final_report

January 22, 2026

1 Badge In/Out Attendance Analysis Report

This report analyzes attendance behavior of students and teachers using badge in/out data collected over multiple months.

1.1 Dataset Overview

- Source: Badge in/out events
- Scale: ~1 million events
- Entities: Students, Teachers
- Time Range: Sept 2024 – April 2025

1.2 Data Model

The system follows a dimensional model: - Dimension tables: Students, Teachers, Calendar - Source table: Raw badge events - Fact table: Daily attendance metrics

1.3 Fact Attendance (Daily)

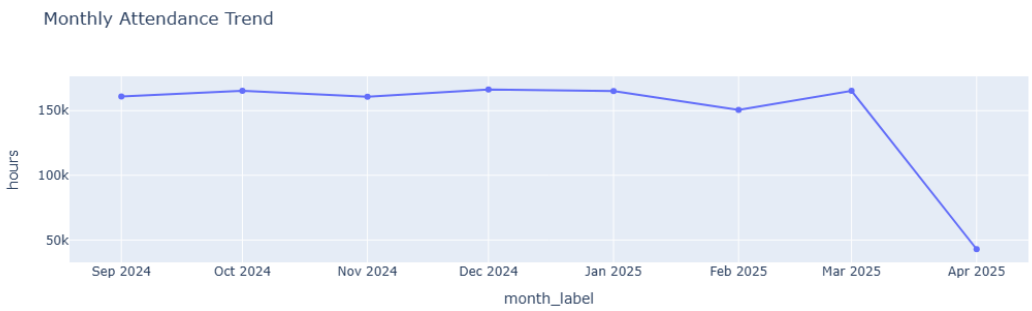
The daily fact table aggregates raw badge events into metrics such as: - Total hours per person per day - Total sessions - First in-time and last out-time - Average session duration

1.4 Monthly Attendance Trend

This line chart shows the overall trend of total attendance hours across months.

[6]:
```
query = """
SELECT
    c.year,
    c.month,
    SUM(f.total_hours) as hours
FROM fact_attendance_daily f
JOIN dim_calendar c
ON f.date_id = c.date_id
GROUP BY c.year, c.month
ORDER BY c.year, c.month
"""

df_month = pd.read_sql(query, conn)
```

```
df_month["month_label"] = df_month["year"].astype(str) + "-" +␣
↪df_month["month"].astype(str)

fig = px.line(df_month,
              x="month_label",
              y="hours",
              title="Monthly Attendance Trend",
              markers=True)

fig.show()
```

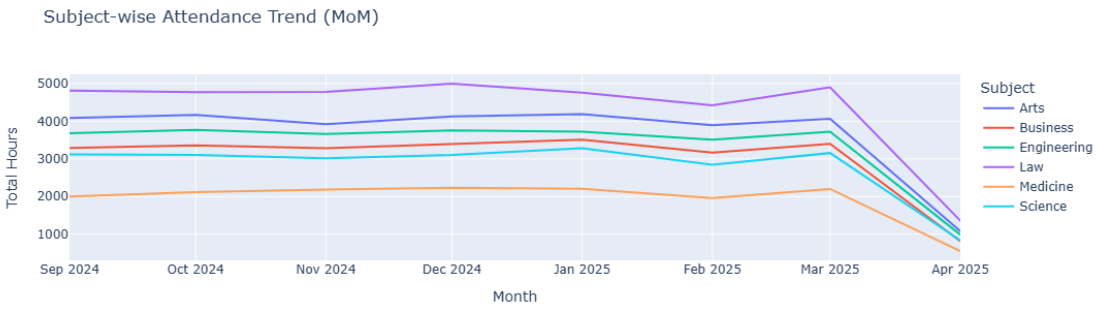


1.5 Subject-wise Attendance Trend (Month-over-Month)

This line chart compares attendance trends across different subjects, showing how total attendance hours vary by department over time.

[8]:
```
query = """
SELECT
    c.year,
    c.month,
    t.department AS subject,
    SUM(f.total_hours) AS total_hours
FROM fact_attendance_daily f
JOIN dim_calendar c
    ON f.date_id = c.date_id
JOIN dim_teachers t
    ON f.badge_id = t.badge_id
GROUP BY c.year, c.month, t.department
ORDER BY c.year, c.month;
"""

df = pd.read_sql(query, conn)
```

```
df["month_label"] = df["year"].astype(str) + "-" + df["month"].astype(str)

fig = px.line(
    df,
    x="month_label",
    y="total_hours",
    color="subject",
    title="Subject-wise Attendance Trend (MoM)"
)

fig.update_layout(
    xaxis_title="Month",
    yaxis_title="Total Hours",
    legend_title="Subject"
)

fig.show()
```

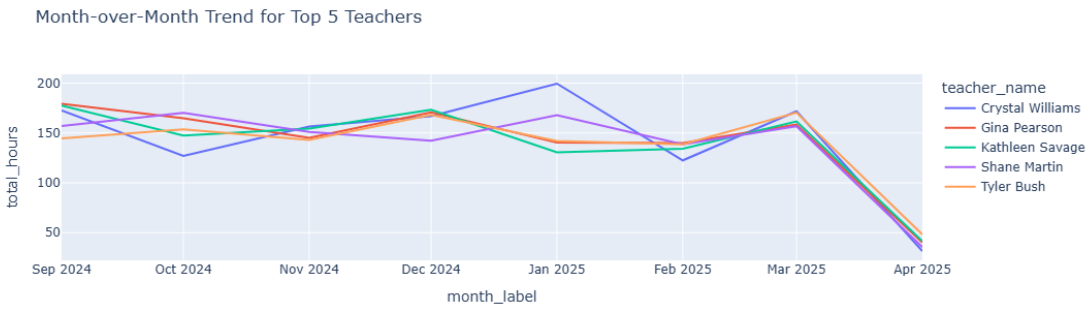


1.6 Teacher-wise Attendance Trend (Month-over-Month)

This line chart shows month-over-month attendance trends for top 5 teachers, illustrating variations in teaching activity over time.

[9]:
```
query = """
SELECT
    c.year,
    c.month,
    t.name AS teacher_name,
    SUM(f.total_hours) AS total_hours
FROM fact_attendance_daily f
JOIN dim_calendar c
    ON f.date_id = c.date_id
JOIN dim_teachers t
    ON f.badge_id = t.badge_id
```

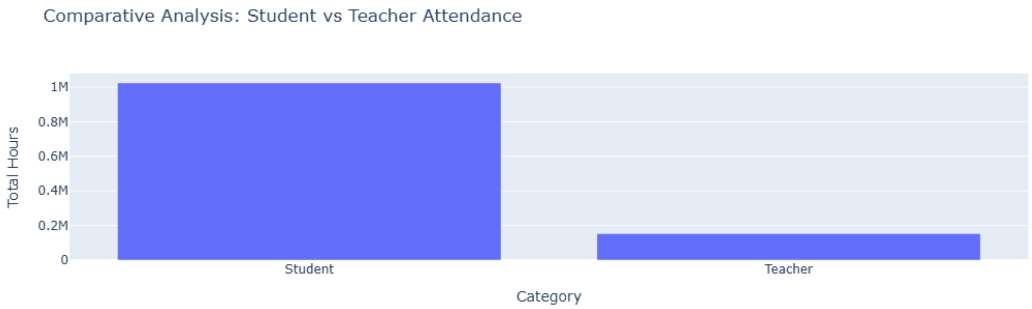
```
GROUP BY c.year, c.month, t.name
ORDER BY c.year, c.month;
"""

df = pd.read_sql(query, conn)

df["month_label"] = df["year"].astype(str) + "-" + df["month"].astype(str)

# Pick only top 5 teachers overall for cleaner chart
top_teachers = (
    df.groupby("teacher_name")["total_hours"]
      .sum()
      .sort_values(ascending=False)
      .head(5)
      .index
)

df_top = df[df["teacher_name"].isin(top_teachers)]

fig = px.line(
    df_top,
    x="month_label",
    y="total_hours",
    color="teacher_name",
    title="Month-over-Month Trend for Top 5 Teachers"
)

fig.show()
```



1.7 Comparative Attendance: Students vs Teachers

This chart compares total attendance hours between students and teachers, providing insight into engagement patterns across roles.

[10]:
```
query = """
SELECT
    CASE
        WHEN f.badge_id IN (SELECT badge_id FROM dim_teachers) THEN 'Teacher'
        ELSE 'Student'
    END AS person_type,
    SUM(f.total_hours) AS total_hours
FROM fact_attendance_daily f
GROUP BY person_type;
"""

df = pd.read_sql_query(query, conn)

fig = px.bar(
    df,
    x="person_type",
    y="total_hours",
    title="Comparative Analysis: Student vs Teacher Attendance",
    labels={"person_type": "Category", "total_hours": "Total Hours"}
)

fig.show()
```

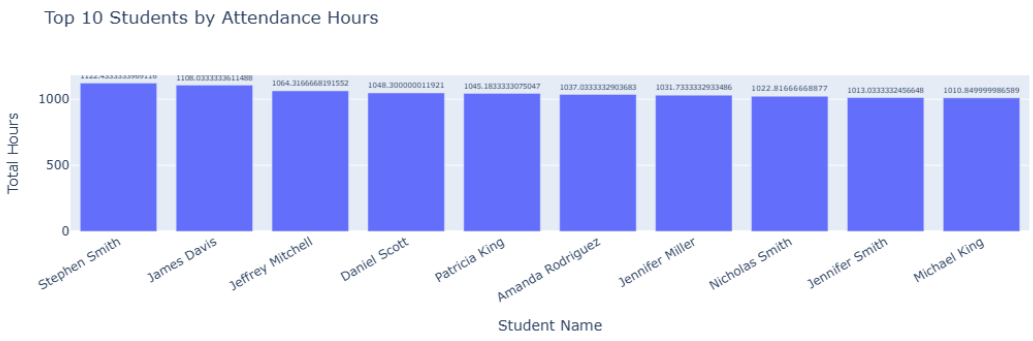


1.8 Top 10 Students by Attendance

This bar chart highlights the top 10 students based on total attendance hours, indicating the most consistently present students on campus.

[11]:
```
query = """
SELECT
    s.name AS student_name,
    SUM(f.total_hours) AS total_hours
FROM fact_attendance_daily f
JOIN dim_students s
```

```
    ON f.badge_id = s.badge_id
GROUP BY s.name
ORDER BY total_hours DESC
LIMIT 10;
"""

df = pd.read_sql_query(query, conn)

fig = px.bar(
    df,
    x="student_name",
    y="total_hours",
    title="Top 10 Students by Attendance Hours",
    labels={"student_name": "Student Name", "total_hours": "Total Hours"},
    text="total_hours"
)

fig.update_traces(textposition="outside")
fig.update_layout(xaxis_tickangle=-30)

fig.show()
```



1.9 Conclusion

This analysis demonstrates how raw event data can be transformed into analytical insights using dimensional modeling and fact tables.