IT Service Desk Analytics Dashboard

Matthew Williams

December 11th, 2025

Executive Summary

This project aims to create a dashboard that provides analysis of IT Service Desk performance with the use of Power BI and SQL Server Management Studio 19. The project extracts data from a SQL database, which is transformed and loaded into Power BI, where the visuals will be created. With a focus on identifying trends, it also aims to understand ticket volume behavior, technician performance, and create visuals for storytelling to allow easier decision-making. The dashboard highlights ticket distribution, resolution timelines, monthly performance metrics, and identifies the fastest resolving technician.

Insights are intended to assist leadership in optimizing the IT department by identifying areas of bottlenecks, service delivery efficiencies, resolution times, and compliance with service level agreements.

Table of Contents

Primary Objectives

Demonstrating the ability to analyze IT Service Desk Operations using SQL and Power BI, the primary goals of this project were as follows:

- SLA tracking and Breach analysis
- Device Uptime Monitoring
- Technician and Team Performance Metrics
- Data Analyst-style thinking and reporting
- Level 1-3 escalation logic
- Ticket volume trends and categorization

A data set of approximately 100 support tickets was generated for this analysis.

System Architecture

This project is composed of three core tables to simulate a real-world IT support environment. **PK** represents the primary key for each table, whereas **FK** represents the foreign key, which communicates how a table forms a relationship with another.

Tickets Table

This table captures incident-level information, including:

- ticket_id **PK**
- date_opened, date_closed
- category (Network, Software, Hardware, & Security)
- priority (Low to High)
- assigned_to **FK** (*references Technicians[technician_id])*
- status (Open, In Progress, Closed)

- sla_hours
- actual_resolution_hours
- department (Finance, HR, IT, Operations, Security, & Executive)
- device_id **FK** (*references Devices[device_id])*

Devices Table

This table captures and tracks asset performance:

- device_id **PK**
- device_type (Server, Laptop, POS, Router, etc.)
- location
- status
- uptime_percentage
- last_checked

Technicians Table

This table provides technician-level information:

- technician_id **PK**
- name
- team (Network, Software, Hardware, Helpdesk)

Data Generation and Insertion

Approximately 100 sample tickets and numerous supporting device/technician records were generated using a combination of the following:

- SQL scripts
- ChatGPT-based synthetic data creation

- Manual adjustments for realistic patterns

The data covers a realistic distribution of the following:

- High-traffic departments
- Different severity levels
- Fast vs slow technician performance
- SLA breaches
- Device uptime variations

SQL Queries Executed

The following SQL scripts were performed to generate insights for the reporting layer of this project:

Create Tables

Create_tbl_IT_query creates the Technicians, Devices, and Tickets table. It also sets the primary and foreign keys for each table to create relationships between the tables.

Add Tickets

Add_tickets_query creates tickets and inserts them into the Ticket table.

Add Devices

Add_devices_query creates devices and inserts them into the Devices table.

Add Technicians

Add_tech_query creates and adds technicians and their team to the Technicians table.

SLA Compliance

SLA_compliance_query calculates if the tickets were resolved within the hours of the SLA agreements.

Group Tickets

Group_tickets_query groups the tickets by categories and then sorts them in count order.

Average Resolution Time

Avg_res_time_query calculates the average resolution hours and then groups them by category.

Devices Uptime

Dev_uptime_query calculates the average uptime of devices.

Extraction, Transformation & Load — Power BI

Within Power BI, the data is extracted from SQL Server Management 19 into Power BI. While being extracted from the SQL Server, it would be transformed to allow the dates to be split from the time into two separate columns. During the transformation period, an Hour_Opened column is created from the split. Additionally, the SLA_Met column is created to record which tickets met the SLA. A calendar table, as well as a time table, is created to allow measures to work properly. A relationship with both of the previously mentioned tables is then connected to the tickets table in a star schema. Following this, the data is loaded into the app, and measures are created.

Tables

Calendar Table

This table covers dates on the calendar; it is automatically created using a DAX formula on Power BI. The Date column forms a relationship with the Tickets table with the date_opened column, the foreign key, a column made in Power BI.

- Date **PK**

- DayOfWeek
- DayOfWeekNumber

Time Table

This table covers the time and is automatically created using a DAX formula on Power BI. The Value column forms a relationship with the Tickets table with the Hour_Opened column, the foreign key, a column made in Power BI.

- HourLabel
- HourText
- Time
- Value **PK**

Measures

Mean_Time_Between_Failures

Calculates the average time between failures for devices.

MostProblematicDeviceType

Calculates the most problematic device in the devices table.

Avg_Resolution_Time_Per_Tech

Calculates the average resolution time for each technician.

Best Team

Calculates the team with the minimum average resolution time.

BestTechnician

Calculates the technician with the minimum average resolution time.

Min_Avg_Resolution_Per_Tech

Calculates the minimum average resolution for each technician.

AvgResolutionTime

Calculates the average resolution time in the tickets table.

Sum of SLA_Met divided by Tickets Count

Calculates SLA compliance.

Tickets Count

Calculates the number of tickets received.

TicketsVolume_MTD

Calculates the number of tickets received from the month beginning to end.

Power BI Dashboard Design

The dashboard design is divided into three pages to mirror a real IT performance reporting system.

Page 1 — January Service Desk Overview

KPIs:

- Total Ticket Volume
- SLA Compliance %
- Average Resolution Time
- Open vs Closed Tickets
- Priority

Visuals:

- Bar Chart: Tickets by Category
- Line Chart: Tickets by Day
- Tree Map: Tickets by Department

- Heat Map: Ticket Volume by Hour of Day

This page aims to answer:

“What is happening in the Service Desk?”

Page 2 — January Technician Performance Overview

KPIs:

- Best Technician
- Best Team
- Average Resolution Time per Technician

Visuals:

- Line Chart: Resolution Speed vs Ticket Volume
- Tree Map: Tickets by Team
- Heat Map: Average Resolution Time by Technicians

This page aims to answer:

“Who is performing well, and who needs support?”

Page 3 — January Infrastructure & Devices Overview

KPIs:

- Average Device Uptime
- Locations
- Most Problematic Device
- Amount of Devices

Visuals:

- Pie Chart: Device Status
- Line Chart: Mean Time Device Failures by Hour

- Heat Map: Failures by Devices

This page aims to answer:

"How healthy is the infrastructure?"

Dashboard Enhancements

To make the dashboard feel enterprise-ready, multiple features were added, which include:

- Conditional formatting
- RAG status indicators (Red/yellow/green)
- Drill-through pages for technicians
- Dynamic tooltips
- Interactive slicers and buttons
- Sorting by performance thresholds

Deliverables Created

A dashboard was created, which can be exported into multiple formats for viewing while achieving the objectives previously mentioned. The dashboard provides the following:

- SLA tracking and Breach analysis
- Device Uptime Monitoring
- Technician and Team Performance Metrics
- Data Analyst-style thinking and reporting
- Level 1-3 escalation logic
- Ticket volume trends and categorization

Limitations & Challenges

Some challenges were faced when attempting to run SQL queries within Power BI; thus, some queries were used to compare the outputs to the measures, using DAX, made in Power BI. Power BI requires an educational or professional Microsoft account to have access to the full version. Consequently, with no access to the full version, other measures for sharing the project were taken instead of publishing it and sharing the link.

Scalability

The project currently has approximately 100 tickets, 20 technicians, and 20 devices. Due to it being connected to a SQL database, this project has the ability to be upscaled or downscaled. This allows smaller or larger companies to follow a similar format for their IT departments. It also has the ability to review larger periods of time with slight adjustments to the data loaded and some DAX formulas. This scalability allows this project to be quite valuable to a diaspora of companies across the globe due to its deliverables and scalability.

Acknowledgements

Thank you goes out to many people who have given feedback or knowledge that has supported this project. Close relatives and colleagues such as Javan Johnson and Shannon Louissaint have given feedback during the brainstorming portion of the project. Colleagues like Trenton Fowler have given knowledge on the topic of SQL and its uses. ChatGPT and Microsoft CoPilot have also assisted with understanding and troubleshooting errors. Professors at the University of The Bahamas from my undergraduate years, Prof. Chaker Eid and Prof. Eardley Grant, assisted in past years with introducing me to SQL. The Bahamas' Upskill program

specifically the Power BI and SQL for Data Analyst course provided exemplary information that was also utilized in this project, alongside knowledge gained from text such as *Database Systems: Design, Implementation and Management* and *The Definitive Guide to Dax Business Intelligence with Microsoft Power BI, SQL Server Analysis Services and Excel* (Coronel & Morris, 2017; Russo & Ferrari, 2020).

References

Coronel, C., & Morris, S. (2017). *Database Systems: Design, Implementation, and Management* (Vol. 12). Boston, Massachusetts, United States of America: Cengage Learning.

Russo, M., & Ferrari, A. (2020). *The Definitive Guide to Dax Business intelligence with Microsoft Power BI, SQL Server Analysis Services, and Excel.* Pearson Education.