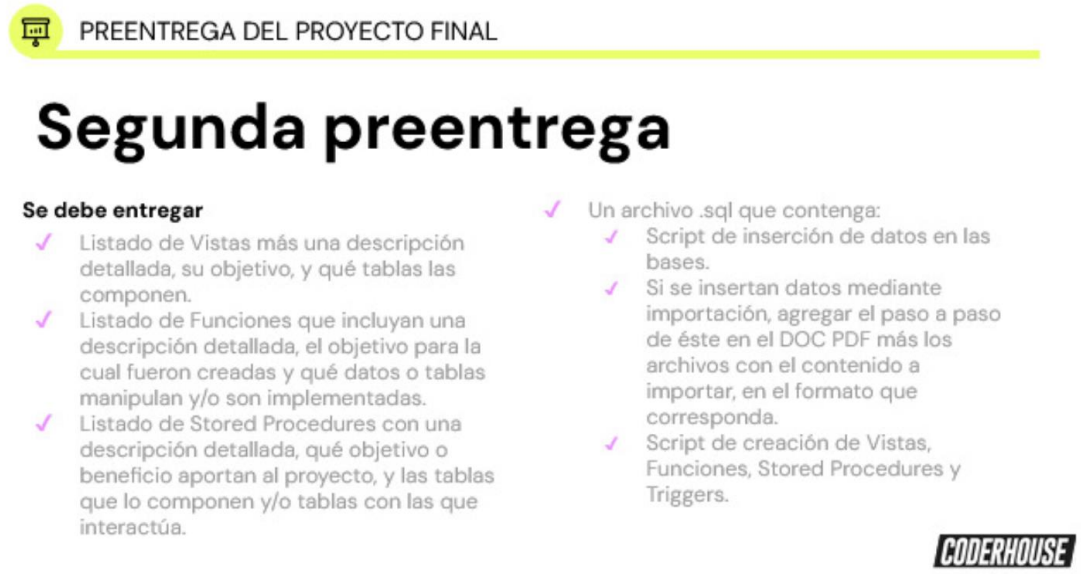
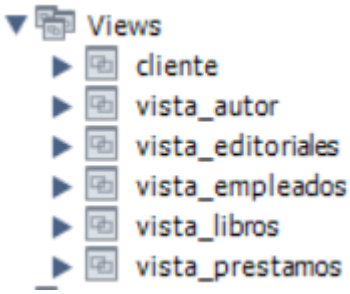


2da entrega

VISTAS

Las vistas que se listan a continuación en el proyecto, sirven para simplificar y optimizar consultas de una manera mas sencilla. Como se ven en la siguiente captura y tambien en el código, ellas son vista cliente, autor, editoriales, empleados, libros y prestamos



Se intenta apuntar a la consulta de ciertos datos, los que resulten mas relevantes para tener un acceso rápido, optimizado y seguro pero siempre más simplificado

1-

```
create or replace view vista_Cliente as
(select nombre, apellido,mail from cliente);

select * from vista_Cliente;
```

2-

```
create view vista_autor as (select nombre, nacionalidad from autor a);
select * from vista_autor va ;
```

3-

```
create view vista_editoriales as (select nombre_editorial from editoriales);

select * from vista_editoriales ve ;
```

4-

```
create view vista_empleados as
(select nombre,nacionalidad from empleados);

select * from vista_empleados ve ;
```

5-

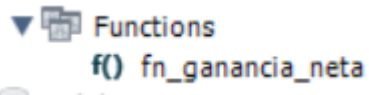
```
create or replace view vista_libros as (select id_libros ,titulo,autor from libros l);

select * from vista_libros vl ;
```

6-

```
create or replace view vista_prestamos as(select id_libros, fecha_prestamo,fecha_devolucion from prestamos);

select * from vista_prestamos vp ;
```

FUNCIONES

La función que se detalla a continuación



Tiene como finalidad generar una operación específica en los datos de la base de datos, en este caso se intento obtener de la tabla “venta de libros” la ganancia que se obtiene de restar del precio de venta del libro el costo, obteniendo la ganancia neta.

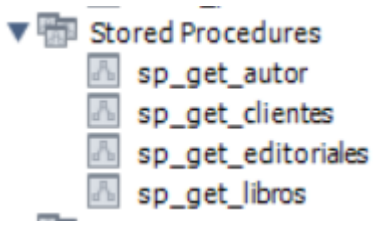
En términos generales la función ayuda a personalizar las script según la necesidad del usuario, evita duplicación de código, acelera la consulta optimizándola y la hace más eficiente.

```
-- CREACION DE FUNCION
DELIMITER $$

CREATE FUNCTION fn_ganancia_neta(venta_id INT) RETURNS decimal (15,2)
deterministic
    no sql
BEGIN
    DECLARE diferenciaPrecio decimal (15,2);
    select
    (precio_venta_libro - costo_libro) INTO diferenciaPrecio
from venta_libros where id_venta = venta_id;

    RETURN diferenciaPrecio;
END $$

DELIMITER ;

select fn_ganancia_neta(2);
```

Stored Procedures

En términos grales los procedimientos almacenados son una herramienta esencial para mejorar la eficiencia, la seguridad y la mantenibilidad de las bases de datos.

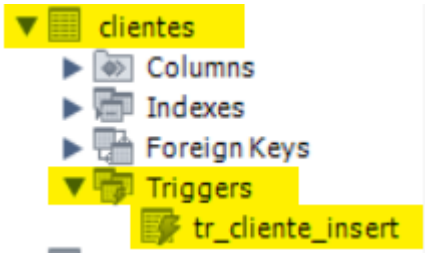
Como se muestra en la siguiente captura , se hicieron en la Tabla autor, clientes, editoriales y libros del proyecto_biblioteca.




```
-- creación de stored procedures
call sp_get_clientes();

call sp_get_editoriales();

call sp_get_autor();

call sp_get_libros();
```

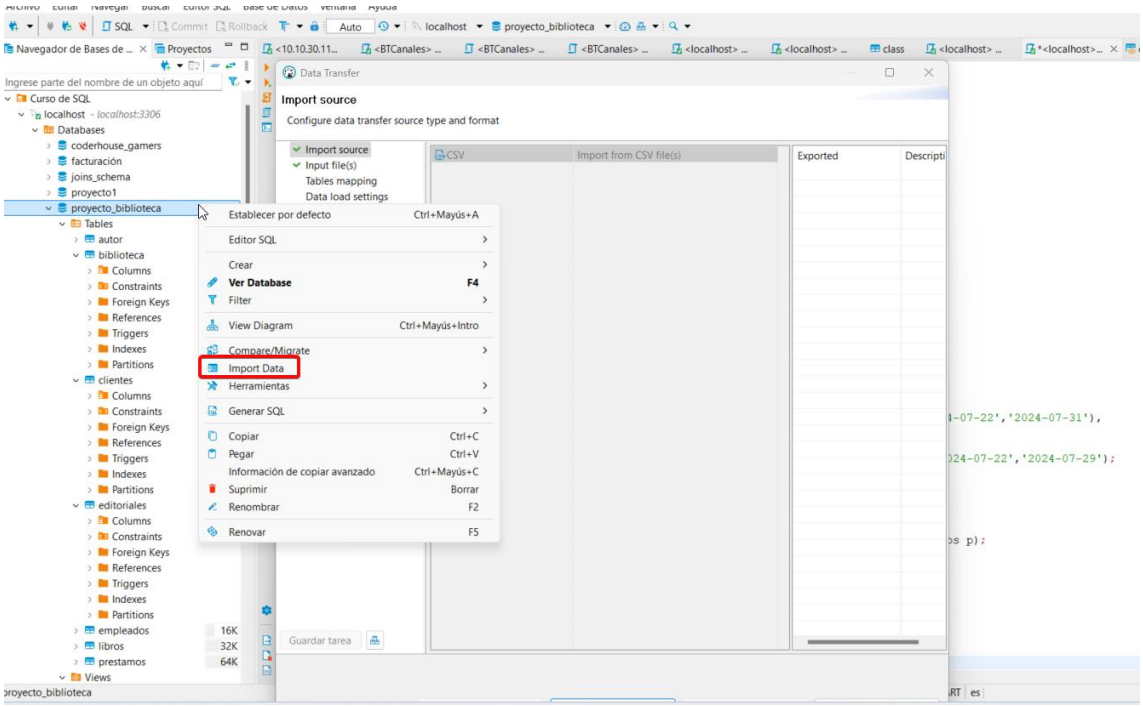
TRIGGERS

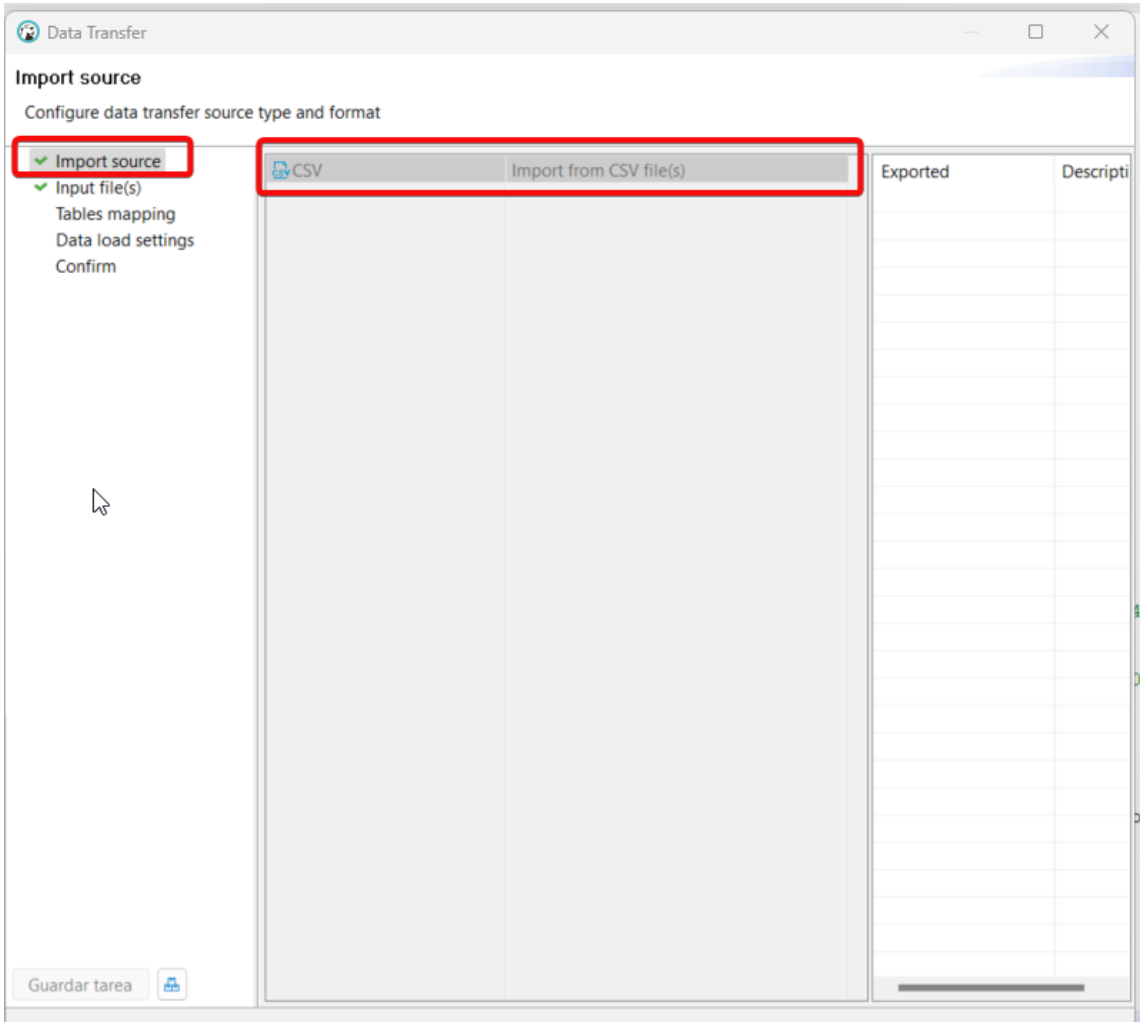
Se inserta en la Tabla Clientes, el siguiente procedimiento almacenado que se ejecutará automáticamente cuando ingrese (INSERT) a la base de datos un nuevo cliente , dejando constancia del ID de cliente y del la fecha y hora en que se registró.

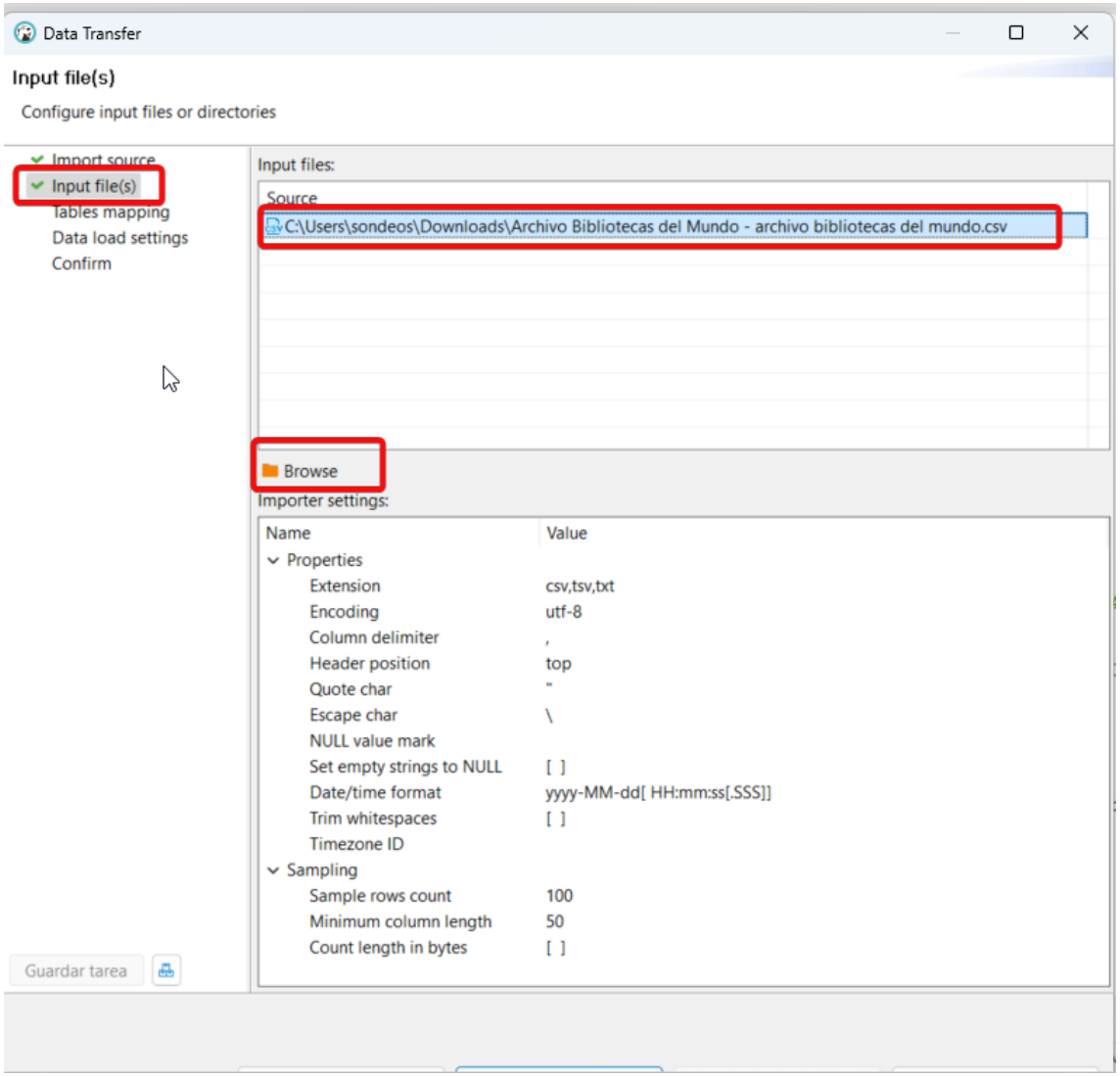


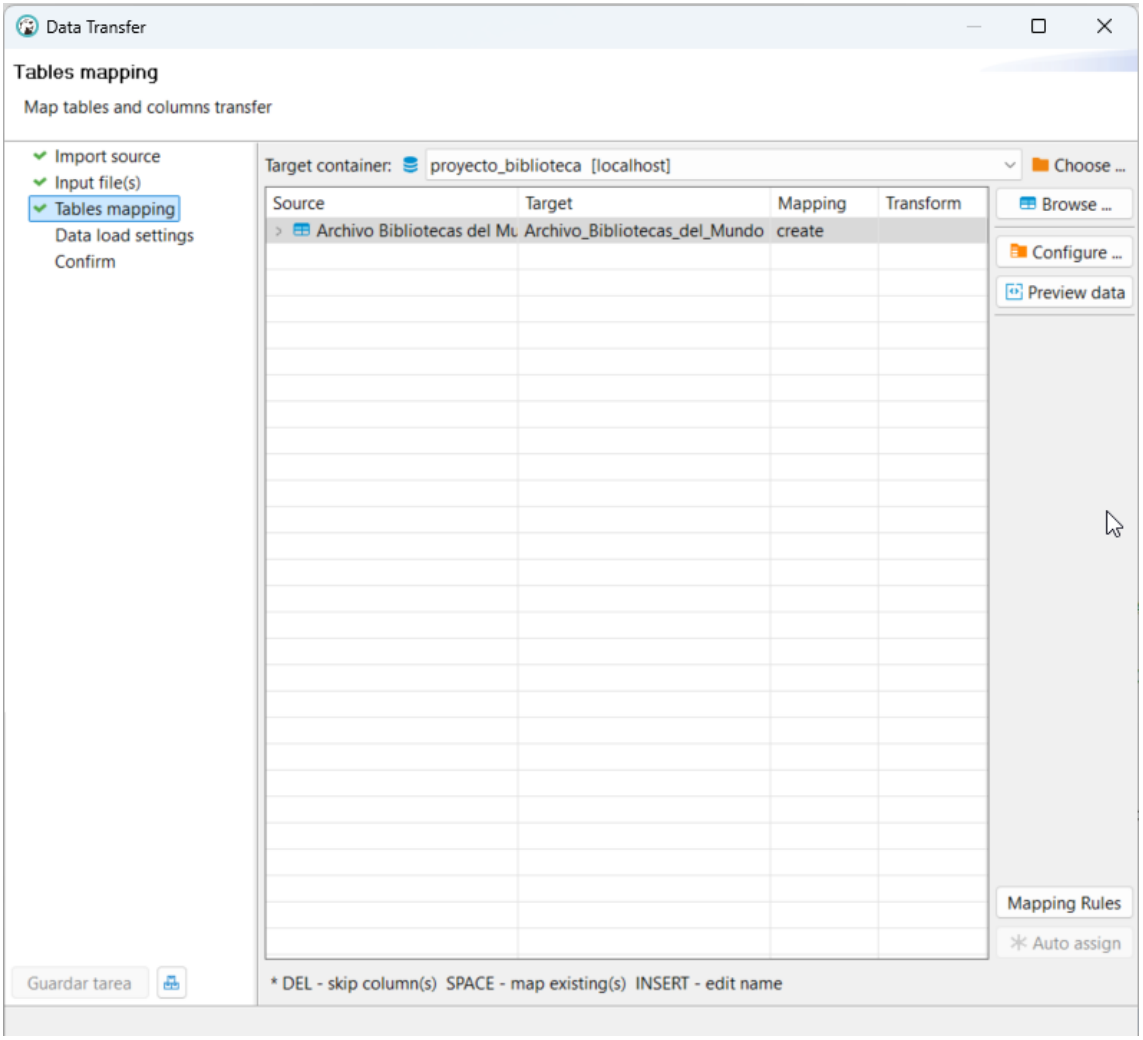

```
-- Crear el trigger
DELIMITER //
CREATE TRIGGER tr_cliente_insert
AFTER INSERT ON clientes
FOR EACH ROW
BEGIN
    -- Acción a realizar cuando se inserta un cliente
    INSERT INTO registro_clientes (id_cliente, fecha_registro)
    VALUES (NEW.id_cliente_lector, NOW());
END;
//
DELIMITER ;
```

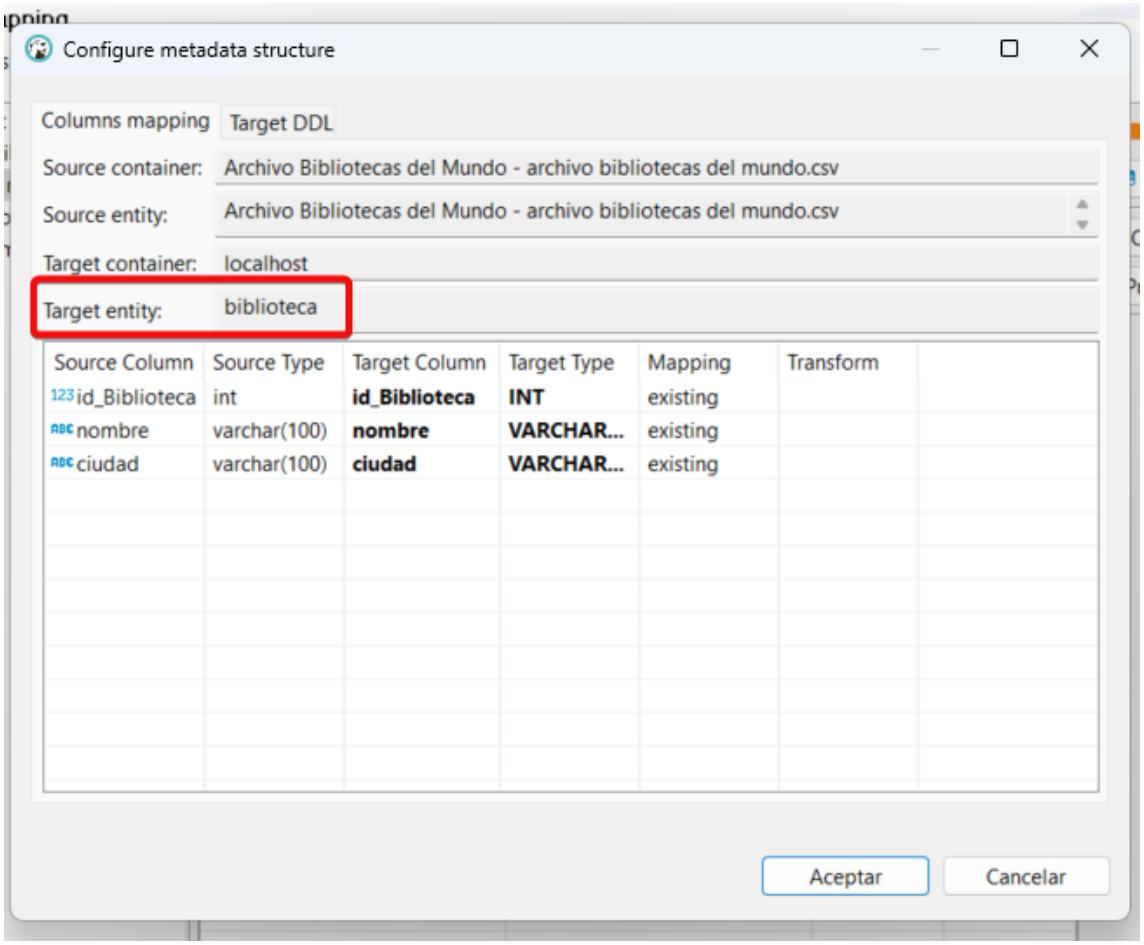
PASOS PARA LA IMPORTACIÓN DE ARCHIVOS

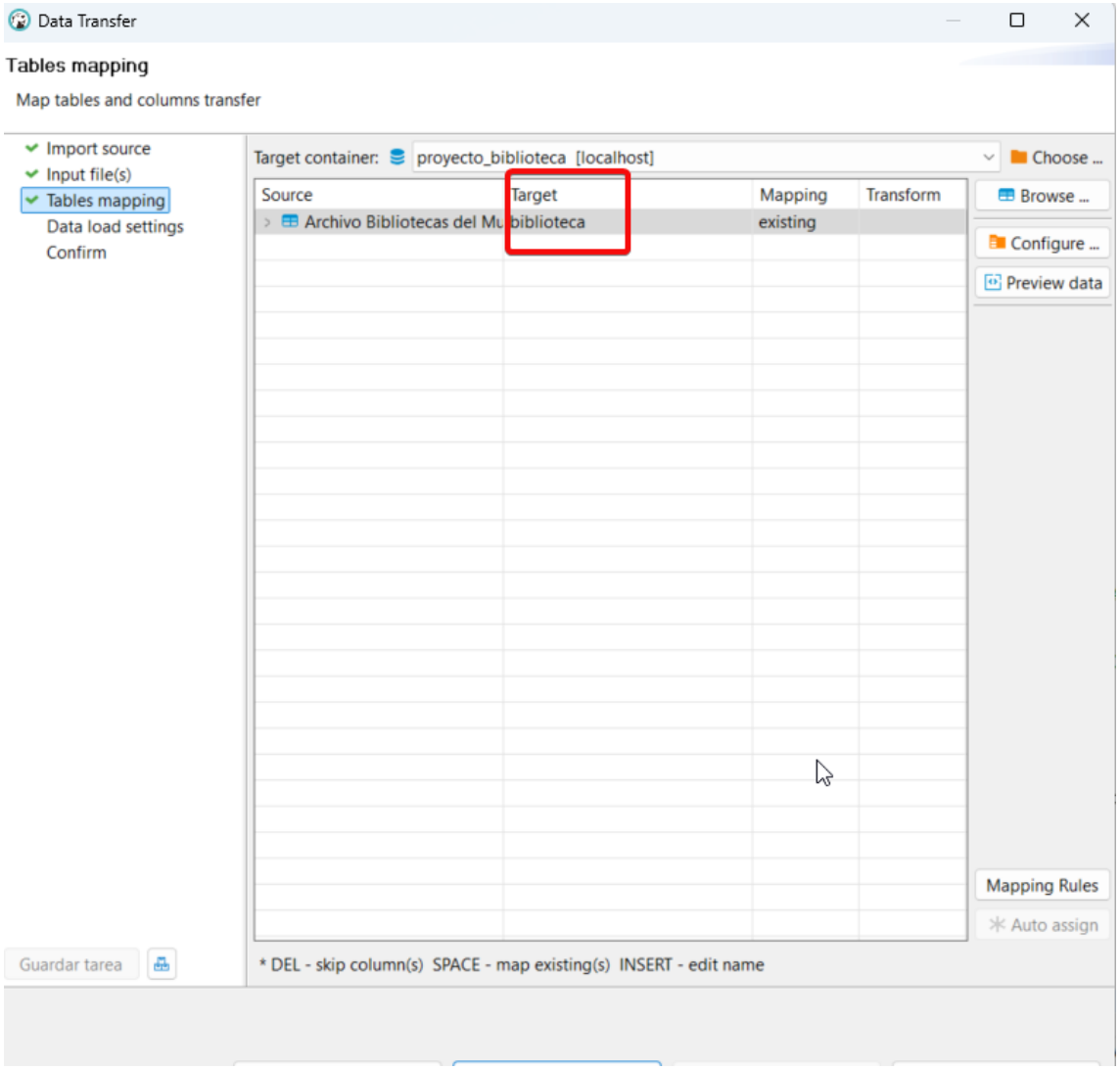


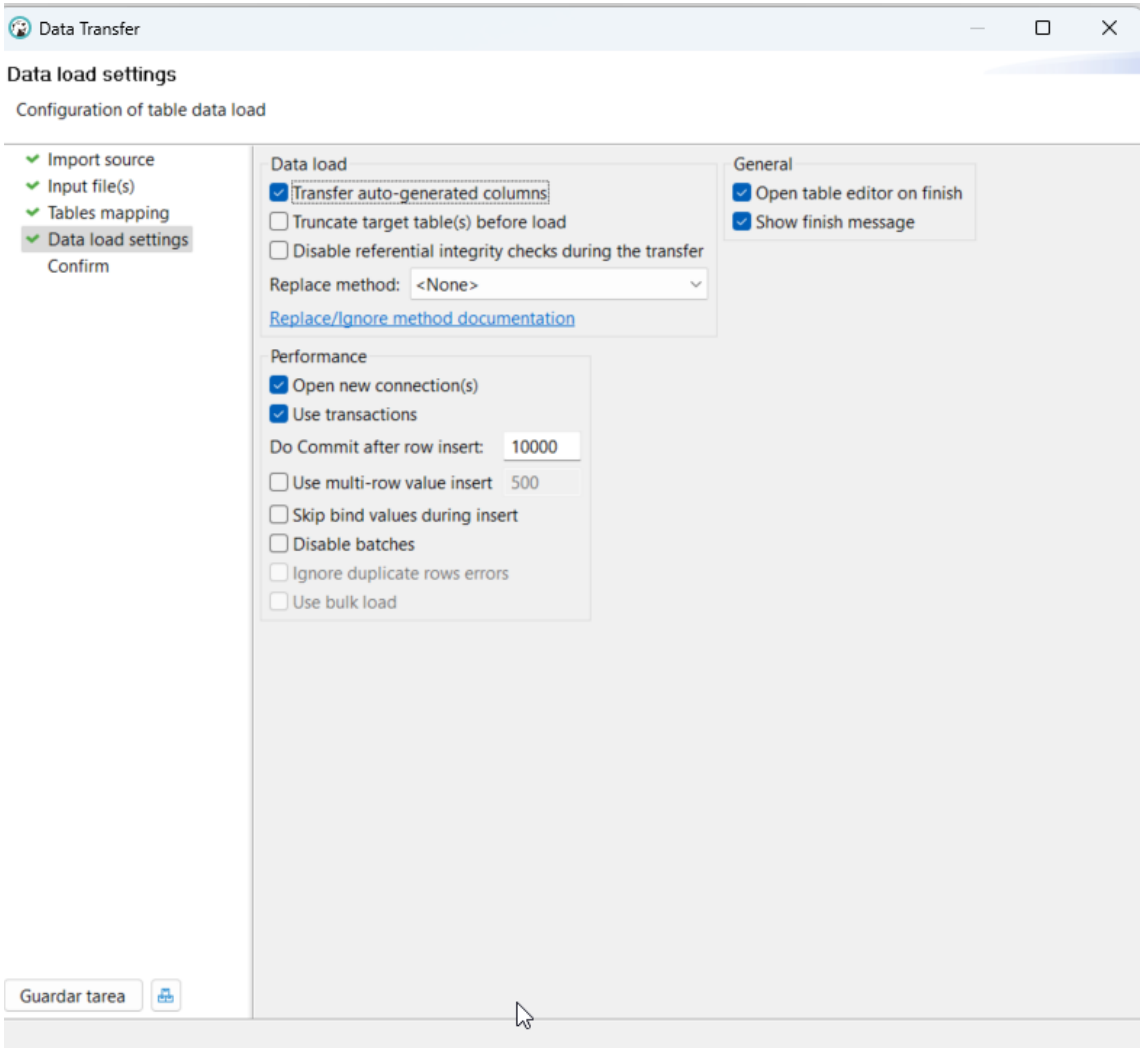

Data Transfer
Import source
Configure data transfer source type and format
Import source
Input file(s)
Tables mapping
Data load settings
Confirm
CSV
Import from CSV file(s)
Exported
Descripti
Guardar tarea

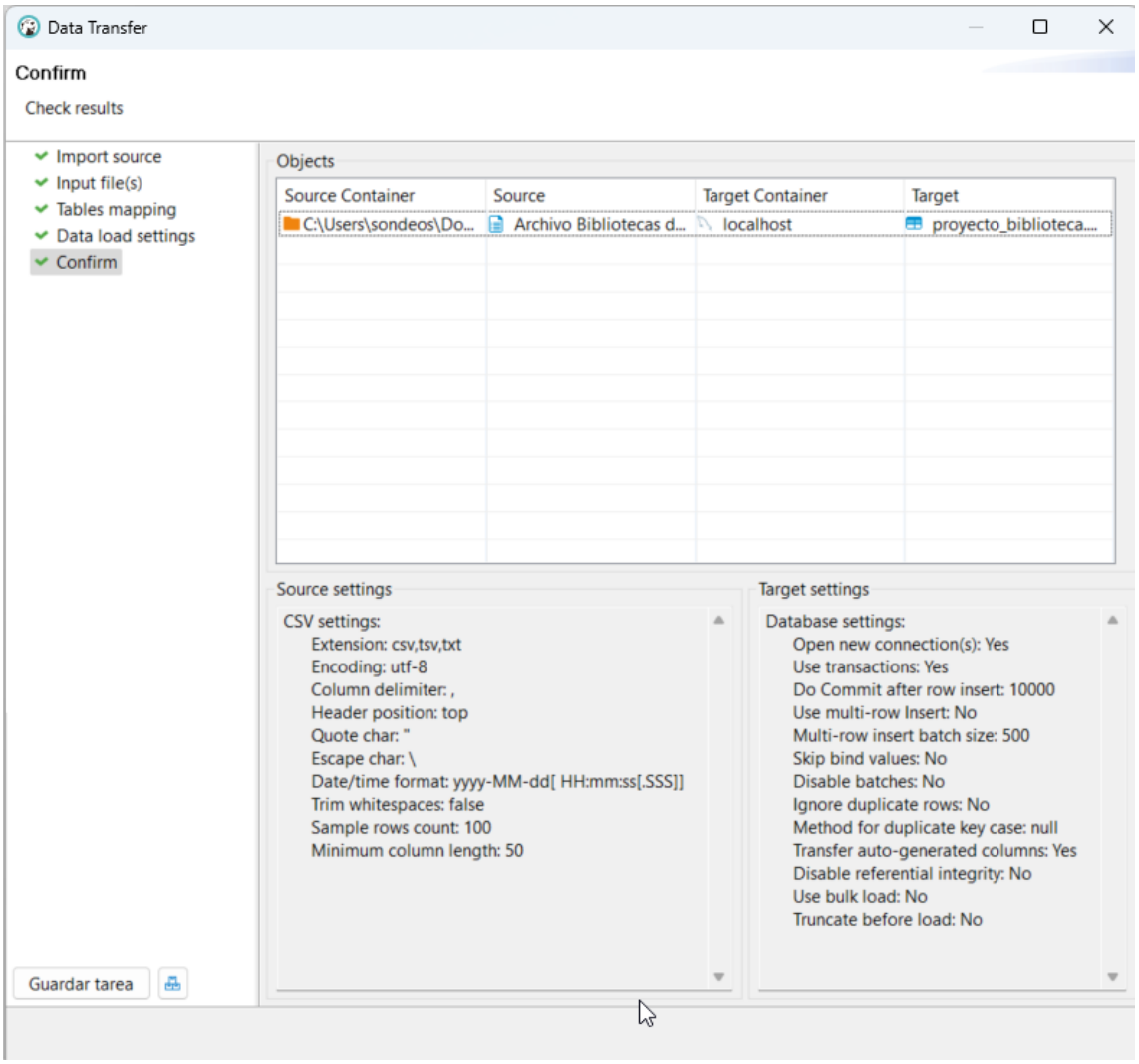

Data Transfer
Input file(s)
Configure input files or directories
Import source
Input file(s)
Tables mapping
Data load settings
Confirm
Input files:
Source
C:\Users\sondeos\Downloads\Archivo Bibliotecas del Mundo - archivo bibliotecas del mundo.csv
Browse
Importer settings:
Name
Value
Properties
Extension
csv,tsv,txt
Encoding
utf-8
Column delimiter
,
Header position
top
Quote char
"
Escape char
\
NULL value mark
Set empty strings to NULL
[ ]
Date/time format
yyyy-MM-dd[HH:mm:ss[.SSS]]
Trim whitespaces
[ ]
Timezone ID
Sampling
Sample rows count
100
Minimum column length
50
Count length in bytes
[ ]
Guardar tarea

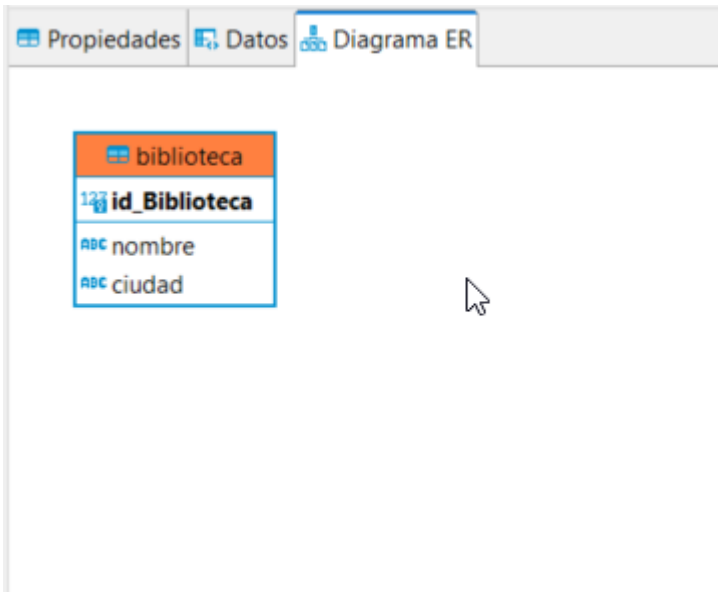

Data Transfer
Tables mapping
Map tables and columns transfer
Import source
Input file(s)
Tables mapping
Data load settings
Confirm
Target container:
proyecto_biblioteca [localhost]
Choose ...
Source
Target
Mapping
Transform
Archivo Bibliotecas del Mu
Archivo_Bibliotecas_del_Mundo
create
Browse ...
Configure ...
Preview data
Mapping Rules
Auto assign
Guardar tarea
* DEL - skip column(s) SPACE - map existing(s) INSERT - edit name


Configure metadata structure
Columns mapping
Target DDL
Source container:
Archivo Bibliotecas del Mundo - archivo bibliotecas del mundo.csv
Source entity:
Archivo Bibliotecas del Mundo - archivo bibliotecas del mundo.csv
Target container:
localhost
Target entity:
biblioteca
Source Column
Source Type
Target Column
Target Type
Mapping
Transform
id_Biblioteca
int
id_Biblioteca
INT
existing
nombre
varchar(100)
nombre
VARCHAR...
existing
ciudad
varchar(100)
ciudad
VARCHAR...
existing
Aceptar
Cancelar


Data Transfer
Tables mapping
Map tables and columns transfer
Import source
Input file(s)
Tables mapping
Data load settings
Confirm
Target container:
proyecto_biblioteca [localhost]
Choose ...
Source
Target
Mapping
Transform
Archivo Bibliotecas del Mu
biblioteca
existing
Browse ...
Configure ...
Preview data
Mapping Rules
Auto assign
Guardar tarea
* DEL - skip column(s) SPACE - map existing(s) INSERT - edit name


Data Transfer
Data load settings
Configuration of table data load
Import source
Input file(s)
Tables mapping
Data load settings
Confirm
Data load
Transfer auto-generated columns
Truncate target table(s) before load
Disable referential integrity checks during the transfer
Replace method: <None>
Replace/Ignore method documentation
General
Open table editor on finish
Show finish message
Performance
Open new connection(s)
Use transactions
Do Commit after row insert: 10000
Use multi-row value insert 500
Skip bind values during insert
Disable batches
Ignore duplicate rows errors
Use bulk load
Guardar tarea


Data Transfer
Confirm
Check results
Import source
Input file(s)
Tables mapping
Data load settings
Confirm
Objects
Source Container
Source
Target Container
Target
C:\Users\sondeos\Do...
Archivo Bibliotecas d...
localhost
proyecto_biblioteca....
Source settings
CSV settings:
Extension: csv,tsv,txt
Encoding: utf-8
Column delimiter: ,
Header position: top
Quote char: "
Escape char: \
Date/time format: yyyy-MM-dd[HH:mm:ss[.SSS]]
Trim whitespaces: false
Sample rows count: 100
Minimum column length: 50
Target settings
Database settings:
Open new connection(s): Yes
Use transactions: Yes
Do Commit after row insert: 10000
Use multi-row Insert: No
Multi-row insert batch size: 500
Skip bind values: No
Disable batches: No
Ignore duplicate rows: No
Method for duplicate key case: null
Transfer auto-generated columns: Yes
Disable referential integrity: No
Use bulk load: No
Truncate before load: No
Guardar tarea


Propiedades
Datos
Diagrama ER
biblioteca
id_Biblioteca
nombre
ciudad

Propiedades | Datos | Diagrama ER

biblioteca | Enter a SQL expression to filter results (use Ctrl+Space)

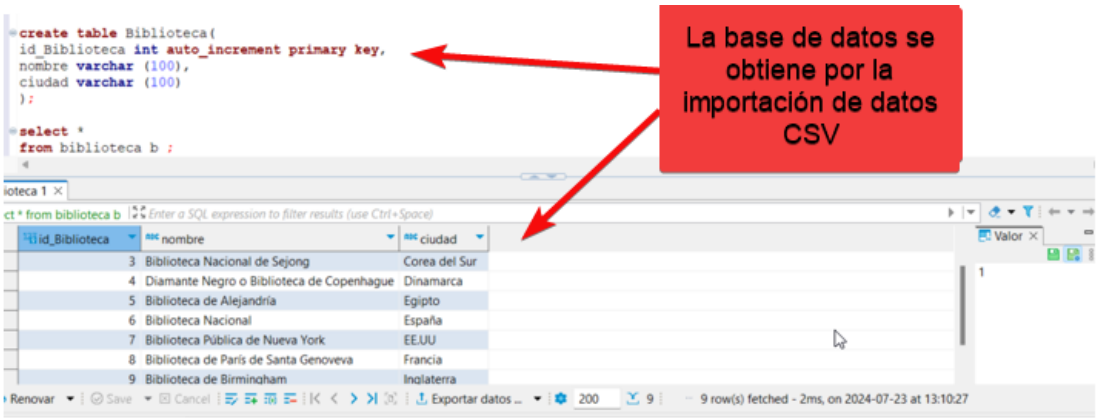
Grilla | Texto

	id_Biblioteca	nombre	ciudad
1	1	Biblioteca pública de Sttugart	Alemania
2	2	Biblioteca de Tianjin Binhai	China
3	3	Biblioteca Nacional de Sejong	Corea del Sur
4	4	Diamante Negro o Biblioteca de Copenhague	Dinamarca
5	5	Biblioteca de Alejandría	Egipto
6	6	Biblioteca Nacional	España
7	7	Biblioteca Pública de Nueva York	EE.UU
8	8	Biblioteca de París de Santa Genoveva	Francia
9	9	Biblioteca de Birmingham	Inglaterra

Table Name: biblioteca Partitioned
Engine: InnoDB
Auto Increment: 0
Charset: utf8mb4
Collation: utf8mb4_0900_ai_ci
Description:

Columns | Constraints | Foreign Keys | References | Triggers | Indexes | Partitions | Statistics | DDL | Virtual

Column Name	#	Data Type	Not Null	Auto Increment	Key	Default	Extra	Express
id_Bibliote...	1	int	[v]	[v]	PRI		auto_incre...	
nombre	2	varchar(100)	[ ]	[ ]				
ciudad	3	varchar(100)	[ ]	[ ]				



id_Biblioteca	nombre	ciudad
1	Biblioteca pública de Sttugart	Alemania
2	Biblioteca de Tianjin Binhai	China
3	Biblioteca Nacional de Sejong	Corea del Sur
4	Diamante Negro o Biblioteca de Copenhague	Dinamarca
5	Biblioteca de Alejandría	Egipto
6	Biblioteca Nacional	España
7	Biblioteca Pública de Nueva York	EE.UU
8	Biblioteca de París de Santa Genoveva	Francia
9	Biblioteca de Birmingham	Inglaterra
NULL	NULL	NULL