



Clase 18

BASE DE DATOS

1era parte





¿Qué es SQL?

El **Lenguaje de Consulta Estructurado** popularmente conocido por sus siglas en inglés como **SQL**, es un tipo de lenguaje de programación que ayuda a solucionar problemas específicos o relacionados con la definición, manipulación e integridad de la información representada por los datos que se almacenan en las bases de datos.



¿Qué es SQL?

- SQL son las siglas de Structured Query Language
 - SQL le permite acceder y manipular bases de datos
 - SQL se convirtió en un estándar del American National Standards Institute (ANSI) en 1986 y de la International Organization for Standardization (ISO) en 1987.
-



¿Qué puede hacer SQL?

- SQL puede ejecutar consultas en una base de datos
- SQL puede recuperar datos de una base de datos
- SQL puede insertar registros en una base de datos
- SQL puede actualizar registros en una base de datos
- SQL puede eliminar registros de una base de datos
- SQL puede crear nuevas bases de datos
- SQL puede crear nuevas tablas en una base de datos
- SQL puede crear procedimientos almacenados en una base de datos
- SQL puede crear vistas en una base de datos
- SQL puede establecer permisos en tablas, procedimientos y vistas



¿Qué es MySQL?

En cuanto a la definición general, MySQL es un sistema de gestión de bases de datos relacionales de código abierto (RDBMS, por sus siglas en inglés) con un modelo cliente-servidor. **RDBMS** es un software o servicio utilizado para crear y administrar bases de datos basadas en un modelo relacional. Ahora, echemos un vistazo más de cerca a cada término:



INTERFACES GRAFICAS PARA MySQL

- **MySQL Workbench.** Debemos comenzar por MySQL Workbench, una aplicación creada por los propios desarrolladores de MySQL. Se trata de una herramienta visual muy completa que, además, es multiplataforma y se puede instalar en sistemas Windows, Linux y Mac. Nos ayudará en cualquier etapa del desarrollo o la administración de la base de datos, desde el diseño del modelo de datos, el acceso a la información y la generación de diagramas, etc. Dispone también un editor de consultas y un módulo dedicado a la optimización de rendimiento, entre otros. Si necesitamos una herramienta profesional y tenemos la capacidad de instalar software de interfaz gráfica en la máquina, MySQL Workbench será, sin duda, la primera opción que debemos tener en cuenta. La única desventaja es que puede resultar una herramienta un poco compleja, por lo que quizás usuarios con menos experiencia prefieran otras alternativas.



INTERFACES GRAFICAS PARA MySQL



MySQL Workbench

Server Intranet MySQL x Mysql@192.168.1.93:3306 x

File Edit View Query Database Server Tools Scripting Help

Navigator: knowledge_areas x

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

SCHEMAS

Filter objects

- knowledge_areas
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers

Information

Object Info Session

SQL Editor

1 • `SELECT * FROM dw2019.knowledge_areas;`

Limit to 1000 rows

Result Grid

	id	name	icon	created_at	updated_at
▶	1	Iniciación	wb_incandescent	NULL	NULL
	2	Diseño	dashboard	NULL	NULL
	3	Backend	dns	NULL	NULL
	4	Frontend	web	NULL	NULL
	5	Apps	apps	NULL	NULL
	6	Sistemas	computer	NULL	NULL

Result Grid Form Editor

Apply Revert

Context Help Snippets

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 1	00:55:05	SELECT * FROM dw2019.knowledge_areas LIMIT 0, 10	6 row(s) returned	0.000 sec / 0.000 sec



INTERFACES GRAFICAS PARA MySQL



- [PhpMyAdmin](#). En segundo lugar nos encontramos este conocido y altamente extendido gestor de MySQL con interfaz web. Es difícil encontrar un desarrollador de PHP que no conozca, use o la haya usado en algún momento o proyecto. Nos ofrece todas las herramientas de administración de MySQL de una forma sencilla, como la generación de tablas, su manipulación, acceso a los datos, usuarios de MySQL, creación y restauración de copias de seguridad, etc. Al tratarse de una interfaz web, es posible instalarla en cualquier servidor con soporte a PHP y disponer así de una herramienta de gestión de base de datos remota o local.



INTERFACES GRAFICAS PARA MySQL



phpMyAdmin

Recent Favorites

- help_relation
- help_topic
- innodb_index_stats
- innodb_table_stats
- ndb_binlog_index
- plugin
- proc
- procs_priv
- proxies_priv
- servers
- slave_master_info
- slave_relay_log_info
- slave_worker_info
- slow_log
- tables_priv
- time_zone
- time_zone_leap_second
- time_zone_name
- time_zone_transition
- time_zone_transition_type
- user**
- performance_schema

Server: localhost:3306 » Database: mysql » Table: user "Users and global privileges"

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Showing rows 0 - 6 (7 total, Query took 0.0005 seconds.)

SELECT * FROM `user`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

				Host	User	Password	Select_priv	Insert_priv
<input type="checkbox"/>	Edit	Copy	Delete	localhost	root	*47B2CD842066CE03F0B8F4BFFEC6889515C37966	Y	Y
<input type="checkbox"/>	Edit	Copy	Delete	bookle-5.local	root	*81F5E21E35407D884A6CD4A731AEBFB6AF209E1B	Y	Y
<input type="checkbox"/>	Edit	Copy	Delete	127.0.0.1	root	*81F5E21E35407D884A6CD4A731AEBFB6AF209E1B	Y	Y
<input type="checkbox"/>	Edit	Copy	Delete	::1	root	*81F5E21E35407D884A6CD4A731AEBFB6AF209E1B	Y	Y
<input type="checkbox"/>	Edit	Copy	Delete	localhost			N	N
<input type="checkbox"/>	Edit	Copy	Delete	bookle-5.local			N	N
<input type="checkbox"/>	Edit	Copy	Delete	%	mik	*A4B6157319038724E3560894F7F932C8886EBFCF	N	N

Check all With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

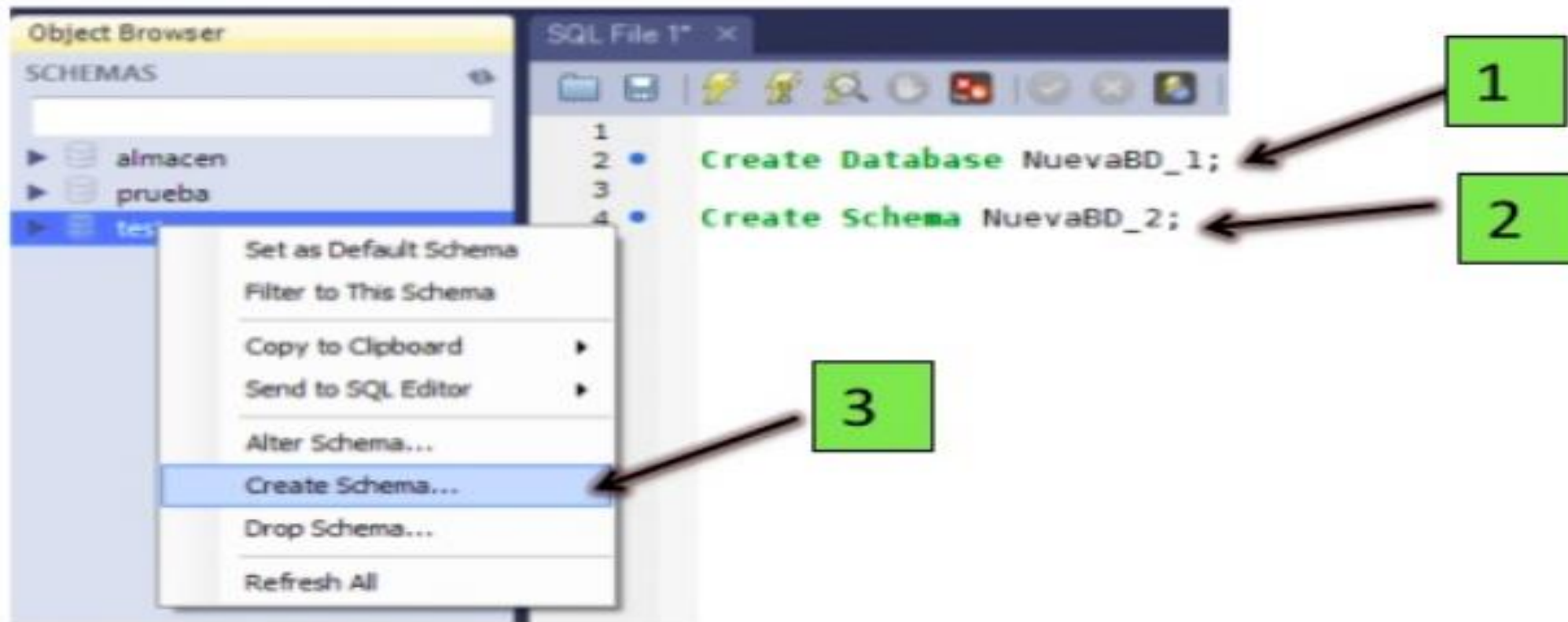
Console Copy to clipboard Export Display chart Create view



CREAR UNA BASE DE DATOS



Dentro de MySQL Workbench hay diferentes formas de crear las bases de datos, tales como:





CREAR UNA BASE DE DATOS



The screenshot shows a database management interface. On the left, the 'Object Browser' displays a list of schemas: 'almacen', 'nuevabd_1', 'nuevabd_2', 'prueba', and 'test'. The schemas 'nuevabd_1' and 'nuevabd_2' are highlighted with an orange box. In the center, the 'SQL File 1*' window shows the following SQL commands:

```
1  
2 • Create Database NuevaBD_1  
3  
4 • Create Schema NuevaBD_2
```

On the right, the 'Output' window shows the execution results:

	Time	Action	Message
✓	1 11:18:39	Create Database NuevaBD_1	1 row(s) affected
✓	2 11:18:39	Create Schema NuevaBD_2	1 row(s) affected

Al ejecutar las Instrucciones vemos que han sido creadas ambas BD

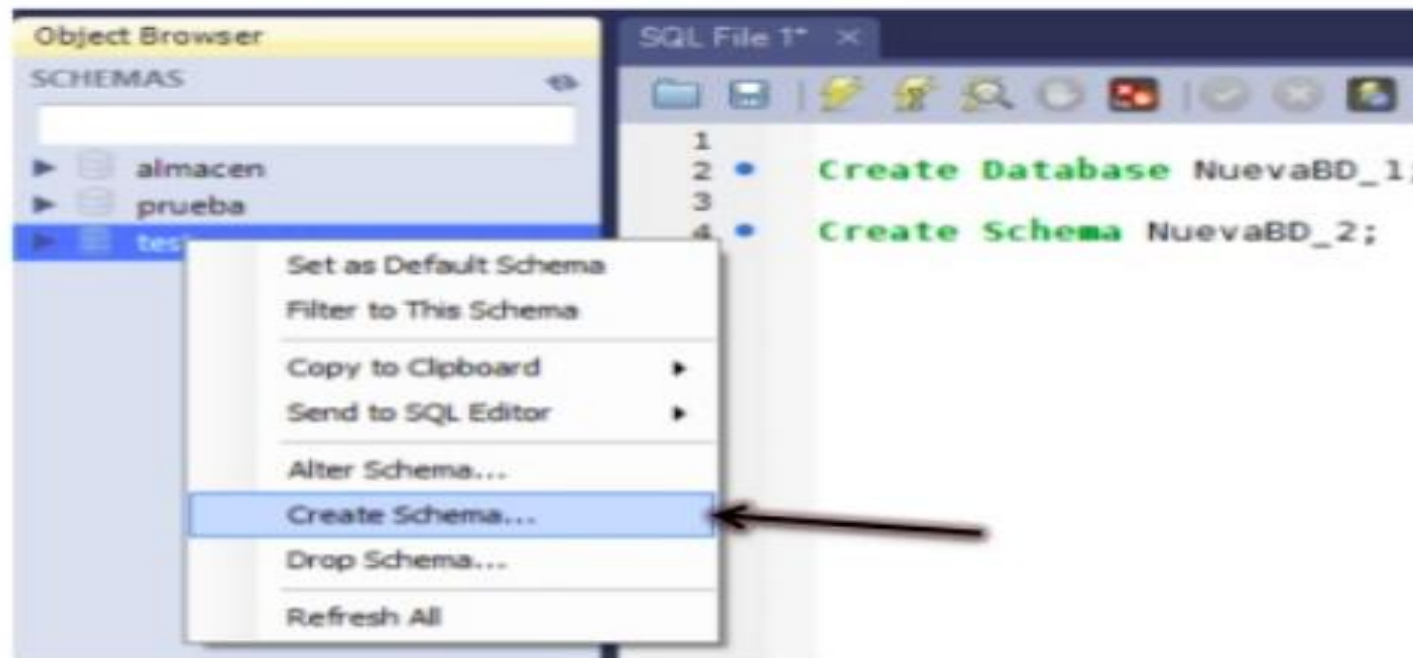


CREAR UNA BASE DE DATOS



Para crear una BD de la Forma

3





CREAR UNA BASE DE DATOS



Damos un Nombre a la BD

SQL File 1* nuevabd_3 x

 Name: nuevabd

Collation: Server Default

The name of the schema. It is recommended to use only alpha-numeric characters. Spaces should be avoided and be replaced by _.

Specifies which charset/collations the schema's tables will use if they do not have an explicit setting. Common choices are Latin1 or UTF8.

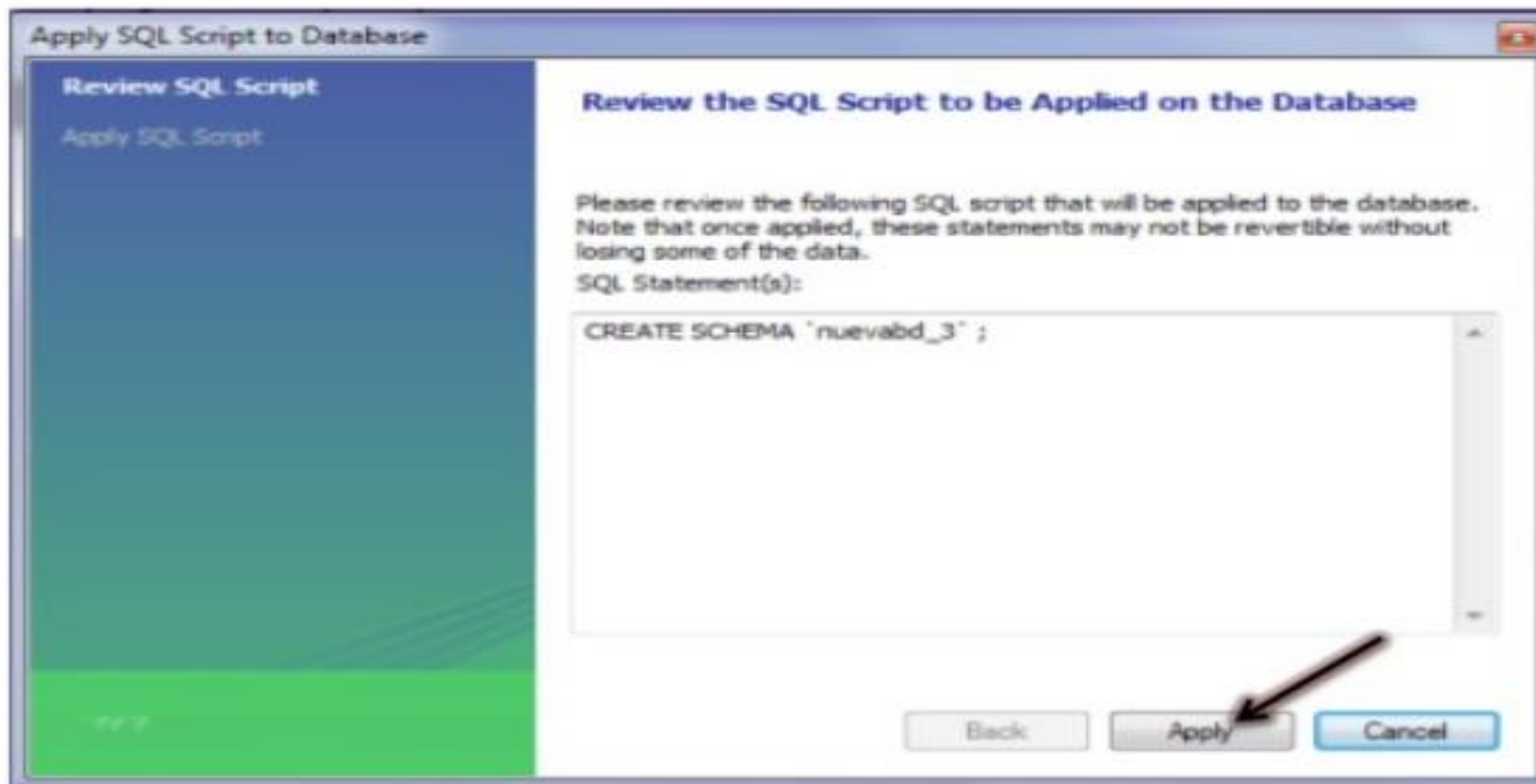
Damos en Aplicar

Schema

Apply Revert

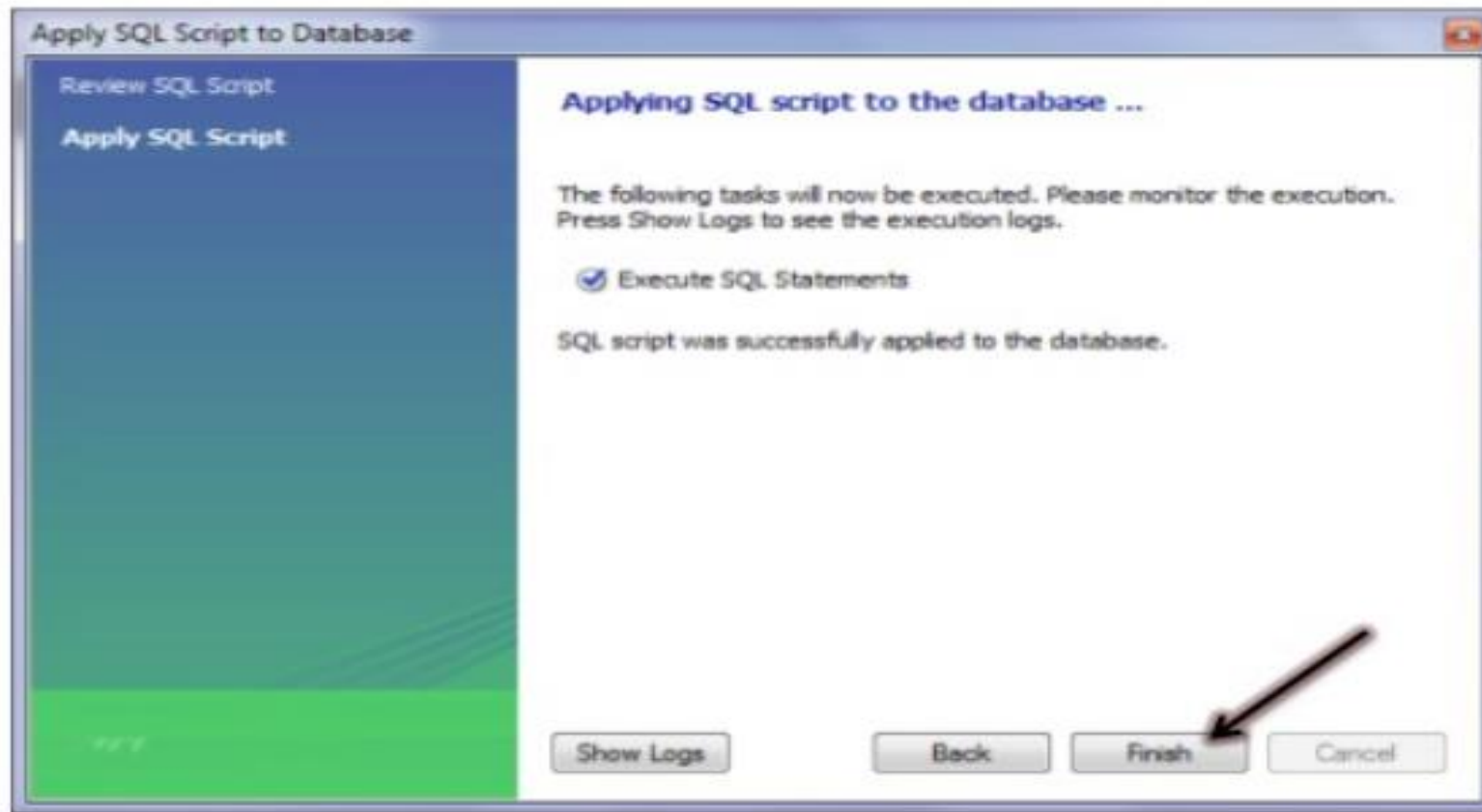


CREAR UNA BASE DE DATOS



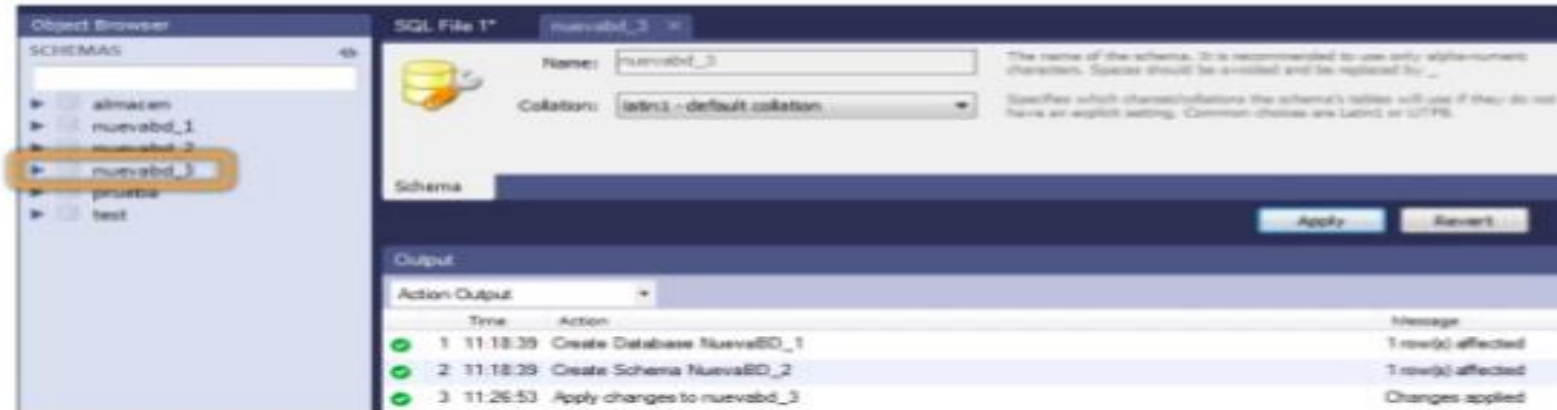


CREAR UNA BASE DE DATOS

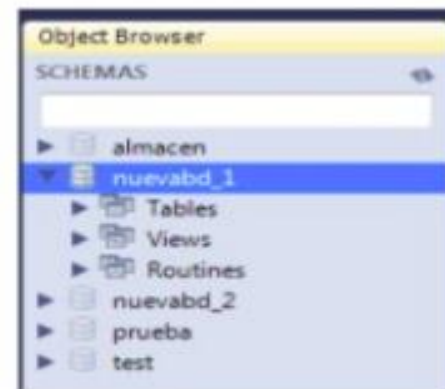




CREAR UNA BASE DE DATOS

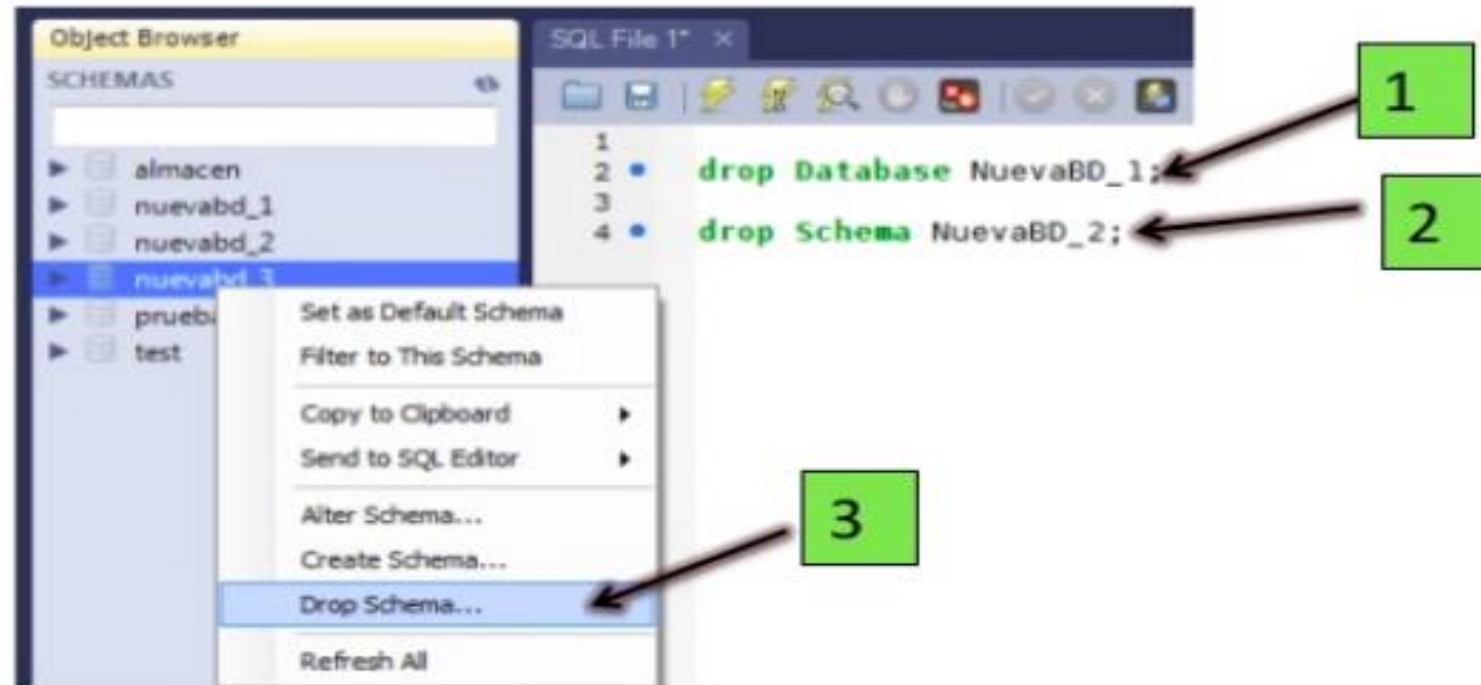


Automáticamente crea la estructura interna de la base de datos en forma de árbol, podremos ver las tablas, vistas y rutinas creadas (triggers, procedimientos, etc.).





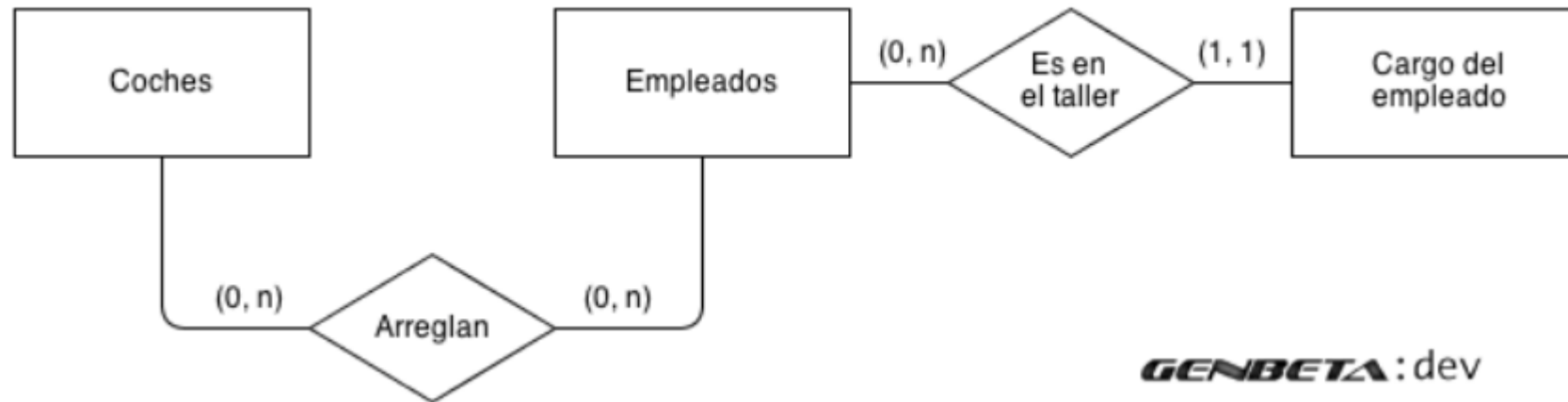
ELIMINAR UNA BASE DE DATOS



https://www.w3schools.com/sql/sql_intro.asp



Modelo entidad-relación



GENBETA:dev

Las **bases de datos** son un gran pilar de la programación actual, ya que nos permiten almacenar y usar de forma rápida y eficiente cantidades ingentes de datos con cierta facilidad. En la actualidad se usa de forma mayoritaria las **bases de datos relacionales** (dominadas por distintos gestores a través del lenguaje **SQL**, en gran medida).



¿Qué es el modelo entidad-relación?

Como ya he comentado este modelo es solo y exclusivamente un método del que disponemos para diseñar estos esquemas que posteriormente debemos de implementar en un gestor de *BBDD* (bases de datos). Este modelo se representa a través de diagramas y está formado por varios elementos.

Este modelo habitualmente, además de disponer de un diagrama que ayuda a entender los datos y como se relacionan entre ellos, debe de ser completado con un pequeño resumen con la lista de los atributos y las relaciones de cada elemento.



Elementos del modelo entidad-relación



Entidad

Las entidades representan *cosas* u *objetos* (ya sean reales o abstractos), que se diferencian claramente entre sí.

Para poder seguir un ejemplo durante el artículo añadiré ejemplos sobre un taller mecánico, donde se podría crear las siguientes entidades:

- **Coches** (objeto físico): contiene la información de cada taller.
- **Empleado** (*objeto* físico): información de los trabajadores.
- **Cargo del empleado** (*cosa* abstracta): información de la función del empleado.

Estas entidades se representan en un diagrama con un rectángulos, como los siguientes.





Atributos

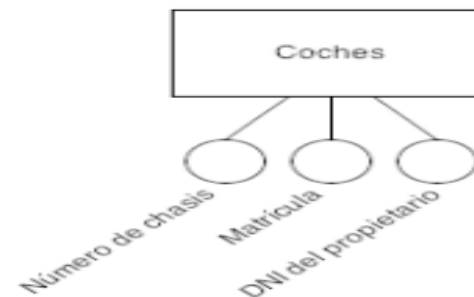


Los atributos definen o identifican las características de entidad (**es el contenido de esta entidad**). Cada entidad contiene distintos atributos, que dan información sobre esta entidad. Estos atributos pueden ser de distintos tipos (numéricos, texto, fecha...).

Siguiendo el ejemplo de antes podemos analizar los atributos de nuestra entidad "**Coches**", que nos darán información sobre los coches de nuestro supuesto taller.

Unos posibles atributos serían los siguientes: *número de chasis*, *matrícula*, *DNI del propietario*, *marca*, *modelo* y muchos otros que complementen la información de cada coche.

Los atributos se representan como círculos que descienden de una entidad, y no es necesario representarlos todos, sino los más significativos, como a continuación.





En un modelo relacional (ya implementado en una base de datos) una ejemplo de tabla dentro de una *BBDD* podría ser el siguiente.

Número de chasis	Matrícula	DNI del propietario
5tfem5f10ax007210	4817 BFK	45338600L
6hsen2j98as001982	8810 CLM	02405068K
5rgsb7a19js001982	0019 GGL	40588860J

Este ejemplo es con tres atributos, pero un coche podría tener cientos (si fuese necesario) y seguirían la misma estructura de columnas, tras implementarlo en una *BBDD*.



Relación



Es un vínculo que nos permite definir una dependencia entre varias entidades, es decir, nos permite exigir que varias entidades compartan ciertos atributos de forma indispensable.

Por ejemplo, los empleados del taller (de la entidad "**Empleados**") tienen un cargo (según la entidad "**Cargo del empleado**"). Es decir, un atributo de la entidad "*Empleados*" especificará que cargo tiene en el taller, y tiene que ser idéntico al que ya existe en la entidad "*Cargo del empleado*".

Las relaciones se muestran en los diagramas como rombos, que se unen a las entidades mediante líneas.



Yo, bajo mi punto de vista, entiendo mejor esto en una tabla (de una implementación en una *BBDD*), por lo que voy a poner el ejemplo de como se representaría (resaltada la relación, que posteriormente veremos como se haría).



Relación



Empleados

Nombre	DNI	Cargo
Carlos Sánchez	45338600L	001
Pepe Sánchez	02405068K	002
Juan Sánchez	40588860J	002

Cargo del empleado

ID del cargo	Descripción
001	Jefe de taller
002	Mecánico



Relaciones de cardinalidad

Podemos encontrar distintos tipos de relaciones según como participen en ellas las entidades. Es decir, en el caso anterior cada empleado puede tener un cargo, pero un mismo cargo lo pueden compartir varios empleados.

Esto complementa a las representaciones de las relaciones, mediante un intervalo en cada extremo de la relación que especifica cuantos *objetos* o *cosas* (de cada entidad) pueden intervenir en esa relación.

Uno a uno: Una entidad se relaciona únicamente con otra y viceversa. Por ejemplo, si tuviésemos una entidad con distintos chasis y otra con matrículas deberíamos de determinar que cada chasis solo puede tener una matrícula (y cada matrícula un chasis, ni más en ningún caso).





Relaciones de cardinalidad

Uno a varios o varios a uno: determina que un registro de una entidad puede estar relacionado con varios de otra entidad, pero en esta entidad existir solo una vez. Como ha sido en el caso anterior del trabajador del taller.



Varios a varios: determina que una entidad puede relacionarse con otra con ninguno o varios registros y viceversa. Por ejemplo, en el taller un coche puede ser reparado por varios mecánicos distintos y esos mecánicos pueden reparar varios coches distintos.



Los indicadores numéricos indican el primero el número mínimo de registros en una relación y posteriormente el máximo (si no hay límite se representa con una " n ").



Claves



Es el atributo de una entidad, al que le aplicamos una restricción que lo distingue de los demás registros (no permitiendo que el atributo específico se repita en la entidad) o le aplica un vínculo (exactamente como comentábamos en las relaciones). Estos son los distintos tipos:

Superclave: aplica una clave o restricción a varios atributos de la entidad, para así asegurarse que en su conjunto no se repitan varias veces y así no poder entrar en dudas al querer identificar un registro.

Clave primaria: identifica inequívocamente un solo atributo no permitiendo que se repita en la misma entidad. Como sería la matrícula o el número de chasis de un coche (no puede existir dos veces el mismo).

Clave externa o clave foránea: este campo tiene que estar estrictamente relacionado con la clave primaria de otra entidad, para así exigir que exista previamente ese clave. Anteriormente hemos hablado de ello cuando comentábamos que un empleado indispensablemente tiene que tener un cargo (que lo hemos representado numéricamente), por lo cual si intentásemos darle un cargo inexistente el gestor de bases de datos nos devolvería un error.



Claves



Es el atributo de una entidad, al que le aplicamos una restricción que lo distingue de los demás registros (no permitiendo que el atributo específico se repita en la entidad) o le aplica un vínculo (exactamente como comentábamos en las relaciones). Estos son los distintos tipos:

Superclave: aplica una clave o restricción a varios atributos de la entidad, para así asegurarse que en su conjunto no se repitan varias veces y así no poder entrar en dudas al querer identificar un registro.

Clave primaria: identifica inequívocamente un solo atributo no permitiendo que se repita en la misma entidad. Como sería la matrícula o el número de chasis de un coche (no puede existir dos veces el mismo).

Clave externa o clave foránea: este campo tiene que estar estrictamente relacionado con la clave primaria de otra entidad, para así exigir que exista previamente ese clave. Anteriormente hemos hablado de ello cuando comentábamos que un empleado indispensablemente tiene que tener un cargo (que lo hemos representado numéricamente), por lo cual si intentásemos darle un cargo inexistente el gestor de bases de datos nos devolvería un error.



Tipos de Datos de Mysql



Los tipos de datos que puede haber en un campo, se pueden agrupar en tres grandes grupos:

1. Tipos numéricos
2. Tipos de Fecha
3. Tipos de Cadena



Tipos de Datos de Mysql



1 Tipos numéricos:

Tipo de Campo	Tamaño de Almacenamiento
TINYINT	1 byte
SMALLINT	2 bytes
MEDIUMINT	3 bytes
INT	4 bytes
INTEGER	4 bytes
BIGINT	8 bytes
FLOAT(X)	4 ú 8 bytes
FLOAT	4 bytes
DOUBLE	8 bytes
DOUBLE PRECISION	8 bytes
REAL	8 bytes
DECIMAL(M,D)	M+2 bytes sí D > 0, M+1 bytes sí D = 0
NUMERIC(M,D)	M+2 bytes if D > 0, M+1 bytes if D = 0



Tipos de Datos de Mysql



BigInt:

Número entero con o sin signo. Con signo el rango de valores va desde -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807. Sin signo el rango va desde 0 a 18.446.744.073.709.551.615.

Float:

Número pequeño en coma flotante de precisión simple. Los valores válidos van desde -3.402823466E+38 a -1.175494351E-38, 0 y desde 1.175494351E-38 a 3.402823466E+38.

xReal, Double:

Número en coma flotante de precisión doble. Los valores permitidos van desde -1.7976931348623157E+308 a -2.2250738585072014E-308, 0 y desde 2.2250738585072014E-308 a 1.7976931348623157E+308

Decimal, Dec, Numeric:

Número en coma flotante desempaquetado. El número se almacena como una cadena



Tipos de Datos de Mysql



TinyInt:

Es un número entero con o sin signo. Con signo el rango de valores válidos va desde -128 a 127. Sin signo, el rango de valores es de 0 a 255

Bit ó Bool:

Un número entero que puede ser 0 ó 1

SmallInt:

Número entero con o sin signo. Con signo el rango de valores va desde -32768 a 32767. Sin signo, el rango de valores es de 0 a 65535.

MediumInt:

Número entero con o sin signo. Con signo el rango de valores va desde -8.388.608 a 8.388.607. Sin signo el rango va desde 0 a 16777215.

Integer, Int:

Número entero con o sin signo. Con signo el rango de valores va desde -2147483648 a 2147483647. Sin signo el rango va desde 0 a 429.4967.295



Tipos de Datos de Mysql



2 Tipos fecha:

Tipo de Campo	Tamaño de Almacenamiento
DATE	3 bytes
DATETIME	8 bytes
TIMESTAMP	4 bytes
TIME	3 bytes
YEAR	1 byte



Tipos de Datos de Mysql



Date:

Tipo fecha, almacena una fecha. El rango de valores va desde el 1 de enero del 1001 al 31 de diciembre de 9999. El formato de almacenamiento es de año-mes-día

DateTime:

Combinación de fecha y hora. El rango de valores va desde el 1 de enero del 1001 a las 0 horas, 0 minutos y 0 segundos al 31 de diciembre del 9999 a las 23 horas, 59 minutos y 59 segundos. El formato de almacenamiento es de año-mes-día horas:minutos:segundos

TimeStamp:

Combinación de fecha y hora. El rango va desde el 1 de enero de 1970 al año 2037. El formato de almacenamiento depende del tamaño del campo:



Tipos de Datos de Mysql



Time:

Almacena una hora. El rango de horas va desde -838 horas, 59 minutos y 59 segundos a 838, 59 minutos y 59 segundos. El formato de almacenamiento es de 'HH:MM:SS'

Year:

Almacena un año. El rango de valores permitidos va desde el año 1901 al año 2155. El campo puede tener tamaño dos o tamaño 4 dependiendo de si queremos almacenar el año con dos o cuatro dígitos.



Tipos de Datos de Mysql



3 Tipos de cadena:

Char(n):

Almacena una cadena de longitud fija. La cadena podrá contener desde 0 a 255 caracteres.

VarChar(n):

Almacena una cadena de longitud variable. La cadena podrá contener desde 0 a 255 caracteres.

Dentro de los tipos de cadena se pueden distinguir otros dos subtipos, los tipo Text y los tipo BLOB (Binary large Object)

La diferencia entre un tipo y otro es el tratamiento que reciben a la hora de realizar ordenamientos y comparaciones. Mientras que el tipo text se ordena sin tener en cuenta las Mayúsculas y las minúsculas, el tipo BLOB se ordena teniéndolas en cuenta.

Los tipos BLOB se utilizan para almacenar datos binarios como pueden ser ficheros.

TinyText y TinyBlob:

Columna con una longitud máxima de 255 caracteres.

Blob y Text:

Un texto con un máximo de 65535 caracteres.



Tipos de Datos de Mysql



MediumBlob y MediumText:

Un texto con un máximo de 16.777.215 caracteres.

LongBlob y LongText:

Un texto con un máximo de caracteres 4.294.967.295. Hay que tener en cuenta que debido a los protocolos de comunicación los paquetes pueden tener un máximo de 16 Mb.

Enum:

Campo que puede tener un único valor de una lista que se especifica. El tipo Enum acepta hasta 65535 valores distintos

Set:

Un campo que puede contener ninguno, uno ó varios valores de una lista. La lista puede tener un máximo de 64 valores.



Tipos de Datos de Mysql



Tipo de campo	Tamaño de Almacenamiento
CHAR(n)	n bytes
VARCHAR(n)	n +1 bytes
TINYBLOB, TINYTEXT	Longitud+1 bytes
BLOB, TEXT	Longitud +2 bytes
MEDIUMBLOB, MEDIUMTEXT	Longitud +3 bytes
LONGBLOB, LONGTEXT	Longitud +4 bytes
ENUM('value1','value2',...)	1 ó dos bytes dependiendo del número de valores
SET('value1','value2',...)	1, 2, 3, 4 ó 8 bytes, dependiendo del número de valores

Diferencia de almacenamiento entre los tipos Char y VarChar

Valor	CHAR(4)	Almacenamiento	VARCHAR(4)	Almacenamiento
"	"	4 bytes	"	1 byte
'ab'	'ab '	4 bytes	'ab'	3 bytes
'abcd'	'abcd'	4 bytes	'abcd'	
'abcdefgh'	'abcd'	4 bytes	'abcd'	5 bytes