

Northlake Therapy — Clinic Performance Report

Build a three-page Power BI report over a multi-site physical therapy clinic, with DAX measures and cross-page drill-through.

Instructions

Open the provided .pbip starter project in Power BI Desktop. The tables and relationships are already in place — build the required visuals and measures on the named report pages. Submit your .pbix. Assignment: Motion Forward Physical Therapy — Insurance Coverage Report Course: Business Intelligence & Data Visualization Total Points: 100 Instructions: 1. Open the .pbip starter file in Power BI Desktop. Do not modify the existing data model relationships unless a task explicitly instructs you to verify or create one. 2. Create the four DAX measures listed in the DAX Measures section before building visuals that depend on them. Define all measures in the patient_insurers table unless otherwise noted. 3. Build each report page exactly as named: 'Patient & Insurer Overview', 'Coverage Timeline Analysis', and 'Drill-Through Patient Policy Detail'. 4. Each page must include at least one slicer as specified in the page instructions. 5. All visuals must have descriptive titles formatted in the Format visual pane (not just the default field name). 6. When finished, save your file as a .pbix and submit via the course portal. 7. Partial credit is awarded for visuals that are partially complete or incorrectly formatted but demonstrate correct intent.

Business Scenario

Motion Forward Physical Therapy operates five outpatient clinics across a metropolitan region, helping patients recover from injuries and manage chronic pain through individualized treatment plans. As the clinic network grows, administrators need clear visibility into which insurers cover their patient population, how long policies have been active, and how enrollment trends shift over time. This report will help the billing and operations teams make informed decisions about insurer partnerships and patient intake planning.

Report Pages

Page 1: Patient & Insurer Overview (28 pts) | Difficulty: easy

Create the Patient & Insurer Overview page with the following visuals. This page provides a high-level snapshot of patient counts and insurer distribution across the clinics. Ensure all

visuals are clearly titled and formatted with consistent colors.

Required slicers: Primary Coverage slicer (is_primary)

Visual 1: Total Unique Patients (8 pts)

Type: card | Difficulty: easy | Skills: create_visual, format_visual, measure

Add a Card visual that displays the total number of unique patients registered in the patient_insurers table. Use the DAX measure 'Total Unique Patients'. Format the card with a bold title font (size 14), a light blue background, and no border. Position the card in the upper-left area of the canvas.

Hint: The Card visual has a 'Fields' field well — make sure you drag a measure (not a raw column) into it so aggregation is handled correctly.

Solution Walkthrough:

Step 1: In the Visualizations pane, click the 'Card' icon to add a Card visual to the canvas.

Step 2: From the Fields pane, expand the patient_insurers table and drag the [Total Unique Patients] measure into the 'Fields' field well. Step 3: With the card selected, open the Format visual pane. Step 4: Under 'Callout value', set the font size to 20 and make it bold. Step 5: Under 'General > Title', type 'Total Unique Patients', set font size to 14, and toggle title on. Step 6: Under 'General > Effects > Background', enable background and select a light blue color. Step 7: Resize and position the card in the upper-left corner of the page.

Fields field well: [Total Unique Patients]

Visual 2: Policy Distribution by Insurer (10 pts)

Type: donutChart | Difficulty: easy | Skills: create_visual, format_visual

Add a Donut Chart that shows how policies are distributed across insurers. Use insurer_id as the legend and a count of patient_insurer_id as the values. Format the chart with data labels showing both the category and percentage. Set the inner radius to 50%. Give the chart a clear title.

Hint: Check that insurer_id is placed in 'Legend' not 'Values' — if both are numeric, Power BI may try to aggregate insurer_id. You may need to right-click and choose 'Don't summarize' or format it as text.

Solution Walkthrough:

Step 1: In the Visualizations pane, click the 'Donut chart' icon to insert a donut chart. Step 2: From the Fields pane, drag patient_insurers[insurer_id] into the 'Legend' field well. Step 3: Drag patient_insurers[patient_insurer_id] into the 'Values' field well; it will auto-aggregate as Count. Step 4: With the visual selected, open Format visual. Step 5: Under 'Detail labels', toggle on and set label contents to 'Category, percent of total'. Step 6: Under 'Slices', adjust the inner radius slider to 50%. Step 7: Under 'General > Title', type 'Policy Distribution by Insurer' and set an appropriate font size. Step 8: Resize and position the visual centrally on the page.

Legend field well: insurer_id

Values field well: COUNT(patient_insurers[patient_insurer_id])

Visual 3: Primary Coverage Filter (10 pts)

Type: slicer | Difficulty: easy | Skills: slicer, format_visual

Add a Slicer visual using the is_primary field from the patient_insurers table. Change the slicer style to 'Tile' so users can click between True and False. Label the slicer clearly as 'Primary Coverage?' and format the selected tile with a contrasting highlight color. This slicer should cross-filter all other visuals on the page.

Hint: If is_primary shows as 0 and 1 instead of True/False, you can change the data type in Power Query or create a calculated column that maps 0/1 to readable labels. Check the Format pane for the slicer style dropdown — it's accessible via the visual header dropdown arrow.

Solution Walkthrough:

Step 1: In the Visualizations pane, click the 'Slicer' icon to insert a slicer. Step 2: From the Fields pane, drag patient_insurers[is_primary] into the 'Field' field well. Step 3: With the slicer selected, click the dropdown arrow in the upper-right corner of the slicer visual and select 'Tile' from the slicer style options. Step 4: Open the Format visual pane. Step 5: Under 'Slicer settings > Selection', ensure 'Single select' is toggled off so both values can be independently filtered. Step 6: Under 'General > Title', type 'Primary Coverage?' and style appropriately. Step 7: Under 'Tiles > Selected', set the background to a contrasting color (e.g., dark teal). Step 8: Position the slicer in a visible area, such as the top-right or a side panel.

Field field well: patient_insurers[is_primary]

Page 2: Coverage Timeline Analysis (42 pts) | Difficulty: medium

Create the Coverage Timeline Analysis page to help clinic administrators understand how insurance policy enrollments have grown over time and explore coverage start trends. This page requires DAX measures and cross-visual interactions. Each visual must have a descriptive title and properly formatted axes.

Required slicers: Year slicer (from coverage_start), Insurer ID slicer

Visual 1: Coverage Start Year (8 pts)

Type: slicer | Difficulty: easy | Skills: slicer, format_visual

Add a Slicer that lets users filter by the year of coverage_start. Use the Year level of the coverage_start date hierarchy (or a calculated column for year). Set the slicer style to 'Dropdown' to save canvas space. Format the slicer header text as bold, and ensure the slicer interacts with the line chart and bar chart on the page.

Hint: If coverage_start does not automatically expand into a hierarchy, check that the column's data type is set to 'Date' in the data model. You can verify this in the Model view or Power Query.

Solution Walkthrough:

Step 1: In the Visualizations pane, click the 'Slicer' icon. Step 2: From the Fields pane, expand coverage_start in the patient_insurers table to find the date hierarchy. Drag the 'Year' level into the 'Field' field well. If no hierarchy exists, use a calculated column: Coverage Year = YEAR(patient_insurers[coverage_start]). Step 3: With the slicer selected, click the dropdown arrow on the slicer visual header and choose 'Dropdown'. Step 4: Open Format visual pane. Step 5: Under 'Slicer header', enable and set the text to bold with font size 12. Step 6: Confirm that Edit Interactions is set so this slicer filters both the line chart and bar chart on the page (View > Edit Interactions). Step 7: Position the slicer neatly at the top or left side of the page.

Field field well: YEAR(patient_insurers[coverage_start])

Visual 2: New Policies Enrolled Over Time (14 pts)

Type: lineChart | Difficulty: medium | Skills: create_visual, format_visual, measure

Add a Line Chart that shows the count of new policies enrolled by month using the coverage_start date. Use the DAX measure 'New Policies Enrolled' on the Y-axis and the Month level of coverage_start on the X-axis. Add a reference line at the average enrollment count. Format the line with rounded line ends, enable markers at each data point, and add a data label for the most recent month only.

Hint: The Analytics pane is separate from the Format pane — look for the icon that looks like a magnifying glass below the Visualizations icons. Reference lines are only available for certain visual types including line charts.

Solution Walkthrough:

Step 1: In the Visualizations pane, click the 'Line chart' icon. Step 2: Drag the 'Month' level of the coverage_start hierarchy into the 'X-axis' field well. Step 3: Drag the [New Policies Enrolled] measure into the 'Y-axis' field well. Step 4: With the visual selected, open the Format visual pane. Step 5: Under 'Lines', set line style to solid, and enable 'Show markers' with a size of 4. Step 6: Under 'Line > Line ends', set to 'Round'. Step 7: To add a reference line, go to the Analytics pane (magnifying glass icon in Visualizations pane). Click 'Average line' and toggle it on. Label it 'Avg Enrollment'. Step 8: Under 'Data labels', toggle on and set the label position to show only the last point by configuring 'Apply settings to: Last data point'. Step 9: Set an appropriate title, axis titles, and adjust gridlines. Step 10: Resize and position centrally on the page.

X_axis field well: coverage_start (Month)

Y_axis field well: [New Policies Enrolled]

Visual 3: Total Policies by Insurer and Primary Status (20 pts)

Type: barChart | Difficulty: medium | Skills: create_visual, format_visual, measure, relationship

Add a Clustered Bar Chart showing the total number of policies per insurer, broken down by primary vs. non-primary coverage. Use insurer_id on the Y-axis, the DAX measure 'Total Policies' on the X-axis, and is_primary as the legend. Before building this visual, verify that any relationship between patient_insurers and related tables (such as insurers) is correctly

established in the Model view. Format the bars with distinct colors for primary vs. non-primary, add data labels, and sort the bars by total descending.

Hint: To verify or create a relationship, use the Model view. If is_primary displays as 0/1 in the legend, consider whether renaming it through a calculated column would improve readability. Make sure the legend field is a category, not a numeric measure.

Solution Walkthrough:

Step 1: Navigate to the Model view (left panel icon) to verify relationships. Confirm that patient_insurers[insurer_id] is connected to the insurers table (if available) with a many-to-one relationship. If missing, drag the insurer_id field from patient_insurers to the corresponding key in the insurers table to create the relationship. Step 2: Return to the Report view. Step 3: In the Visualizations pane, click the 'Clustered bar chart' icon. Step 4: Drag patient_insurers[insurer_id] into the 'Y-axis' field well. Step 5: Drag the [Total Policies] measure into the 'X-axis' field well. Step 6: Drag patient_insurers[is_primary] into the 'Legend' field well. Step 7: In Format visual, under 'Bars', assign distinct colors to each legend value (e.g., teal for primary, orange for non-primary). Step 8: Enable 'Data labels' and set their position to 'Outside end'. Step 9: Click the ellipsis (...) on the visual and select 'Sort axis > Total Policies > Sort descending'. Step 10: Set the chart title and format axis titles. Step 11: Resize and position the chart.

Y_axis field well: insurer_id

X_axis field well: [Total Policies]

Legend field well: is_primary

Page 3: Drill-Through Patient Policy Detail (22 pts) | Difficulty: hard

Create the Drill-Through Patient Policy Detail page. This page serves as a drill-through destination that clinic staff can reach by right-clicking a specific insurer on other report pages. It must show granular patient-level policy information. Configure drill-through correctly and include advanced formatting and measures. This page should have a 'Back' button to return to the source page.

Required slicers: Policy Number search slicer

Visual 1: Search by Policy Number (6 pts)

Type: slicer | Difficulty: medium | Skills: slicer, format_visual

Add a Slicer using the policy_number field. Set the slicer style to 'Text search' so users can type part of a policy number to filter the table below. Format the slicer with a rounded border and a placeholder prompt text. Ensure the slicer filters the detail table on this page.

Hint: The text search style only appears when the field placed in the slicer is a text/string field. If policy_number is categorized differently, check its data type in Power Query or the Model view.

Solution Walkthrough:

Step 1: In the Visualizations pane, click the 'Slicer' icon. Step 2: Drag patient_insurers[policy_number] into the 'Field' field well. Step 3: Click the dropdown arrow on the slicer visual header and select 'Search' (text search style). Step 4: Open Format visual pane. Step 5: Under 'Slicer settings', confirm the search box is enabled. Step 6: Under 'General > Border', enable border and set corner radius to 5 for rounded corners. Step 7: Under 'General > Title', enter 'Search by Policy Number'. Step 8: Position the slicer above the detail table on the page. Step 9: Use View > Edit Interactions to confirm this slicer filters the table visual below it.

Field field well: patient_insurers[policy_number]

Visual 2: Patient Policy Detail Table (16 pts)

Type: table | Difficulty: hard | Skills: create_visual, format_visual, drill_through, measure

Add a Table visual that shows patient-level policy records including patient_id, insurer_id, policy_number, coverage_start, is_primary, and the DAX measure 'Coverage Duration (Days)'. Configure this page as a drill-through destination: in the Visualizations pane drill-through well, add insurer_id so users can right-click an insurer on other pages and drill through here. Enable conditional formatting on the 'Coverage Duration (Days)' column using a color scale (short durations red, long durations green). Add a Back button using Insert > Buttons > Back. Format the table with alternating row colors and bold column headers.

Hint: Drill-through is configured on the destination page, not the source page. The field you place in the drill-through well on this page determines what field users right-click on source pages to trigger navigation. Make sure the same field (insurer_id) exists on both pages.

Solution Walkthrough:

Step 1: In the Visualizations pane, click the 'Table' icon to insert a table. Step 2: Drag the following fields into the 'Columns' field well in order: patient_insurers[patient_id], patient_insurers[insurer_id], patient_insurers[policy_number], patient_insurers[coverage_start], patient_insurers[is_primary], and the [Coverage Duration (Days)] measure. Step 3: To configure drill-through, scroll down in the Visualizations pane to find the 'Drill through' section (it appears below the field wells when on a report page). Drag patient_insurers[insurer_id] into the 'Add drill-through fields here' well. Power BI will automatically add a Back button; if not, proceed to Step 4. Step 4: Go to Insert tab in the ribbon > Buttons > Navigator > Back. Position the Back button in the upper-left of the page. Step 5: With the table selected, open Format visual. Step 6: Under 'Style presets', choose 'Alternating rows' or manually set alternating row colors under 'Grid > Alternating row color'. Step 7: Under 'Column headers', enable bold and set a background color. Step 8: To add conditional formatting, click the dropdown arrow next to 'Coverage Duration (Days)' in the field well and select 'Conditional formatting > Background color'. Set the format style to 'Gradient', minimum color to red, maximum color to green. Step 9: Adjust column widths for readability. Step 10: Test drill-through by going to the Coverage Timeline Analysis page, right-clicking a bar for an insurer, and selecting 'Drill through > Drill-Through Patient Policy Detail'.

Columns field well: patient_id, insurer_id, policy_number, coverage_start, is_primary, [Coverage Duration (Days)]

DAX Measures

Create the following DAX measures in your Power BI report:

Total Unique Patients (3 pts) | Difficulty: easy

Counts the number of distinct patient IDs in the patient_insurers table, representing the total number of unique patients who have at least one insurance record.

Formula hint: Use DISTINCTCOUNT on the patient_id column in the patient_insurers table

Total Policies (3 pts) | Difficulty: easy

Returns the total number of policy records in the patient_insurers table, representing all insurer associations across all patients.

Formula hint: Use COUNTROWS on the patient_insurers table to count all policy records

New Policies Enrolled (4 pts) | Difficulty: medium

Counts the number of policy records within the current filter context, intended to be used with coverage_start on a time axis to show how many new policies were enrolled per time period.

Formula hint: Use COUNTROWS combined with the patient_insurers table; this measure counts rows and relies on the date filter context from coverage_start to show monthly enrollment when placed on a line chart with a date axis

Coverage Duration (Days) (8 pts) | Difficulty: hard

Calculates the number of days between the policy coverage_start date and today's date, representing how long each policy has been active. Should return BLANK if coverage_start is missing.

Formula hint: Use DATEDIFF between coverage_start and TODAY() with DAY as the interval; handle cases where coverage_start might be blank using IF or ISBLANK