SQL CASE Statement

Basic syntax

CASE WHEN <condition 1> **THEN** <result 1>
WHEN <condition 2> **THEN** <result 2>

-
-
-

ELSE <default result> **END as** <alias name>

Useful for

- Categorizing
- Filtering
- Aggregations

Categorizing – Let's say we want to ***categorize*** the April 2018 unemployment rate for TN counties as low, medium, or high according to a specified business rule:

< 3.0 is low
between 3.0 and 4.0 is medium
> 4.0 is high

```
SELECT DISTINCT county, value,
    CASE WHEN value < 3.0 THEN 'low unemployment'
         WHEN value >= 3.0 and value < 4.0 THEN 'medium unemployment'
         ELSE 'high unemployment' END as level_of_unemployment
FROM unemployment
WHERE year = '2018' and period_name = 'April'
ORDER BY county;
```

	county text	value numeric	level_of_unemployment text
1	Anderson	3.3	medium unemployment
2	Bedford	3.2	medium unemployment
3	Benton	4.5	high unemployment
4	Bledsoe	5.1	high unemployment
5	Blount	2.7	low unemployment
6	Bradley	3.2	medium unemployment
7	Campbell	4.3	high unemployment
8	Cannon	2.7	low unemployment

Filtering – Next we want to look at the FastTrack Job Training Assistance Program (FJTAP) money and the associated new jobs in the Economic and Community Development data (ecd) for Davidson County.

Here we will use a CASE statement in the SELECT statement to ***categorize*** job counts and copy it again to the WHERE clause to ***filter out*** all rows where fjtap is null.

```
SELECT company, landed, new_jobs, fjtap,
  CASE WHEN county = 'Davidson' and fjtap < '$2,000,000' THEN 'small grant'
       WHEN county = 'Davidson' and fjtap >= '$2,000,000' THEN 'large grant' END as grant_size
FROM ecd
WHERE county = 'Davidson' and
  CASE WHEN fjtap < '$2,000,000' THEN 'small grant'
       WHEN fjtap >= '$2,000,000' THEN 'large grant'
       END IS NOT null;
```

	company text	landed date	new_jobs integer	fjtap money	grant_size text
1	Philips North America, Inc.	2017-08-24	815	$7,400,000.00	large grant
2	Parallon Business Solutions, LLC.	2012-09-27	800	$500,000.00	small grant
3	ARAMARK	2013-07-17	1500	$6,000,000.00	large grant
4	Asurion, Inc.	2013-10-16	800	$1,400,000.00	small grant
5	Service Source Delaware, Inc.	2011-12-29	925	$2,405,000.00	large grant
6	HCA, Inc.	2012-02-22	210	$210,000.00	small grant
7	Oberto Brands	2012-11-16	310	$310,000.00	small grant
8	Carlex Glass America, LLC	2011-03-31	50	$1,800,000.00	small grant

Aggregation – next we use a CASE statement in combination with COUNT() to report the number of small grants and large grants for each county.

```
SELECT county,
    COUNT(CASE WHEN fjtap < '$2,000,000' THEN 'small grant' END) as small_grant_count,
    COUNT(CASE WHEN fjtap >= '$2,000,000' THEN 'large grant' END) as large_grant_count
FROM ecd
WHERE fjtap IS NOT null
GROUP BY county
ORDER BY large_grant_count DESC;
```

	county text	small_grant_count bigint	large_grant_count bigint
1	Davidson	68	3
2	Giles	7	2
3	Shelby	62	1
4	Maury	22	1
5	Montgome…	7	1
6	Rutherford	29	1
7	Hamilton	42	1
8	Sumner	27	1

Exercises

1. Using the population table in the **ecd** database, write a query that selects the county, 2017 population, and uses a case statement to characterize the 2017 population (pop_category) according to the following business rule:
   Greater than or equal to 500,000 - *high population*
   Between 100,000 and 500,000 - *medium population*
   Less than or equal to 100,000 - *low population*

2. Using the ecd table in the **ecd** database, write a query that selects the company, landed date, number of new jobs, and a case statement to classify observations (rows) in the table where the project type is 'New Startup' according to the following business rule:
   Fewer than 50 new jobs – *small startup*
   Between 50 and 100 new jobs – *midsize startup*
   More than 100 new jobs – *large startup*

3. Using the population table in the **ecd** database, write a query that uses a case statement to find the total population for 2010 and 2017. Call these Total_Pop_2010 and Total_Pop_2017.