Project Documentation

# SPREADSHEET (Excel)

## 1. Creating a Job Code Column

Created a job code column to detect repeated jobs. This was done using:

=[@[ihs\_id]]&" - "&[@fault]

This formula combines the ihs\_id and fault columns, allowing easy detection of duplicate jobs.

## 2. Job Classification

To classify jobs based on fault descriptions, I used:   
 =IF(ISNUMBER(SEARCH("Movement", [@fault])),  
 IFERROR(INDEX(dimensions.xlsx!ihs\_price[Group], MATCH(TRUE, ISNUMBER(SEARCH(dimensions.xlsx!ihs\_price[Group], [@fault])), 0)), "Crosscheck"),  
 IF(ISNUMBER(SEARCH("MDG", [@fault])),  
 IFERROR(INDEX(dimensions.xlsx!ihs\_price[Group], MATCH(TRUE, ISNUMBER(SEARCH(dimensions.xlsx!ihs\_price[Group], [@fault])), 0)), "Crosscheck"),  
 XLOOKUP([@fault], dimensions.xlsx!ihs\_price[fault], dimensions.xlsx!ihs\_price[Group], "Crosscheck")  
 )  
 )   
This helps categorize the jobs based on keywords in the fault field, providing easy tracking.

## 3. Payment Classification

Classified the payment reference based on job closure methods, Created another formula:  
 =IF([@[payment\_ref]]="MRF", "Warehouse",  
 IF(LEFT([@[payment\_ref]], 3)="PO-", "Outsource",  
 IF(LEFT([@[payment\_ref]], 6)="Vendor", "Outsource",  
 IF(LEFT([@[payment\_ref]], 3)="IHS", "Employee",  
 IF(LEFT([@[payment\_ref]], 5)="Raise", "Employee",  
 IF(LEFT([@[payment\_ref]], 2)="AR", "Employee", "Employee"))))))  
This formula divides jobs into "Warehouse", "Outsource", or "Employee", based on the job's payment\_ref.

# PYTHON (Pandas)

## 1. Data Loading and Inspection

First, I loaded the data and checked for missing values using pandas:   
 import pandas as pd  
 # Load Data  
 df = pd.read\_excel('Excel//fact.xlsx')  
 # Display the column names and check for missing values  
 print(df.columns)  
 print(df.isnull().sum())   
Here, columns like approval\_date, closure\_date, and payment\_ref were found to have missing values.

## 2. Data Cleaning

Converted the approval\_date column to a datetime format and handled missing values:  
 df['approval\_date'] = pd.to\_datetime(df['approval\_date'], errors='coerce')  
 df = df.dropna(subset=['approval\_date'])  
Reduced the dataset to 8,406 rows.

## 3. Date Conversion for Analysis

Converted the rto\_validation and revenue\_month columns to month/year format to facilitate month-to-month analysis:  
 df['rto\_validation'] = pd.to\_datetime(df['rto\_validation'], errors='coerce').dt.to\_period('M')  
 df['revenue\_month'] = pd.to\_datetime(df['revenue\_month'], errors='coerce').dt.to\_period('M')

## 4. Data Renaming

Renamed the columns for clarity and better understanding:  
 df.rename(columns={  
 'jobcode': 'Job\_Code',  
 'request\_date': 'Request\_Date',  
 # other renaming operations...  
 }, inplace=True)

## 5. EDA

We performed some basic exploratory analysis using pandas:

# Total number of requests (8406)  
 total\_requests = df.shape[0]  
 print(f"Total Number of Requests: {total\_requests}")

# Total number of closed jobs (7972)  
 total\_closed\_jobs = df[df['Job\_Status'] == 'Closed'].shape[0]  
 print(f"Total Number of Closed Jobs: {total\_closed\_jobs}")

# Total approved amount  
 total\_approved\_amount = df['Total\_Approved'].sum()  
 print(f"Total Approved Amount: {total\_approved\_amount}")

From our analysis, the total approved amount is approximately 604 million.

# SQL Section

## 1. Database Setup

Created a database for telecom maintenance:

CREATE DATABASE TelecomMaintenance;

## 2. Data Import and Joins

The cleaned\_data.csv file was imported into the nr table, and a join operation with the job class table (class) was performed:

SELECT nr.Request\_date, nr.Site\_ID, nr.Job\_Type, nr.Job\_Description, class.project  
 FROM nr  
 LEFT JOIN class ON nr.Job\_Description = class.fault;

This merged data with job classifications, filling in missing values using COALESCE:

COALESCE(class.project, 'Other Capex') AS project

## 3. Handling Unit Price Issues

Rows with a unit price of zero were excluded for accurate financial analysis:

SELECT \* FROM nr WHERE Unit\_Price <> 0;

## 4. EDA (sql)

Several analyses were conducted, such as checking project distributions and revenue trends:

# Most Frequent Jobs  
 SELECT TOP 10 Job\_Description, COUNT(\*) AS Job\_Count  
 FROM nr  
 GROUP BY Job\_Description  
 ORDER BY Job\_Count DESC;

# Revenue Trend  
 SELECT DATEPART(MONTH, Request\_Date) AS Month, SUM(Qty\_Approved \* Approved\_Unit\_Price) AS Monthly\_Revenue  
 FROM nr  
 GROUP BY DATEPART(MONTH, Request\_Date)  
 ORDER BY Monthly\_Revenue DESC;

# Average profit per job type ratio count of job  
 SELECT Job\_Type, COUNT(Job\_Type) AS Distribution, AVG(Qty\_Used \* Unit\_Price) AS Avg\_Expense, AVG(Qty\_Approved \* Approved\_Unit\_Price) AS Avg\_Revenue, (AVG(Qty\_Approved \* Approved\_Unit\_Price) - AVG(Qty\_Used \* Unit\_Price)) AS Profit, ((AVG(Qty\_Approved \* Approved\_Unit\_Price) - AVG(Qty\_Used \* Unit\_Price)) / COUNT(Job\_Type)) AS Ratio  
 FROM nr  
 GROUP BY DATEPART(MONTH, Request\_Date)  
 ORDER BY Monthly\_Revenue DESC;

# Power BI

## 1. Import Data from SQL

Provided Power BI with Server and database details, used direct query. The imported data is named “non routine”

## 2. Transformation in Power Query

Cleanup of non routine table (fact table) in power query, also addition of another table “matrix” (dimension table)   
# Used proper data types for each column (But we found out we could make these changes with direct query, so we switch to import mode)  
# Two new columns made in power query for Total Expenses and Total Approval. Used “Columns from example” to reduce the size of the site\_ids, since they all start with same 7 characters (example MYPROJ\_ABE\_0008B = ABE\_0008B), did the same in the matrix dimension table (this is needed to model the data)  
# Editing of some job type names to reduce vagueness, for example we have (site cleanup/fe, site cleanup/tower painting, spares – charging alternator), we would change them to just fire extinguisher, tower paining, charging alternator respectively.

## 3. Model the Data

# Create a calendar table for time intelligence purposes, Add Year, Month Quarter and other columns  
 Calender = CALENDAR(DATE(2022,01,01),TODAY())  
# Join the matrix dimension to the non-routine fact table on “site\_id” with a one to many relationship as we already made sure site\_ids are unique values, and can be used as primary key (one) to be linked with site\_ids in non routine table which occurs multiple times (many). Also joined the calendar to all date related columns in the non-routine table, one active and the rest inactive relationships, only one active relationship at a time.

## 4. Create Visualizations to Answer Business Questions

# The main reason for the project is about to be answered with a few visualizations and DAX expressions.

# [Profitability Analysis] Which job types and regions have generated most profit? *Approach: Calculate total revenue vs. total expense by job type and region.* – we create total revenue and total expense measures alongside profit measure, this will enable us to see top 5 and bottom 5, we discovered that tower painting has produced significantly most profit of 52m while replacement of oil seals have generated the least with 32k. We also discovered that Ogun region has produced most profit with 66m, closely followed by Kaduna with 51m.

Measures used  
Total Revenue = SUM('non routine (fact)'[Total\_Approved])  
Total Expense = SUM('non routine (fact)'[Total\_Expense])  
Profit = [Total Revenue] -  [Total Expense]  
A stack bar chart was used to depict this information  
# [Job Efficiency] What is the average time taken to complete different types of jobs? Approach: Analyze the time difference between job request and closure dates. – we create days to execute calculated column   
Days to execute = VAR DaystoExecute = DATEDIFF('non routine (fact)'[Approval\_Date], 'non routine (fact)'[Closure\_Date], DAY)  
RETURN IF(DaystoExecute < 0, 0, DaystoExecute)}.   
The average is matched with top type, we have handled the negative values But we have some extremely high values, some job types also have occurrence of less than 20 - so we consider those as outliers and filter those out. We discovered that the palisade fence repairs are the quickest to close with an average of 5 days while fortification took the longest with an average of 86 days. A bar chart was used to depict this information

# [Resource Allocation] Are there patterns in the frequency and types of job requests across different regions and times of the year? *Approach: Use scatter plots or trend analysis to identify hotspots and peak times.*– we create a scatter chart comparing the numbers of jobs per each month using the bubble size and separating in the different job types by colors, we discovered that site cleanup/environs is done on a big scale in January while site cleanup/fe, security lights and aviation light is also done on a big scale in June. December is where tasks are rarely done.

# [Cost Optimization] How can the company optimize the use of spares to reduce costs without compromising service quality? Approach: Identify spares with the highest usage and cost, then explore alternative procurement strategies i.e bulk purchasing. – we create a measure to check job closure count per month using   
JobsPerMonth\_ClosureDate = CALCULATE(COUNT('non routine (fact)'[Job\_Description]),USERELATIONSHIP('non routine (fact)'[Closure\_Date], 'calender (dim)'[Date]))  
We used “userelationship” because the active relationship is the request date. We used a cluster column chart to display the different jobs over the period of 12 months and used top N filter to show only top 5, we this we can see what items are mostly changed in each period and the company can properly plan

Project Summary  
  
This project aims to streamline and optimize telecom maintenance processes, classifying job types, analyzing costs, and improving resource allocation. Key objectives include detecting duplicated jobs, classifying faults and payments, cleaning and preparing data, and conducting data analyses using Excel, SQL, Pandas, and Power BI. By addressing inefficiencies in job handling, the project provides insights for profitability, job efficiency, and cost optimization.