



## Enterprise PMO Performance Analytics

### Business Problem & Objectives

Large organizations running multiple projects struggle to answer key PMO-level questions such as:

- Which projects are on track, delayed, or over budget?
- How is the overall portfolio performing across cost, schedule, and risk?
- Are resources being utilized efficiently across locations?
- Which projects or locations require management attention?

This project addresses these questions by building a PMO analytics solution using SQL, Excel, and Power BI, following industry-standard data modeling, ETL, and performance measurement practices.

### Table of Content

1. Step 1: Database Creation & Power BI Integration
2. Step 2: ETL, Data Modeling, & DAX Analytical Preparation
3. Step 3: Visualization & Business Insights

### Step 1 Database Creation & Power BI Integration

For this project, I designed and implemented a **MySQL database** to simulate a real-world **Enterprise PMO (Project Management Office) analytics environment**.

#### 1. Database Design

I created a dedicated MySQL database named **enterprise\_pmo\_performance\_analytics** to centrally store structured project data.

Within this database, I implemented three fact tables, each aligned to a specific PMO analytics domain:

The screenshot shows the MySQL Workbench Navigator interface. The 'SCHEMAS' section is open, displaying the 'enterprise\_pmo\_performance\_analytics' schema. Under this schema, there are three tables: 'fact\_project\_performance', 'fact\_resource\_assignment', and 'fact\_stakeholder'. Additionally, there are sections for 'Views', 'Stored Procedures', and 'Functions'. A separate entry for 'practice1 db' is also visible.

- **fact\_Project\_Performance**

Captures project-level cost and schedule performance metrics (PV, EV, AC, planned vs actual)

duration), enabling Earned Value Management (EVM) analysis.

```

1 • INSERT INTO fact_project_performance
2   (Perf_ID, Project_ID, Reporting_Date_ID, Risk_Key, PV, EV, AC, Percent_Complete, Planned_Duration,
3    VALUES
4   ('FP001', 'P001', 202401, 'R005', 120000, 110000, 125000, 18, 540, 120, 'L001'),
5   ('FP002', 'P002', 202401, 'R001', 100000, 100000, 100000, 22, 480, 105, 'L003'),
6   ('FP003', 'P003', 202402, 'R005', 150000, 165000, 185000, 20, 600, 95, 'L002'),

```

- **Fact\_Resource\_Assignment**

Stores resource allocation, utilization, and cost data at the project and reporting-period level, supporting workload, utilization, and cost efficiency analysis.

```

1 • INSERT INTO fact_resource_assignment
2   (Resource_Assign_ID, Team_ID, Project_ID, Reporting_Date_ID, Assigned_Role, Planned_Hours, Actual_Hours,
3    VALUES
4   ('FRA001', 'T001', 'P001', 202401, 'Project Manager', 160, 150, 94.00, 80, 12000, 'On Track'),
5   ('FRA002', 'T002', 'P001', 202401, 'Business Analyst', 140, 135, 96.00, 60, 8100, 'On Track'),
6   ('FRA003', 'T003', 'P001', 202401, 'Senior Developer', 180, 190, 106.00, 70, 13300, 'Overload'),
7   ('FRA004', 'T004', 'P001', 202401, 'QA Lead', 160, 165, 103.00, 65, 10725, 'On Track'),

```

- **Fact\_Stakeholder**

Tracks stakeholder approvals, feedback scores, escalations, and delays, enabling stakeholder engagement and governance monitoring.

```

1 • INSERT INTO Fact_Stakeholder
2   (FPS_ID, Project_ID, Stakeholder_ID, Reporting_Date_ID, Approval_Status, Feedback_Score, Escalation_Status,
3    VALUES
4   ('FPS001', 'P001', 'S001', 202404, 'Approved', 5, 'No', 0),
5   ('FPS002', 'P001', 'S006', 202404, 'Approved', 5, 'No', 0),
6   ('FPS003', 'P002', 'S001', 202405, 'Delayed', 3, 'Yes', 6),
7   ('FPS004', 'P002', 'S005', 202405, 'Pending', 3, 'Yes', 5).

```

All tables were designed at the fact-grain level, using consistent keys such as Project\_ID and Reporting\_Date\_ID to support a star schema model in Power BI.

## 2. Power BI Integration

The MySQL database was then connected directly to Power BI Desktop using the MySQL connector. Each fact table was imported using explicit SELECT queries, ensuring:

- Clean and controlled data extraction.
- Easy traceability between SQL and Power BI.
- High compatibility with Power BI's data model and DAX calculations.

Verified the Select STATEMENT in MySQL

```

9   Utilization_Percent,
10  Hourly_Billable_Cost_USD,
11  Resource_Cost_USD,
12  Assignment_Status
13  FROM Fact_Resource_Assignment;
14

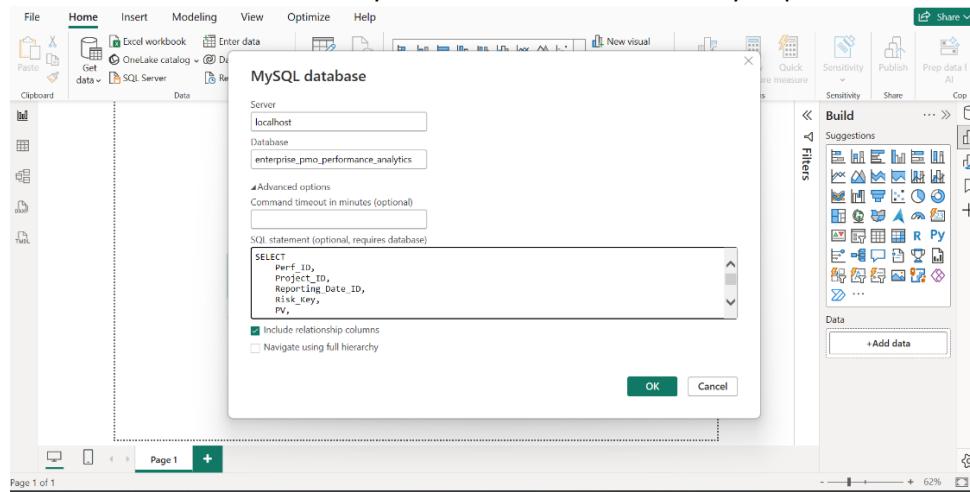
```

Resource_Assign_ID	Team_ID	Project_ID	Reporting_Date_ID	Assigned_Role	Planned_Hours	Actual_Hours	Utilization_Percent
FRA001	T001	P001	202401	Project Manager	160	150	94.00
FRA002	T002	P001	202401	Business Analyst	140	135	96.00
FRA003	T003	P001	202401	Senior Developer	180	190	106.00
FRA004	T004	P001	202401	QA Lead	160	165	103.00

Output:

#	Time	Action	Message	Duration / Fetch
1	21:38:41	SELECT FPS_ID, Project_ID, Stakeholder_ID, Reporting_Date_ID...	50 row(s) returned	0.000 sec / 0.000 sec
2	21:52:26	SELECT Perf_ID, Project_ID, Reporting_Date_ID, Risk_Key, PV...	67 row(s) returned	0.094 sec / 0.000 sec
3	21:55:20	SELECT Resource_Assign_ID, Team_ID, Project_ID, Reporting_D...	80 row(s) returned	0.000 sec / 0.000 sec

Used SELECT statement directly into connector to extract only required columns



### 3. Outcome

This approach demonstrates an end-to-end BI workflow, from database creation and SQL data handling to Power BI modeling and visualization, closely reflecting how enterprise PMO analytics systems are built in real organizations.

Perf_ID	Project_ID	Reporting_Date_ID	Risk_Key	PV	EV	RPY
FP001	P001	202401	R005	120000		
FP002	P002	202401	R001	100000		
FP003	P003	202402	R005	150000		
FP004	P004	202402	R001	80000		
FP005	P005	202403	R004	140000		
FP006	P006	202403	R001	110000		
FP007	P007	202404	R005	160000		
FP008	P008	202404	R003	95000		
FP009	P009	202405	R005	175000		
FP010	P010	202405	R004	100000		
FP011	P011	202406	R001	130000		
FP012	P012	202406	R005	145000		
FP013	P013	202407	R004	150000		
FP014	P014	202407	R001	120000		
FP015	P015	202408	R005	165000		
FP016	P016	202408	R004	90000		
FP017	P017	202409	R005	180000		
FP018	P018	202409	R003	210000		
FP019	P019	202410	R005	140000		
FP020	P020	202410	R004	100000		
FP021	P001	202411	R005	120000		
FP022	P002	202411	R001	100000		

### Connecting with Dimension Tables

- > Dim\_Date
- > Dim\_location
- > Dim\_Project
- > Dim\_Risk
- > Dim\_Stakeholder
- > Dim\_team
- > Fact\_Project\_Stakehold...
- > Fact\_Project\_Performa...
- > Fact\_Resource\_Assign...

## NEXT STEP:

Connecting to Power BI and perform ETL operation, create Data model, and perform DAX to create Calculated Coloumns and Calculated measures.

## Step 2: ETL, Data Modeling, & DAX Analytical Preparation

After loading the fact tables from MySQL and the dimension tables from Excel, I performed a series of ETL (Extract, Transform, Load) operations within Power BI to prepare the data for analysis.

### 1. ETL & Data Cleaning

- Removed empty and duplicate rows across fact and dimension tables
- Standardized column names and data types for consistency
- Merged and split columns where required (e.g., status fields, percentage values)
- Applied basic normalization rules to avoid redundancy and improve model clarity

These steps ensured high data quality and reliable downstream analysis.

### 2. Data Modeling (Star Schema)

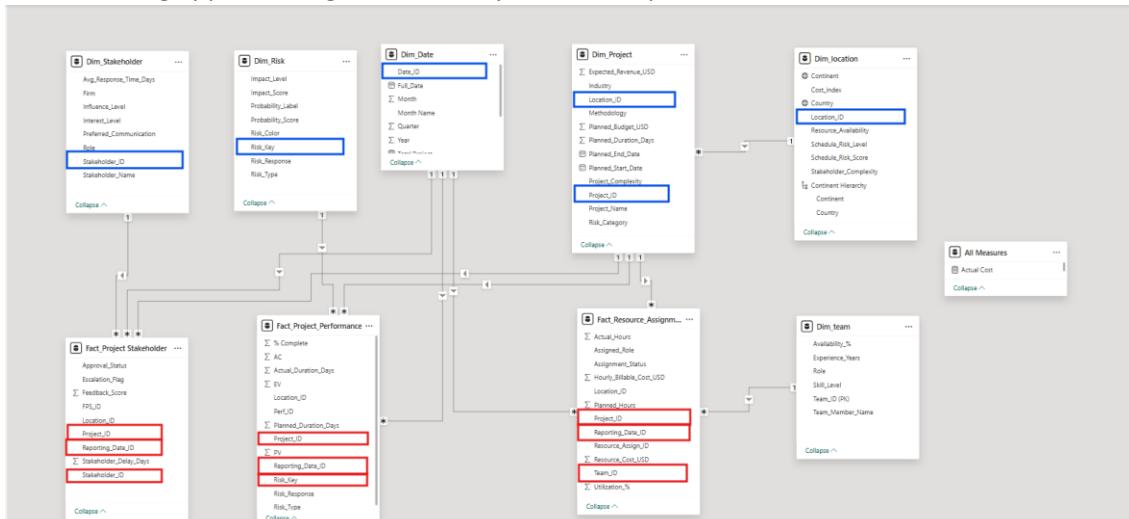
A star-schema data model was created to support scalable analytics:

- **3 Fact Tables**
  - Fact\_Project\_Performance
  - Fact\_Resource\_Assignment
  - Fact\_Stakeholder
- **6 Dimension Tables**
  - Dim\_Project
  - Dim\_Date
  - Dim\_Location
  - Dim\_Stakeholder
  - Dim\_Team
  - Dim\_Risk

Primary keys from each dimension were connected to corresponding foreign keys in the fact tables, ensuring:

- One-to-many relationships
- Single-direction filtering
- No ambiguity or many-to-many relationships

This modeling approach aligns with enterprise BI best practices.



## 2. DAX – Calculated Columns

Calculated columns were created **inside fact tables** to support row-level analysis:

- **Fact\_Project\_Performance**
  - Schedule Variance Days (Planned vs Actual Duration)
  - Cost Overrun Flag (AC vs EV)
- **Fact\_Resource\_Assignment**
  - Resource Over-Utilization Flag (Actual Hours > Planned Hours)

These columns enabled conditional logic and filtering in visuals.

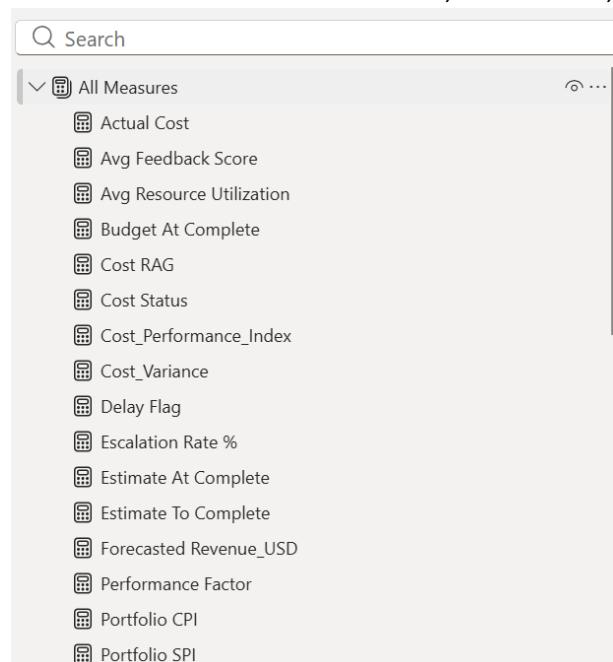
---

## 3. DAX – Measures (Aggregations)

Analytical measures were created using DAX aggregation and calculation functions, including:

- SUM() – Planned Value, Earned Value, Actual Cost, Resource Cost
- AVERAGE() – Resource Utilization %, Feedback Score
- COUNT() / COUNTROWS() – Projects, Resources, Stakeholder Records
- DIVIDE() – CPI, SPI, Utilization Rate (safe division)
- CALCULATE() with filters – Approval Rate, Escalation Rate

All KPIs were calculated as measures, not columns, to ensure dynamic filtering and slicer interaction.



The screenshot shows a Power BI data model interface. At the top is a search bar with a magnifying glass icon and the word 'Search'. Below it is a tree view of measures. The root node is 'All Measures', which is expanded, showing the following list of measures:

- Actual Cost
- Avg Feedback Score
- Avg Resource Utilization
- Budget At Complete
- Cost RAG
- Cost Status
- Cost\_Performance\_Index
- Cost\_Variance
- Delay Flag
- Escalation Rate %
- Estimate At Complete
- Estimate To Complete
- Forecasted Revenue\_USD
- Performance Factor
- Portfolio CPI
- Portfolio SPI

## **Step 3: Visualization & Business Insights**

After completing ETL, data modeling, and DAX calculations, the final step focused on building interactive Power BI dashboards to translate data into actionable business insights. These dashboards are designed for PMO leaders, project managers, and senior stakeholders.

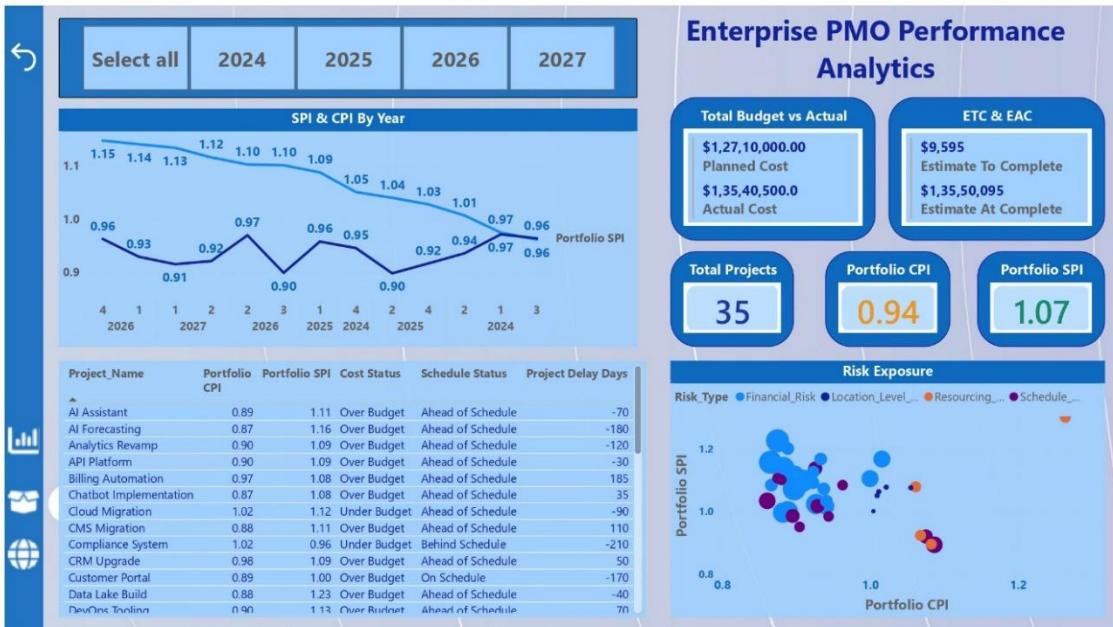
---

### **Dashboard 1: Executive Portfolio Overview**

**Purpose:** Enable leadership to instantly assess portfolio health

- Provides a high-level view of the entire project portfolio
- KPIs include overall cost health, schedule status, risk exposure, and stakeholder approvals
- Answers key business questions:
  - Which projects are on track, delayed, or over budget?

- What is the overall health of the portfolio?
- Where should leadership prioritize attention or intervention?



## Dashboard 2: Project Performance & Earned Value Analysis

**Purpose:** Enable PMs and PMO to control execution and forecast outcomes

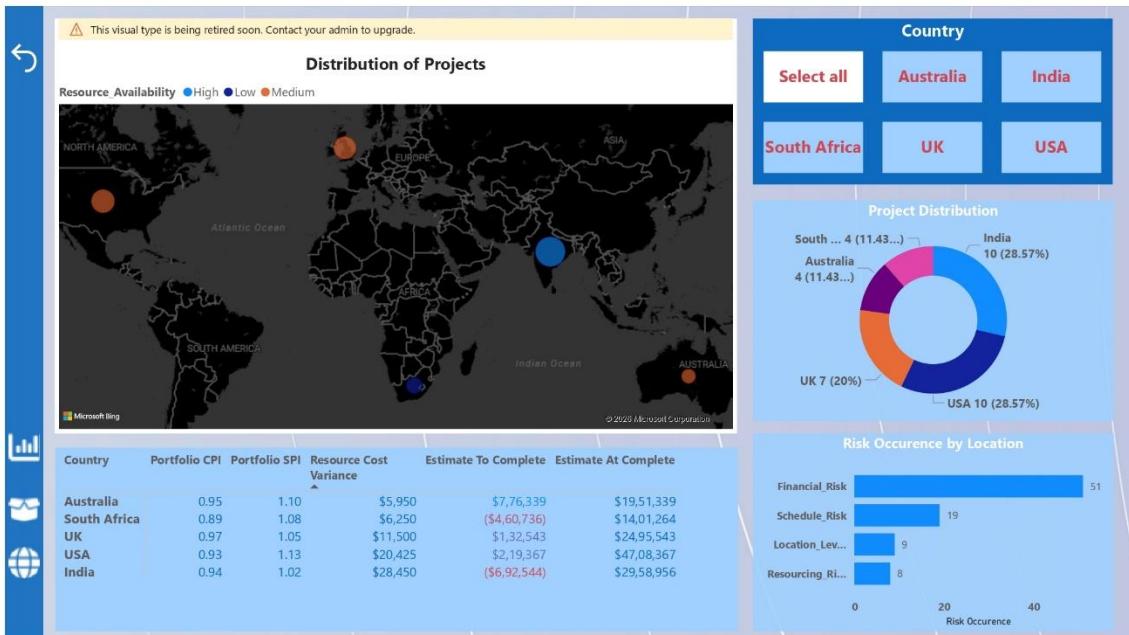
- Focuses on detailed project-level performance
- Key metrics: PV, EV, AC, Schedule Variance (SV), Cost Variance (CV), CPI, SPI
- Answers key business questions:
  - Are projects performing as per plan?
  - Which projects are ahead or behind schedule?
  - Which projects are overrunning costs?



## Dashboard 3: Location & Resource Efficiency Analysis

**Purpose:** Enable Regional managers and Stakeholders to look projects and resources perform best

- Analyzes project execution across locations and resource utilization
- KPIs include resource utilization %, planned vs actual effort, cost efficiency, estimate vs forecast revenue across location
- Answers key business questions:
  - Which locations are most efficient in delivering projects?
  - Are resources over-allocated or under-utilized?
  - How can resource distribution and risk type be optimized across locations?



## Key Outcomes

- Delivered a 360° PMO analytics solution
- Demonstrated end-to-end skills in:
  - SQL
  - Data modeling
  - ETL
  - DAX
  - Power BI visualization
- Enabled data-driven answers to real PMO business questions