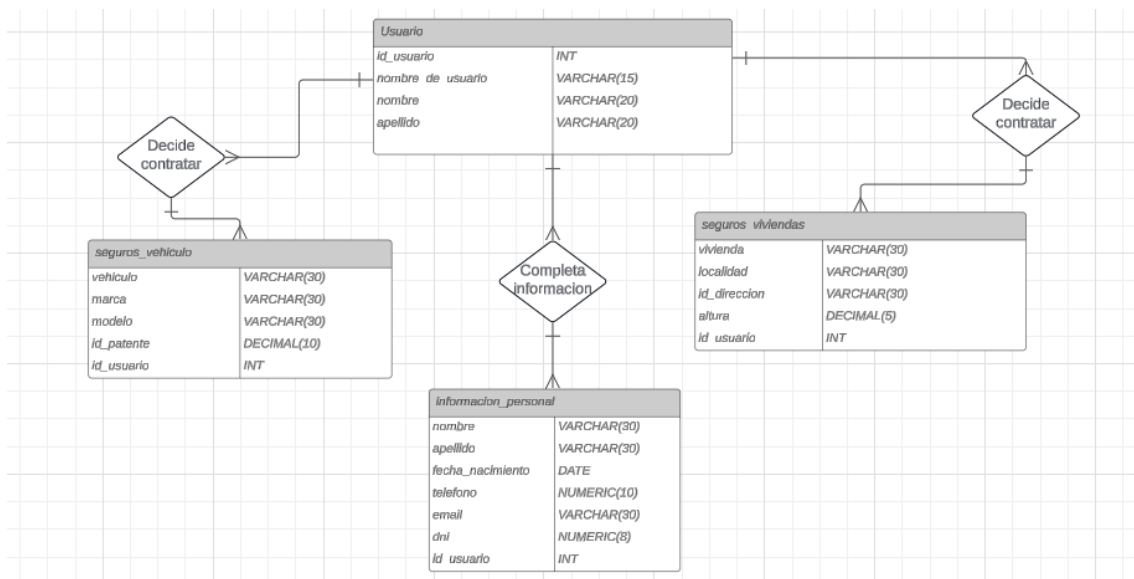


Primera PreEntrega – Agustin Alvarez

Descripción de la temática de la base de datos:

Mi idea es hacer una base de datos relacionada a una agencia de seguros que opera desde internet, en la cual para poder ser cliente primero deberían registrarse en el sitio web, rellenar alguna información personal adicional y luego poder elegir entre un seguro de vivienda o vehículo. En si esa es mi idea introductoria, con el pasar del tiempo le agregare más cosas.

Diagrama de entidad relación (DER):



Listado de tablas que componen la base de datos:

TABLA	CAMPOS	DETALLE DEL CAMPO	PK	FK	TIPO DE DATO
-------	--------	-------------------	----	----	--------------

Usuario	Id_usuario	Identificador de cada usuario	PK		INT
	nombre_de_usuario	Nombre del usuario			VARCHAR(15)
	nombre	Nombre real del usuario			VARCHAR(20)
	apellido	Apellido del usuario			VARCHAR(20)

Seguros Vehículos	vehículo	Tipo de vehículo			VARCHAR(30)
	marca	Marca del vehículo			VARCHAR(20)
	modelo	Modelo del vehículo			VARCHAR(20)
	Id_patente	Numero de patente	PK		FLOAT(10)
	Id_usuario	Identificador del usuario		FK	INT

Seguros Viviendas	pais	Pais en el que se encuentra la propiedad.			VARCHAR(30)
	localidad	Localidad en la que se encuentra la vivienda			VARCHAR(50)
	direccion	Dirección exacta del lugar	PK		VARCHAR(50)
	altura	Altura de la calle			NUMERIC(10)
	Id_usuario	Identificador del usuario		FK	INT

Informacion Personal	Nombre	Nombre y segundo nombre (sies que tiene) del usuario			VARCHAR(40)
	apellido	Apellido del usuario			VARCHAR(40)
	teléfono	Teléfono del usuario			FLOAT(20)
	email	Email del usuario			VARCHAR(30)
	DNI	DNI del usuario	PK		NUMERIC(20)
	Id_usuario	Identificador del usuario	PK	FK	INT

Servicios	Id_producto	Identificador del servicio a contratar	PK		INT
	Tipo	Tipo de servicio que se contrato(autos, motos, viviendas)			VARCHAR(150)
	Precio_servicios	Precio de los servicios			INT
	Id_usuario	Identificador del usuario		FK	INT

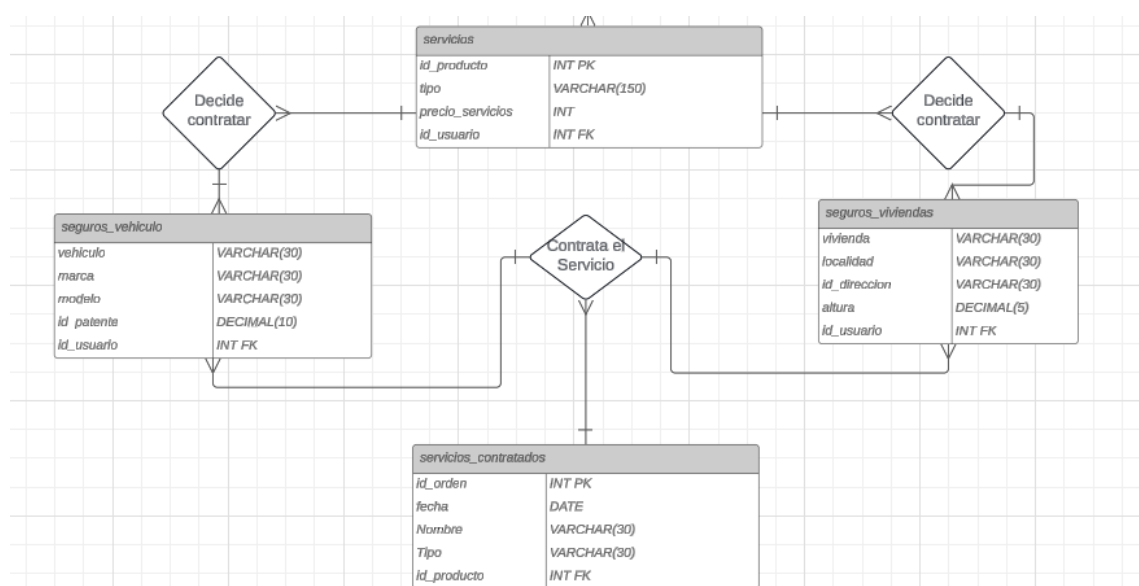
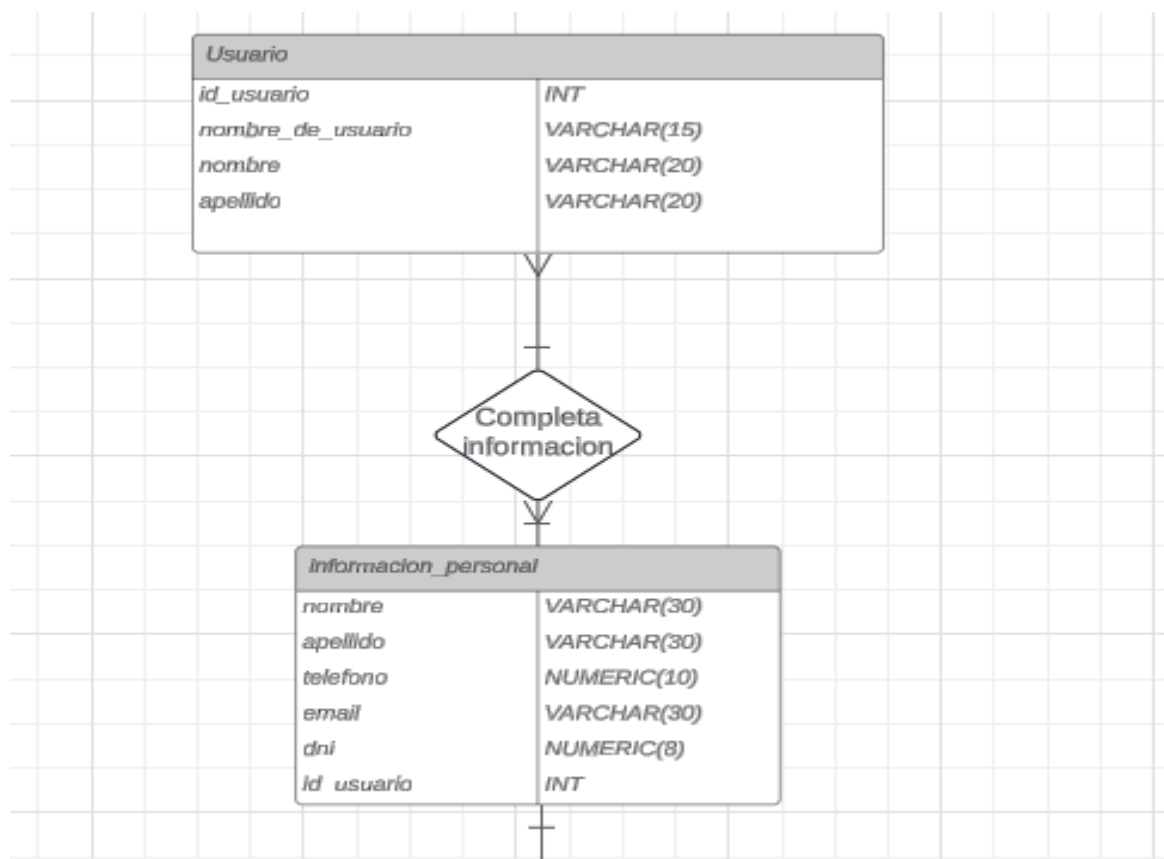
Servicios Contratados	Id_orden	Identificador de cada pedido entrante	PK		INT
	Fecha	Fecha en la que se solicito			DATE
	Nombre	Nombre de quien contrato el servicio			VARCHAR(30)
	Tipo	Tipo de servicio contratado(auto, motos, vivienda)			VARCHAR(30)
	Id_producto	Identificador del producto contratado		FK	INT

Actualizado SegundaPreEntrega

Diagrama de entidad relación (DER):

Link para verlo mejor:

https://lucid.app/lucidchart/ba7d6576-20cd-4f02-821b-daf5eb74f000/edit?viewport_loc=-1818%2C-582%2C1997%2C873%2C0_0&invitationId=inv_91685e3b-1af0-4e1f-a7e7-c29b5f9383e0



SEGUNDA PRE ENTREGA

AGUSTIN ALVAREZ

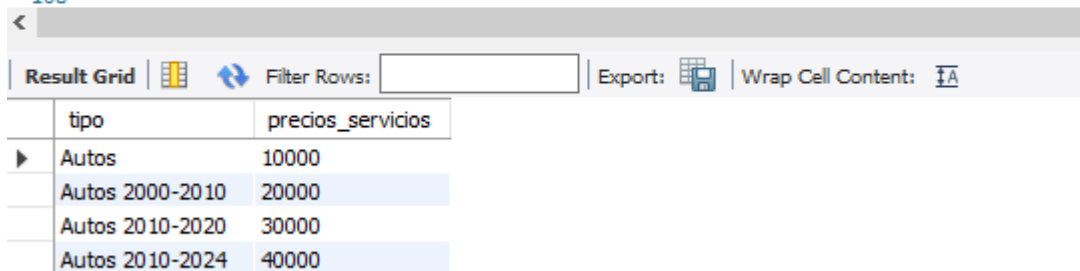
Listado de Vistas :

- Servicios_autos_vw
- Servicios_motos_vw
- Servicios_viviendas_vw

Estas 3 vistas están pensadas para simplificar la visualización de los precios de los distintos tipos de servicios, a cada una de ellas la componen la tabla "Servicios". Toma la información solicitada, encuentra y muestra los distintos precios.

Vista para autos:

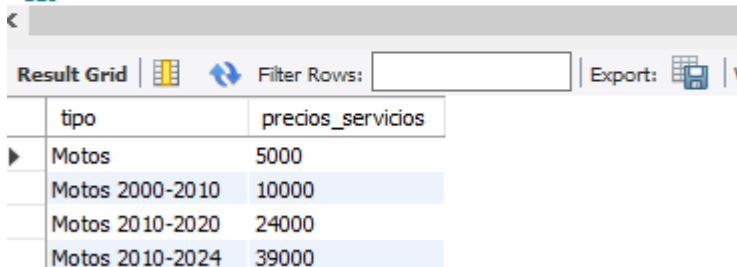
```
104 • CREATE OR REPLACE VIEW servicios_autos_vw AS
105     SELECT tipo, precios_servicios
106     FROM servicios
107     WHERE tipo LIKE "%Autos%";
108
```



	tipo	precios_servicios
▶	Autos	10000
	Autos 2000-2010	20000
	Autos 2010-2020	30000
	Autos 2010-2024	40000

Vista para motos:

```
121 • CREATE OR REPLACE VIEW servicios_motos_vw AS
122     SELECT tipo, precios_servicios
123     FROM servicios
124     WHERE tipo LIKE "%Motos%";
125
```



	tipo	precios_servicios
▶	Motos	5000
	Motos 2000-2010	10000
	Motos 2010-2020	24000
	Motos 2010-2024	39000

Vista para viviendas:

```
126 • CREATE OR REPLACE VIEW servicios_viviendas_vw AS
127     SELECT tipo, precios_servicios
128     FROM servicios
129     WHERE tipo LIKE "%Viviendas%";
130
```

Result Grid		Filter Rows:	Export:	Wrap
tipo	precios_servicios			
Viviendas	25000			
Viviendas Monoambiente	100000			
Viviendas 4 habitaciones	150000			
Viviendas 4 habitaciones, patio y garage	350000			

- Servicios_contratado_vw

Y mi cuarta vista fue una en la que podamos ver que servicios fueron contratados, toma información de la tabla "Usuario" y de la tabla "Servicios" haciendo de los datos recolectados la siguiente vista:

```
135 -- SE CREA LA VISTA PARA VER QUE SERVICIO CONTRATO CADA USUARIO --
136 • CREATE OR REPLACE VIEW servicio_contratado_vw AS
137     SELECT u.nombre_de_usuario, u.nombre, s.id_producto, s.tipo
138     FROM usuario AS u JOIN servicios AS s ;
139
```

Result Grid					Filter Rows:	Export:	Wrap Cell Content:
nombre_de_usuario	nombre	id_producto	tipo				
InnoZ	Kelsey	103	Viviendas				
InnoZ	Kelsey	102	Motos				
InnoZ	Kelsey	101	Autos				
Edgeblab	Nicole	112	Viviendas 4 habitaciones, patio y garage				
Edgeblab	Nicole	111	Viviendas 4 habitaciones				
Edgeblab	Nicole	110	Viviendas Monoambiente				
Edgeblab	Nicole	109	Motos 2010-2024				

Listado de Funciones

Utilice esta función para hacer el cálculo al que se tenían que vender algunos servicios fijando un monto y un cargo.

```
-- CREAMOS FUNCION PARA AVERIGUAR EL PRECIO DE VENTA DE UN SERVICIO
DELIMITER $$
CREATE FUNCTION `precio_servicio_venta_final_fn` (monto DECIMAL(11,2), cargo DECIMAL(4,2))
RETURNS DECIMAL (11,2)
NO SQL
BEGIN
    DECLARE resultado DECIMAL(11,2);
    SET resultado = monto + monto * (cargo/100);
    RETURN resultado;
END$$
DELIMITER ;
```

Luego utilice estas otras dos para calcular el precio final + IVA de algunos servicios:

Primero cree esta para utilizarla en la próxima función, en la que también debo de fijar un impuesto.

```
DROP function if exists calcular_iva_venta_fn;
DELIMITER $$
CREATE FUNCTION calcular_iva_venta_fn(monto DECIMAL(11,2))
RETURNS DECIMAL(11,2)
NO SQL
BEGIN
    DECLARE resultado DECIMAL(11,2);
    DECLARE impuesto DECIMAL(11,2);
    SET impuesto = 15.00;
    SET resultado = monto * (impuesto / 100);
    RETURN resultado;
END$$
DELIMITER ;
```

Y en esta otra junto el monto y la función de arriba, para que me de como resultado el precio + IVA:

```
DELIMITER $$
CREATE FUNCTION calcular_total_venta_fn(monto DECIMAL(11,2))
RETURNS DECIMAL (11,2)
NO SQL
> BEGIN
    DECLARE resultado DECIMAL(11,2);
    SET resultado = monto + calcular_iva_venta_fn(monto);
    RETURN resultado;
~ END$$
DELIMITER ;
```

Listado de Stored Procedures :

Utilice este Stored Procedure para ver los datos en orden, como se ve en la imagen hay que insertar que tabla, el campo y el orden en el que se desea que se muestre.

Se pueden utilizar e interactuar con todas las tablas.

```
CREATE PROCEDURE ordenar_tablas_sp (IN tabla VARCHAR (20), IN campo VARCHAR (20), IN orden VARCHAR (4))
) BEGIN
SET @ordenar = CONCAT( 'SELECT * FROM', ' ', tabla, ' ', 'ORDER BY', ' ', campo, ' ', orden);
PREPARE consulta FROM @ordenar;
EXECUTE consulta;
DEALLOCATE PREPARE consulta;
- END $$
DELIMITER ;

-- TABLA SERVICIOS ORDENADA CON LOS PRECIOS DE FORMA DESCENDENTE --
CALL ordenar_tablas_sp ('servicios', 'precios_servicios', 'DESC');
-- TABLA SERVICIOS ORDENADA CON LOS PRECIOS DE FORMA ASCENDENTE --
CALL ordenar_tablas_sp ('servicios', 'precios_servicios', 'ASC');
```

En este otro Stored Procedure lo utilice para poder registrar la fecha, el tipo de servicio, el usuario que lo contrato y marcar un orden en cada servicio que va entrando.

```
-- REGISTRAR LAS CONTRATACIONES A MEDIDA QUE SE LAS CONTRATAN --
DELIMITER $$
CREATE PROCEDURE servicios_contratados_sp (IN orden INT, IN sp_fecha DATE, IN sp_nombre VARCHAR (30), in sp_tipo VARCHAR (30))
) BEGIN
INSERT INTO servicios_contratados
(id_orden, fecha, nombre, tipo)
VALUES
(orden, sp_fecha, sp_nombre, sp_tipo);
- END $$
DELIMITER ;

-- INSERTAR SERVICIOS CONTRATADOS --
CALL servicios_contratados_sp (1, '2024-08-15', 'Roombo', "Autos");
```

Este último lo utilice para crear usuarios y que se vayan guardando en su tabla (Usuario),

Solicitando los 3 datos que la componen.


```

-- STORED PROCEDURE PARA INSERTAR NUEVOS USUARIOS
DELIMITER $$
CREATE PROCEDURE insertar_nuevo_usuario_sp(
    IN p_nombre_de_usuario VARCHAR(15),
    IN p_nombre VARCHAR(20),
    IN p_apellido VARCHAR(20))
BEGIN
    INSERT INTO usuario (nombre_de_usuario, nombre, apellido)
    VALUES (p_nombre_de_usuario, p_nombre, p_apellido);
END $$
DELIMITER ;

```

Listado de Triggers:

- clientes_nuevos_tr: Este trigger tiene la tarea de que a cada servicio vendido, almacenar en una nueva tabla a los clientes. Monitorea la tabla “servicios_contratados”, luego de que se realice un INSERT y registra todo en la tabla “clientes”
- up_pedido_tr: Este trigger tiene que funcionar a medida que los pedidos se modifiquen, ya sea que hubo un error de seleccion o algo, guarde la informacion antes del cambio. Monitorea la tabla “servicios_contratados”, luego de que se realice un UPDATE registra todo en la tabla “up_pedido”

Inserción de datos en formato CSV

Para las tablas usuario, seguros_viviendas e información_personal importe los datos de las siguientes tablas.

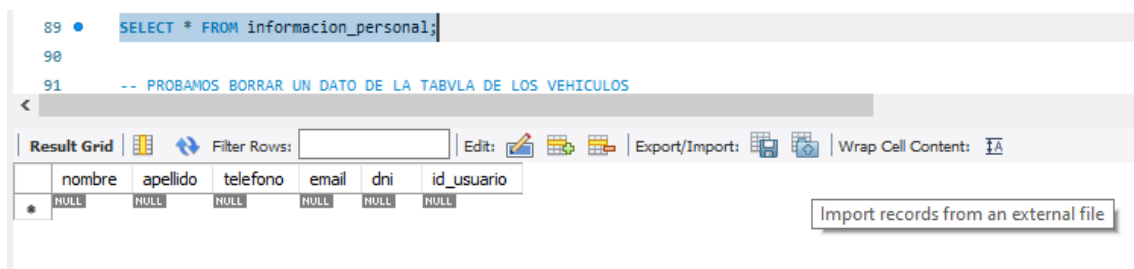
- Primero cree la tabla:

```

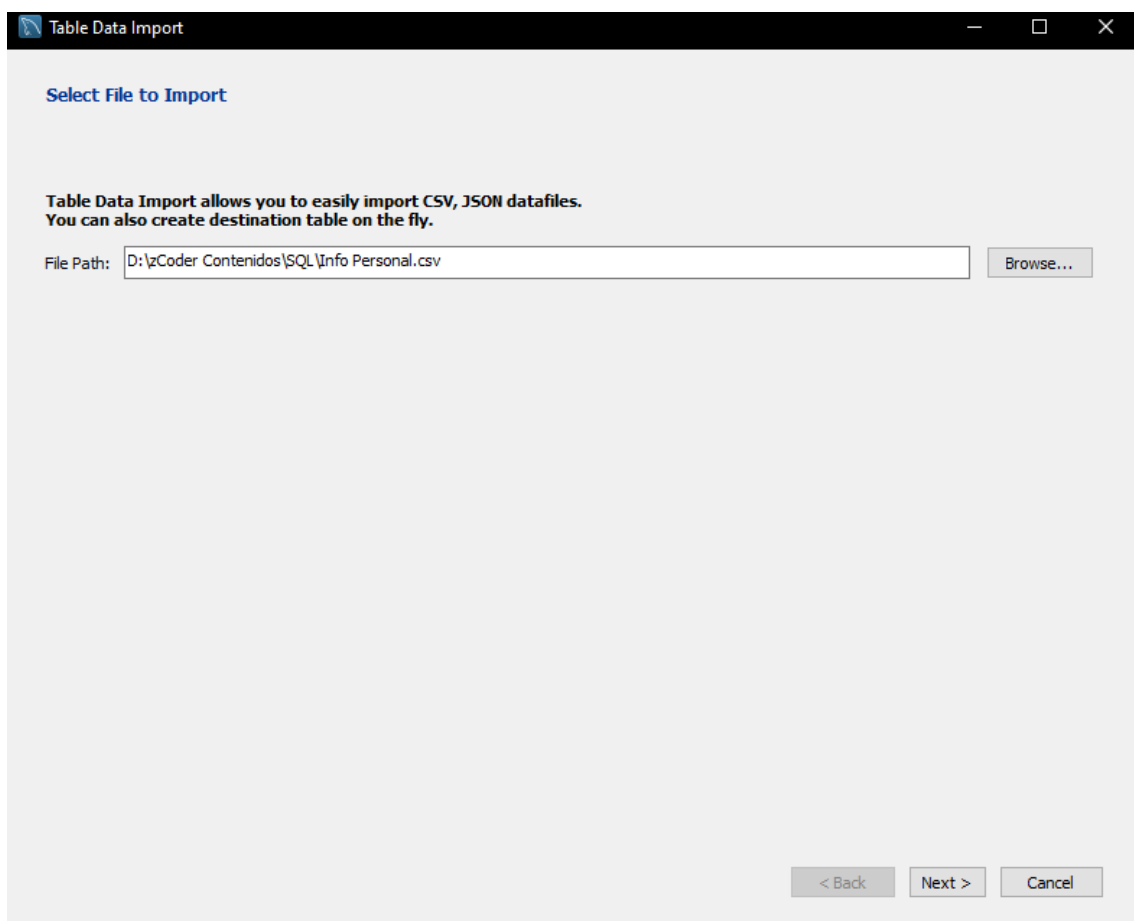
CREATE TABLE IF NOT EXISTS informacion_personal (
    nombre VARCHAR(40) NOT NULL,
    apellido VARCHAR(40) NOT NULL,
    telefono FLOAT(20) NOT NULL,
    email VARCHAR(30) NOT NULL,
    dni NUMERIC(20) NOT NULL,
    id_usuario INT AUTO_INCREMENT,
    PRIMARY KEY (id_usuario, dni),
    FOREIGN KEY (id_usuario) REFERENCES usuario(id_usuario));

```

- Luego inserte los datos yendo al dibujo de la carpeta que dice “Import records from an external file”:



- Una vez adentro busque el documento .CSV para poder hacer la inserción:



- Luego seleccione la tabla a la que quería que se importen los datos:

Select destination table and additional options.

☒ Use existing table:

☐ Create new table: .

☐ Truncate table before import

- Puse siguiente y corroboré que estén todos los datos en orden:

Table Data Import

Configure Import Settings

Detected file format: csv

Encoding:

Columns:

<input checked="" type="checkbox"/> Source Column	Dest Column
<input checked="" type="checkbox"/> nombre	<input type="text" value="nombre"/>
<input checked="" type="checkbox"/> apellido	<input type="text" value="apellido"/>
<input checked="" type="checkbox"/> telefono	<input type="text" value="telefono"/>
<input checked="" type="checkbox"/> email	<input type="text" value="email"/>
<input checked="" type="checkbox"/> dni	<input type="text" value="dni"/>

nombre	apellido	telefono	email	dni
Dagny	Vawton	943036183	dvawton0@...	850220765
Maure	Guilliatt	36562657	mguilliatt1...	695399943
Maryanna	Neeves	840391457	mneeves2...	635002238
Murdoch	Lexa	62025359	mlexa3@dii...	491248950
Carline	Ricioppo	659265671	cricioppo4...	859807773

< Back Next > Cancel

- Pulse Next hasta que se importaron los datos:

File D:\zCoder Contenidos\SQL\Info Personal.csv was imported in 0.533 s

Table agencia_aseguradora.informacion_personal has been used

10 records imported

- Finalizado el proceso fui a revisar que este toda la información en la tabla de mi db:

```
89 • SELECT * FROM informacion_personal;
90
```

Result Grid						
Filter Rows:						
Edit:						
Export/Import:						
	nombre	apellido	telefono	email	dni	id_usuario
▶	Dagny	Vawton	943036000	dvawton0@networksolutions.com	850220765	1
	Maure	Guilliatt	36562700	mguilliatt1@tripod.com	695399943	2
	Maryanna	Neeves	840391000	mneeves2@discuz.net	635002238	3
	Murdoch	Lexa	62025400	mlexa3@diigo.com	491248950	4
	Carline	Ricioppo	659266000	cricioppo4@comcast.net	859807773	5
	Leticia	Shrubshall	123471000	lshrubshall5@1688.com	98896917	6
	Bren	Luchelli	171838000	bluchelli6@opensource.org	357657411	7
	Julie	Oxteby	828692000	joxteby7@gizmodo.com	297418034	8
	Wilton	Merton	731660000	wmerton8@amazon.de	206114829	9
	Fallon	Fairholme	679161000	ffairholme9@addtoany.com	233336126	10
*	NULL	NULL	NULL	NULL	NULL	NULL

Y listo, este procedimiento lo repetí en las tablas “usuario” y “seguros_viviendas”.

Los Excel que utilice están en el link del repositorio.

REPOSITORIO EN GITHUB:

<https://github.com/AgustinAlvarez04/SQL-SegundaPreEntega/tree/main>