

5 Lenguaje Definición y Manipulación de Datos

Links de interés:

CREATE TABLE

CREATE [OR REPLACE] [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name (create_definition,...) [table_options]... [partition_options] CREATE [OR REPLACE] [TEMPORARY]

🔗 <https://mariadb.com/kb/en/create-table/>

MariaDB Create Table

Summary: in this tutorial, you will learn how to use the MariaDB create table statement to create a new table in a database. The create table statement allows you to create a new table in a

🔗 <https://mariadbtutorial.com/mariadb-basics/mariadb-create-table/>

CREATE DATABASE

CREATE [OR REPLACE] {DATABASE | SCHEMA} [IF NOT EXISTS] db_name [create_specification] ... create_specification: [DEFAULT] CHARACTER SET [=] charset_name | [DEFAULT]

🔗 <https://mariadb.com/kb/en/create-database/>

SQL CREATE DATABASE Statement

The CREATE DATABASE statement is used to create a new SQL database. CREATE DATABASE databasename; The

🔗 https://www.w3schools.com/sql/sql_create_db.asp



CREATE INDEX

CREATE [OR REPLACE] [UNIQUE|FULLTEXT|SPATIAL] INDEX [IF NOT EXISTS] index_name [index_type] ON tbl_name (index_col_name,...) [WAIT n | NOWAIT] [index_option]

🔗 <https://mariadb.com/kb/en/create-index/>

MariaDB Create Index Statement By Practical Examples

Summary: in this tutorial, you will learn how to use the MariaDB create index statement to create a new index to improve the speed of queries. A database index is similar to a book index

🔗 <https://mariadbtutorial.com/mariadb-index/mariadb-create-index/>

Table Pullout Optimization

Table pullout is an optimization for Semi-join subqueries. Sometimes, a subquery can be re-written as a join.

🔗 <https://mariadb.com/kb/en/table-pullout-optimization/>

EXPLAIN

EXPLAIN tbl_name [col_name | wild] Or EXPLAIN [EXTENDED | PARTITIONS | FORMAT=JSON] {SELECT select_options | UPDATE update_options | DELETE delete_options} The

🔗 <https://mariadb.com/kb/en/explain/>

INSERT

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY]
[IGNORE] [INTO] tbl_name [PARTITION (partition_list)] [(col,...)]
{VALUES | VALUE} {(expr | DEFAULT),...},{...},... [ ON
```

<https://mariadb.com/kb/en/insert/>

Learn MariaDB Insert Statement By Practical Examples

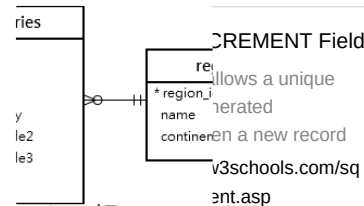
Summary: in this tutorial, you will learn how to use the MariaDB insert statement to insert a row into a table. The insert statement allows you to add a new row to a table. The

<https://www.mariadbtutorial.com/mariadb-basics/mariadb-insert/>

MariaDB Insert Into Select Statement By Practical Examples

Summary: in this tutorial, you will learn how to use the MariaDB insert into select statement to insert result sets of a query into a table. The value list of an insert statement can be either literal

<https://www.mariadbtutorial.com/mariadb-basics/mariadb-insert-into-select/>



*Una sesión de base de datos representa el diálogo de una aplicación con una base de datos relacional. Una sesión activa es una conexión que ha enviado trabajo al motor de base de datos y está esperando una respuesta.

*Una sesión está activa cuando se ejecuta en la CPU o a la espera de que un recurso esté disponible para que pueda continuar. Por ejemplo, una sesión activa puede esperar a que se lea una página en la memoria y, a continuación, consumir CPU mientras lee los datos de la página.

*Las restricciones (o constraints) se pueden especificar cuando se crea la tabla con la instrucción **CREATE TABLE** o después de que se crea la tabla con la instrucción **ALTER TABLE**

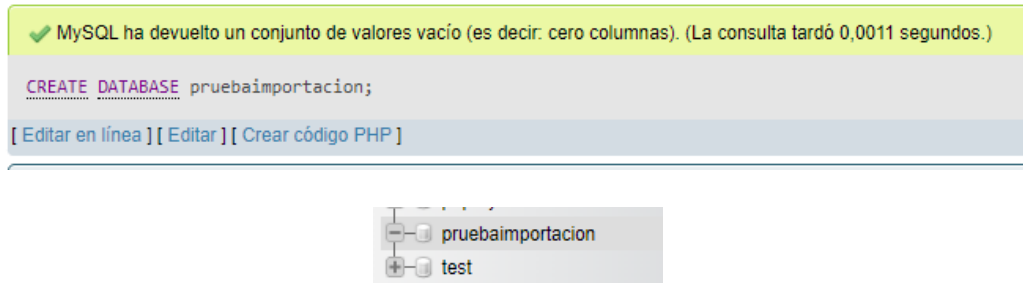
*Se puede crear una tabla o una tabla temporal en una consulta

*Crear una tabla virtual cuyo contenido (columnas y filas) se define mediante una consulta. Utilice esta instrucción para crear una vista de los datos de una o varias tablas de la base de datos. Por ejemplo, una vista se puede utilizar para lo siguiente:

- Para centrar, simplificar y personalizar la percepción de la base de datos para cada usuario.
- Como mecanismo de seguridad, que permite a los usuarios obtener acceso a los datos por medio de la vista, pero no les conceden el permiso de obtener acceso directo a las tablas base subyacentes de la vista.
- Para proporcionar una interfaz compatible con versiones anteriores para emular una tabla cuyo esquema ha cambiado.

*Create view tiene un problema porque es una consulta que se tiene que ejecutar y puede estar siendo alterada en cualquier momento y puede influir en el rendimiento de las consultas sobre las que se hagan sobre esta vista (Cuando tengas mucha experiencia, usa las Create view) Hay alternativas a las vistas que son procedimientos almacenados

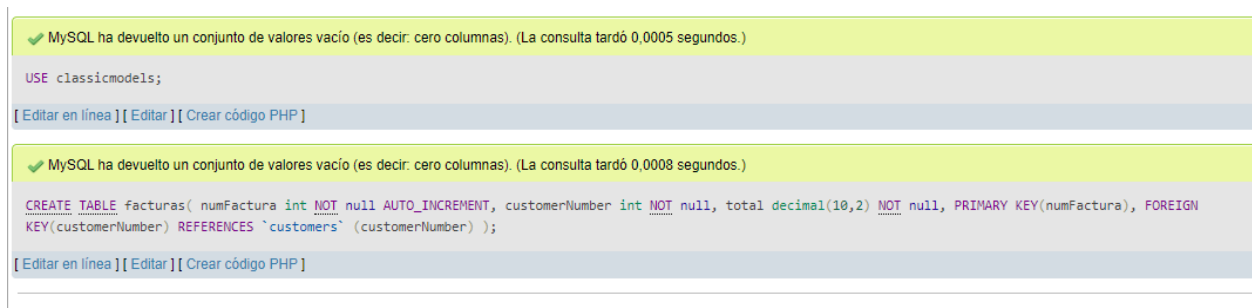
- **CREATE DATABASE** **pruebaimportacion**; (Así creamos una base de datos en donde luego podemos crear las tablas)



- **USE classicmodels;**

```
CREATE TABLE facturas(
    numFactura int NOT null AUTO_INCREMENT,
    customerNumber int NOT null,
    total decimal(10,2) NOT null,
    PRIMARY KEY(numFactura),
    FOREIGN KEY(customerNumber) REFERENCES customers (customerNumber)
);
```

(Creamos una tabla con integridad referencial hacia la tabla de customers. Siempre hay que decir en qué base de datos hay que hacerlo con el USE al principio)



(Así es como se ve ahora nuestra tabla facturas)

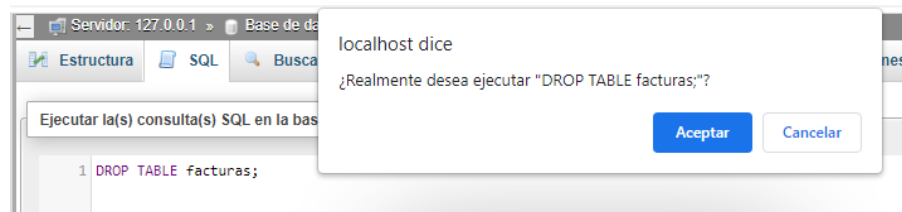
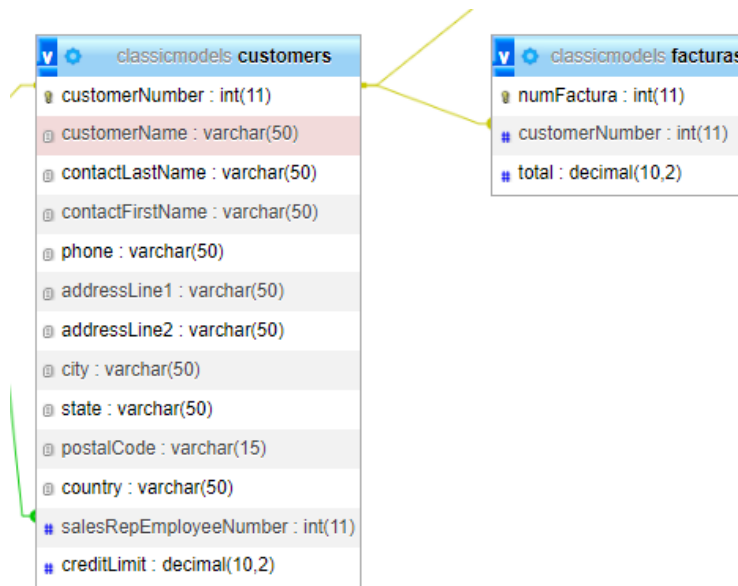
#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 numFactura	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
<input type="checkbox"/>	2 customerNumber	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	3 total	decimal(10,2)			No	Ninguna			Cambiar Eliminar Más

(Y cómo se relaciona con las demás tablas:

Vemos como al haberle puesto:

```
FOREIGN KEY(customerNumber) REFERENCES customers (customerNumber)
```

hemos hecho que se relacione facturas con customers con el customerNumber como clave foránea de facturas)



- **DROP TABLE facturas;** (Borramos la tabla. Pide confirmación en la página web de phpmyadmin, pero eso no tiene que ser siempre así)

- **CREATE OR REPLACE TABLE facturas (**

numFactura int,

customerNumber int NOT null,

total decimal(10,2) NOT null,

PRIMARY KEY(numFactura),

FOREIGN KEY(customerNumber) REFERENCES customers (customerNumber));

(Si usamos la opción create or replace, borraría la tabla preexistente si existiera, y crearía una desde cero con las características que pusiéramos entre paréntesis)

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/> 1	numFactura	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 2	customerNumber	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 3	total	decimal(10,2)			No	Ninguna			Cambiar Eliminar Más

- **CREATE TABLE** personas(
IdPersona **INT(11)** **AUTO_INCREMENT PRIMARY KEY**,
Nombre **VARCHAR(50)** **NOT NULL**,
Apellido **VARCHAR(50)** **NOT NULL**,
Direccion **VARCHAR(50)**,
FechaAlta **DATE** **NOT NULL**,
Email **VARCHAR(100)** **NOT NULL**
);

(Creamos una tabla nueva pero esta vez, no tiene ninguna relación con las demás ya que no le hemos añadido a la consulta "FOREIGN KEY...REFERENCES...")

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1	IdPersona	int(11)		No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
<input type="checkbox"/>	2	Nombre	varchar(50)	latin1_swedish_ci	No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	3	Apellido	varchar(50)	latin1_swedish_ci	No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	4	Direccion	varchar(50)	latin1_swedish_ci	Sí	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	5	FechaAlta	date		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	6	Email	varchar(100)	latin1_swedish_ci	No	Ninguna			Cambiar Eliminar Más

classicmodels personas
IdPersona : int(11)
Nombre : varchar(50)
Apellido : varchar(50)
Direccion : varchar(50)
FechaAlta : date
Email : varchar(100)

- **CREATE TABLE** pedidos(
IdPedidos **INT(11)** **AUTO_INCREMENT PRIMARY KEY**,
NombreArticulo **VARCHAR(50)** **NOT NULL**,
Coleccion **VARCHAR(50)** **NOT NULL**,
idPersona **INT(11)** **NOT NULL**,
idOficina **VARCHAR(10)** **NOT NULL**,
FOREIGN KEY(idPersona) **REFERENCES** **personas** (idPersona),
FOREIGN KEY(idOficina) **REFERENCES** **offices** (officeCode)
);

(Creamos la tabla pedidos conectada a la tabla personas y offices. Podemos ver que tenemos que crear los atributos que luego serán las FOREIGN KEYs aunque no se tienen que llamar igual a las primary keys de las tablas como podemos ver en: **FOREIGN KEY(idOficina) REFERENCES offices (officeCode)**)

	#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1	IdPedidos	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
<input type="checkbox"/>	2	NombreArticulo	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	3	Coleccion	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	4	idPersona	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	5	idOficina	varchar(10)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más

- **USE classicmodels;**

ALTER TABLE facturas ADD COLUMN tipoIVA decimal(4,2); (Cambiaría/Alteraría la estructura de la tabla añadiendo un nuevo campo llamado tipo de IVA)

	#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1	numFactura	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	2	customerNumber	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	3	total	decimal(10,2)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	4	tipoIVA	decimal(4,2)			Sí	NULL			Cambiar Eliminar Más

- **USE classicmodels;**

ALTER TABLE facturas MODIFY COLUMN numFactura int AUTO_INCREMENT; (Cambiaría/Alteraría la estructura de la tabla modificando la columna numFactura haciéndola autoincremental. Hay que poner siempre al modificar una tabla, el tipo de valor que es el atributo, en este caso si no ponemos "int" en "numFactura", la consulta no funciona)

	#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1	numFactura	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
<input type="checkbox"/>	2	customerNumber	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	3	total	decimal(10,2)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	4	tipoIVA	decimal(4,2)			Sí	NULL			Cambiar Eliminar Más

- **USE classicmodels;**

ALTER TABLE facturas ADD COLUMN descripcion Varchar(255);

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 numFactura	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
<input type="checkbox"/>	2 customerNumber	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	3 total	decimal(10,2)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	4 tipoIVA	decimal(4,2)			Sí	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	5 descripcion	varchar(255)	latin1_swedish_ci		Sí	NULL			Cambiar Eliminar Más

- **INSERT INTO facturas (descripcion, customerNumber, total, tipoIva) VALUES ("naranjas", 103, 12.67, 4.0), ("mandarinas", 103, 133.9, 12.0)** (Así añadimos nuevos registros a nuestra tabla, en este caso vemos que se añaden dos entre comas. Podemos ver que no escribimos ninguna referencia a numFactura que es la primary key porque al ser autoincremental, ya se va numerando automáticamente por orden los registros)

(Esto es lo que te devuelve la consulta:)

2 filas insertadas.
 La Id de la fila insertada es: 4 (La consulta tardó 0,0127 segundos.)

```
INSERT INTO facturas (descripcion, customerNumber, total, tipoIva) VALUES ("naranjas", 103, 12.67, 4.0), ("mandarinas", 103, 133.9, 12.0);
```

[\[Editar en línea \]](#)
[\[Editar \]](#)
[\[Crear código PHP \]](#)

(Y esto es lo que se ve en Facturas:)

Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0023 segundos.)

```
SELECT * FROM `facturas`
```

☐ Perfilando
 [\[Editar en línea \]](#)
[\[Editar \]](#)
[\[Explicar SQL \]](#)
[\[Crear código PHP \]](#)
[\[Actualizar \]](#)

☐ Mostrar todo
 Número de filas:
 Filtrar filas:
 Ordenar según la clave:

Opciones extra

	numFactura	customerNumber	total	descripcion	tipolva
<input type="checkbox"/> Editar Copiar Borrar	3	103	12.67	naranjas	4
<input type="checkbox"/> Editar Copiar Borrar	4	103	133.90	mandarinas	12

- **INSERT INTO personas (Nombre, Apellido, Direccion, FechaAlta, Email)**
SELECT * FROM
 (SELECT e.firstName Nombre, e.lastName Apellido, null Direccion, NOW() FechaAlta, e.email Email
 FROM classicmodels.employees e WHERE e.jobTitle NOT LIKE '%President%'
 UNION
 SELECT c.contactFirstName Nombre, c.contactLastName Apellido, c.addressLine1 Dirección,
 NOW() FechaAlta, CONCAT(c.customerNumber, '@classicmodelcars.com') Email

FROM classicmodels.customers c)

personasImportadas; (Añadimos nuevos registros a la tabla persona, pero esta vez elegimos todos los registros de la tabla clientes y de la tabla empleados que no sean Presidentes. Aquí lo que hacemos es: Usar **INSERT INTO** para seleccionar la tabla a la que queremos añadir los registros y entre paréntesis, los atributos sobre los que añadiremos nueva información [todos menos la clave primaria]. Hacemos un **SELECT** de todo y una subconsulta en el from que tiene el alias "personasImportadas". En la subconsulta primero seleccionamos de empleados, los atributos que coincidan con la tabla personas, por el mismo orden que los de personas [usamos **NULL** en dirección porque los empleados no tienen ningún atributo de dirección, y usamos el operador **NOW()** para la fechaAlta, ya que registra la fecha de alta del día que se ha añadido el registro], luego usamos la cláusula **WHERE** para evitar que se añada al presidente y se hace una **UNION** y se selecciona la tabla clientes, con los atributos ordenados como los de personas [Hacemos un **CONCAT** para el Mail poniendo su clave primaria y un **STRING** para crear un mail, ya que los clientes no tienen mail])(Podemos ver que tanto los clientes como los empleados se han añadido. **Así es como añades registros de una tabla a otra**)

<input type="checkbox"/>	 Editar	 Copiar	 Borrar	19 Tom	King	NULL	2023-02-25	tking@classicmodelcars.com
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	20 Mami	Nishi	NULL	2023-02-25	mnishi@classicmodelcars.com
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	21 Yoshimi	Kato	NULL	2023-02-25	ykato@classicmodelcars.com
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	22 Martin	Gerard	NULL	2023-02-25	mgerard@classicmodelcars.com
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	23 Carine	Schmitt	54, rue Royale	2023-02-25	103@classicmodelcars.com
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	24 Jean	King	8489 Strong St.	2023-02-25	112@classicmodelcars.com
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	25 Peter	Ferguson	636 St Kilda Road	2023-02-25	114@classicmodelcars.com

- **USE** pruebaimportacion;

CREATE TABLE clientes AS SELECT * FROM classicmodels.customers ; (Esta última sentencia crea la tabla clientes en la base de datos pruebaimportacion, a partir de la tabla customers de la base de datos classicmodels. Podemos ver que no solo se exportan los atributos de la tabla customers de classicmodels sino también todos los registros. La consulta puede ser lo compleja que queramos.)

Estructura de tabla

Vista de relaciones

	#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1	customerNumber	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	2	customerName	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	3	contactLastName	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	4	contactFirstName	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	5	phone	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	6	addressLine1	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	7	addressLine2	varchar(50)	latin1_swedish_ci		Sí	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	8	city	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	9	state	varchar(50)	latin1_swedish_ci		Sí	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	10	postalCode	varchar(15)	latin1_swedish_ci		Sí	NULL			Cambiar Eliminar Más
<input type="checkbox"/>	11	country	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	12	salesRepEmployeeNumber	int(11)			Sí	NULL			Cambiar Eliminar Más

Mostrando filas 0 - 24 (total de 122, La consulta tardó 0,0003 segundos.)

SELECT * FROM `clientes`

Perfilando

Editar en línea

Editar

Explicar SQL

Crear código PHP

Actualizar

1 > >>

Mostrar todo

Número de filas: 25

Filtrar filas:

Opciones extra

customerNumber	customerName	contactLastName	contactFirstName	phone	addressLine1	addressLine2	city	state	postalCode	country	salesRepEmployeeNumber
103	Atelier graphique	Schmitt	Carine	40.32.2555	54, rue Royale	NULL	Nantes	NULL	44000	France	1370
112	Signal Gift Stores	King	Jean	7025551838	8489 Strong St.	NULL	Las Vegas	NV	83030	USA	1166
114	Australian Collectors, Co.	Ferguson	Peter	03 9520 4555	636 St Kilda Road	Level 3	Melbourne	Victoria	3004	Australia	1611
119	La Rochelle Gifts	Labruno	Janine	40.67.8555	67, rue des Cinquante Otages	NULL	Nantes	NULL	44000	France	1370
121	Baane Mini Imports	Bergulfsen	Jonas	07-98 9555	Erling Skakkes gate 78	NULL	Stavern	NULL	4110	Norway	1504
124	Mini Gifts Distributors Ltd.	Nelson	Susan	4155551450	5677 Strong St.	NULL	San Rafael	CA	97562	USA	1166

- **UPDATE** personas **SET** Email='jfirrelli2@classicmodelcars.com' **WHERE** IdPersona = 2; (Esto cambia el correo de la persona con IdPersona = 2 a 'jfirrelli2@classicmodelcars.com')

Antes:

Ahora:

				IdPersona	Nombre	Apellido	Direccion	FechaAlta	Email
<input type="checkbox"/>				1	Mary	Patterson	NULL	2023-02-25	mpatterso@classicmodelcars.com
<input type="checkbox"/>				2	Jeff	Firrelli	NULL	2023-02-25	jfirrelli2@classicmodelcars.com

- **UPDATE** personas **p SET** Email=**CONCAT**(**TRIM**(p.Nombre), **TRIM**(p.Apellido), '@classicmodelcars.com') **WHERE** Email **REGEXP** '^([0-9]{3})@classicmodelcars.com'; (Así editamos todos los mails que empiecen con tres números y contengan @classicmodelcars.com, y hacemos que tengan su Nombre+Apellido+@classicmodelcars.com. Usamos **TRIM**() para quitar espacios u otras cosas) (Cuidado con los updates masivos, siempre que se haga un update hay que hacer una restricción (Cuantas más restricciones mejor), revisar todo y hacerlo en una transición o con otras cosas que veremos en otro momento)

Antes:

<input type="checkbox"/>				19	Tom	King	NULL	2023-02-25	tking@classicmodelcars.com
<input type="checkbox"/>				20	Mami	Nishi	NULL	2023-02-25	mnishi@classicmodelcars.com
<input type="checkbox"/>				21	Yoshimi	Kato	NULL	2023-02-25	ykato@classicmodelcars.com
<input type="checkbox"/>				22	Martin	Gerard	NULL	2023-02-25	mgerard@classicmodelcars.com
<input type="checkbox"/>				23	Carine	Schmitt	54, rue Royale	2023-02-25	103@classicmodelcars.com
<input type="checkbox"/>				24	Jean	King	8489 Strong St.	2023-02-25	112@classicmodelcars.com
<input type="checkbox"/>				25	Peter	Ferguson	636 St Kilda Road	2023-02-25	114@classicmodelcars.com

Ahora:

<input type="checkbox"/>				19	Tom	King	NULL	2023-02-25	tking@classicmodelcars.com
<input type="checkbox"/>				20	Mami	Nishi	NULL	2023-02-25	mnishi@classicmodelcars.com
<input type="checkbox"/>				21	Yoshimi	Kato	NULL	2023-02-25	ykato@classicmodelcars.com
<input type="checkbox"/>				22	Martin	Gerard	NULL	2023-02-25	mgerard@classicmodelcars.com
<input type="checkbox"/>				23	Carine	Schmitt	54, rue Royale	2023-02-25	CarineSchmitt@classicmodelcars.com
<input type="checkbox"/>				24	Jean	King	8489 Strong St.	2023-02-25	JeanKing@classicmodelcars.com
<input type="checkbox"/>				25	Peter	Ferguson	636 St Kilda Road	2023-02-25	PeterFerguson@classicmodelcars.com

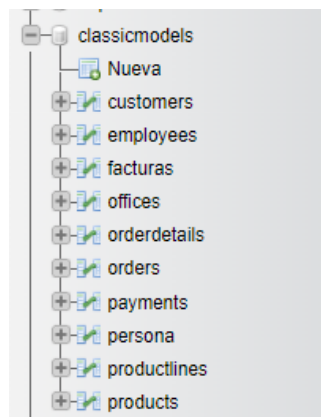
- **DELETE FROM** personas **WHERE** IdPersona=1; (Esto elimina el primer registro de Mary Patterson)

				IdPersona	Nombre	Apellido	Direccion	FechaAlta	Email
<input type="checkbox"/>				2	Jeff	Firrelli	NULL	2023-02-25	jfirrelli2@classicmodelcars.com
<input type="checkbox"/>				3	William	Patterson	NULL	2023-02-25	wpatterson@classicmodelcars.com

- **DELETE FROM personas WHERE IdPersona IN(SELECT IdPersona FROM personas p JOIN classicmodels.employees e ON (e.firstName=p.Nombre AND e.lastName=p.Apellido) WHERE e.officeCode=1);** (Esto es para borrar todos los registros de la tabla personas que tenga el officeCode=1. Como en DELETE siempre el WHERE tiene que ser con la primary key de la tabla, tenemos que poner IN y luego la condición en una subconsulta)

		IdPersona	Nombre	Apellido	Direccion	FechaAlta	Email
<input type="checkbox"/>	Editar Copiar Borrar	3	William	Patterson	NULL	2023-02-25	wpatterson@classicmodelcars.com
<input type="checkbox"/>	Editar Copiar Borrar	4	Gerard	Bondur	NULL	2023-02-25	gbondur@classicmodelcars.com
<input type="checkbox"/>	Editar Copiar Borrar	8	Julie	Firrelli	NULL	2023-02-25	jfirrelli@classicmodelcars.com
<input type="checkbox"/>	Editar Copiar Borrar	9	Steve	Patterson	NULL	2023-02-25	spatterson@classicmodelcars.com
<input type="checkbox"/>	Editar Copiar Borrar	10	Foon Yue	Tseng	NULL	2023-02-25	ftseng@classicmodelcars.com
<input type="checkbox"/>	Editar Copiar Borrar	11	George	Vanauf	NULL	2023-02-25	gvanauf@classicmodelcars.com

- **RENAME TABLE personas TO persona** (Esto cambia el nombre de la tabla personas para que se llame persona)



- **EXPLAIN SELECT * FROM customers c WHERE c.customerName='Atelier graphique'** (Para buscar un solo cliente tiene que pasar por una estimación de 122 filas (Es una búsqueda secuencial). La instrucción "EXPLAIN" se utiliza para obtener información sobre cómo se está ejecutando una consulta SQL en la base de datos. Cuando se utiliza la instrucción "EXPLAIN" antes de una consulta SQL, la base de datos proporciona información sobre cómo planea ejecutar la consulta y cómo se accede a las tablas y los índices involucrados en la consulta. La información que se devuelve de la consulta "EXPLAIN" puede ayudar a los desarrolladores y administradores de bases de datos a identificar problemas de rendimiento y optimizar las consultas para que se ejecuten más rápidamente. **En el caso de la consulta SQL "SELECT**

* **FROM customers c WHERE c.customerName='Atelier graphique'**", la instrucción "EXPLAIN" se utiliza para obtener información sobre cómo se planea ejecutar la consulta y qué índices se están utilizando para acceder a los datos en la tabla "customers". El resultado de la consulta "EXPLAIN" puede proporcionar información valiosa sobre cómo mejorar el rendimiento de la consulta y optimizar la base de datos en general.)

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	c	ALL	NULL	NULL	NULL	NULL	122	Using where

Si la base de datos es muy grande, poner un índice es importante, pero poner un índice tienes que pensar que es algo que luego puede impactar bastante en el rendimiento. Decidir crear un índice es un asunto complicado y hay que medirse mucho.

Ahora lanzamos el índice

- **CREATE INDEX customerCustomerName ON customers(customerName);**

Y ahora una consulta

- **EXPLAIN SELECT * FROM customers WHERE customerName='Atelier graphique';**

Vemos que ahora solo consulta una fila gracias al índice

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	customers	ref	customerCustomerName	customerCustomerName	52	const	1	Using index condition

- **CREATE UNIQUE INDEX indiceEmail ON personas(Email);** (Ahora creamos un índice único en el que cada mail solo puede haber estado registrado una vez y no se puede volver a repetir)

VISTAS:

CREATE VIEW nombredelavista AS SELECT

por ejemplo:

USE classicmodels;

CREATE VIEW pagadoPorCliente AS SELECT c.customerNumber, SUM(p.amount) totalPagado FROM customers c JOIN payments p USING(customerNumber) GROUP BY c.customerNumber;

crearia una vista con lo pagado por cada cliente. Sobre esta vista podría hacer un join con la tabla de customers para mostrar el nombre de los clientes ademas por ejemplo.

```
INSERT INTO employees (employeeNumber,lastName,firstName,extension,email,officeCode,reportsTo)
select
customerNumber+10000,contactLastName,contactFirstName,'X8888','customer@gmail.com',666,null
from customers

update employees set jobTitle='customer' where officeCode=666

delete from employees where officeCode=666
```