Cryptocurrency Exchange

Fiddy Cent is a digital currency exchange headquartered in Neo Tokyo. They broker exchanges of Bitcoin, Bitcoin Cash, Ethereum, and Litecoin with fiat currencies in around 50 countries.

Help them analyze their January ledger data using SQL aggregate functions! You are given the `transactions` table, which contains both money-in and money-out transactions.

Let's get started!

If you get stuck during this project or would like to see an experienced developer work through it, click "**Get Help**" to see a **project walkthrough video**.

1. Let's start by checking out the whole `transactions` table:

```
SELECT *
FROM transactions;
```

What are the column names?

2. The `money_in` column records the amount (in USD) the user bought. What is the total `money_in` in the table?

```
select sum(money_in)
from transactions;
```

3. The `money_out` column records the amount (in USD) the user sold. What is the total `money_out` in the table?

```
select sum(money_out)
from transactions;
```

4. It was reported that Bitcoin dominates Fiddy Cent's exchange. Let's see if it is true within these dates by answering two questions:
   - How many `money_in` transactions are in this table?
   - How many `money_in` transactions are in this table where 'BIT' is the `currency`?

```
select count(money_in)
from transactions;
```

```
select count(money_in)
from transactions
where currency = 'bit';
```

5. What was the largest transaction in this whole table? Was it `money_in` or `money_out`?

```
select max(money_in)
from transactions;
```

```
select max(money_out)
from transactions;
```

6. What is the average `money_in` in the table for the currency Ethereum ('ETH')?

```
select avg(money_in)
from transactions
where currency = 'ETH';
```

7. Let's build a ledger for the different dates. Select `date`, average `money_in`, and average `money_out` from the table. And group everything by `date`.

```
select date, avg(money_in), avg(money_out)
from transactions
group by 1
order by 1 desc
limit 10;
```

8. To make the previous query easier to read, round the averages to 2 decimal places.
Give the column aliases using `AS` for readability.

```
select date, round(avg(money_in), 2) as avg_income, round(avg(money_out), 2) as
avg_expense
from transactions
group by 1
order by 1 desc
limit 10;
```