

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Feb 14 21:01:34 2026

@author: evebarr20
"""
import pandas as pd

def get_transactions_by_month(engine, month, year):
    """
    Retrieve all transactions for a specified month and year.

    Parameters
    -----
    engine : sqlalchemy.engine.Engine
        Active SQLAlchemy database connection to the bank database.
    month : int
        Month value (1-12) used to filter transactions.
    year : int
        Year value (>= 1800) used to filter transactions.

    Returns
    -----
    pandas.DataFrame
        DataFrame containing all transaction records for the specified
        month and year (excluding account_id). If the month or year
        values are invalid, a DataFrame with the correct columns and
        a single row of -1 values is returned.

    """
    # Validate input month and year values
    if ((0 < month <= 12) and (year >= 1800)):
        # SQL query to retrieve transactions for the given month and year
        # account_id is intentionally excluded
        sql = """SELECT txn_date, txn_id, txn_type_cd, amount, teller_emp_id,
                        execution_branch_id, funds_avail_date, trial104
                FROM transaction
                WHERE EXTRACT(MONTH FROM txn_date) = %s
                AND EXTRACT(YEAR FROM txn_date) = %s
                """
        return pd.read_sql_query(sql, engine, params=(month, year))
    # Handle invalid month/year input by returning a
    # single-row DataFrame filled with -1 values
    print("Invalid month or year")
    data = {
        "txn_date": [-1], "txn_id": [-1], "txn_type_cd": [-1], "amount": [-1],
        "teller_emp_id": [-1], "execution_branch_id": [-1], "funds_avail_date": [-1],
        "trial104": [-1]
    }
    # return dataframe with all columns and one row of -1

```

```
return pd.DataFrame(data)
```