```python
import csv, sqlite3

if __name__ == '__main__':
    # open and read from the cleaned csv file
    infile = open('cleaned_gwas_ancestry.csv', 'r')
    data_reader = csv.reader(infile, delimiter=',')
    # we move the reader to the line after the header (column names) since
    # we are going to create the column names using the sql command below
    _ = next(data_reader, None)

    # connect to the sqlite database
    conn = sqlite3.connect('gwas_associations.db')
    # get the current cursor
    cursor = conn.cursor()

    # run an sql command to remove an already existing ancestry table
    cursor.execute("""DROP TABLE IF EXISTS ancestry;""")

    # create an sql command to create the ancestry table
    create_table_sql_cmd = """
    CREATE TABLE ancestry (
                            PUBMEDID INTEGER,
                            'FIRST AUTHOR' CHAR,
                            CATEGORY VARCHAR,
                            'COUNTRY OF ORIGIN' VARCHAR
                            );
    """

    # execute the above sql command to create an ancestry table
    cursor.execute(create_table_sql_cmd)

    # insert records into the ancestry table
    insert_records_sql_cmd = """
    INSERT INTO ancestry(
                        'PUBMEDID',
                        "FIRST AUTHOR",
                        CATEGORY,
                        "COUNTRY OF ORIGIN"
                        )
                        VALUES (?, ?, ?, ?)
    """
    cursor.executemany(insert_records_sql_cmd, data_reader)

    # query for records as required by exercise 11.2
    sql_query_cmd = """
                    SELECT *
                        FROM association
                    WHERE PUBMEDID IN
                    (SELECT PUBMEDID FROM ancestry
                    WHERE "COUNTRY OF ORIGIN" LIKE "Mexico%")
                    """
    cursor.execute(sql_query_cmd)

    # print some information about the results obtained for the previous query
    col_names = [col[0] for col in cursor.description]
    print(col_names)
    records = cursor.fetchall()

    # print the first 10 records (rows)
    for record in records:
        print(record)

    print("number of rows: {}".format(len(records)))

    # save all changes made to the database
    conn.commit()

```

```
68      # close the database and disconnect
69      conn.close()
70
71      # close the csv file
72      infile.close()
```