Cohort Analysis with examples in SQL

Owen Price

A cohort is a group of discrete entities which share some characteristics

Owen Price

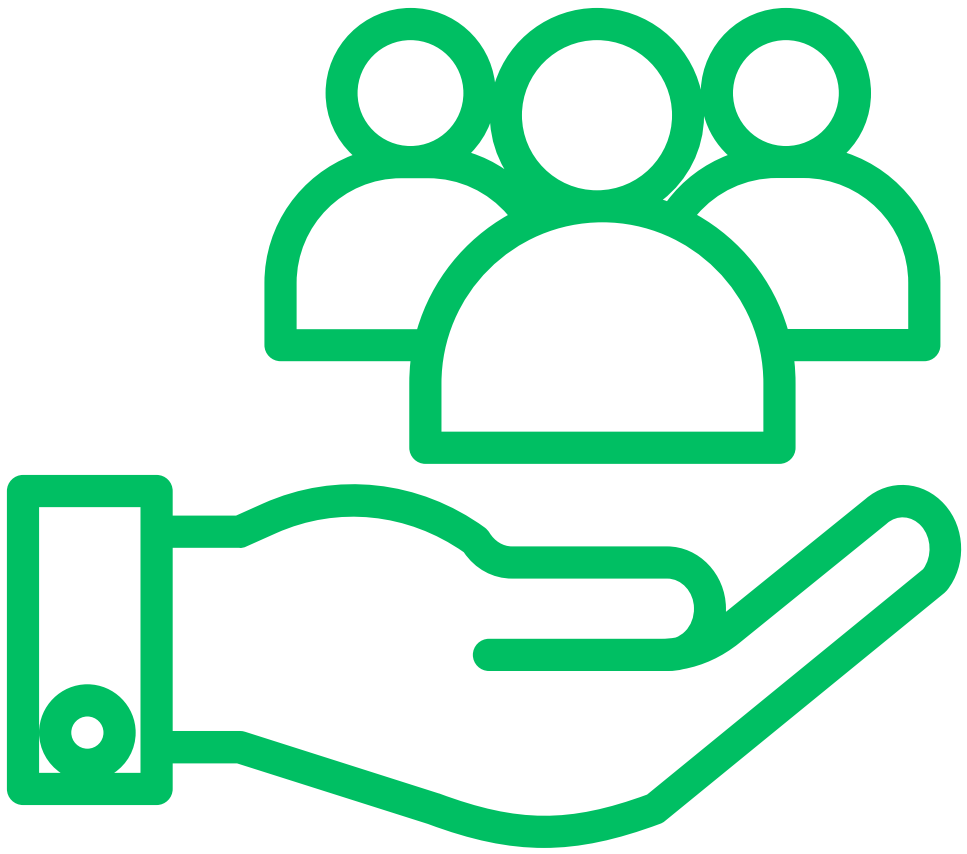
Some examples might be:

- Retail outlets
- Customers
- Airports
- Schools

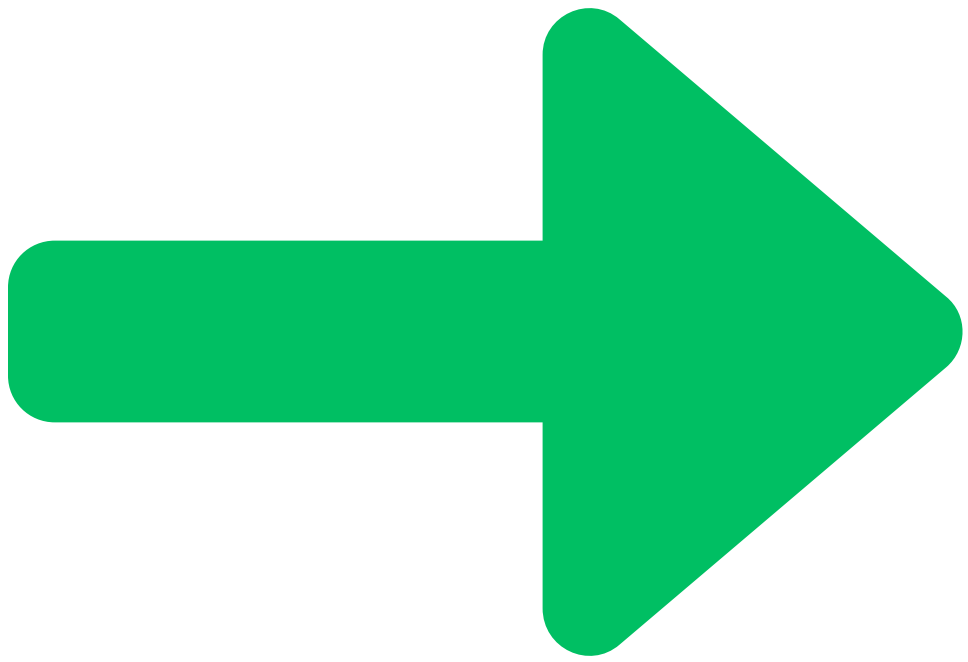
In cohort analysis, we track a cohort over time to uncover behavior and trends related to the shared characteristics of the group

Owen Price

Let's consider customers of a store



We might define one cohort as those customers who first shopped in a store in January of 2012



First purchase 2012-01

```sql
USE ADVENTUREWORKSDW2019;

SELECT CustomerKey
FROM FactInternetSales fis
GROUP BY CustomerKey
HAVING MIN(OrderDate) BETWEEN '2012-01-01' AND '2012-01-31'

SELECT @@ROWCOUNT AS cohort_size;

```

	CustomerKey
1	13986
2	20710
3	21002
4	12575
5	12343
6	12572
7	19173
8	26020

	cohort_size
1	252

Note that we use
HAVING MIN(OrderDate) etc.
to find the first month they purchased

From SQL Server Sample Database
AdventureWorksDW 2019

Owen Price

Tracked over 24 months

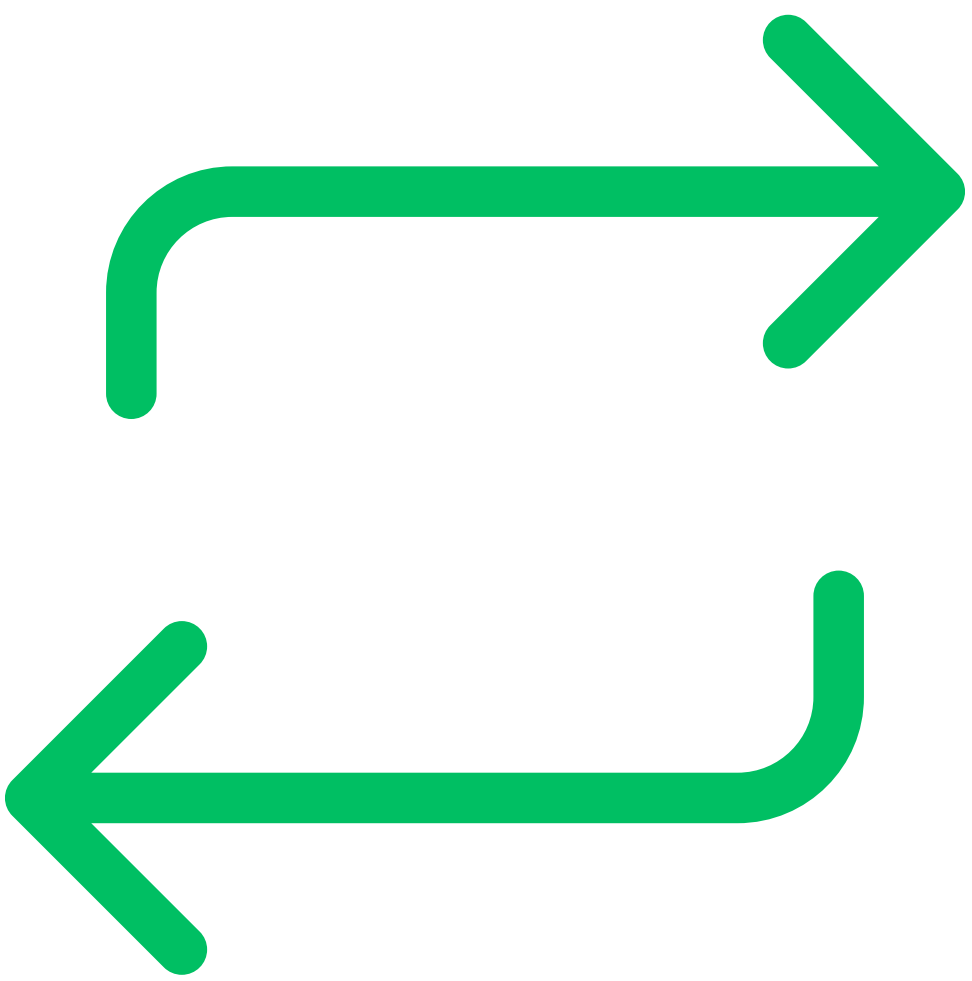
```sql
USE ADVENTUREWORKSDW2019;

DROP TABLE IF EXISTS #cohort;

SELECT CustomerKey
INTO #cohort
FROM FactInternetSales fis
GROUP BY CustomerKey
HAVING MIN(OrderDate) BETWEEN '2012-01-01' AND '2012-01-31';

DROP TABLE IF EXISTS #cohort_sales;

SELECT fis.*
INTO #cohort_sales
FROM FactInternetSales fis
    INNER JOIN #cohort c ON fis.CustomerKey = c.CustomerKey
WHERE fis.OrderDate BETWEEN '2012-01-01' AND '2013-12-31';

SELECT @@ROWCOUNT AS cohort_sales_rowcount;
```

	cohort_sales_ro...
1	1042

This cohort has a total of 1042 sales line items in the 24 month period beginning with the cohort month

Once we have the cohort sales, we can do any analytics we want, perhaps drawing some inference about the cohort in the process

But what if we want to repeat the process with a different month?



2012-02 cohort

```sql
USE ADVENTUREWORKSDW2019;

DECLARE @cohort_size int;

DROP TABLE IF EXISTS #cohort;

SELECT CustomerKey
INTO #cohort
FROM FactInternetSales fis
GROUP BY CustomerKey
HAVING MIN(OrderDate) BETWEEN '2012-02-01' AND '2012-02-28';

SET @cohort_size = @@ROWCOUNT;

DROP TABLE IF EXISTS #cohort_sales;

SELECT fis.*
INTO #cohort_sales
FROM FactInternetSales fis
    INNER JOIN #cohort c ON fis.CustomerKey = c.CustomerKey
WHERE fis.OrderDate BETWEEN '2012-02-01' AND '2014-01-31';

SELECT 'Cohort size' AS stat, @cohort_size AS cohort_size
UNION
SELECT 'Sales rowcount', @@ROWCOUNT AS cohort_sales_rowcount;
```

RESULTS

	stat	cohort_size
1	Cohort size	258
2	Sales rowcount	976

Owen Price

Since this process may be needed at different times in the year, it would be useful to parameterize the inputs

Proc GetCohortSales - 1

We parameterize the components most likely to change

```
4   CREATE PROCEDURE GetCohortSales (@cohort_start_date DATE, @cohort_tracking_months INT)
5   AS
6   BEGIN
7       SET NOCOUNT ON
8
9       DECLARE @cohort_month_end_date DATE
10          = DATEADD(day, -1,
11              DATEADD(month, 1, @cohort_start_date)
12          );
13
14      DECLARE @cohort_end_date DATE
15          = DATEADD(day, -1,
16              DATEADD(month, @cohort_tracking_months, @cohort_start_date)
17          );
```

We calculate the end of the tracking period dynamically from the two parameters

Proc GetCohortSales - 2

This is the same business logic defining the cohort and returning the sales

```
19      DROP TABLE IF EXISTS #cohort;
20
21      SELECT CustomerKey
22      INTO #cohort
23      FROM FactInternetSales fis
24      GROUP BY CustomerKey
25      HAVING MIN(OrderDate)
26          BETWEEN @cohort_start_date AND @cohort_month_end_date;
27
28      SELECT fis.*
29      FROM FactInternetSales fis
30          INNER JOIN #cohort c ON fis.CustomerKey = c.CustomerKey
31      WHERE fis.OrderDate
32          BETWEEN @cohort_start_date AND @cohort_end_date;
33
34      RETURN
35  END
```

Using RETURN on its own will return the result-set of the most recent SELECT

This is an incredibly simple example but it can now be called reliably with a single line of code

```
39  EXECUTE GetCohortSales '2012-01-01', 24;
```

Owen Price

The tricky part about cohort analysis is in the definition of what defines a cohort.

Owen Price

In the preceding example we used:

"Customers whose first purchase was in a given month"

Owen Price

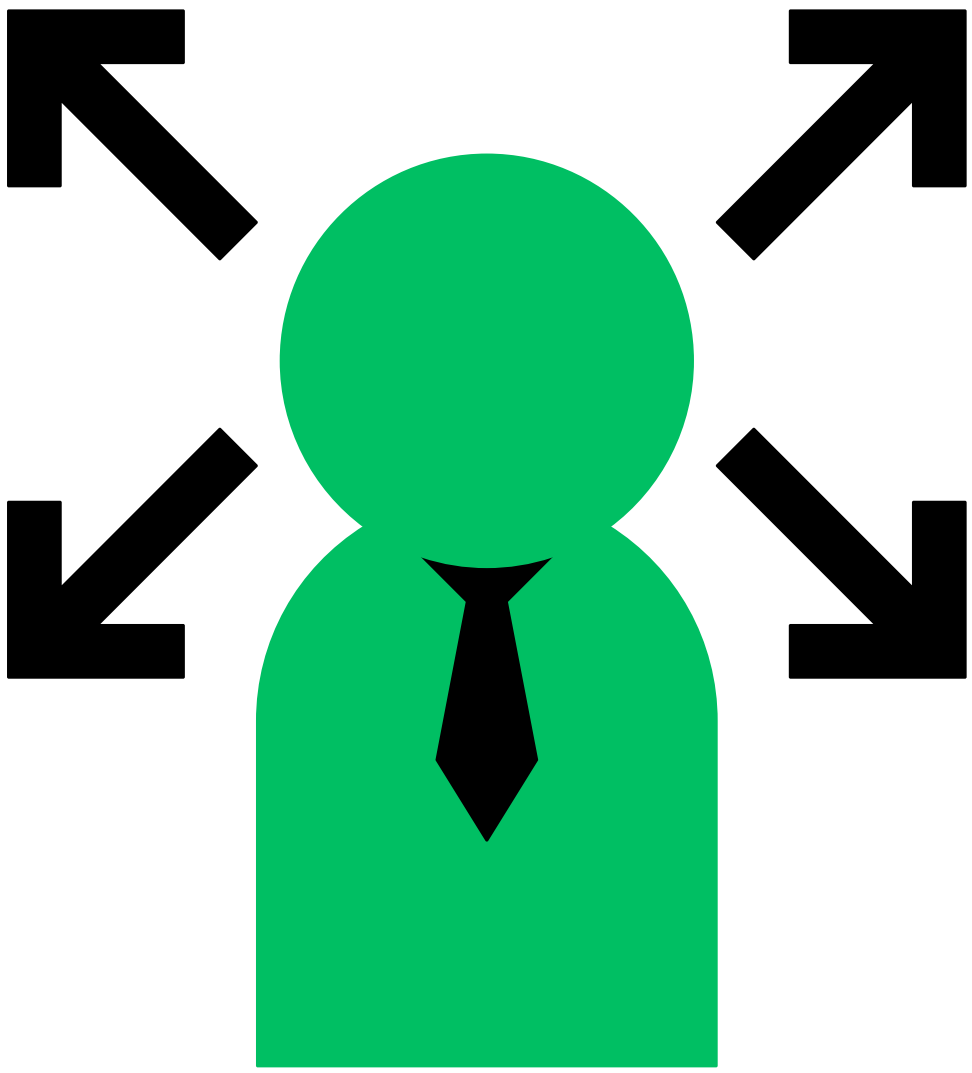
We could also have used:

"Customers whose first purchase was in a given month and who bought Product X within 2 months of their first purchase"

Or even:

"Customers whose first purchase was in the first three months following a given month, who spent at least $1000 in their first 6 months of activity and nothing afterwards"

Ultimately what defines a cohort will be driven by the characteristics of the market



Owen Price

Key takeaways

You will need:

1. *A solid understanding of the structure and availability of data in your database*

2. A good understanding of market behavior - it's usually a good idea to do some exploratory queries befo behavior -

3. A collaborative relationship with your stakeholders - you will likely find that several iterations of cohort definition may be needed. Devise an output infrastructure to test, then compare and constrast the results

Owen Price