

Technical Project Report: Institutional Data Analysis via SQL

Developer: Chinenye J. Orji

Organisation: Nigerian College of Aviation Technology (NCAT)

Project Type: Data Analysis & Relational Querying

Tool: SQL (MySQL)

1. Project Overview

This project focuses on the analysis of an existing institutional database at the Nigerian College of Aviation Technology. The objective was to perform complex data retrieval to solve specific administrative and operational challenges. The project is divided into two distinct analytical tasks.

2. Task 1: Academic Performance & Graduation Audit

Objective: To identify departments with declining graduation rates and low average GPAs to facilitate early academic intervention.

Technical Approach:

- **Multi-table Joins:** Linked student demographic tables with academic result logs to create a unified view of the student lifecycle.
- **Conditional Aggregation:** Utilised CASE statements within SUM and AVG functions to calculate graduation percentages and withdrawal counts programmatically.
- **Logic:** Filtered by enrolment years (2015–2016) to focus on recent institutional trends.

SQL Query Logic:

```
1 SELECT
2     Department,
3     COUNT(StudentID) AS Total_Enrolments,
4     ROUND(AVG(GPA), 2) AS Average_GPA,
5     SUM(CASE WHEN Status = 'Withdrawn' THEN 1 ELSE 0 END) AS Withdrawal_Count,
6     ROUND((SUM(CASE WHEN Status = 'Graduated' THEN 1 ELSE 0 END) * 100.0 / COUNT(StudentID)), 2) AS Graduation_Rate_PCT
7 FROM NCAT_Student_Records
8 GROUP BY Department
9 ORDER BY Graduation_Rate_PCT ASC;
```

Figure 1: Screenshot of Task 1 Query

3. Task 2: Resource & Capacity Optimisation

Objective: To identify over-utilised courses and flag resource bottlenecks where enrolment exceeds classroom or instructor capacity.

Technical Approach:

- **Threshold Filtering:** Applied the HAVING clause to focus only on courses with significant enrolment (greater than 10 students).
- **Automated Alerting:** Created a "Resource Status" column using nested CASE logic to categorise courses as "Normal", "Warning", or "Urgent" based on capacity thresholds.
- **Operational Insight:** Designed the output to assist the bursary and registrar departments in classroom allocation for the upcoming semester.

SQL Query Logic:

```
1 SELECT
2     CourseName,
3     Instructor,
4     MaxCapacity,
5     COUNT(EnrolmentID) AS Current_Enrolment,
6     (MaxCapacity - COUNT(EnrolmentID)) AS Available_Seats,
7     CASE
8         WHEN COUNT(EnrolmentID) >= MaxCapacity THEN 'URGENT: Over Capacity'
9         WHEN COUNT(EnrolmentID) >= (MaxCapacity * 0.8) THEN 'Warning: High Demand'
10        ELSE 'Normal'
11    END AS Resource_Status
12 FROM NCAT_Course_Logistics
13 GROUP BY CourseName, Instructor, MaxCapacity
14 HAVING COUNT(EnrolmentID) > 10
15 ORDER BY Current_Enrolment DESC;
```

Figure 2: Screenshot of Task 2 Query

4. Reflection & Learning

This two-task analysis demonstrated the power of SQL as a tool for **Institutional Intelligence**.

Task 1 reinforced my ability to use **Conditional Aggregates** to transform raw status logs into high-level performance metrics. This is vital for government and academic reporting where stakeholders need to see "Success Rates" rather than just row-level data.

Task 2 improved my understanding of **Operational Data Analysis**. By creating a traffic-light warning system within the SQL query, I learned how to provide actionable insights rather than just static reports. I realised that as a Data Analyst, my goal is to help the organisation anticipate problems, such as classroom overcrowding, before they impact the student experience.