

# API N°3

Materia: Base de datos y Big Data

Alumno: Héctor H. Vergara

## Tabla de contenidos

<b>Situación.....</b>	<b>2</b>
<b>Introducción.....</b>	<b>3</b>
Repositorio.....	3
Dependencias.....	3
<b>Cómo empezar.....</b>	<b>3</b>
Creación de bases de datos y scripts SQL.....	4
<b>Diseño de entidades.....</b>	<b>5</b>
Lineamientos de diseño.....	6
ERDs.....	6
farispley_stock.....	6
farispley_stock_dw.....	6
farispley_dm_proveedores.....	7
farispley_dm_scanning.....	7
Diagrama de arquitectura de alto nivel.....	8
<b>Creación de las tablas (stock y DW).....</b>	<b>8</b>
<b>Insertión de datos de prueba.....</b>	<b>10</b>
<b>Resolución de consignas.....</b>	<b>10</b>
1. Generar SQL Queries.....	10
2. Crear las relaciones entre las tablas del DW y DM.....	11



## Situación

*Farispley ya cuenta con un modelo de datos maestros, el cual ha sido generado para realizar la gestión sobre la información y, con esto, acercarse cada vez más a la gran y anhelada transformación digital. Todos los usuarios de negocio se encuentran ansiosos por empezar a mirar datos, y para ello falta generar los DATAMARTs necesarios para hacer disponible la información en la capa de acceso a la cual toda la compañía podrá entrar a consultar.*

*Este es el primer requerimiento formal de explotación de datos en el data warehouse de la compañía. Como no todos los usuarios de negocio tienen acceso a la capa maestra, es necesario generar una capa que se abstraiga de los conceptos técnicos del modelo anterior y, a su vez, darle una mirada más funcional.*

*La base de datos transaccional es un SQL Server y está compuesta por las siguientes tablas:*

MNIdStockFisico		SIP..SIPdStocks		MNADARTICULOS		MNAdTipoArticulos	
IdStock	(PK, Integer)	Id	(PK, Integer)	Id	(PK, Integer)	Mna	(PK, VARCHAR)
Descripcion	(VARCHAR)	UxB	(Integer)	IdArticulo	(Integer)	Descripcion	(VARCHAR)
Apto	(VARCHAR)	IdStock	(FK, Integer)	IdNumero	(Integer)	TipoCotizacion	(VARCHAR)
Usuario	(VARCHAR)	CantStock	(Integer)	Mna	(VARCHAR)	Ts	(TIMESTAMP)
Terminal	(VARCHAR)	Unidades	(Integer)	SectSecc	(Integer)		
Ts	(TIMESTAMP)	Usuario	(VARCHAR)	Usuario	(VARCHAR)		
		Terminal	(VARCHAR)	Terminal	(VARCHAR)		
		Ts	(TIMESTAMP)	Ts	(TIMESTAMP)		
				CodigoSap	(Integer)		



## Introducción

Este trabajo práctico propone crear un datawarehouse y sus datamarts asociados, para el almacenamiento de los cambios históricos en el stock de la empresa de retail “Farispley”.

Para ello crearemos 4 bases de datos:

- **farispley\_stock**: Base de datos de uso diario por la empresa
- **farispley\_stock\_dw**: Datawarehouse diseñado utilizando la metodología Vault
- **farispley\_dm\_proveedores**: Datamart aislado para proveedores
- **farispley\_dm\_scanning**: Datamart para la información de scanning de productos.

## Repositorio

Para la practicidad y almacenamiento del código construido para este ejercicio, se ha creado un repositorio en github, al que puedes acceder para ejecutar este trabajo por completo:

 [Link al repositorio](#)

## Dependencias

Para poder realizar la ejecución de este ejercicio, debes tener instalado docker y docker-compose en tu ordenador.

También se recomienda utilizar una herramienta como DBeaver para la gestión de la base de datos.

Si no tienes DBeaver instalado, puedes utilizar "**Adminer**" que se ejecuta junto con la base de datos MySQL y te permite realizar consultas SQL directamente desde el navegador.

Para más información acerca de:

 [MySQL en Docker](#)

 [DBeaver](#)

 [Adminer](#)

## Cómo empezar

Descargue el repositorio ejecutando:

```
$ git clone https://github.com/hhvergara/datawarehouse_model.git
```

Para inicializar este repositorio de práctica, simplemente debe ejecutar el archivo **docker-compose.yml** haciendo lo siguiente:

```
$ docker-compose up
```



Una vez que inicie el servidor, abra su navegador web y vaya a <http://localhost:8080> y podrá ver la página de inicio de "Adminer":

The screenshot shows the Adminer 4.8.0 login interface. It includes a language dropdown set to 'Español', a 'Login' button, and a form with fields for 'Motor de base de datos' (MySQL), 'Servidor' (db), 'Usuario' (admin), 'Contraseña' (masked), and 'Base de datos' (default). There are 'Login' and 'Guardar contraseña' checkboxes at the bottom.

Use el usuario "root" y las credenciales almacenadas en el archivo .env para iniciar sesión en Adminer:

The screenshot shows the Adminer 4.8.0.1 interface after logging in as root. It displays the 'Select database' page with a table of available databases. The table has columns for Database, Collation, Tables, and Size. The 'default' database is selected.

Database - Refresh	Collation	Tables	Size - Compute
<input type="checkbox"/> default	utf8mb4_0900_ai_ci	?	?
<input type="checkbox"/> information_schema	utf8mb3_general_ci	?	?
<input type="checkbox"/> mysql	utf8mb4_0900_ai_ci	?	?
<input type="checkbox"/> performance_schema	utf8mb4_0900_ai_ci	?	?
<input type="checkbox"/> sys	utf8mb4_0900_ai_ci	?	?

## Creación de bases de datos y scripts SQL

La carpeta **db\_scripts** almacena todos los scripts SQL necesarios para crear y realizar operaciones básicas en la base de datos, tales como:

- Creación de la base de datos
- Creación de entidades para cada base de datos
- Consultas de inserción para completar las tablas

De manera predeterminada, no se crean bases de datos para el ejercicio, debe crearlas utilizando el Archivo **00\_database\_creation.sql**

Diríjase a la pestaña "SQL command" en Adminer e importe el archivo:

The screenshot shows the Adminer 4.8.0.1 'SQL command' page. It features a large text area for entering SQL commands, an 'Execute' button, and options to 'Limit rows', 'Stop on error', and 'Show only errors'.



← → ↻ 🏠 ⓘ localhost:8080/?server=db&username=root&import=

Language: English MySQL » db » Import

Adminer 4.8.0 4.8.1

DB:

SQL command **Import** Export

File upload  
SQL[.gz] (< 128MB): Choose Files 00\_database\_creation.sql Execute

From server  
Webserver file adminer.sql[.gz] Run file

☒ Stop on error ☒ Show only errors

Después de la ejecución:

← → ↻ 🏠 ⓘ localhost:8080/?server=db&username=root&import=

Language: English MySQL » db » Import

Adminer 4.8.0 4.8.1

DB:

SQL command **Import** Export

4 queries executed OK. (0.098 s)

File upload  
SQL[.gz] (< 128MB): Choose Files No file chosen Execute

From server  
Webserver file adminer.sql[.gz] Run file

☒ Stop on error ☒ Show only errors

Ahora, vaya a la página de inicio y verá todas las bases de datos de Farispley creadas:

← → ↻ 🏠 ⓘ localhost:8080/?server=db&username=root

Language: English MySQL » db

Adminer 4.8.0 4.8.1

DB:

SQL command Import **Export**

Select database

[Create database](#) [Privileges](#) [Process list](#) [Variables](#) [Status](#)

MySQL version: **9.0.1** through PHP extension **PDO\_MySQL**

Logged as: **root@172.19.0.3**

	Database - Refresh	Collation	Tables	Size - Compute
<input type="checkbox"/>	default	utf8mb4_0900_ai_ci	?	?
<input type="checkbox"/>	farispley_dm_proveedores	utf8mb4_0900_ai_ci	?	?
<input type="checkbox"/>	farispley_dm_scanning	utf8mb4_0900_ai_ci	?	?
<input type="checkbox"/>	farispley_stock	utf8mb4_0900_ai_ci	?	?
<input type="checkbox"/>	farispley_stock_dw	utf8mb4_0900_ai_ci	?	?
<input type="checkbox"/>	information_schema	utf8mb3_general_ci	?	?
<input type="checkbox"/>	mysql	utf8mb4_0900_ai_ci	?	?
<input type="checkbox"/>	performance_schema	utf8mb4_0900_ai_ci	?	?
<input type="checkbox"/>	sys	utf8mb4_0900_ai_ci	?	?

Selected (0)

Drop

## Diseño de entidades

Para este ejercicio, tomamos la información del archivo

“Diccionario-de-datos-API-2(Diccionario API 2).xlsx” que guardamos en la carpeta

**data\_model** en el repositorio.

A partir de esta información creamos el diseño de las entidades para las distintas bases.



## Lineamientos de diseño

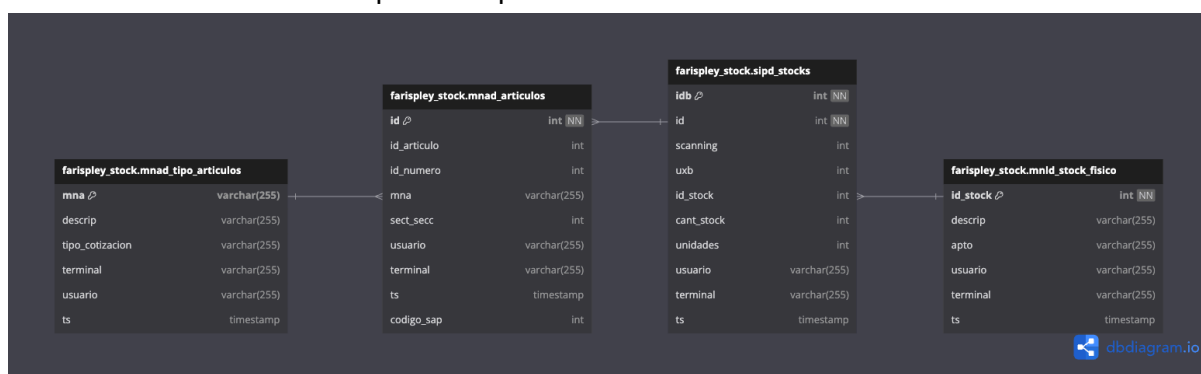
Para todos los casos se utilizará snake\_case para los nombres de las tablas y sus campos, en excepción de las tablas que responden a la metodología Vault en donde se utilizará la primera letra en mayúscula para indicar si las tablas son Hubs, Links o Sats.

## ERDs

A continuación listamos los diseños propuestos para los diagramas de entidad relación (ERDs)

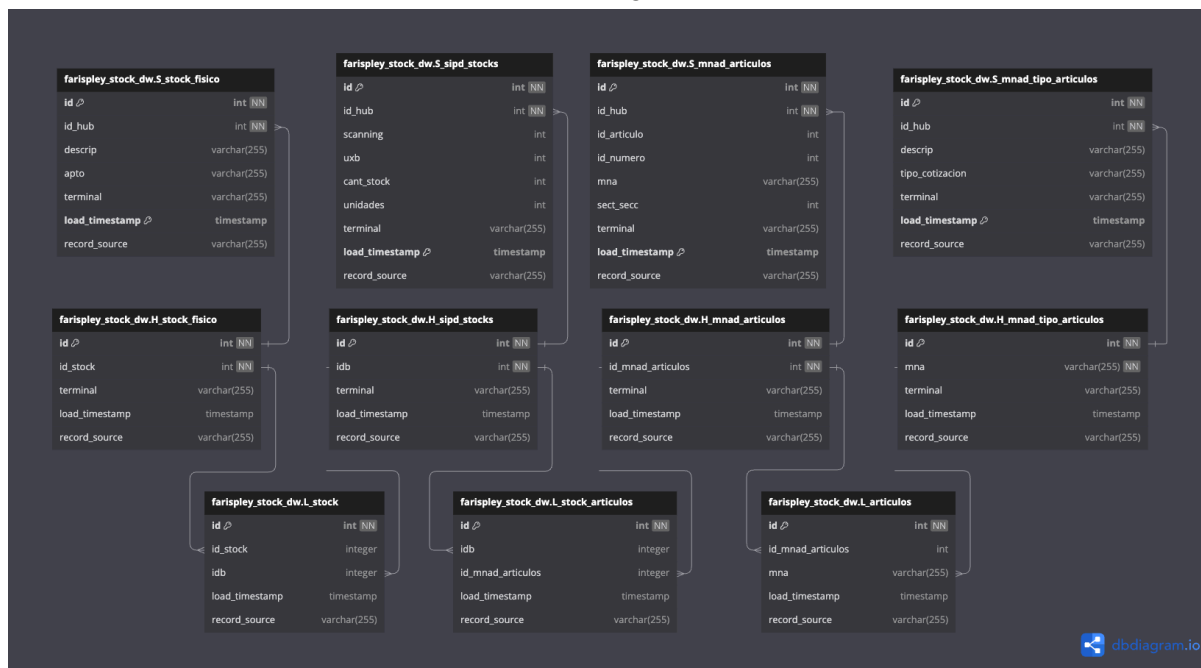
### farispley\_stock

Base de datos de uso diario por la empresa.



### farispley\_stock\_dw

Datawarehouse diseñado utilizando la metodología Vault

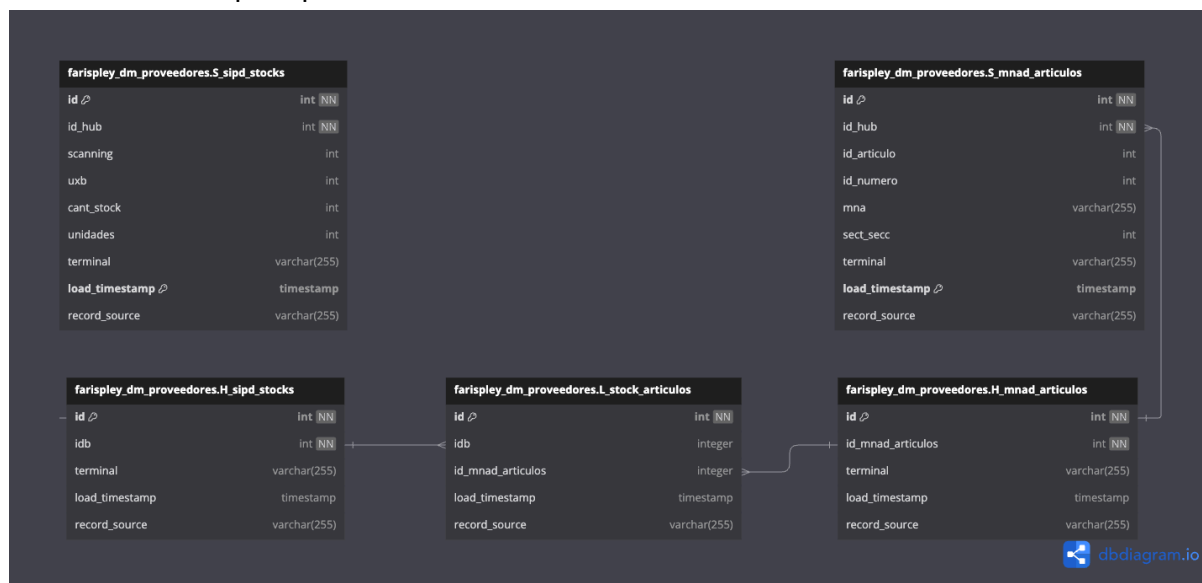


Este datawarehouse fue diseñado en la API2, para más detalle consulte el informe de API2 [aquí](#)



## farispley\_dm\_proveedores

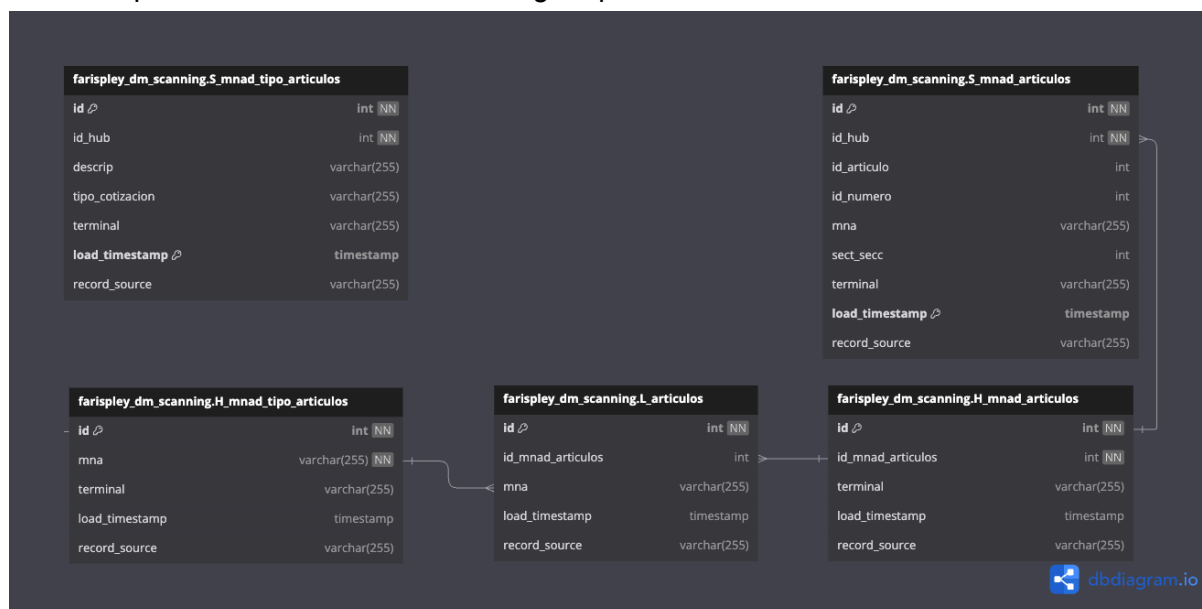
Datamart aislado para proveedores



Para el diseño de este datamart consideramos que la información útil para los proveedores incluye la de cantidad de stock por producto en un tiempo dado, por ejemplo el stock resultante para el día X, con sus datos y la sucursal (campo ubx según la documentación provista).

## farispley\_dm\_scanning

Datamart para la información de scanning de productos.



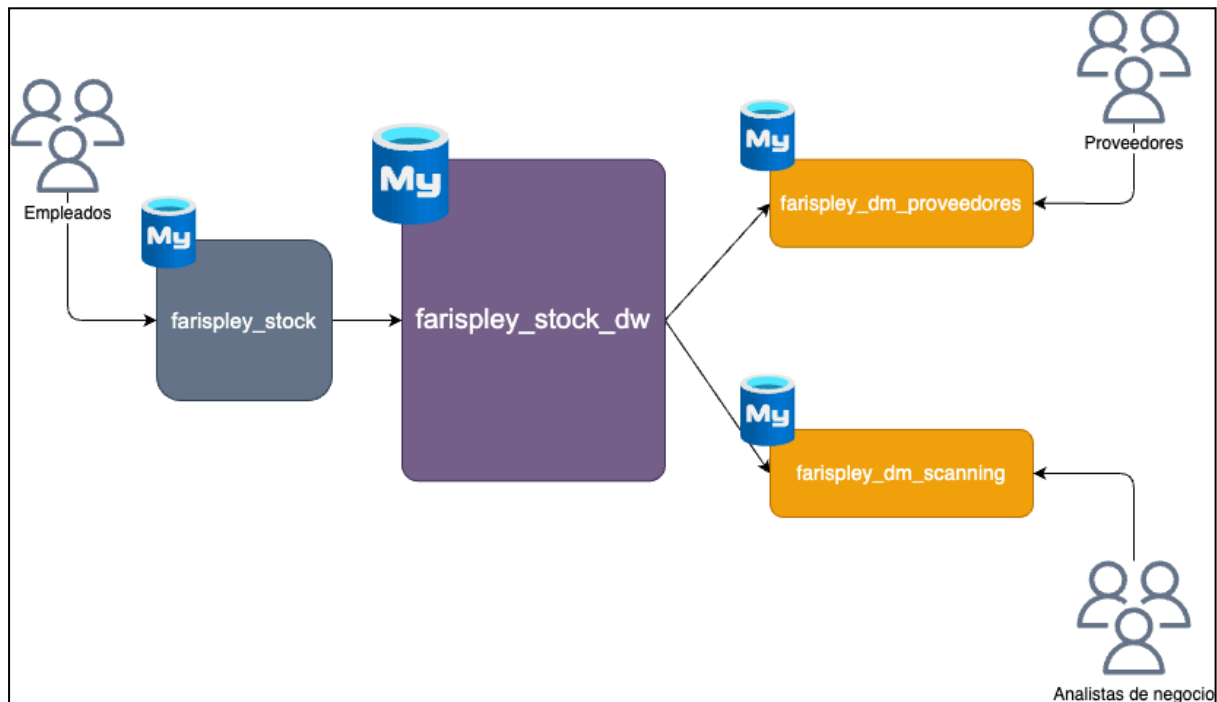
Para el diseño de este datamart lo que intentamos responder a través de los datos es ¿Qué productos fueron introducidos? ¿En qué día? y ¿En qué sector?

Con el objetivo de tener la trazabilidad de la reposición de qué tipo de productos y en qué lugares. Esto permite determinar qué productos fueron más vendidos, en qué lugar, o por ejemplo ver la canibalización entre productos.



## Diagrama de arquitectura de alto nivel

Aquí presentamos el diagrama de alto nivel del flujo de los datos:



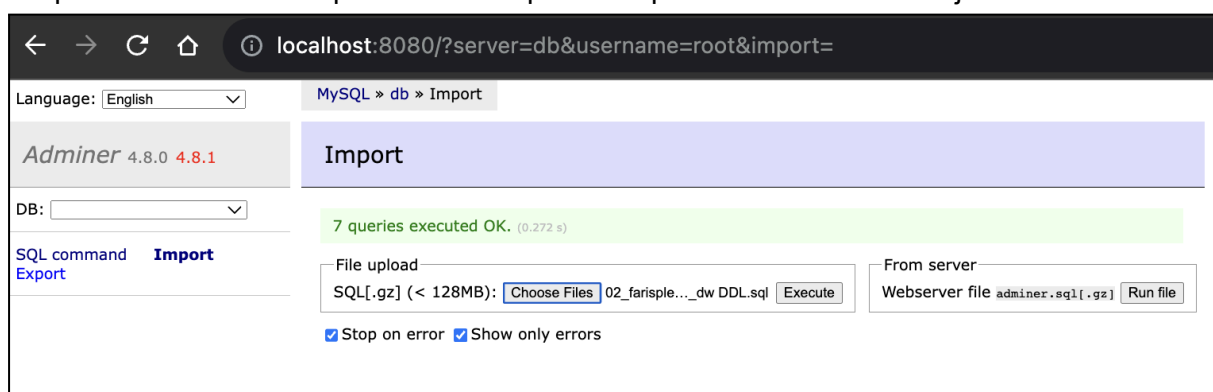
## Creación de las tablas (stock y DW)

Para crear las tablas en las bases de datos de stock diario y nuestro datawarehouse, preparamos una serie de queries que nos ayudaran con este propósito, estas queries son:

- [01 farispley\\_stock DDL.sql](#)
- [02 farispley\\_stock\\_dw DDL.sql](#)

La ejecución de estos scripts pueden realizarse utilizando Adminer, DBeaver o cualquier otro administrador de bases de datos compatible con MySQL.

Simplemente vamos a la pantalla de "Import" e importamos el archivo a ejecutar:



Al terminar, veremos las tablas creadas si seleccionamos la base de datos:





← → ↻ 🏠 ⓘ localhost:8080/?server=db&username=root&db=farispley\_stock

Language: English MySQL » db » Database: farispley\_stock

**Adminer 4.8.0 4.8.1**

DB: farispley\_stock

SQL command Import Export Create table

select mnad\_articulos  
select mnad\_tipo\_articulos  
select mnld\_stock\_fisico  
select sipd\_stocks

**Database: farispley\_stock**

[Alter database](#) [Database schema](#) [Privileges](#)

**Tables and views**

Search data in tables (4)

<input type="checkbox"/>	Table	Engine?	Collation?	Data Length?	Index Length?	Data Free?	Auto Increment?	Rows?	Comment?
<input type="checkbox"/>	mnad_articulos	InnoDB	utf8mb4_0900_ai_ci	16,384	16,384	0		1	0
<input type="checkbox"/>	mnad_tipo_articulos	InnoDB	utf8mb4_0900_ai_ci	16,384	0	0		0	0
<input type="checkbox"/>	mnld_stock_fisico	InnoDB	utf8mb4_0900_ai_ci	16,384	0	0		0	0
<input type="checkbox"/>	sipd_stocks	InnoDB	utf8mb4_0900_ai_ci	16,384	32,768	0		1	0
	<b>4 in total</b>	InnoDB	utf8mb4_0900_ai_ci	65,536	49,152	0			

Selected (0)

Analyze Optimize Check Repair Truncate Drop

Move to other database: farispley\_stock Move Copy ☐ overwrite

[Create table](#) [Create view](#)

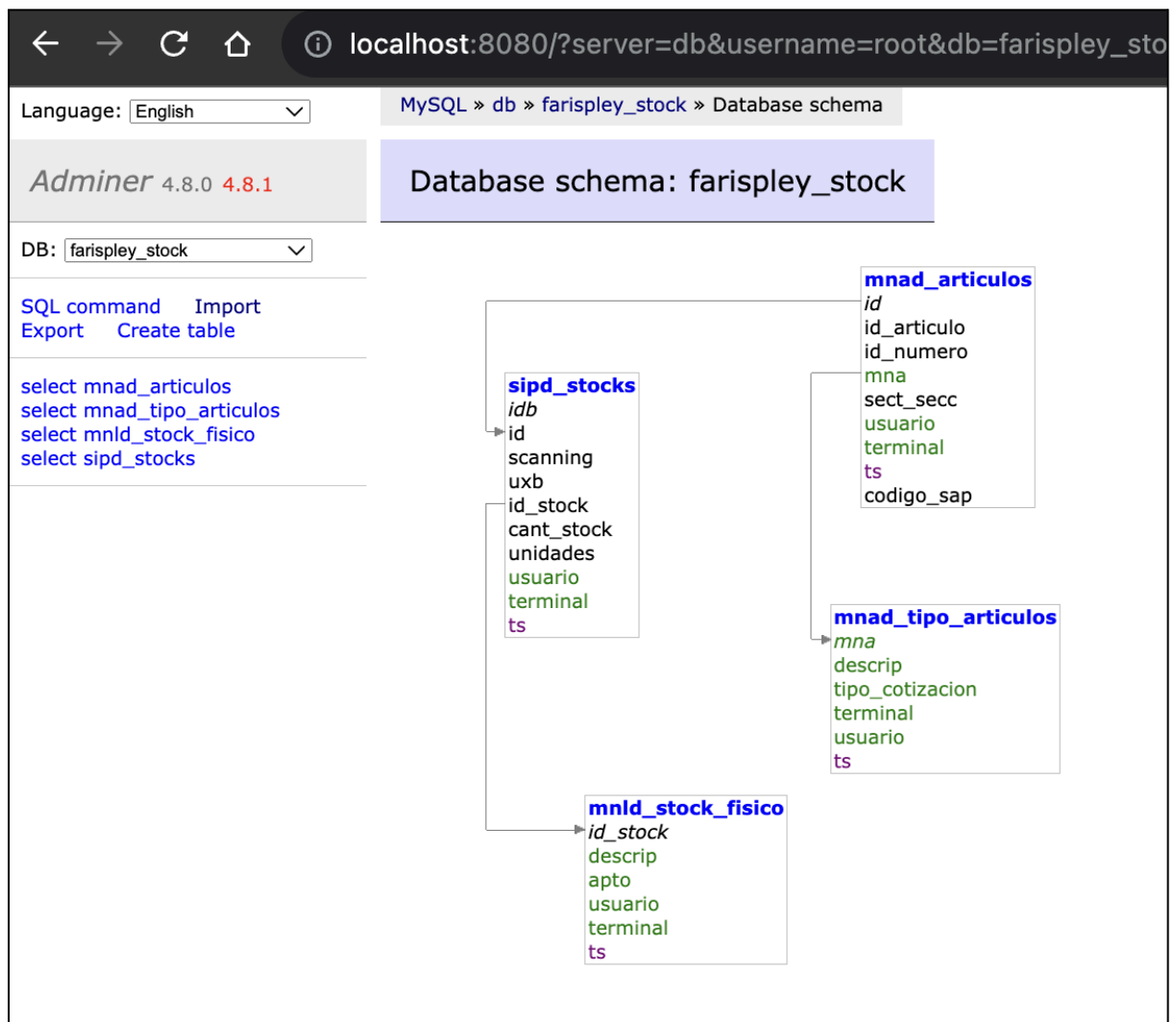
**Routines**

[Create procedure](#) [Create function](#)

**Events**

[Create event](#)

Y podremos ver el ERD si vamos a “Database schema”:





## Inserción de datos de prueba

Para poder trabajar con estas bases de datos, simulamos una cantidad de datos que pueden ser insertados por medio de las queries:

- [03\\_stock\\_db\\_mocked\\_inserts.sql](#)
- [04\\_dw\\_insertion\\_from\\_stock\\_db.sql](#)

Debemos seguir los mismos pasos para importar estos archivos a adminer y ejecutarlos en el orden presentado, primero el archivo 03 y luego el 04. Una vez realizado el import, podremos ver los datos:

The screenshot shows the Adminer web interface. The browser address bar displays 'localhost:8080/?server=db&username=root&db=farispley\_stock\_dw&select='. The interface includes a language dropdown set to 'English', a breadcrumb trail 'MySQL » db » farispley\_stock\_dw » Select: H\_stock\_fisico', and the Adminer version '4.8.0 4.8.1'. The database selected is 'farispley\_stock\_dw'. On the left, there are links for 'SQL command', 'Import', 'Export', and 'Create table'. The main area shows a 'Select data' tab with buttons for 'Select', 'Search', 'Sort', 'Limit' (set to 50), 'Text length' (set to 100), and 'Action' (set to 'Select'). Below these is the SQL query: 'SELECT \* FROM `H\_stock\_fisico` LIMIT 50 (0.001 s)'. The query result is displayed in a table with columns: 'id', 'id\_stock', 'terminal', 'load\_timestamp', and 'record\_source'. The table contains 5 rows of data. At the bottom, there are options for 'Whole result' (5 rows), 'Modify' (Save), 'Selected (0)' (Edit, Clone, Delete), and 'Export (5)'.

	id	id_stock	terminal	load_timestamp	record_source
<input type="checkbox"/> edit	1	1	terminal1	2024-09-18 02:51:30	farispley_stock.mnld_stock_fisico
<input type="checkbox"/> edit	2	2	terminal2	2024-09-18 02:51:30	farispley_stock.mnld_stock_fisico
<input type="checkbox"/> edit	3	3	terminal3	2024-09-18 02:51:30	farispley_stock.mnld_stock_fisico
<input type="checkbox"/> edit	4	4	terminal4	2024-09-18 02:51:30	farispley_stock.mnld_stock_fisico
<input type="checkbox"/> edit	5	5	terminal5	2024-09-18 02:51:30	farispley_stock.mnld_stock_fisico

## Resolución de consignas

### 1. Generar SQL Queries

Para cada script, se ha creado un archivo en el repositorio, dado que son extensos y pueden tomar varias páginas. Se deja el link a cada uno de estos archivos en el repositorio.

Generar SQL Queries compatibles con MySQL para:

- Obtener la foto del stock del día anterior: [LINK](#)

Se puede ejecutar la query desde Adminer utilizando:



Language: English MySQL » db » farispley\_stock\_dw » SQL command

Adminer 4.8.0 4.8.1

DB: farispley\_stock\_dw

SQL command Import Export Create table

```

select H_mnad_articulos
select H_mnad_tipo_articulos
select H_slpd_stocks
select H_stock_fisico
select L_articulos
select L_stock
select L_stock_articulos
select S_mnad_articulos
select S_mnad_tipo_articulos
select S_slpd_stocks
select S_stock_fisico

```

```

-- ** REEMPLAZA LOS CAMPOS DATE PARA FILTRAR POR DIA DESEADO
-- ** REEMPLAZA LOS mna PARA EL OPERADOR "IN" PARA FILTRAR POR ARTICULO,
-- COMENTALO PARA TRAER LA LISTA COMPLETA DE ARTICULOS

SELECT
sma.mna,
ssf.cont_stock AS total_stock
FROM farispley_stock_dw.s_slpd_stocks ssf
LEFT JOIN farispley_stock_dw.s_mnad_articulos sma
ON ssf.id_hub = sma.id_hub
WHERE
DATE(ssf.load_timestamp) = '2024-09-18'
AND sma.mna IN ('mna_01', 'mna_02')
AND ssf.load_timestamp = (
SELECT MAX(s2.load_timestamp)
FROM farispley_stock_dw.s_slpd_stocks s2
WHERE s2.id_hub = ssf.id_hub
AND DATE(s2.load_timestamp) = '2024-09-18'
)
ORDER BY sma.mna

```

Warning: PDO::query(): SQLSTATE[42S02]: Base table or view not found: 1146 Table 'farispley\_stock\_dw.sma' doesn't exist in /var/www/html/adminer.php on line 185

mna	total_stock
mna_01	500
mna_02	300

Warning: PDO::query(): SQLSTATE[42S02]: Base table or view not found: 1146 Table 'farispley\_stock\_dw.ssf' doesn't exist in /var/www/html/adminer.php on line 185

Se puede variar el día de la consulta y los productos que se quieren consultar.

- Crear un datamart aislado para proveedores: [LINK](#)
- Crear un datamart para la información de scanning de productos: [LINK](#)

## 2. Crear las relaciones entre las tablas del DW y DM

Crear las relaciones entre las tablas HUB, SAT y LINK para que los datamarts puedan abstraerse de los conceptos técnicos del modelo anterior y ser funcionales para los usuarios de negocio. Los datamarts deben estar en un esquema aparte para simplificar el acceso y uso por parte de los usuarios de negocio.

Respuesta:

La creación de las queries para la construcción de los datamarts en el punto anterior incluyen las relaciones entre las tablas, tomando por ejemplo el [datamart de proveedores](#)

Para crear estas tablas solo debemos importar los scripts:

- [06 farispley\\_dm\\_proveedores.sql](#)
- [07 farispley\\_dm\\_scanning.sql](#)

Al terminar, podremos ver las tablas creadas:

