

Healthcare Hackathon Guide: Building a Healthcare KPI Dashboard in Microsoft Fabric

Introduction

In this hackathon challenge, teams will design and build a Healthcare KPI Dashboard using Microsoft Fabric. The goal is to bring together basic clinical, operational, and other data into a simple example of a unified view that empowers decision-makers. Using Fabric's Data Lakehouse, data engineering, and Power BI experiences, participants will ingest sample healthcare datasets, model them for performance, and deliver an interactive dashboard highlighting KPIs such as patient wait times, readmission rates, bed utilization, and staffing efficiency.

The result will showcase how Microsoft Fabric can break down data silos in healthcare, streamline reporting, and enable leaders to make real-time, data-driven decisions that improve patient care and operational outcomes.

Before You Start

1. Sign in to **<https://app.fabric.microsoft.com>** with your Microsoft account.
2. Ensure you have these CSV files saved locally in the Test Environment AVD:
 - Beds.csv
 - Encounter.csv
 - Finance.csv
 - Observation.csv
 - Patient.csv
 - Practitioner.csv
 - QualityMetrics.csv
 - Staffing.csv

Step 1: Create a Workspace

1. In Fabric, select **Workspaces** from the left menu.
2. Click the **+ New Workspace** button at the bottom of the workspace panel.
3. Enter a name such as "HealthcareHackathon".
4. Click **Apply** to create the workspace.

Step 2: Create a Lakehouse

1. Inside the new workspace, click **+ New Item** at the top.
2. From the list of item types, select **All items** and then **Lakehouse**.
3. Enter a name such as "HackathonLakehouse".
4. Click **Create**.
5. When the Lakehouse opens, you will see **Tables** and **Files** sections on the left.

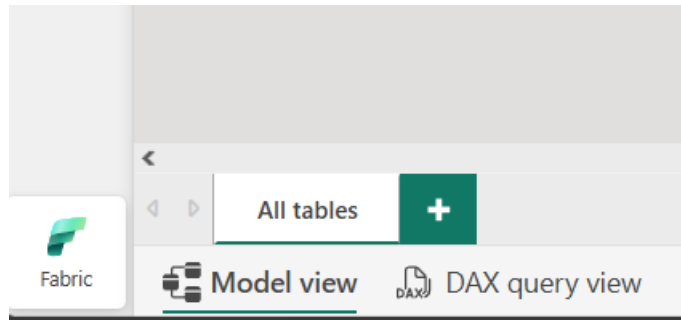
Step 3: Upload CSV Files

1. Right-click **Files** in the left-hand menu of the **Lakehouse**.
2. Choose **Upload Files** (or **Upload Folder** if you have all CSV files in a folder).
3. Select the CSV files and click **Open**.
4. After the files are uploaded, right-click each file name and select **Load to Tables**
→ **New Table**.
5. Keep the default table name and click **Load**.
6. Repeat for each CSV file until all tables are created.

Step 4: Build the Semantic Model

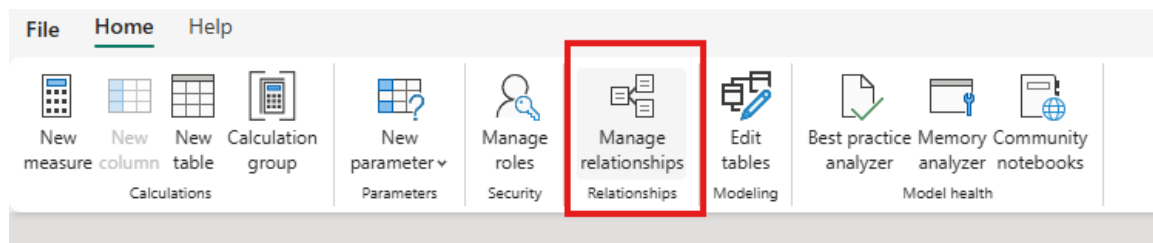
Build the Semantic Model (Cardinality & Measures)

1. In the **HackathonLakehouse**, click **New Semantic Model**.
2. Enter a Semantic Model Name like "HackathonModel".
3. Select all the CSV tables you loaded and click **Confirm**.
4. Go to **Model View** (icon with three tables at the bottom left corner of your screen).



Create Relationships

Click the button in the **Home** ribbon **Manage relationships**.



Click **New relationship**. For each line below, set the **From table** and **To table** with the appropriate cardinality, selecting the column in each table (see the screenshots below).

From Table	From Column	To Table	To Column	Cardinality	Cross Filter
QualityMetrics	Month	Finance	Month	**1 : 1 **	Both
QualityMetrics	Month	Staffing	Date	**1 : * **	Single
QualityMetrics	Month	Beds	Date	**1 : * **	Single
Patient	PatientID	Encounter	PatientID	**1 : * **	Single
Encounter	EncounterID	Observation	EncounterID	**1 : * **	Both



Microsoft



← New relationship

×

Select tables and columns that are related.

From table

Select a table to preview its data

To table

Cardinality

Cross-filter direction

☐ Make this relationship active

☐ Assume referential integrity

Save

Cancel

← New relationship

×

Select tables and columns that are related.

From table

qualitymetrics

ALC_Days HAPI_Count LWBS_Count **Month** OrgID ReadmissionR...

A preview of this table isn't available

To table

finance

AdminCosts ClinicalCosts **Month** OrgID Revenue

A preview of this table isn't available

Cardinality

One to one (1:1)

Cross-filter direction

Both

☒ Make this relationship active

☐ Assume referential integrity [Learn more](#)

1

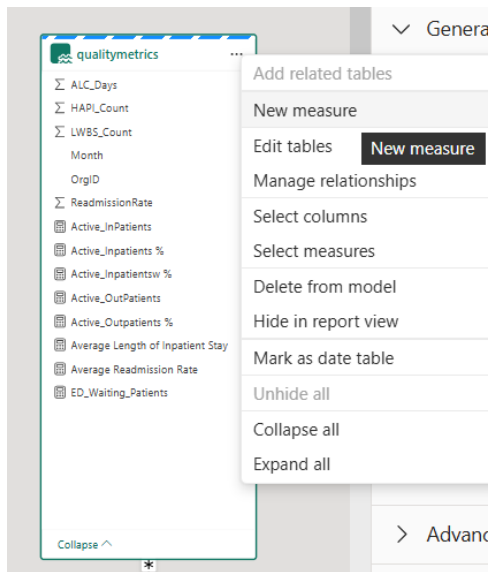
Direct Lake tables don't show data previews or validate relationships cardinality and cross-filter direction. Cardinality is determined by table row count and single cross-filter direction is always populated but these properties may need to be changed manually to reflect the relationship correctly.

Save

Cancel

Create Key Measures

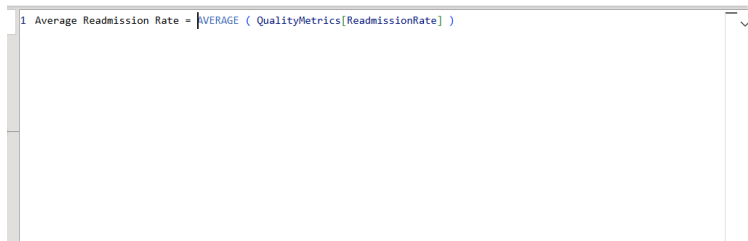
1. In the **Model** view, right-click "QualityMetrics" table and select **New Measure**.



2. Copy and Paste each of these DAX formulas as a **New measure** and press **Enter** after each formula (ensure you copy and paste the entire measure):

Measure 1

Average Readmission Rate =
AVERAGE (QualityMetrics[ReadmissionRate])



Measure 2

Active_InPatients =
VAR SelectedOrgs = VALUES(QualityMetrics[OrgID])

```
VAR MinDate = MIN(QualityMetrics[Month])
VAR MaxDate = MAX(QualityMetrics[Month]) + 1 // Correctly calculates the day after the last
selected date

VAR Result =
CALCULATE(
    COUNTROWS(
        FILTER(
            Encounter,
            Encounter[Status] = "active" &&
            Encounter[Type] = "Inpatient" &&
            Encounter[Start] < MaxDate &&
            Encounter[Start] >= MinDate
        )
    ),
    // 2. Organization Filter: Apply the list of ALL selected organizations to the unrelated Encounter
    table
    TREATAS(SelectedOrgs, Encounter[OrgID])
)
RETURN
COALESCE (Result,0)
```

Measure 3

```
Active_Inpatients % =
VAR SelectedOrgs = VALUES(QualityMetrics[OrgID])
VAR MinDate = MIN(QualityMetrics[Month])
VAR MaxDate = MAX(QualityMetrics[Month]) + 1 // Correctly calculates the day after the last
selected date
VAR Total = CALCULATE(
    COUNTROWS(
        FILTER(
            Encounter,
            encounter[Type] = "Inpatient" &&
            Encounter[Start] < MaxDate &&
            Encounter[Start] >= MinDate
        )
    ),
    // 2. Organization Filter: Apply the list of ALL selected organizations to the unrelated Encounter
    table
    TREATAS(SelectedOrgs, Encounter[OrgID])
)
VAR Active = CALCULATE(
    COUNTROWS(
        FILTER(
            Encounter,
```

```
        Encounter[Status] = "active" &&
        Encounter[Type] = "Inpatient" &&
        Encounter[Start] < MaxDate &&
        Encounter[Start] >= MinDate
    )
),
// 2. Organization Filter: Apply the list of ALL selected organizations to the unrelated Encounter
table
    TREATAS(SelectedOrgs, Encounter[OrgID])
)
RETURN
COALESCE ( DIVIDE ( Active, Total, 0 ), 0 )
```

Measure 4

```
Active_OutPatients =
VAR SelectedOrgs = VALUES(QualityMetrics[OrgID])
VAR MinDate = MIN(QualityMetrics[Month])
VAR MaxDate = MAX(QualityMetrics[Month]) + 1 // Correctly calculates the day after the last
selected date

VAR Result =
CALCULATE(
    COUNTROWS(
        FILTER(
            Encounter,
            Encounter[Status] = "active" &&
            Encounter[Type] = "Outpatient" &&
            Encounter[Start] < MaxDate &&
            Encounter[Start] >= MinDate
        )
    ),
// 2. Organization Filter: Apply the list of ALL selected organizations to the unrelated Encounter
table
    TREATAS(SelectedOrgs, Encounter[OrgID])
)
RETURN
COALESCE ( Result, 0 )
```

Measure 5

```
Active_Outpatients % =
VAR SelectedOrgs = VALUES(QualityMetrics[OrgID])
VAR MinDate = MIN(QualityMetrics[Month])
```

```
VAR MaxDate = MAX(QualityMetrics[Month]) + 1 // Correctly calculates the day after the last
selected date
VAR Total = CALCULATE(
    COUNTROWS(
        FILTER(
            Encounter,
            encounter[Type] = "Outpatient" &&
            Encounter[Start] < MaxDate &&
            Encounter[Start] >= MinDate
        )
    ),
    // 2. Organization Filter: Apply the list of ALL selected organizations to the unrelated Encounter
table
    TREATAS(SelectedOrgs, Encounter[OrgID])
)
VAR Active = CALCULATE(
    COUNTROWS(
        FILTER(
            Encounter,
            Encounter[Status] = "active" &&
            Encounter[Type] = "Outpatient" &&
            Encounter[Start] < MaxDate &&
            Encounter[Start] >= MinDate
        )
    ),
    // 2. Organization Filter: Apply the list of ALL selected organizations to the unrelated Encounter
table
    TREATAS(SelectedOrgs, Encounter[OrgID])
)
RETURN
COALESCE ( DIVIDE ( Active, Total, 0 ), 0 )
```

Measure 6

```
Average Length of Inpatient Stay =
VAR SelectedOrg = MAX(QualityMetrics[OrgID])
VAR MinDate = MIN(QualityMetrics[Month])
VAR MaxDate = MAX(QualityMetrics[Month]) + 1 // Correctly calculates the day after the last
selected date
VAR Result =
CALCULATE(
    AVERAGEX(
        FILTER(
            Encounter,
            Encounter[OrgID] = SelectedOrg &&
            Encounter[Type] = "Inpatient" &&
```



```
        Encounter[Status] = "finished" &&
        Encounter[Start] < MaxDate &&
        Encounter[Start] >= MinDate
    ),
    DATEDIFF(Encounter[Start], Encounter[End], DAY)
)
)
RETURN
COALESCE ( Result, 0 )
```

Measure 7

```
ED_Waiting_Patients =
VAR SelectedOrgs = VALUES(QualityMetrics[OrgID])
VAR MinDate = MIN(QualityMetrics[Month])
VAR MaxDate = MAX(QualityMetrics[Month]) + 1 // Correctly calculates the day after the last
selected date

RETURN
CALCULATE(
    COUNTROWS(
        FILTER(
            Encounter,
            Encounter[Status] = "active" &&
            Encounter[Type] = "Emergency" &&
            Encounter[Start] < MaxDate &&
            Encounter[Start] >= MinDate
        )
    ),
    // 2. Organization Filter: Apply the list of ALL selected organizations to the unrelated Encounter
table
    TREATAS(SelectedOrgs, Encounter[OrgID])
)
```

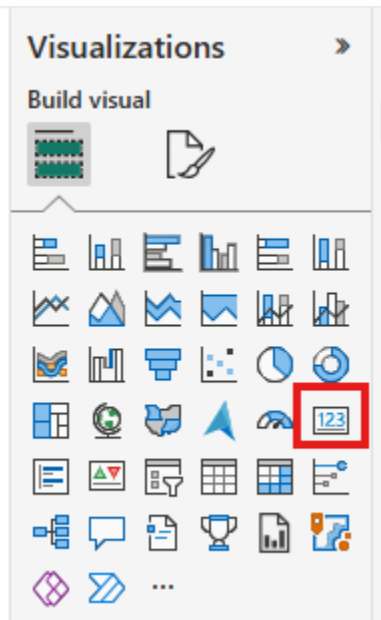
3. The new measures will appear under the table where you created them.
You can drag them into visuals just like fields.

Step 5: Create the Power BI Report

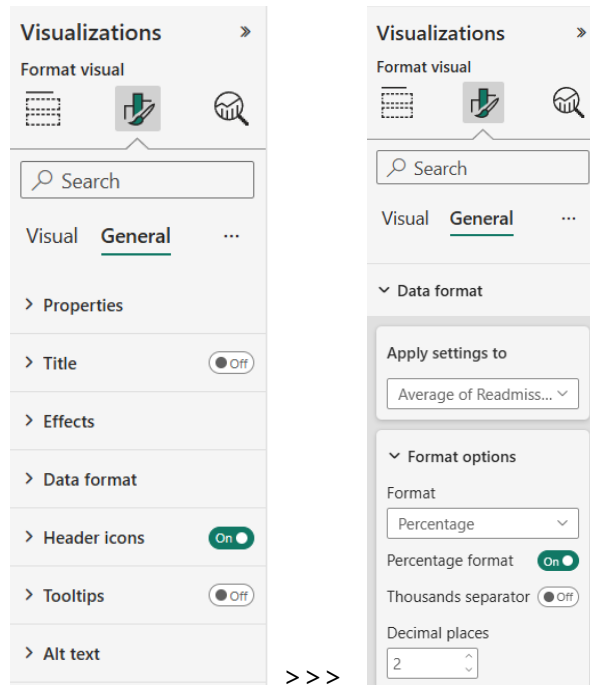
1. In the **Semantic Model**, go to the top menu and click **File** → **Create New Report**.
2. The report canvas opens with a blank page and your tables listed on the right.

3. Add visuals:

KPI Card:



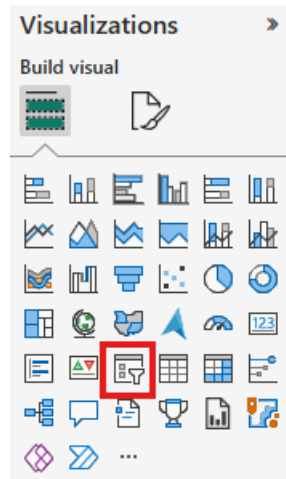
1. Select the **Card** visual from **Visualizations**.
2. Drag "ReadmissionRate" measure from the Table into the **Fields** box.
3. Ensure aggregation is set to **Average** (not Sum).
4. On **Visualization** pane, switch to the middle button with the **paint brush**, select **General**. Under **Data format**, select **Percentage** from the **Format** option dropdown.



5. Add a **Card** visual from **Visualizations** for each of the below, drag each of the measures to the Card **Fields** box:
 - a. Active_InPatients,
 - b. Active_InPatients %,
 - c. Active_OutPatients,
 - d. Active_OutPatients %,
 - e. Average Length of Inpatient Stay,
 - f. ED_Waiting_Patients into the Fields box.

Slicer:

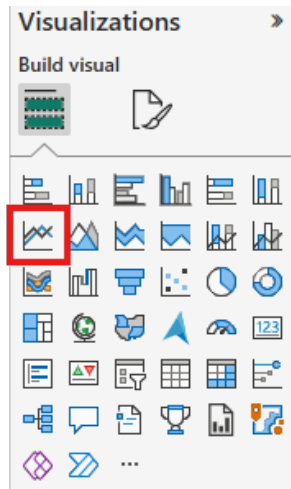
1. Select **Slicer** from **Visualizations**.



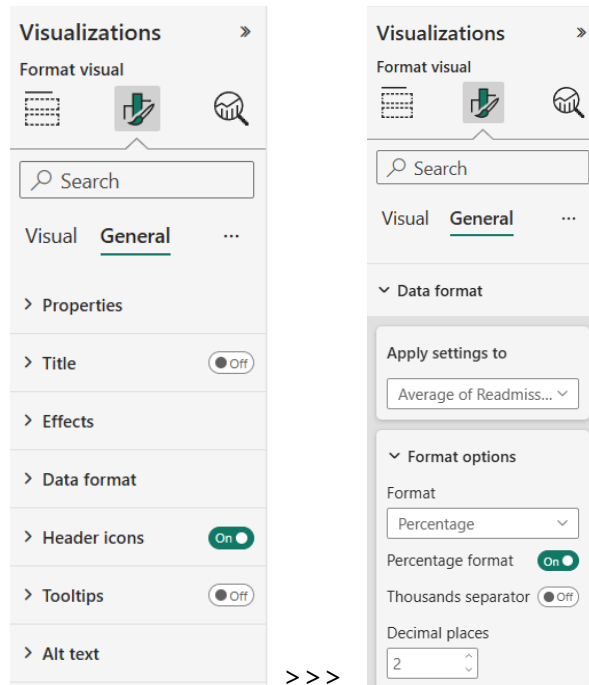
2. From the "qualitymetrics" table select "Month" and drag to the **Field** box.
3. Select another **Slicer** from Visualizations.
4. From the "qualitymetrics" table select "OrgId" and drag to the **Field** box.

Line Chart:

1. Select the **Line chart** visual.

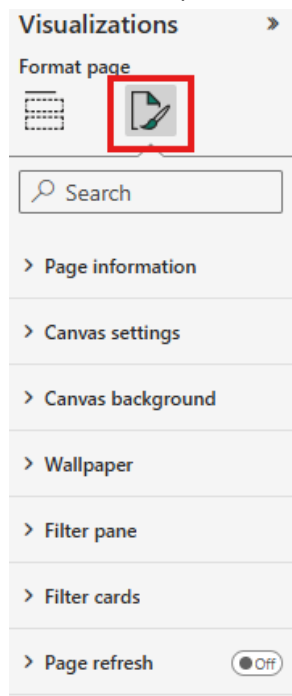


2. Drag "Month" to the **X-axis**.
3. Drag "ED_Waiting_Patients" to the **Y-axis**.
4. Select the **Line chart** visual.
5. Drag "Month" to the **X-axis**.
6. Drag "Active_OutPatients" to the **Y-axis**.
7. Select the **Line chart** visual.
8. Drag "Month" to the **X-axis**.
9. Drag "Active_InPatients" to the **Y-axis**.
10. Select the **Line chart** visual.
11. Drag "Month" to the **X-axis**.
12. Drag "ReadmissionRate" to the **Y-axis**.
13. Ensure **aggregation** is set to **Average** (not Sum).

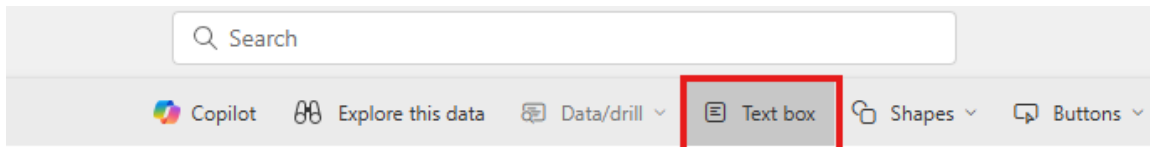


Step 6: Style the Dashboard

1. Click on the Visual you want to format.
2. On the right-hand side, you'll see two tabs:
Build visual (chart icon) and **Format visual** (paint brush icon)



3. Click the **Format** visual tab and select the dropdown you want to use to style.
4. To add a **Title**, Click **Text Box** on the top bar. Add a title such as "Healthcare KPIs", select the preferred font size and color of the text and place it at the top of the report.



Step 7: Save and Share

1. Click **File** → **Save** to save the report. Enter a name like "Healthcare KPI Dashboard".
2. Click **Save** again to confirm.
3. *PRO TIP: To share, click **Share** and provide access to anyone who needs to access your dashboard. If they want to edit your dashboard you will need to ensure they also have access to the Lakehouse and Semantic Model.*

