Database Schema Design

To create `Traders` table:

```
USE TRADING;

CREATE TABLE dbo.Traders(
    trader_id INT IDENTITY(1, 1) PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
);
```

It has two columns:

- `trader_id` which stores the primary key for each trader (type `int`)
- `name` which is the unique name of each trader (type `varchar`)

We insert each trader like so:

```
USE TRADING;

INSERT INTO dbo.Traders(name)
VALUES
('trader_1'),
('trader_2'),
('trader_3'),
('trader_4'),
('trader_5');
```

SQL will automatically assign each trader with their corresponding `trading_id`.

To create the `Trades` table, we simply import the data.

We must assign a `trade_id` to each of the 100 trades, we use the following query. We also set it as the primary key:

```
ALTER TABLE dbo.Trades
ADD trade_id INT IDENTITY(1,1) PRIMARY KEY
```

The columns have the following data types:

- `currency_pair` (type `nvarchar(50)`)
- `amount` (type `money`)
- `price` (type `decimal(18,4)`)
- `trade_date` (type `datetime2(7)`)
- `trade_name` (type `nvarchar(50)`)
- `identifier`

Here is the final schema as discussed in the specifications. Here was have a one-to-many relationship between trades and traders (i.e. one trader per trade, a trader can have multiple trades):