Database Design

HARSHIL VARIA

Section 1: Overview of Your Dataset:

➔ **Description OF DataSet:**

The IRS publishes migration data for the US population based on individual tax returns, tracking several key factors year by year. These factors include where people are moving from (their prior state of residency), where they are moving to (their new state of residency), the number of tax returns filed (approximating the number of migrating households), the number of exemptions (approximating the number of individuals). The default dataset Contains 1 Table -> irs_migration_flow with the following fields:-

- Y1: Refers to the first year (the origin of migration).
- Y1_STATE_FIPS: The FIPS code of the state in Y1.
- Y1_STATE_ABBR: The two-letter abbreviation of the state in Y1.
- Y1_STATE_NAME: The name of the state in Y1.
- Y2: Refers to the second year (the destination of migration, always the year following Y1).
- Y2_STATE_FIPS: The FIPS code of the state in Y2.
- Y2_STATE_ABBR: The two-letter abbreviation of the state in Y2.
- Y2_STATE_NAME: The name of the state in Y2.
- NUM_RETURNS: The number of tax returns filed for migration between these states.
- NUM_EXEMPTIONS: The number of exemptions claimed in the tax returns.

As the dataset has approx. 80000 lines, all other tables are taken from this and made compact which can be used to have clear concept and generate information according to our understanding.

➔ **Source for the data:**

Data imported from Kaggle. Available - https://www.kaggle.com/datasets/wumanandpat/irs-migration-data-1992-to-2020

NOTE: the dataset used has

License

ODC Public Domain Dedication and Licence (PDDL)

This can allow me to use data freely - The Rightsholder realises that once these rights are relinquished, that the Rightsholder has no further rights in Copyright and Database Rights over the Work, and that the Work is free and open for others to Use

➔ **Usage:**

The IRS migration data has been used in past for understanding population migration trends in the United States. Researchers, policymakers, and analysts have leveraged this data to study patterns of migration, analyze demographic shifts, and inform policy decisions. While specific citations or URLs are not provided, the IRS migration data is widely recognized as a valuable resource in demographic and economic research.

➔ **Generation Of Data:**

- Table 1 irs_migration_info (constructed by me data imported from Kaggle)
- Table 2 state_info. (constructed by me data imported from Kaggle)
- Table 3 year_info (constructed by me dataset imported from Kaggle)
- Table 4 region. (constructed by me dataset imported from chatgpt)

```
 prompt-  This is my states_info table CREATE TABLE state_info (
state_fips INT PRIMARY KEY ,
state_abbr VARCHAR(2),
state_name VARCHAR(50),
FOREIGN KEY (state_abbr) REFERENCES region(state_abbr)
); generate all different states in us and thier region as SQL insertion
```

➔ **Plan:**

Example question 1- What were the top states people were migrating to in a specific year?

Example question 2- Did migration patterns change over the years in terms of income levels?

Regional Analysis: Can we identify patterns of migration within specific regions of the country?

Section 2: Description of Your Tables:

However, the raw data on the IRS website displays evolving patterns in data recording, making it challenging to track changes over time. To address this, the current dataset normalizes the record layout, standardizes naming conventions, and consolidates the annual data into a coherent dataset hence modifying the dataset giving us the following tables-

1) **Table: region** stores information about the geographic regions of each state in the United States. It acts as a reference table for the state_info and year tables through foreign key relationships.

   Attributes:
   state_abbr: Two-letter abbreviation of the state.
   state_name: Full name of the state.
   region: Geographical region of the state.
   Primary Key:
   Primary key: state_abbr.
   Foreign Keys:
   No foreign keys.
   Dimensions:
   Rows: 50 (each state in the U.S.).
   Columns: 3 (state_abbr, state_name, region).

2) **Table: year** records economic and population growth data for each state in different years. It has foreign key relationships with the region table to link state information.

   Attributes:
   year: Year representing economic and population growth.
   state_abbr: Two-letter abbreviation of the state.
   gdp_growth: GDP growth rate for the state.
   population_growth: Population growth rate for the state.
   Primary Key:
   Composite primary key: (year, state_abbr).
   Foreign Keys:
   Foreign key: state_abbr references region(state_abbr).
   Dimensions:
   Rows: 2005 (assuming one entry per state per year, but this might need clarification from the dataset).
   Columns: 4 (year, state_abbr, gdp_growth, population_growth).

3) **Table: state_info** stores general information about each state, including FIPS code, abbreviation, and name. It is connected to the region table to associate each state with its respective region.

Attributes:

state_fips: FIPS code of the state.
state_abbr: Two-letter abbreviation of the state.
state_name: Full name of the state.

Primary Key:

Primary key: state_fips.

Foreign Keys:

Foreign key: state_abbr references region(state_abbr).

Dimensions:

Rows: 52 (number of states including Washington, D.C.).
Columns: 3 (state_fips, state_abbr, state_name).

4) **Table: irs_migration_info** contains data about migration between states, including the source and destination states, along with associated migration statistics.

Attributes:

year: Year representing migration data.
source_state_fips: FIPS code of the source state.
destination_state_fips: FIPS code of the destination state.
num_return: Number of returns.
num_exemption: Number of exemptions.

Primary Key:

Composite primary key: (year, source_state_fips, num_return).

Foreign Keys:

Foreign keys:
year references year(year).
source_state_fips references state_info(state_fips).
destination_state_fips references state_info(state_fips).

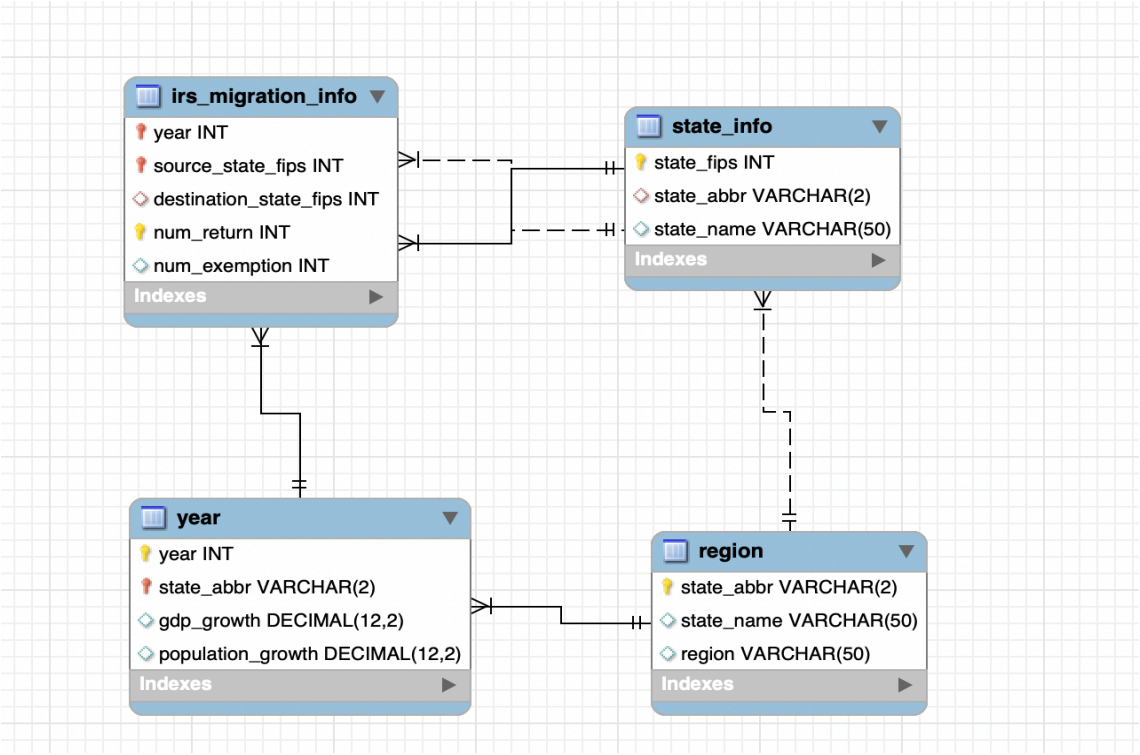
Dimensions:

Rows: 23840 (assuming one entry per migration record, but this might need clarification from the dataset).
Columns: 5 (year, source_state_fips, destination_state_fips, num_return, num_exemption).

Section 3: Internal Schema and Normalization:

The internal schema for the normalized database comprises four interconnected tables. The "year" table acts as the primary repository for information related to different years, encompassing attributes such as the year itself, GDP growth, and population growth with year and state abbreviation as the primary key. The "state_info" table holds details about states, including a unique state identifier (state_fips), state abbreviation (state_abbr), and full state name (state_name).

Additionally, table "irs_migration_info," capture migration data. Both tables a foreign key relationship with the "year" table, connecting migration data to specific years. Moreover, it is a reference to the "state_info" table to link source and destination states with their respective FIPS codes. Lastly region table serves the function to include regional information of a particular state with state abbreviation as primary key and it is a refrence to state,year (state_abbr) tables. This schema is designed to efficiently organize and represent data concerning yearly, state-specific, and migration-related information, is facilitating a comprehensive analysis within a normalized relational database structure



Dependency diagram:

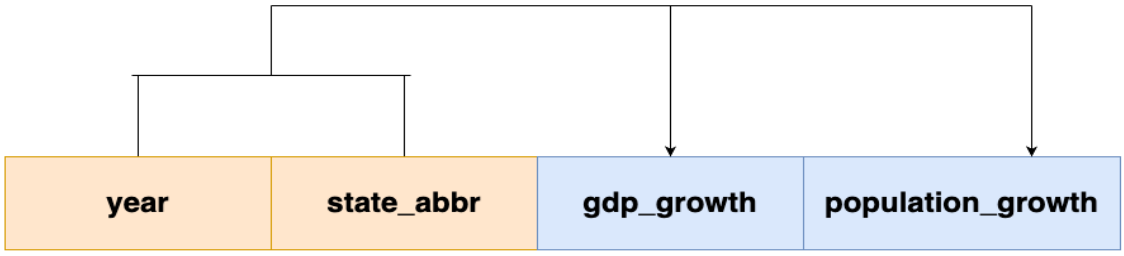


Table Name : Year

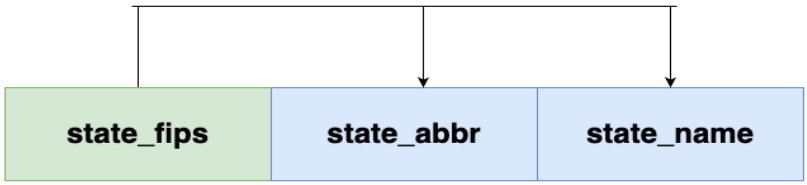


Table Name : state_info

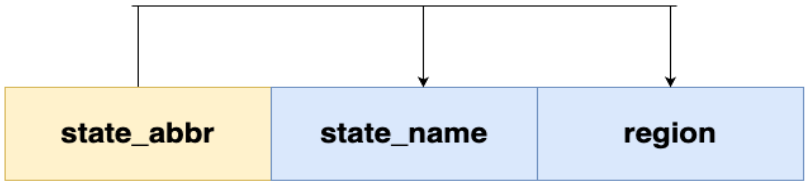


Table Name : region

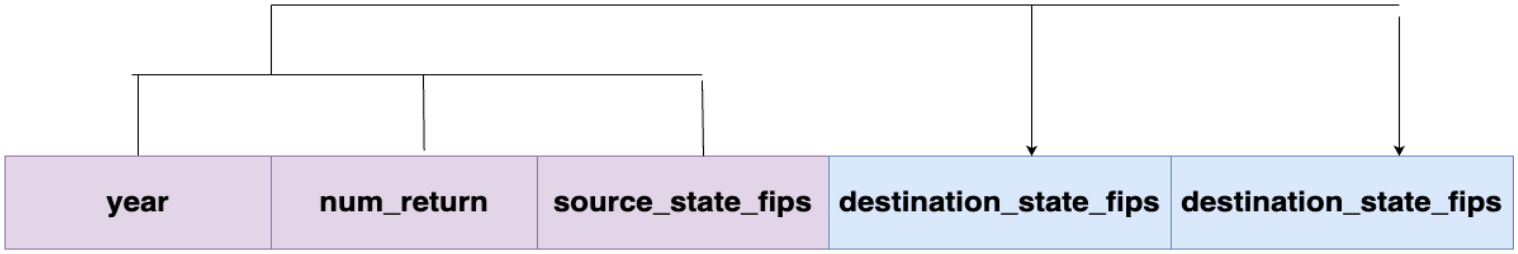


Table Name : irs_migration_info

Each Table is in 3NF as

1: There are no duplicate rows in table. (1NF)
2: All attributes have single values and no partial dependency (2NF)
3. All transitive independent that is no non-prime attribute is dependent on other non-prime attribute
4. Every non-prime attribute is fully dependent on the primary key.

Section 4: Documentation

Data Source: The database integrates information related to economic and demographic aspects of U.S. states. It includes data on GDP growth, population growth, and migration patterns.

License Information:

Data imported from Kaggle. Available - https://www.kaggle.com/datasets/wumanandpat/irs-migration-data-1992-to-2020

NOTE: the dataset used has

License

ODC Public Domain Dedication and Licence (PDDL)

This can allow me to use data freely - The Rightsholder realises that once these rights are relinquished, that the Rightsholder has no further rights in Copyright and Database Rights over the Work, and that the Work is free and open for others to Use

Number of Tables And Number of Attributes:

1) Region Table: 3 attributes
2) Year Table: 4 attributes
3) State_Info Table: 3 attributes
4) IRS_Migration_Info Table: 5 attributes

Business Rules:

Rule 1: Non-Negative GDP Growth and Population Growth
Table Constraint: CHECK constraints on the gdp_growth and population_growth columns in the year table to ensure values are non-negative.

Rule 2: Primary Key Uniqueness Table Constraint:
Primary key constraints on state_abbr in the region and year tables to enforce uniqueness.

Rule 3: Valid Foreign Key References Table Constraint:
Foreign key constraints in the year and irs_migration_info tables referencing state_abbr and state_info(state_fips) to ensure valid references.

Use Of Queries:

Query 1 selects the state abbreviation (state_abbr), GDP growth (gdp_growth), and uses a CASE statement to create the derived attribute growth_category based on different growth ranges.

- ➔ **Categorizing GDP Growth.**
  This query groups states based on their GDP growth for the year 2022 into 'High Growth,' 'Moderate Growth,' or 'Low Growth.'

Query 2 selects the year, GDP growth, population growth, and state name, joining the year and state_info tables on the common state_abbr attribute.

- ➔ **Joining Year and State Information**
  Retrieve combined information from the year and state_info tables.

Query 3 selects the distinct year, state name, and counts the number of returns (num_return) for each state, grouping by year and state.

- ➔ **Counting Total Returns by State and Year**
  Count the total number of returns for each state in each year from the irs_migration_info table.

Query 4 selects state_abbr and total_population

- ➔ **Using Subquery In From Clause**
  uses a subquery to calculate the total population for each state and then selects states with a total population exceeding one million.

Query 5 creates a view named view_state_migration by joining irs_migration_info with state_info twice to get source and destination state names corresponding to state FIPS codes. This view provides a more readable representation of migration data.

- ➔ **Creating a View for State Migration**
  Create a view combining migration data with source and destination state names.

Stored Procedure:

1) UpdateAndInsertProcedure

Input Parameters:

stateAbbreviation (IN): A two-letter abbreviation representing the state for which population growth is to be updated or a new record is to be inserted.
newPopulation (IN): The new population growth rate percentage to be applied, specified as a percentage.
success (OUT): An output parameter that indicates the success of the procedure. It will be set to 1 if the operation is successful.

Usage:

Call the Stored Procedure: Use the CALL statement to invoke the stored procedure.

“ CALL UpdateAndInsertProcedure('CA', 5.0, @success);”

Parameters:
Provide the required input parameters:
stateAbbreviation: Two-letter abbreviation of the state.
newPopulation: New population growth rate in percentage.

Output:
The procedure updates the population growth for the specified state. If the state is not found, it inserts a new record with the provided information. The success flag (@success) is an output parameter that will be set to 1 if the operation is successful.

Transaction Handling:
The procedure is designed to work within a transaction. If the update and insert operations are successful, the transaction is committed. Otherwise, it is rolled back to maintain data consistency

2) DeleteProcedure

Input Parameters:

IN stateAbbreviation VARCHAR(2): Specifies the state abbreviation for which the record should be deleted.

Output Parameters:

OUT success INT: An output parameter indicating the success of the operation. If success is set to 1, the transaction was successful; otherwise, it was rolled back.

Procedure Steps:

The procedure begins by starting a transaction to ensure that the entire sequence of statements either completes successfully or is rolled back in case of an issue. It then performs a DELETE operation on the year table, removing the record where the state_abbr matches the provided stateAbbreviation. The success flag is set to 1 to indicate a successful transaction.

Transaction Handling:
The procedure either commits the changes (if successful) or rolls back the transaction (if unsuccessful).

Usage:
Call the Stored Procedure: Use the CALL statement to invoke the stored procedure.
“ DeleteProcedure('NY', @success);"