

# DESCRIPCIÓN DE TABLAS

ALUMNO: ALAN SHALEM - 42.025.710

CURSO: SQL

COMISIÓN 43425

# CONSIGNA

## DESCRIPCIÓN DE TABLAS

**Formato:** Archivo en **formato .PDF** nombrado como **"Diagrama+Apellido"**.

**Sugerencia:** Si lo publican en su Drive Personal, recuerden compartir el acceso.

Desafío  
entregable



### >> Consigna:

- Armar en formato grilla, los nombres de las tablas de nuestro proyecto final. *(pueden armarlo en una planilla de cálculo y luego exportar su formato a PDF).*

### >> Aspectos a incluir en el entregable:

- Definir las tablas de su proyecto
- Incluir una descripción de cada tabla
- Deben incluir los campos que las componen, detallando:
  - claves primarias
  - claves foráneas
  - campos abreviados
  - nombre del campo completo
  - tipos de datos de cada campo

**CODER HOUSE**

# TABLA USER

user

id	integer	PK
first_name	varchar(64)	
last_name	varchar(64)	
user_name	varchar(64)	
email	varchar(320)	
password	varchar(64)	
confirmation_code	varchar(128)	
datetime_registered	datetime	
datetime_confirmed	datetime	
mobile_no	varchar(45)	
country_id	integer	FK
account_balance	decimal	

La tabla principal es la tabla user. Por cada usuario, almacenaremos:

- *first\_name* y *last\_name* - Nombre y apellido del usuario
- *user\_name* y *password* - El nombre de usuario y password (hash) elegidos por el usuario. El campo *user\_name* puede almacenar solamente valores ÚNICOS.
- *email* - La dirección de email del usuario, la misma sera utilizada para completar el proceso de registro. Solamente puede almacenar valores ÚNICOS
- *confirmation\_code* - El código enviado al usuario para completar el proceso de registro.
- *datetime\_registered* y *datetime\_confirmed* - Fecha y hora cuando el usuario se registro y completo el proceso de registro validando su identidad (KYC)
- *country\_id* - El país donde vive el usuario
- *account\_balance* - El balance de activos del usuario.

# TABLA COUNTRY

country		
id	integer	PK
<hr/>		
name	varchar(45)	

La tabla *country* almacena un diccionario con todos los países relevantes. Además de la PK (id), contiene un atributo que tiene un **ÚNICO** *nombre* de país.

# TABLA PORTFOLIO

portfolio

id		PK
user_id	integer	FK
stock_id	integer	FK
stock_quantity	integer	

Una lista de todos los activos que un usuario tiene se almacena en la tabla portfolio, para cada par ÚNICO de *user\_id* y *stock\_id* guardamos la cantidad que tiene en *stock\_quantity*.

# TABLA ORDER

order		
id		PK
user_id	integer	FK
stock_id	integer	FK
quantity	integer	
price_per_unit	integer	
order_type	char	
timestamp	timestamp	
status_id	integer	FK

Asumimos que cada usuario (trader) puede colocar una orden para comprar o vender activos (stocks) a un determinado precio. Cuando una orden de compra y una orden de venta coinciden, el evento de intercambio ocurre.

Almacenaremos un registro de todas las órdenes en la tabla order.

- user\_id y stock\_id - Referencian al user que metio la orden y el stock que quiere comprar o vender
- quantity - La cantidad que quiere comprar o vender
- price\_per\_unit - El precio deseado de compra/venta
- order\_type - 0 para buy y 1 para sell
- timestamp - El timestamp para cuando se ingreso la orden
- status\_id - El id del status actual de la orden

# TABLA STATUS

status

id	integer	PK
----	---------	----

code	varchar(45)
------	-------------

En la tabla *status*, almacenamos el estado actual de la orden, el mismo lo referenciamos por un *id* y un *code*, pueden ser:

```
status:  
- failed  
- in-progress  
- completely-filled  
- partially-filled
```

# TABLA STOCK

stock		
id	integer	PK
stock_code	varchar(45)	
stock_metadata	varchar(255)	

La tabla stock la utilizamos para guardar la información del stock que el usuario quiere comprar o vender.

- *id* - Referencia al stock que se quiere comprar o vender
- *stock\_code* - Cada stock o acción tiene un código de referencia, también llamado *stock\_symbol*, el mismo es un arreglo de caracteres -generalmente letras- que representa una compañía, un índice, o una moneda públicamente.
- *stock\_metadata* - Contiene datos de ese activo, similar a lo que encontraríamos en la siguiente página:

<https://finance.yahoo.com/quote/AAPL/history?p=AAPL>

**Apple Inc. (AAPL)**  
NasdaqGS - NasdaqGS Real Time Price. Currency in USD [Follow](#)

**172.07** 0.00 (0.00%) **171.13** -0.94 (-0.54%)  
At close: May 16 04:00PM EDT Pre-Market: 8:33AM EDT

[Summary](#) [Chart](#) [Conversations](#) [Statistics](#) [Historical Data](#) [Profile](#) [Financials](#) [Analysis](#) [Options](#) [Holders](#) [Sustainability](#)

Time Period: May 17, 2022 - May 17, 2023 Show: Historical Prices Frequency: Daily [Apply](#)

Currency in USD [Download](#)

Date	Open	High	Low	Close*	Adj Close**	Volume
May 16, 2023	171.99	173.14	171.80	172.07	172.07	42,071,100
May 15, 2023	173.16	173.21	171.47	172.07	172.07	37,266,700
May 12, 2023	173.62	174.06	171.00	172.57	172.57	45,497,800



# TABLA PRICE

price		
id		PK
stock_id	integer	FK
currency_id	integer	FK
buy_price	decimal	
sell_price	decimal	
timestamp	timestamp	

La tabla price hace referencia al precio actual o historico de un activo (stock). Para definir el precio de una accion necesitamos de los siguientes campos:

- *id* - Identificador UNICO del precio en un determinado momento
- *stock\_id* - Hace referencia al activo del cual es el precio
- *currency\_id* - Es el id de la moneda en la cual cotiza el activo
- *buy\_price* - Precio de compra
- *sell\_price* - Precio de venta
- *timestamp* - El timestamp para saber de que momento es el precio

# TABLA CURRENCY

currency		
id	integer	PK
currency_code	varchar(45)	

La tabla *currency* almacena un diccionario con todos los países relevantes. Además de la PK (id), contiene un atributo que tiene un **ÚNICO código** de moneda.

# TABLA MATCHED\_ORDER

matched\_order

id	integer	PK
sale_quantity	integer	
sale_price	decimal	
buyer_id	integer	FK
seller_id	integer	FK
timestamp	timestamp	

Asumimos que cada usuario (trader) que haya colocado una orden previamente puede haber sido matcheada y ejecutada a un determinado precio de venta.

Almacenaremos un registro de todas las órdenes que hayan sido completadas en la tabla *matched\_order*

- *sale\_quantity* - Cantidad de stocks que se compraron/vendieron
- *sale\_price* - Precio al cual se completó la orden
- *buyer\_id* - Id del usuario comprador
- *seller\_id* - Id del usuario vendedor
- *timestamp* - El timestamp para saber en qué momento se completó la orden