Projects Data DataFrame as `name_types`

```sql
-- Use this table for the answer to question 1:
-- List the overall top five names in alphabetical order and find out if each name is "Classic" or "Trendy."

-- Select first_name, the sum of babies who have ever had that name, and popularity_type
SELECT first_name, SUM(num),
-- Classify first names as 'Classic' or 'Trendy'
    CASE WHEN COUNT(year) > 50 THEN 'Classic'
        ELSE 'Trendy' END AS popularity_type
FROM baby_names
-- Group by first_name to use aggregate functions
GROUP BY first_name
-- Order the results alphabetically by first_name
ORDER BY first_name
-- Limit to the first 5 names
LIMIT 5;
```

index	first_name	sum	popularity_type
0	Aaliyah	15870	Trendy
1	Aaron	530592	Classic
2	Abigail	338485	Trendy
3	Adam	497293	Trendy
4	Addison	107433	Trendy

Rows: 5 Expand

Projects Data DataFrame as `top_`

```sql
-- Use this table for the answer to question 2:
-- What were the top 20 male names overall, and how did the name Paul rank?

-- RANK names by the sum of babies who have ever had that name (descending), aliasing as name_rank
SELECT
    RANK() OVER(ORDER BY SUM(num) DESC) AS name_rank,
    first_name, SUM(num)
FROM baby_names
-- Filter the data for results where sex equals 'M'
WHERE sex = 'M'
-- Group by first name, oder by rank, and limit to the top 20
GROUP BY first_name
ORDER BY name_rank
LIMIT 20;
```

	n.	firs…	…
0	1	James	47481…
1	2	John	45107…
2	3	Robert	44951…
3	4	Michael	42788…
4	5	William	36144…
5	6	David	35714…
6	7	Richard	24148…
7	8	Joseph	23613…
8	9	Thomas	21668…
9	10	Charles	21123…
10	11	Christopher	20127…
11	12	Daniel	18242…
12	13	Matthew	15672…
13	14	Anthony	13443…
14	15	Donald	12802…
15	16	Mark	12659…

Rows: 20 Expand

Projects Data DataFrame as a_na

```sql
-- Use this table for the answer to question 3:
-- Which female names appeared in both 1920 and 2020?

-- Select first name and total occurrences
SELECT a.first_name, (a.num + b.num) AS total_occurrences
FROM baby_names a
JOIN baby_names b
-- Join on first name
ON a.first_name = b.first_name
-- Filter for the years 1920 and 2020 and sex equals 'F'
WHERE a.year = 1920 AND a.sex = 'F'
AND b.year = 2020 AND b.sex = 'F';
```

	fi...	total_occurre...
0	Emma	20818
1	Evelyn	23283
2	Elizabeth	23125
3	Eleanor	14832
4	Grace	12741
5	Hazel	12765

Rows: 6

Expand