# STRV | Prague STRV\_Case\_Study

## Popular articles and sources

<https://www.babycenter.com/baby-names/most-popular/top-baby-names-2014>

<https://www.ssa.gov/oact/babynames/>

## Document verification (MD5 hash) – identical ✅

Current ouput md5 hashes:

MD5 (output/NationalNames.csv) = ea4786e4f7843c53e5de8cfd0ea9879d

MD5 (output/StateNames.csv) = 46d7e87b679bd97f594b6878edfa5705

MD5 (output/database.sqlite) = 4ade7c04b352a570782418970e8bda41

MD5 (output/NationalReadMe.pdf) = 8c0f02e34055160af46669777a1ba14d

MD5 (output/StateReadMe.pdf) = ed4fa3a742e0e0d6284955324f40f6a3

My Output:

MD5 (NationalNames.csv) = ea4786e4f7843c53e5de8cfd0ea9879d

MD5 (StateNames.csv) = 46d7e87b679bd97f594b6878edfa5705

MD5 (database.sqlite) = 4ade7c04b352a570782418970e8bda41

MD5 (NationalReadMe.pdf) = 8c0f02e34055160af46669777a1ba14d

MD5 (StateReadMe.pdf) = ed4fa3a742e0e0d6284955324f40f6a3

## NationalNames vs Sum of StateNames – identical ✅

**Table 1 - StateNames Sophie Sum**

A screenshot of a computer

AI-generated content may be incorrect.**Table 2 - NationalNames Sophie**

A screenshot of a computer

AI-generated content may be incorrect.

## Parents point of view

### **1. Understand What Parents Care About**

Parents often consider the following factors when naming their child:

* **Popularity trends:** Is the name rising or falling in popularity?
* **Uniqueness:** How common or rare is the name? A black background with white text

  AI-generated content may be incorrect.
* **Gender association:** Is the name strongly associated with one gender or unisex?
* **Cultural significance:** Does the name have historical or cultural importance?
* **Regional trends:** How popular is the name in their state or region compared to nationally?
* **Meaning or sentiment:** What does the name signify or represent?

### **2. Create Name Insights and Tools**

You can build features or visualizations to help parents explore and decide on names. Some ideas:

#### **a. Popularity Over Time**

* Show trends for a name over the years (e.g., a line chart showing how popular "Emma" has been nationally or in a specific state).
* Allow users to compare multiple names to see which is trending more strongly.

#### **b. Regional Popularity**

* Provide a map or chart showing the popularity of a name in different states.
* Example: "Is the name ‘Olivia’ more popular in California or Texas?"

#### **c. Name Uniqueness Score**

* A black background with white text

  AI-generated content may be incorrect.Rank names based on how rare or common they are, either nationally or within a state.

#### **d. Gender Neutrality**

* Identify names that are more gender-neutral (e.g., "Jordan," "Taylor") by calculating the ratio of male vs. female babies given that name.
* Provide recommendations for unisex names.

#### **f. Baby Name Explorer**

* Build a tool where parents can input criteria like:
  + Popularity range (e.g., Top 10, Top 100, or below the Top 500).
  + Starting or ending letters (e.g., names starting with "A").
  + Year or decade of origin.
  + Gender preference or unisex names.
* Show matching names with data insights.

### **3. Make Data Visual and Engaging**

Use intuitive, interactive visualizations to present data. Some ideas:

* **Interactive Trend Charts:** Parents can type a name and see its popularity over time.
* **Heat Maps:** Display name popularity geographically.
* **Word Clouds:** Show trending names in different states or years.
* **Personalized Recommendations:** Based on their favorite names, suggest others with similar characteristics.

### **4. Link Data Insights to Startup’s Business Goals**

The startup wants to sell personalized clothing. Use name insights to engage parents and nudge them toward products:

* **Create a Name Preview Tool:** Let parents preview how their chosen name would look on customized baby clothes, like onesies or blankets.
* **Celebrate Rarity:** Highlight rare or unique names with badges like "Exclusive Choice" to make personalized products feel more special.
* **Offer Gift Suggestions:** “Name popularity changes, but this keepsake with [Name] never will!”
* **Run Campaigns for Popular Names:** Promote products like “Top 10 Baby Names of the Year” collections.

## 4 questions to answer:

### **1. How did the name** Ida **change period-over-period nationally?**

#### Steps:

* **Define "period-over-period":** Clarify whether periods are defined as decades, years, or another interval.
* **Query NationalName for 'Ida':**

sql

CopyEdit

SELECT year, SUM(count) AS total\_count

FROM NationalName

WHERE name = 'Ida'

GROUP BY year

ORDER BY year;

* **Analyze trends:**
  + Create a time series plot (e.g., line chart) showing the frequency of "Ida" across years.
  + Compute percentage changes in usage year-over-year or between periods.
    - Percentage change formula: Percentage Change=Countcurrent−CountpreviousCountprevious×100\text{Percentage Change} = \frac{\text{Count}\_{\text{current}} - \text{Count}\_{\text{previous}}}{\text{Count}\_{\text{previous}}} \times 100Percentage Change=Countprevious​Countcurrent​−Countprevious​​×100

### **2. How did the name** Ida **change period-over-period in California?**

#### Steps:

* **Filter data for California:** Query the StateName table for the name "Ida" in California.

sql

CopyEdit

SELECT year, SUM(count) AS total\_count

FROM StateName

WHERE name = 'Ida' AND state = 'CA'

GROUP BY year

ORDER BY year;

* **Perform analysis similar to the national level:**
  + Compare California-specific trends to national trends.
  + Visualize changes (e.g., overlapping line charts showing "Ida" nationally vs. in California).

### **3. What name is the most unisex?**

#### Approach:

A "unisex" name has a nearly equal distribution between genders. This means calculating the gender ratio for each name.

#### Steps:

1. **Group by name and gender, calculate total counts:**

sql

CopyEdit

SELECT name, gender, SUM(count) AS total\_count

FROM NationalName

GROUP BY name, gender;

1. **Pivot data to get counts for both genders side-by-side:**

sql

CopyEdit

SELECT name,

SUM(CASE WHEN gender = 'M' THEN count ELSE 0 END) AS male\_count,

SUM(CASE WHEN gender = 'F' THEN count ELSE 0 END) AS female\_count

FROM NationalName

GROUP BY name;

1. **Calculate gender ratio:**
   * Compute the absolute difference between male and female counts: Ratio Difference=∣male\_count−female\_count∣\text{Ratio Difference} = | \text{male\\_count} - \text{female\\_count} |Ratio Difference=∣male\_count−female\_count∣
   * Rank names with the smallest ratio difference:

sql

CopyEdit

SELECT name, male\_count, female\_count,

ABS(male\_count - female\_count) AS ratio\_difference

FROM (

SELECT name,

SUM(CASE WHEN gender = 'M' THEN count ELSE 0 END) AS male\_count,

SUM(CASE WHEN gender = 'F' THEN count ELSE 0 END) AS female\_count

FROM NationalName

GROUP BY name

) AS subquery

ORDER BY ratio\_difference ASC

LIMIT 1;

### **4. Which names are common nationally but rare at the state level?**

#### Approach:

You’re looking for names that appear frequently in the **NationalName** table but have a low total count in most states in **StateName**.

#### Steps:

1. **Calculate national totals:**

sql

CopyEdit

SELECT name, SUM(count) AS total\_count\_national

FROM NationalName

GROUP BY name;

1. **Calculate state totals:**

sql

CopyEdit

SELECT name, state, SUM(count) AS total\_count\_state

FROM StateName

GROUP BY name, state;

1. **Find names with high national count but low state-level counts:**
   * Calculate an average state count for each name.
   * Compare national total counts to state-level averages:

sql

CopyEdit

SELECT n.name, n.total\_count\_national, AVG(s.total\_count\_state) AS avg\_state\_count

FROM (

SELECT name, SUM(count) AS total\_count\_national

FROM NationalName

GROUP BY name

) AS n

LEFT JOIN (

SELECT name, state, SUM(count) AS total\_count\_state

FROM StateName

GROUP BY name, state

) AS s ON n.name = s.name

GROUP BY n.name

HAVING n.total\_count\_national > 1000 AND AVG(s.total\_count\_state) < 10;

### **General Tips for Approach**

1. **Data Exploration First:**
   * Understand the distribution of names by year, gender, and state.
   * Use visualizations like line charts, bar charts, and scatterplots to spot trends and anomalies.
2. **Tools to Use:**
   * Use SQL for querying the data.
   * Export data into a data analysis tool (e.g., Python with Pandas, R, or Excel) for deeper analysis and visualizations.
3. **Document Insights:**
   * Clearly explain trends, patterns, and your reasoning in a report or presentation.
   * Add visualizations to support your findings.
4. **Validation:**
   * Cross-check results for reasonableness. For example, confirm the accuracy of trends for "Ida" by comparing different datasets (if available).

## My Graphs

1. **Trend of Names State (**Line Graph,X – year and name, Y – count (sum), By – name, gender, year, state**).**

**A screenshot of a graph

AI-generated content may be incorrect.**

1. **Trend of Names Nation (**Line Graph,X – year and name, Y – count (sum), By – name, gender, year, state**).**

**A graph on a white background

AI-generated content may be incorrect.**

1. **Map of Names (**Map Chart, X – longitude and name, Y – latitude, By – state, name**).**

## A map of the united states AI-generated content may be incorrect.

1. **Top Unisex** NatinonalNames by Unisex\_score and Count of Names (Filter - year) and StateNames by Unisex\_score and Count of Names (Filter – year, state).
2. **Top50 NationalNames** (both Genders) by Count of Names (Filter - year) and StateNames (both Genders) by Count of Names (Filter – year, state).
3. **Nationally > State**

**National > State**

SELECT n.name, n.total\_count\_national, AVG(s.total\_count\_state) AS avg\_state\_count, n.year

FROM (

SELECT name, SUM(count) AS total\_count\_national, year

FROM NationalNames

GROUP BY name, year

) AS n

LEFT JOIN (

SELECT name, state, SUM(count) AS total\_count\_state, year

FROM StateNames

GROUP BY name, state, year

) AS s ON n.name = s.name

GROUP BY n.name

--HAVING n.total\_count\_national > 1000 AND AVG(s.total\_count\_state) < 10;

**National > State top 10 nation**

WITH NationalCommonNames AS (

SELECT

Name,

Year,

SUM(Count) AS national\_count

FROM NationalNames

WHERE Year =1960

GROUP BY Name, Year

),

Top10PercentCommon AS (

SELECT

Name,

Year,

national\_count,

round(PERCENT\_RANK() OVER (PARTITION BY Year ORDER BY national\_count DESC), 4) AS rank

FROM NationalCommonNames

),

top10National AS (

SELECT

Name,

Year,

national\_count,

rank

FROM Top10PercentCommon

WHERE rank <=0.1)

select \*

from top10National

**National > State bottom 10 state**

with StateLevelNames AS (

SELECT

Name,

State,

Year,

SUM(Count) AS state\_count

FROM StateNames

WHERE Year =1960

GROUP BY Name, Year, State

),

Bottom10PercentStateRare AS (

SELECT

Name,

State,

Year,

state\_count,

PERCENT\_RANK() OVER (PARTITION BY State, Year ORDER BY state\_count ASC) AS rank

FROM StateLevelNames

),

bottom10State AS (

SELECT

Name,

Year,

State,

state\_count,

rank

FROM Bottom10PercentStateRare

WHERE rank >= 0.9 )

select \*

from bottom10State

## Future enhancements

* Data up from 1910 year and above
* Future prediction of line trend for upcoming years
* Look up to 10 different names at the same time
* Add National > State Name graph

## Streamlit

* Create a venv (.hlib\_strv) to avoid package dependecny issues.
* Create a requirements.txt file with all the libraries and their versions.
* pip install -r requirements.txt.
* Make interactive vizs combining different formats of graphs
* Create SQL views and convert to csv

**SQL**

**DROP VIEW IF EXISTS "main"."Top\_30\_Unisex\_Names\_By\_State";**

**CREATE VIEW Top\_30\_Unisex\_Names\_By\_State AS**

**WITH RankedNames AS (**

**SELECT**

**Name,**

**Year,**

**State,**

**SUM(CASE WHEN Gender = 'M' THEN Count END) AS male\_count,**

**SUM(CASE WHEN Gender = 'F' THEN Count END) AS female\_count,**

**ROUND(**

**CASE**

**WHEN SUM(CASE WHEN Gender = 'M' THEN Count END) = SUM(CASE WHEN Gender = 'F' THEN Count END) THEN 100**

**WHEN SUM(CASE WHEN Gender = 'M' THEN Count END) IS NULL OR SUM(CASE WHEN Gender = 'F' THEN Count END) IS NULL THEN 0**

**WHEN SUM(CASE WHEN Gender = 'M' THEN Count END) < SUM(CASE WHEN Gender = 'F' THEN Count END)**

**THEN (SUM(CASE WHEN Gender = 'M' THEN Count END) \* 100.0 / NULLIF(SUM(CASE WHEN Gender = 'F' THEN Count END), 0))**

**ELSE (SUM(CASE WHEN Gender = 'F' THEN Count END) \* 100.0 / NULLIF(SUM(CASE WHEN Gender = 'M' THEN Count END), 0))**

**END, 2 -- Round to 2 decimal places**

**) AS unisex\_score**

**FROM StateNames**

**WHERE Year >= 1960**

**GROUP BY Name, Year, State**

**),**

**Ranked AS (**

**SELECT**

**ROW\_NUMBER() OVER (PARTITION BY Year, State ORDER BY unisex\_score DESC, (male\_count + female\_count) DESC) AS rank,**

**Name,**

**Year,**

**State,**

**male\_count,**

**female\_count,**

**unisex\_score**

**FROM RankedNames**

**)**

**SELECT \***

**FROM Ranked**

**WHERE rank <= 30 -- Get Top 30 for each year and state**

**ORDER BY Year, State, rank**

**DROP VIEW IF EXISTS "main"."Top\_50\_Unisex\_Names\_National";**

**CREATE VIEW Top\_50\_Unisex\_Names\_National AS**

**WITH RankedNames AS (**

**SELECT**

**Name,**

**Year,**

**SUM(CASE WHEN Gender = 'M' THEN Count END) AS male\_count,**

**SUM(CASE WHEN Gender = 'F' THEN Count END) AS female\_count,**

**ROUND(**

**CASE**

**WHEN SUM(CASE WHEN Gender = 'M' THEN Count END) = SUM(CASE WHEN Gender = 'F' THEN Count END) THEN 100**

**WHEN SUM(CASE WHEN Gender = 'M' THEN Count END) IS NULL OR SUM(CASE WHEN Gender = 'F' THEN Count END) IS NULL THEN 0**

**WHEN SUM(CASE WHEN Gender = 'M' THEN Count END) < SUM(CASE WHEN Gender = 'F' THEN Count END)**

**THEN (SUM(CASE WHEN Gender = 'M' THEN Count END) \* 100.0 / NULLIF(SUM(CASE WHEN Gender = 'F' THEN Count END), 0))**

**ELSE (SUM(CASE WHEN Gender = 'F' THEN Count END) \* 100.0 / NULLIF(SUM(CASE WHEN Gender = 'M' THEN Count END), 0))**

**END, 2 -- Round to 2 decimal places**

**) AS unisex\_score**

**FROM NationalNames**

**WHERE Year >= 1960**

**GROUP BY Name, Year**

**),**

**Ranked AS (**

**SELECT**

**ROW\_NUMBER() OVER (PARTITION BY Year ORDER BY unisex\_score DESC, (male\_count + female\_count) DESC) AS rank,**

**Name,**

**Year,**

**male\_count,**

**female\_count,**

**unisex\_score**

**FROM RankedNames**

**)**

**SELECT \***

**FROM Ranked**

**WHERE rank <= 50 -- Get Top 50 for each year**

**ORDER BY Year, rank**