Project: Trade Management System

Objective:
Develop a system to manage and track trades. This will involve the creation and integration of two SQL tables, **trades** and **traders**, and their associated operations through an API.

Database Schema Design:

traders: This table should store information about traders involved in trades.

- trader_id: Primary key for each trader.
- name: Unique name of the trader.

trades: This table should store information about the executed trades.

- trade_id: Primary key for each trade.
- currency_pair: The currency pair involved in the trade (e.g., EURUSD).
- amount: The amount of the trade.
- price: The execution price of the trade.
- trade_date: The date and time when the trade was executed.
- trader_id: Foreign key linking the 'traders' table.
- identifier: Unique key from the execution platform

Each trade is linked to a single trader, but a trader can be associated with multiple trades. Include indexes or any other items that you feel would be beneficial in the design of the tables.

Stored Procedures:

- Insert new traders into the 'traders' table by passing required fields.
- Insert new trades into the 'trades' table by passing required fields.
- Query 'traders' table by passing in name and return all trader table fields.
- Query ‘trades’ table by passing in a trade_id and return all trade table fields.

API (FastAPI or Flask) Integration:

- Create endpoint to add a new trader with stored procedure.
- Create endpoint to add a new trade with stored procedure.
- Create endpoint to query traders with stored procedure.
- Create endpoint to query trades with stored procedure.

Deliverables:

- SQL scripts for creating tables and stored procedures.
- Source code for the API application.
- Documentation and testing instructions.