**Customer Journey:**

**Everything done to Orderlist in R:**

1. In the Orderlist workbook, there are over 380 separate sheets, and the first step is to bind together all of the sheets that were orders that went OUT so that we have one big dataset that is all of the orders that have gone out.
2. The primary column that is used to get an idea of what product was shipped is the “Accessories/Details” column, which contains the product number. If there is a missing value in the “Accessories/Details” column, it is replaced with the corresponding value in the “PRODUCTS DESCRIPTION” column, and if there is still a missing value, from the “PRODUCTS” column.
3. Sales Order numbers are replaced in the same way.
4. The same process is repeated for all the worksheets that have “in” in the title.
5. When binding rows together, sometimes extra columns are introduced when there is no match in the column names. All the extra columns were checked and removed.
6. Removed rows:
   1. Some extra rows in the worksheets were also introduced, and in the “outs” data, if there was no Sales Order number AND no Source listed under the order, the row was removed because every order has a source and a sales order number. If the order does not have a SO#, there is not a lot of analysis that can be done because there is no way to join it to an order in the “ins.”
   2. In both the “outs” and the “ins,” if the Source column contained “Repair” or the SO# started with “RMA” then the row was removed because we are looking at sales, not repairs/maintenance.
7. A left join was performed on the outs to the ins by SO#, meaning for every order that went out, it was matched by the entry that was placed in the ins. This operation is necessary because we want to match the orders that were shipped out (outs) with the dates from when those orders were ordered (ins).
8. If there was anywhere in the dataset that contained the word “commission” or “refund” that row was removed as well.
9. To take care of any duplicate values from joining the two datasets together, the dataset was grouped by SO# and NET, so that for every unique order, the net revenue is preserved, and the revenue is not inflated by virtue of the left join. If there is an order that has different revenue associated with the same SO#, all those rows will be preserved.
10. Any rows that have missing values for the dates are removed. There were 300 of these, which means that this is probably the biggest potential source of missing data. Some of these are orders that were shipped in early 2011, before which in Orderlist there were no “in” worksheets. Most of these were also parts or tubes that had no SO# associated with it.
11. A few other rows were removed, when the dates had the year as 2029, etc.
12. In order to get an idea of what product is being sold, the first three digits in the “Accessories/Details” column (outs) or the “Products” column (ins) are matched, and then a brand (CID, Felix, Interscan) is associated with the three numbers.
13. The Support database was used in order to get a good match for standardizing the name of the product (i.e. “202” turns into “CI-202 Portable Laser Leaf Area Meter”) and also gets rid of some ambiguity with the F-751s for example.
14. A customer column is introduced that is the “END USER” column, and if there is no end user, replaced with “CUSTOMER’S NAME” column. If the “END USER” column is listed as “same” it is also replaced with “CUSTOMER’S NAME”.
15. A previous customer column is also introduced, which goes through the dataset chronologically, and indicates TRUE if the customer has not been seen before, and FALSE if the name has been seen in the dataset already.

**Queries in Power BI**

Orderlist Revenue

1. Nothing here changed from the R script other than changing datatypes and adding a column for the month and year of the purchase.

Orderlist Unit Sales: Used for the making of the “Sum of Unit Sales” line graph and the Felix Products metric.

1. When changing data type, errors are introduced when converting “NA” values to numeric, which are replaced with 0 in the revenue columns (“ACTUAL SHIPPING”, “NET”, “INVOICE”, etc.)
2. An R script is run here in Power BI to get a better idea of unit sales. Sometimes in the “Accessories/Details” column, there are multiple products shipped for the same Sales Order number, and same NET revenue etc. The rows are duplicated to reflect the number of unit sales, and the duplication is for every time there is a new line character entered, or there are five consecutive spaces, because this is how the product details are split. This step cannot be done before calculating revenue metrics because the NET revenue column is duplicated as well, and there are inconsistencies between the QTY column and the actual number of products that were shipped, so it is hard to get an idea of how much revenue comes from each individual product sale.

Unit Sales Line Graph: Query that was used for making “Sum of Unit Sales” line graph.

1. Duplicate of the Orderlist Unit Sales query. Then, grouping by company (CID, Felix, Interscan) and date (month/year).

ARPA (Monthly): Used for the making of the “Sum of Revenue” line graph.

1. First, duplicate Orderlist Revenue query. Then, group by customer, date (month/year), and company (CID, Felix, Interscan), then take the sum of NET revenue for each group.
2. The missing values in the Company column are removed, which can be a big potential source of error as well.
3. Then, group again by just date and company, taking the sum of revenue from date and company, along with the number of unique customers (accounts) for each month of every year.
4. The Sum of Revenue line graph is made using the sum of revenue for each month of each year.

ARPA (Yearly): Used for the making of the ARPA line graph, corrected from the previous query.

1. First, duplicate Orderlist Revenue query. Then, group by customer, year, and company (CID, Felix, Interscan), then take the sum of NET revenue for each group.
2. The missing values in the Company column are removed, which can be a big potential source of error as well.
3. Then, group again by just year and company, taking the sum of revenue from year and company, along with the number of unique customers (accounts) for each month of every year.
4. The ARPA line chart is made from the sum of revenue divided by the sum of number of accounts.

ARPA (Overall): Used for making the CLV card.

1. First, duplicate Orderlist Revenue query. Then, group by customer and company (CID, Felix, Interscan), then take the sum of NET revenue for each group.
2. The missing values in the Company column are removed, which can be a big potential source of error.
3. Then, group again by just the company, taking the sum of revenue and counting the number of rows. Then, you have three rows for the overall ARPA for CID, Felix, and Interscan.

ACV: Used for the making of the ACV line graph.

1. First, duplicate Orderlist Revenue query. The error values in the NET column when changing data type are replaced with 0.
2. The data is grouped by date (month/year) and the company, while counting the number of rows (number of contracts) and the sum of revenue.
3. The missing values in the Company column are removed, which can be a big potential source of error as well.
4. The ACV bar chart is made from the sum of revenue divided by the sum of the number of contracts.

CLV: Used to make the CLV card.

1. First, duplicate the Orderlist Revenue query. Then, group by customer and company, taking the earliest date that an order was placed and the last date that an order was placed for each customer.
2. Then, take the difference between the last date and the earliest date, which gives the Customer Lifetime in days.
3. The CLV card is made from the measures from ARPA and the average Customer Lifetime to get an overall CLV for the entirety of the company.

All Opportunities:

1. Then, filter for rows that do not contain “Open” in the status column (only considering closed opportunities).
2. Calculate the “Company” column by first using the beginnings of the product column. If the product starts with CI, then it is a CID instrument, and if it starts with F, it is a Felix instrument. If it contains the words, “Halimeter,” “Sensor,” or “GasD,” then it is an Interscan product.

AcquisitionRate: Used for the Acquisition Rate bar chart.

1. Duplicate the All Opportunities query.
2. Then, group by the date (month/year), company (CID, Felix, Interscan), and the status of the opportunity (Won or Lost), and count the rows to get the number of won or lost opportunities by company every month.
3. Pivot the Status column to get two columns, one for won opportunities and one for lost opportunities per month. This makes it easier to put into a Power BI measure.
4. Lastly, the dates are filtered for dates after September 2018, because a lot of the data was transferred to the new database reporting system that month, so the numbers will be very off.

Lead Source (Opportunities): Used to make the Lead Source tables.

1. Query duplicated from All Opportunities. Filtered out missing values in the Products column because for this lead source table we are looking at where the products come from.
2. The data is grouped by date, company, and source campaign where the number of opportunities is counted.
3. Missing values are removed.

Lead Source (Leads): Used to make the Lead Source tables.

1. Run the R script in order to unnest tokens, since there are many leads that have listed multiple products. So, each lead will now reflect one product per lead.
2. Calculate the “Company” column by listing all leads that start with CI- as CID, each lead that starts with F- as Felix, and the rest as Interscan.
3. Group by the date (month/year), company (CID, Felix, Interscan), source campaign, and product. This way there is a lot of drill down potential in the table that results.

Repeat Customers: Used for the Repeat Customer Ratio bar chart.

1. Query duplicated from Orderlist Revenue. The data is grouped by company, date, whether or not they were a previous customer, and the name of the customer. This is so that the customer is only counted once within a month.
2. Then, the data is grouped again by date, company, and whether or not they were a previous customer, and the rows are counted.
3. The indicator column for whether or not the customer was a previous customer was pivoted, the values coming from the count, so there was a column for the number of TRUEs and a column for the number of FALSEs. Nulls were replaced with 0s.
4. A measure was created to calculate the number of TRUEs divided by the total number of unique customers per month.
5. Note: If a customer orders at least one product per month, that will count for one repeat customer every month. However, when aggregated over the years, for every year it will count as a repeat customer 12 times rather than just one. This may inflate the repeat customer metric on the scope of years, but per month it is more accurate.

PublishedPapersTermFrequency: Used for the Published Papers metric on CID dashboard.

1. Imported directly from Published Papers transformations in R.
2. Filtered out certain rows in the CI-710 because of the one-letter instances (G, PI, BI, etc.)

Felix Products: Used for the distribution of Felix products on the market for the Felix dashboard.

1. Query duplicated from Orderlist Unit Sales.
2. The data is grouped by date and product, and the rows are counted, so the result is the distribution of the number of Felix products that were shipped out.

Sensor Express: Used for the Interscan Sensor Express bar chart.

1. Filtered the sensor allocation spreadsheet for all sensors that came with the Sensor Express subscription.
2. Ran an R script that extracts the number of sensors that were ordered for each order in allocation.
3. Ran another R script that unnests tokens to get a better idea of the part numbers that associate with the Sensor Express subscription.
4. Ran another R script to group together the Order numbers and part numbers to get a better calculation for the quantity of part for each order.
5. If there was a null value for quantity, assumed 1.

All Cases:

1. Split the “Case Title” column by the transition between numbers and characters to identify the product for support cases that have no data.
2. Then, replaced all values of the numbers (340, 110, 900, etc) with their product names (CI-340, CI-110, F-900, etc) to create a company column.
3. Filtered rows for those values that had a product.
4. Added column for time until resolution, which is resolution date – create date.

Distribution of Cases: Used for the distribution of support cases bar graphs.

1. Duplicate the All Cases query.
2. Group by the product and the company, then count the number of rows of each group.

Time Until Resolution: Used for the time until resolution bar graphs.

1. Duplicate the All Cases query.
2. Group by the product and the company, then take the average of the time until resolution for each group, resulting in the average time until resolution for each product.

RMA Cases: Used for the ratio of RMA cases bar graphs.

1. Duplicate the All Cases query.
2. Create a new column to determine whether or not the support case is an RMA case by detecting if the string “RMA” appears within the description column or the case title column.
3. Filtering such that the status of the support case is resolved.
4. Group by the RMA indicator (True/False), company (CID, Felix, Interscan), and the product, counting the number of cases in each group.
5. Pivot the RMA column, getting two columns for the number of cases resolved with and without RMA.