

CURSO SQL - CODERHOUSE

- **ALUMNO: ALAN LEONEL SHALEM**
- **DNI: 42025710**
- **PROFESOR: CESAR ARACENA**
- **TUTOR: NANCY VILLENA**
- **COMISION: 43425**
- **FECHA DE ENTREGA:**
- **REPOSITORIO GITHUB:**
<https://github.com/alanshalem/curso-sql>
- **PRIMERA ENTREGA DEL PROYECTO FINAL**

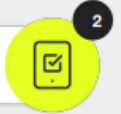
1. CONSIGNA:

PRIMERA ENTREGA DEL PROYECTO FINAL

Formato: Documento PDF con el nombre "Entrega1+Apellido".

Sugerencia: activar la posibilidad de realizar comentarios en el archivo que subis como Entrega.

Proyecto
Final



>> Se debe entregar:

- Descripción de la **temática de la base de datos**
- Diagramas de **entidad relación** de la base de datos
- Listado de las **tablas que comprenden la base de datos**, con descripción de cada tabla, listado de campos, abreviaturas de nombres de campos, nombres completos de campos, tipos de datos, tipo de clave (*foránea, primaria, índice(s)*)

Un archivo **.sql** que contenga:

- Script en SQL de creación de la base de datos y tablas. Este puede estar publicado en un repositorio github, con lo cual el documento pdf debe contener los links de las publicaciones.

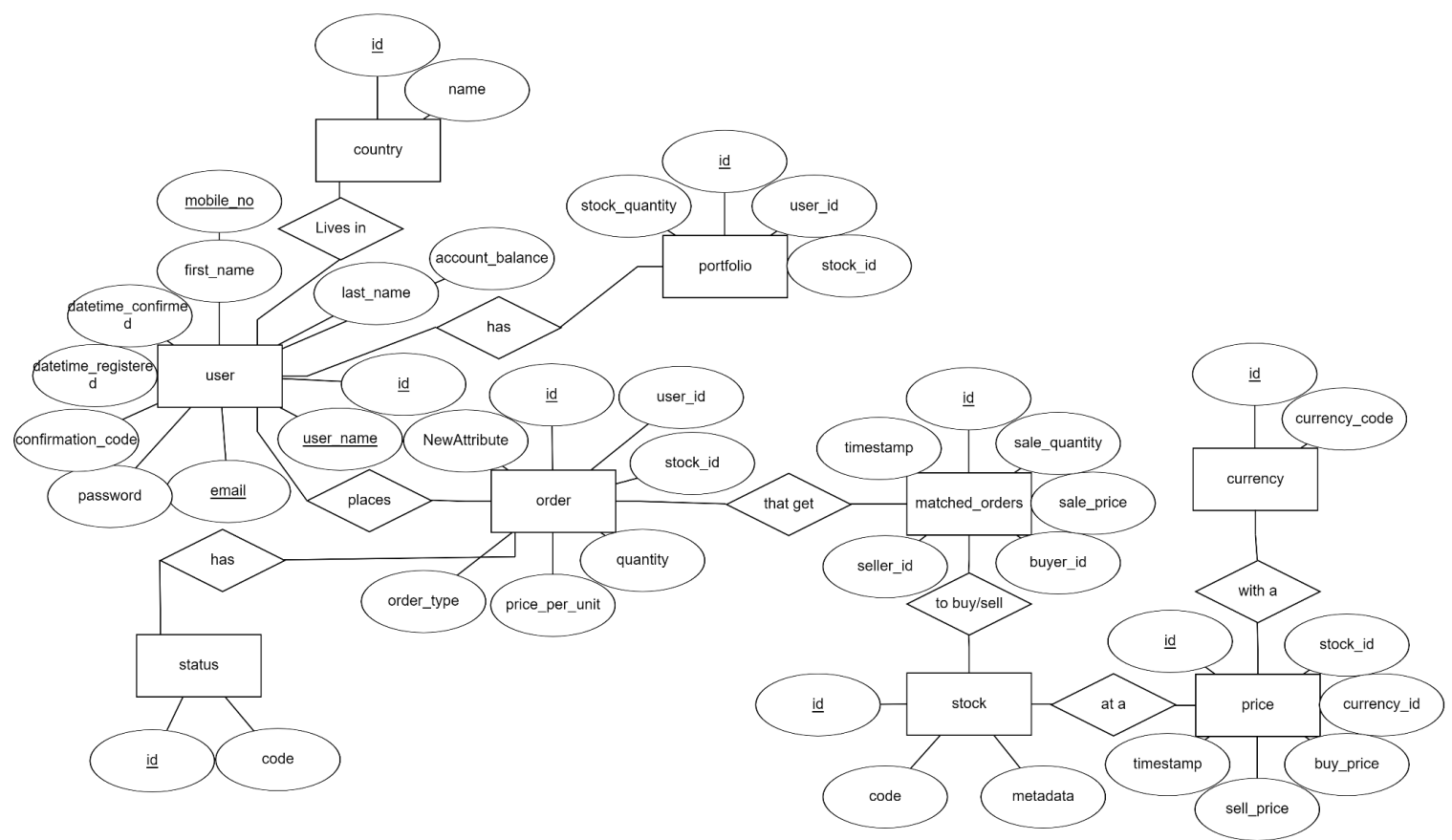
CODER HOUSE

2. DESCRIPCION DE LA TEMATICA DE LA BASE DE DATOS:

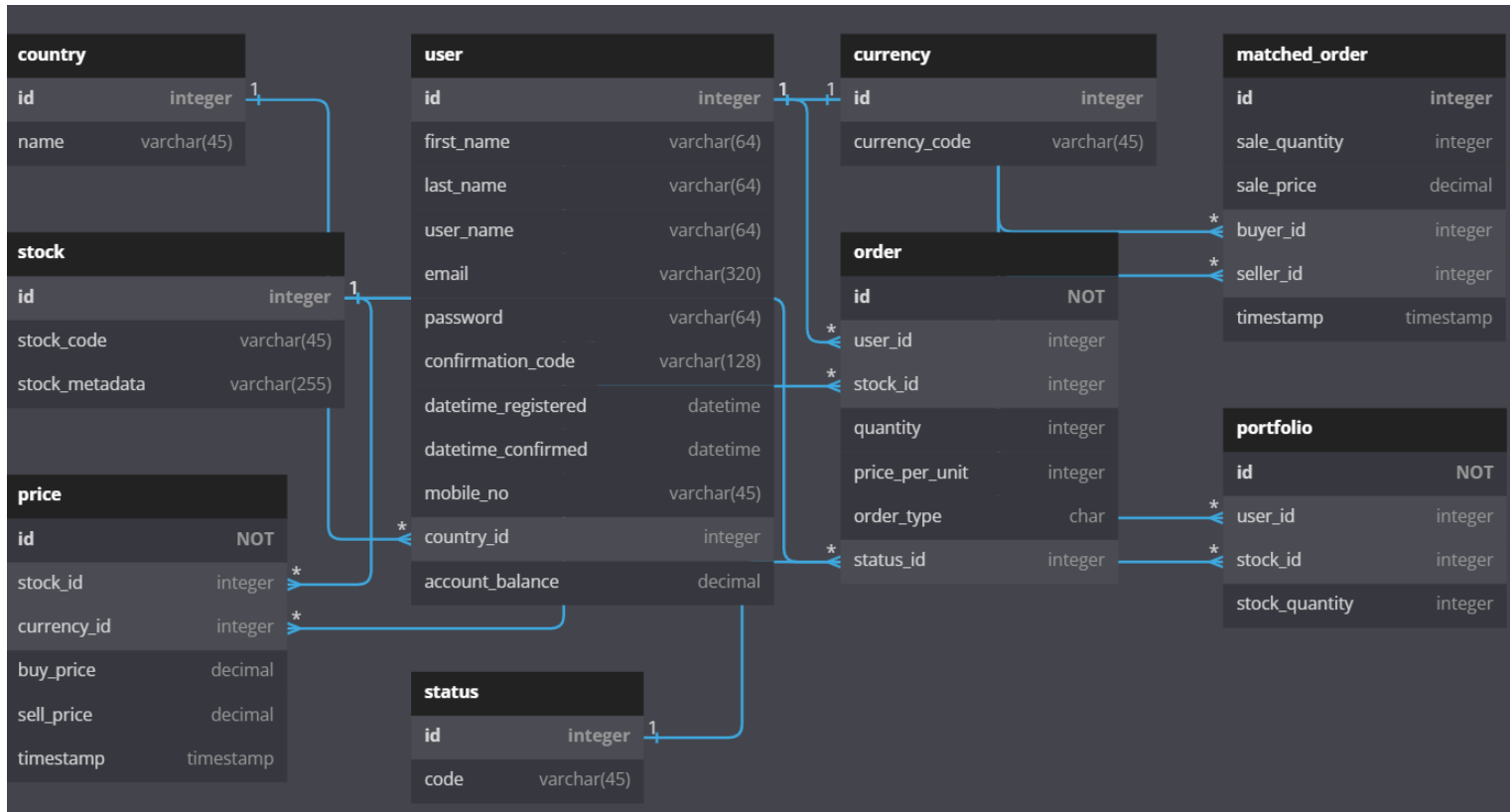
Un broker es una entidad o empresa financiera que ejecuta órdenes de compra y venta; y cobra comisiones por este servicio. Un broker o corredor de bolsa cuenta con licencia para la compra y venta de acciones en los mercados bursátiles. Los traders necesitan de ellos para operar en estos mercados. Una base de datos de un broker online (como puede ser TD Ameritrade o Interactive Brokers) se centra en almacenar y gestionar información relacionada con la operativa y gestion de inversiones financieras. Estas bases de datos son fundamentales para respaldar el funcionamiento del broker y brindar una plataforma segura y confiable para los clientes. Si bien, la implementacion real de cada caso puede variar segun las necesidades y los requisitos de la plataforma. Podemos considerar que todas siguen una estructura de datos claves.

1. Users: La tabla contiene información sobre los usuarios registrados en la plataforma, contiene datos como nombres, información de contacto, credenciales de inicio de sesion y el balance.
2. Portfolio: Esta tabla esta dedicada a la gestion de la cuenta del usuario. Contiene información sobre el saldo de la cuenta, historial de transacciones, actividades comerciales y posiciones de inversion.
3. Price (Market Data): Contiene información sobre los instrumentos financieros en tiempo real y datos historicos como precio, volúmenes de negociacion, datos de nivel 2, etc.
4. Stock: Contiene los SYMBOLS de las acciones que se pueden operar en la plataforma.
5. Country: Contiene los paises de los que puede provenir un usuario.
6. Currency: Contiene las monedas posibles en las que un usuario puede operar
7. Order: Contiene las ordenes que ejecutan los usuarios, estas pueden estar en distintos estados.
8. Status: Contiene los posibles estados en los que pueden estar las ordenes.
9. Matched Order: Contiene las ordenes que efectivamente fueron matcheadas y ejecutadas por el algoritmo de matcheo que tiene el backend.

3. DIAGRAMA ENTIDAD-RELACION DE LA BASE DE DATOS:



4. TABLAS QUE COMPRENDEN LA BASE DE DATOS:



1. Tabla "country":

- "id" (integer): Clave primaria (pk)
- "name" (varchar(45)): Nombre completo

2. Tabla "user":

- "id" (integer): Clave primaria (pk)
- "first_name" (varchar(64)): Nombre completo del usuario
- "last_name" (varchar(64)): Apellido completo del usuario
- "user_name" (varchar(64)): Nombre de usuario
- "email" (varchar(320)): Correo electrónico
- "password" (varchar(64)): Contraseña
- "confirmation_code" (varchar(128)): Código de confirmación
- "datetime_registered" (datetime): Fecha y hora de registro
- "datetime_confirmed" (datetime): Fecha y hora de confirmación
- "mobile_no" (varchar(45)): Número de móvil
- "country_id" (integer): Clave foránea (FK) a la tabla "country"
- "account_balance" (decimal): Saldo de la cuenta

3. Tabla "currency":

- "id" (integer): Clave primaria (pk)
- "currency_code" (varchar(45)): Código de la moneda

4. Tabla "matched_order":

- "id" (integer): Clave primaria (pk)
- "sale_quantity" (integer): Cantidad de venta
- "sale_price" (decimal): Precio de venta
- "buyer_id" (integer): Clave foránea (FK) al usuario comprador en la tabla "user"
- "seller_id" (integer): Clave foránea (FK) al usuario vendedor en la tabla "user"
- "timestamp" (timestamp): Marca de tiempo

5. Tabla "stock":

- "id" (integer): Clave primaria (pk)
- "stock_code" (varchar(45)): Código del stock
- "stock_metadata" (varchar(255)): Metadatos del stock

6. Tabla "status":

- "id" (integer): Clave primaria (pk)
- "code" (varchar(45)): Código del estado

7. Tabla "order":

- "id" (no especifica): No es una clave primaria
- "user_id" (integer): Clave foránea (FK) al usuario en la tabla "user"
- "stock_id" (integer): Clave foránea (FK) al stock en la tabla "stock"
- "quantity" (integer): Cantidad
- "price_per_unit" (integer): Precio por unidad
- "order_type" (char): Tipo de orden
- "status_id" (integer): Clave foránea (FK) al estado en la tabla "status"

8. Tabla "portfolio":

- "id" (no especifica): No es una clave primaria
- "user_id" (integer): Clave foránea (FK) al usuario en la tabla "user"
- "stock_id" (integer): Clave foránea (FK) al stock en la tabla "stock"
- "stock_quantity" (integer): Cantidad de stock

9. Tabla "price":

- "id" (no especifica): No es una clave primaria
- "stock_id" (integer): Clave foránea (FK) al stock en la tabla "stock"
- "currency_id" (integer): Clave foránea (FK) a la moneda en la tabla "currency"
- "buy_price" (decimal): Precio de compra
- "sell_price" (decimal): Precio de venta
- "timestamp" (timestamp): Marca de tiempo