```python
from project_functions.connect import Connect
from project_functions.transaction import Transaction
from seaborn import barplot

class DBManager():
    """
    Top level database connection and query manager.
    """

    def __init__(self, vault_path:str):
        """
        Takes a path to the vault file to create a connection and query engine.
        """
        self.vault_path = vault_path
        self.connection = Connect()
        self.connection.connect(self.vault_path)
        self.transaction = Transaction(self.connection)

    def plot_transactions(self, month:str, year:str):
        """
        Plot the transaction amount by date. Utilize data retreived by the transaction class.
        """
        #Use the private method to retreive the transaction data
        transaction_data = self._retreive_transactions(month, year)

        #Group and aggregate data
        transaction_data = transaction_data[["txn_date","amount"]].groupby("txn_date").sum()

        #Create the plot and assign titles
        plot = barplot(transaction_data, x="txn_date", y="amount")
        plot.set_title(label="Sum of Transaction Amount by Date")
        plot.set_ylabel("Sum of Transaction Amount (in USD)")
        plot.set_xlabel("Transaction Date")

    def _retreive_transactions(self, month:str, year:str):
        """
        Take in a month and year value in MM and YYYY format, respectively, and return a dataframe of all transactions in that time frame.

        Invalid queries should return a one row dataframe with -1 for all values.

        This function is a wrapper around the Transaction class, which internally handles verification.
        """

        return self.transaction.retreive_transactions(month, year)
```