total de registros, para así obtener el promedio buscado.

## HAVING

Es una sentencia SQL utilizada para devolver filas donde los valores agregados cumplan con condiciones previamente especificadas.

## GROUP BY

Sentencia SQL que permite agrupar un conjunto de registros de acuerdo a una determinada característica o valor.

## Funciones de agrupación

Son aquellas funciones que resumen las filas de un grupo de registros en un solo valor. [MIN\(\)](#), [MAX\(\)](#), [COUNT\(\)](#) son ejemplos de funciones de agrupación o agregación.

## Tipos de datos

Es la forma en la cual se define la información estructurada que se almacena en una tabla SQL. Los mismos pueden variar entre tipos de datos que almacenan texto, valores alfanuméricos, numéricos, de fecha, booleanos, entre otros tantos.

## Operador LIKE

Sentencia SQL utilizada en conjunto con WHERE, que permite comparar a un campo si este cumple con determinada condición. Dicha condición esta definida como algo parecido y no como una condición absoluta. El o los valores que se están evaluando se suelen representar en conjunto con un caracter comodín.

## Caracteres comodín

El caracter comodín, comúnmente identificado con el símbolo \* (*asterisco*), % (*por ciento*), \_ (*guión bajo*) o ? (*signo de pregunta*), permite reemplazar el resto o parte de la porción de un dato al momento de realizar una consulta que debe cumplir con determinada condición. Estos caracteres se suelen utilizar en conjunto con el operador LIKE.

## Subconsultas SELECT

La subconsulta SELECT forma parte de DML y, utilizada dentro de una subconsulta, se ocupa de obtener información clave o adicional de otra tabla, que servirá de utilidad para complementar a la consulta SELECT principal.

## DDL CREATE

Sentencia SQL perteneciente a DDL para crear objetos en una base de datos. Se utiliza tanto para crear bases de datos nuevas como también para crear tablas, vistas, o procedimientos almacenados.

## DDL ALTER

Sentencia SQL perteneciente a DDL para modificar, agregar o eliminar objetos de una tabla o vista de una base de datos. Se suele utilizar en combinación con las sentencias ADD, MODIFY o DROP.

## ALTER ADD

Sentencia SQL perteneciente a DDL que permite agregar elementos a un objeto de base de datos. Suele utilizarse como complemento de la sentencia ALTER.

## ALTER MODIFY

Sentencia SQL perteneciente a DDL que permite modificar elementos pertenecientes a un objeto de base de datos. Suele utilizarse como complemento de la sentencia ALTER.

## DDL DROP

Sentencia SQL perteneciente a DDL que permite eliminar elementos pertenecientes a un objeto de base de datos. Suele utilizarse como complemento de la sentencia ALTER.

## DDL TRUNCATE

Sentencia SQL perteneciente a DDL que permite eliminar la información almacenada en una tabla de base de datos. Es utilizada cuando se debe eliminar

toda la información junta, por encima de la sentencia DELETE, dado que TRUNCATE es más efectiva y rápida en la eliminación de registros.

## NOT NULL

Parámetro que suele aplicarse en la definición de un campo de las tablas de una base de datos en el momento de crearlas, para especificar que dicho campo o columna no puede aceptar valores nulos cuando almacena información.

## AUTO\_INCREMENT

Parámetro que suele aplicarse en la definición de un campo de las tablas de una base de datos en el momento de crearlo, para especificar que dicho campo, de carácter numérico, debe generar de forma automática y consecutiva su valor, cada vez que un nuevo registro es agregado a dicha tabla.

## USE "Schema"

Sentencia SQL que se escribe en una ventana de SCRIPT, para configurar de forma anticipada sobre qué base de datos se trabajará a continuación. La palabra "schema" se debe reemplazar por el nombre efectivo de dicha base de datos.

## Funciones Escalares

Son funciones predefinidas en el motor de base de datos que permiten convertir un valor desde un tipo de datos a otro, como también procesar valores referentes al tiempo (fecha/hora).

## Funciones de transformación

Son funciones predefinidas en el motor de base de datos que permiten transformar los datos de un tipo de datos específico en otro tipo de datos.

## Vistas

Son objetos de una base de datos que obtienen los valores de una o más tablas y los muestran de forma homogénea, aplicando las relaciones, selección de

campos, y ordenamiento previamente definido. Las vistas solo obtienen y muestran datos en tiempo real, no almacenan los mismos.

## Stored Procedures

Son programas del servidor de datos que permiten ser contruidos mediante [TRANSACTION SQL](#), y que operan los datos de una o más tablas y/o vistas, combinan el uso de triggers y/o funciones propias de SQL o personalizadas. Es la forma de programar bases de datos a un nivel avanzado, realizando todo el proceso del lado del servidor y evitando que aplicaciones de software de terceros deban ser responsables de todo el manejo de la información del lado del cliente.

## Funciones

Se denomina funciones a bloques de programación de aplicaciones de servidor que permiten realizar una tarea o procesamiento de registros específica del lado del servidor. Se llaman o utilizan igual que las funciones escalares y/o de transformación.

## Tablas de hecho

Las tablas de hecho (del inglés: fact tables) están conformadas por la tabla principal de un modelo dimensional, que contienen campos unidos a las tablas dimensionales. También almacenan métricas específicas para el análisis de la información.

## Tablas transaccionales

Son tablas que permiten llevar a cabo un gran número de transacciones breves en línea. Permiten el procesamiento de consultas de forma rápida, manteniendo la integridad de los datos en aquellos entornos de acceso concurrente, mientras que garantizan elevados niveles de efectividad.

## Tablas dimensionales

Las tablas dimensionales describen los aspectos de un proceso de negocio, como ser la determinación de objetivos de las ventas corporativas. En estos casos, las tablas dimensionales almacenan los atributos de dichos objetivos en sí mismas.

## DESCRIBE

Sentencia SQL que nos permite visualizar la estructura de una tabla, listando la información de sus campos, claves, valores predeterminados, tipos de datos, etcétera.

## Índices

Los índices son la forma de normalizar la información y permitir búsquedas más rápidas y eficientes en las tablas de una base de datos. Están relacionados directamente con las [Claves Primarias](#), [Claves Foráneas](#), [Claves Candidatas](#) y [Claves Concatenadas](#).

## INSERT

Sentencia SQL que se utiliza para insertar, o agregar, datos a una tabla. Se utiliza en combinación con la cláusula INTO para indicar qué tabla será la cual recibirá los datos, y con la cláusula VALUES, a través de la cual se definen los valores que se agregan. INSERT puede agregar un solo registro a una tabla, o un conjunto de registros agrupando a cada uno de ellos entre paréntesis, y separando a los registros subsiguientes por una coma.

## UPDATE

Sentencia SQL que se utiliza para actualizar datos de uno o más registros de una tabla. UPDATE se combina con la cláusula SET, especificando cuál es el campo y registro a modificar. UPDATE se puede combinar con la sentencia WHERE, para determinar cuál o cuáles registros de la tabla serán actualizados.

## DELETE

Sentencia SQL que permite eliminar uno o más datos de una tabla. Esta sentencia se puede combinar con la cláusula WHERE, para determinar cuáles serán los registros que se eliminarán. En el caso que no se utilice WHERE, se eliminarán todos los registros de la tabla en cuestión.

## BEGIN

Palabra reservada que se utiliza en Transact-SQL para iniciar el cuerpo de una función o procedimiento almacenado. Dentro de este cuerpo se escribirán las sentencias que le darán vida a dicho objeto.

## END

Palabra reservada que se utiliza en Transact-SQL para finalizar el cuerpo de una función o procedimiento almacenado. Determina cuál es el final del cuerpo que contiene las sentencias que le dan vida a dicho objeto.

## DECLARE

Palabra reservada que se utiliza en Transact-SQL para definir la creación de una variable.

## SET

Palabra reservada que se utiliza en Transact-SQL para establecer el valor de una variable.

## CREATE FUNCTION

Cláusula DDL para la creación de funciones personalizadas. Se utiliza tanto en una ventana de script dentro del SGBD como también en la línea de comandos.

## RETURN

Palabra reservada que se utiliza para devolver o retornar el valor almacenado en una variable, desde una función o procedimiento almacenado.

## Comentarios

Forma a través de la cual se establece la descripción de determinadas sentencias SQL dentro de una Vista, Función o Procedimiento almacenado. Los comentarios son utilizados en cualquier lenguaje de programación para explicar brevemente el comportamiento de una rutina de código.



En Transact-SQL se utilizan varios caracteres para la generación de comentarios. Entre estos destacamos a:

-- (doble guión)

# (numeral o almohadilla)

/\* (permite definir varias líneas de comentarios encerradas dentro de este juego de caracteres) \*/

Todo comentario encerrado o iniciado dentro de estos caracteres, no es tenido en cuenta por el intérprete del lenguaje Transact-SQL al momento de ejecutar la Vista, Función o Stored Procedure.

## CREATE PROCEDURE

Es la sentencia que se utiliza para crear un procedimiento almacenado, o Stored Procedure.

## SET @variable = 'valor';

Se utiliza para definir una variable temporal dentro de un Stored Procedure. La misma puede tener un valor del tipo string, numérico o toda una cláusula SQL completa.

## CALL

Comando que se utiliza en una ventana de script para invocar o ejecutar un Stored Procedure.

## IN

Forma de declarar un parámetro de entrada en un Stored Procedure.

## OUT

Forma de declarar un parámetro de salida en un Stored Procedure.

## PREPARE FROM

Comando que permite convertir una cláusula o consulta SQL en formato string a un formato del tipo objeto SQL, interpretable por el motor de la base de datos.

PREPARE objetoSQL FROM variableString;

## EXECUTE

Ejecuta dentro de un Stored Procedure un objeto SQL convertido desde un string.

EXECUTE objetoSQL;

## DEALLOCATE PREPARE

Permite deshacer un objeto SQL una vez que el mismo se utilizó dentro de un Stored Procedure. Es el equivalente al Garbage Collector de cualquier lenguaje de programación, usado para liberar de la memoria objetos que no se utilizarán.

## IF THEN

Cláusulas condicionales que permiten o no ejecutar una sentencia de acuerdo al resultado de una expresión evaluada.

## ELSE

Equivalente que se utiliza junto a IF THEN, para ejecutar una sentencia alternativa, en el caso que el resultado de la expresión evaluada por IF, no dé el resultado esperado.

## TRIGGER

Aplicación o programa almacenado en el servidor de base de datos, que se ejecuta de forma automática cuando uno o más eventos específicos ocurren en la bb.dd. Un trigger detecta las operaciones INSERT, UPDATE, DELETE, y realiza una acción al respecto. La acción puede definirse para que ocurra antes o después de la operación DML en cuestión.

## AFTER

Acción que se inicia después de una operación DML, detectada por un Trigger.

## BEFORE

Acción que se inicia previo a una operación DML, detectada por un Trigger.

## CREATE TRIGGER

Comando que permite definir la creación de un [Trigger](#).

## FOR EACH ROW

Comando que permite recorrer un SET de datos dentro de una aplicación de servidor de base de datos.

## NEW

Palabra reservada que se ocupa de detectar o ubicar cada nuevo registro agregado en un set de datos. No se utiliza a nivel registro (o fila) completo, sino a nivel campo.

## Funciones de Fecha y Hora

Son un conjunto de funciones propias de Mysql, que permiten obtener datos relacionados a la fecha y hora del momento en el cual se ejecutan.

## Funciones de Usuario

Son un conjunto de funciones propias de Mysql, que permiten obtener información del usuario de la sesión, del sistema o usuario en general, para utilizarse dentro del motor de la base de datos.

## Funciones de Plataforma

Son un conjunto de funciones propias de Mysql, que permiten obtener información del nombre de la base de datos y su versión.

## SYS

Base de datos del sistema donde se almacena información referente al motor de base de datos en sí, performance de éste y demás cuestiones propias de un software.

## Mysql.user

Tabla que almacena la información de usuarios de la bb.dd. junto a cada uno de los permisos para trabajar sobre la bb.dd. y sus objetos.

## CREATE USER

Sentencia [TCL](#) que permite gestionar la creación de un usuario dentro de la base de datos.

## IDENTIFIED BY

Comando que se utiliza junto a la sentencia CREATE USER, para determinar el password o contraseña del usuario que se esta creando.

## ALTER USER

Comando para modificar un usuario o sus datos asociados, previamente creado con la sentencia CREATE USER.

## RENAME USER

Comando para cambiar el nombre de un usuario, previamente creado con la sentencia CREATE USER.

## DROP USER

Comando para eliminar un usuario, previamente creado con la sentencia CREATE USER.

## GRANT

Sentencia que permite definir los permisos de escritura, modificación, eliminación y/o lectura de datos y uso de objetos de la bb.dd. para un usuario determinado.

## REVOKE

Sentencia que permite eliminar uno o más permisos de escritura, modificación, eliminación y/o lectura de datos y uso de objetos de la bb.dd. para un usuario determinado.

## SHOW GRANT

Permite verificar los permisos otorgados a un usuario de bb.dd.

## GRANT ALL

Permite establecer todos los permisos DML sobre un usuario de bb.dd. previamente creado.

## REVOKE ALL

Permite eliminar todos los permisos DML sobre un usuario de bb.dd. previamente creado.

## START TRANSACTION

Comando del tipo [TCL](#), que permite definir el inicio de una transacción, previo a la ejecución de una o más operaciones del tipo [DML](#), en una tabla de datos.

## COMMIT

Comando del tipo [TCL](#), que permite confirmar o aplicar los cambios realizados mediante operaciones del tipo [DML](#), en una tabla de datos.

## ROLLBACK

Comando del tipo [TCL](#), que permite deshacer cambios realizados mediante operaciones del tipo [DML](#), en una tabla de datos.

## @@AUTOCOMMIT

Variable global de la base de datos Mysql, que permite establecer su valor en 0, para evitar confirmar automáticamente una operación del tipo [DML](#), o establecer su valor en 1, para confirmar automáticamente estas operaciones.

## SAVEPOINT

Comando del tipo [TCL](#), que permite establecer un punto, marca, o bookmark cada determinado punto de una transacción masiva iniciada por una operación del tipo [DML](#), en una tabla de datos.

## ROLLBACK TO <savepoint>

Permite deshacer las operaciones del tipo [DML](#) hasta un determinado bookmark, generado previamente por SAVEPOINT.

## RELEASE <savepoint>

Permite eliminar un determinado bookmark, generado previamente por SAVEPOINT.

## BACKUP

Copia de seguridad, respaldo o reserva, que se realiza sobre los datos originales, con fin de disponer de un medio alternativo para recuperarlos, en caso de una falla o pérdida de éstos.

## RESTORE

Recuperación de datos perdidos o alterados erróneamente, a partir de una copia de seguridad realizada con antelación.

## mysqldump

Aplicación de Terminal o Línea de Comandos utilizada para poder iniciar una copia de seguridad de una base de datos o Schema determinado.

## \*nix

Referencia utilizada para mencionar de forma generalizada a los sistemas operativos del tipo UNIX, incluyendo dentro de éstos a: Mac OS, Unix, Minix, Xenix, Linux, BSD, o cualquier otro S.O. derivado de estos.