



Informatica Intelligent Data Management Cloud

Course Project on Petrol Pump

Bachelor of Engineering

in

COMPUTER SCIENCE AND ENGINEERING

Submitted By

<i>Pragatilaxmi Itigowni</i>	<i>01FE22BCI013</i>
<i>Narendra Kaliwal</i>	<i>01FE22BCI047</i>
<i>Khushi Chalageri</i>	<i>01FE22BCI023</i>
<i>Apeksha Desai</i>	<i>01FE22BCS138</i>

Team No: 9

Faculty In charges

Mr. Channabasappa M

Ms.Neha.T

**DEPARTMENT OF COMPUTER SCIENCE (AI)&
ENGINEERING**

HUBLI-580 031 (India).

Academic year 2024-25

Table of Content

Chapters	Page No
1. Introduction 1.1 Preamble 1.2 Problem Definition 1.3 Objectives	3
2. ER diagram	4
3. Data set description	5-7
4. Transformations (screenshots) with explanation	8-23

5. Conclusion (insights from data – Business Intelligence)

13

Appendix

Roles and responsibility

1. Introduction

1.1 Preamble

Fuel is one of the most critical and high-value resources in the petroleum industry. Accurate tracking of fuel stock and sales is essential to ensure operational efficiency, cost control, and to prevent losses due to leakage, theft, or reporting errors. With the growing need for automation and digital monitoring, data-driven solutions have become increasingly important. Informatica Intelligent Cloud Services (IICS) provides powerful data transformation capabilities that enable real-time discrepancy detection. This project leverages IICS to analyze daily fuel stock and sales data, aiming to identify anomalies and ensure accurate inventory records.

1.2 Problem Definition

In many fuel distribution centers and petrol pumps, inconsistencies often arise between the recorded opening stock, the quantity of fuel sold, and the reported closing stock. These discrepancies may result from theft, leakage, human error, or system faults. Without an automated system to detect and flag such mismatches, significant losses can go unnoticed.

This project addresses the lack of a mechanism to automatically identify discrepancies where:

$$\text{Expected_Closing_Stock} \neq \text{Reported_Closing_Stock}$$

1.3 Objectives

1. Detect Discrepancies & Anomalies:

Identify mismatches in stock, revenue, and branch performance using automated checks.

2. Track Profitability & Expenditure:

Calculate daily profits and flag unusually high expenses for financial control.

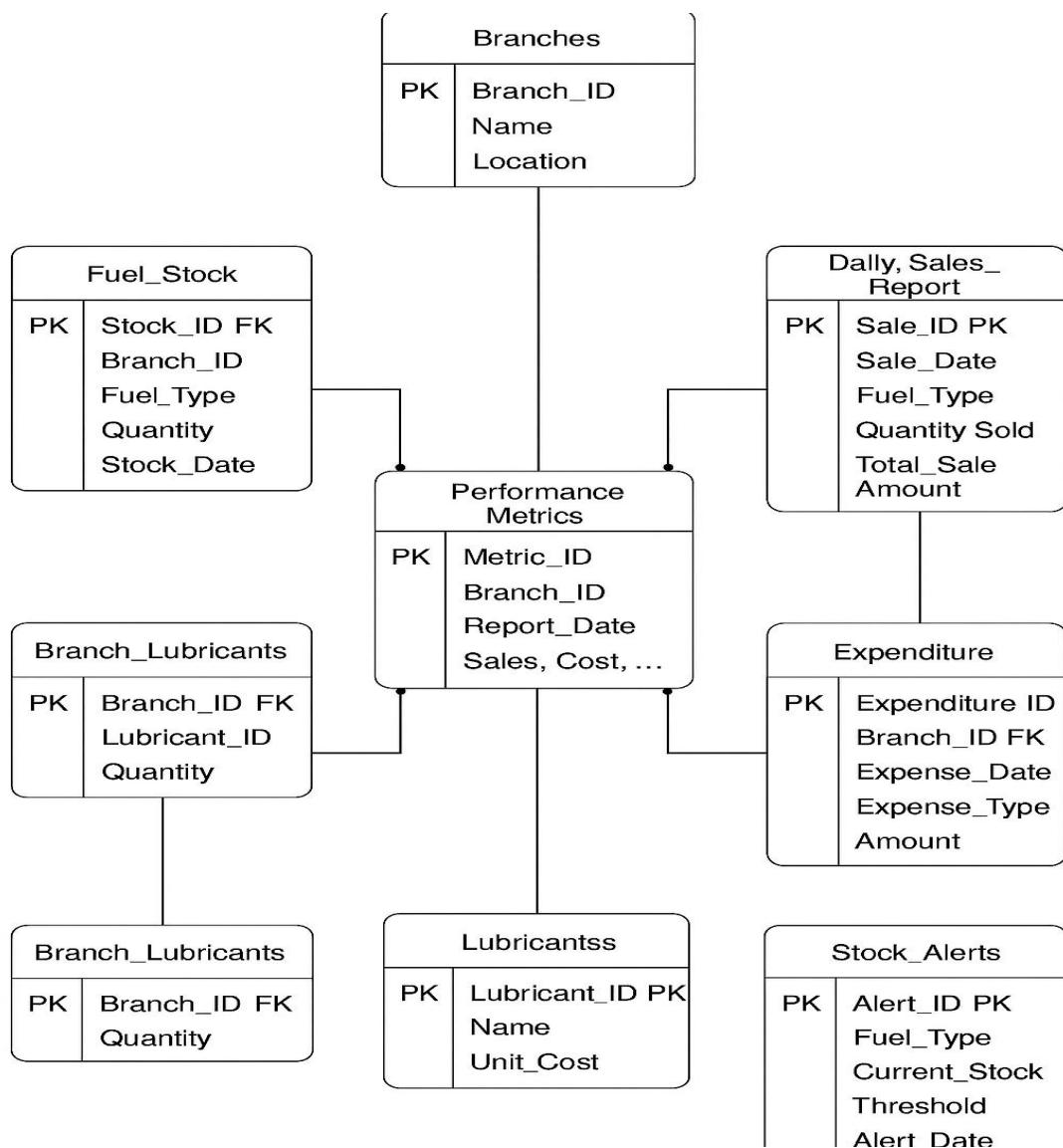
3. Manage Inventory Smartly:

Automate lubricant reordering and stock alerts to ensure smooth operations.

4. Generate Actionable Insights:

Integrate and analyze operational data for reports, customer insights, and branch rankings.

2. ER Diagram



3. Table Relationships Explanation.

1. Branches

- **Primary Key:** Branch_ID
 - Central table referenced by multiple other tables.
- **Related To:**
 - Fuel_Stock, Daily_Sales_Report, Expenditure, Branch_Lubricants, Performance_Metrics
- **Purpose:** Stores basic details of each petrol pump branch.
- **Relationships:**
 - One-to-Many** with Fuel_Stock (one branch has many fuel stock entries)
 - One-to-Many** with Daily_Sales_Report (one branch reports multiple sales)
 - One-to-Many** with Expenditure (one branch incurs multiple expenses)
 - One-to-Many** with Branch_Lubricants (one branch stores many lubricants)
 - One-to-Many** with Performance_Metrics (one branch has multiple performance records)

2. Fuel_Stock

- **Foreign Key:** Branch_ID → Branches
- **Related To:** Performance_Metrics
- **Purpose:** Tracks daily fuel stock levels per branch and fuel type.
- **Relationships:**

Many-to-One with Branches

Many-to-One with Performance_Metrics (implied through aggregation)

3. Daily_Sales_Report

- **Foreign Key:** Branch_ID
- **Related To:** Performance_Metrics
- **Purpose:** Records daily fuel sales by fuel type, including quantity and revenue.
- **Relationships:**

Many-to-One with Branches

Many-to-One with Performance_Metrics (sales data contributes to performance metrics)

4. Expenditure

- **Foreign Key:** Branch_ID → Branches
- **Purpose:** Captures branch-level expenses, categorized by type and date.

Relationships:

Many-to-One with Branches

Many-to-One with Performance_Metrics (expense data contributes to performance)

5. Branch_Lubricants

- **Composite Primary Key:** Branch_ID + Lubricant_ID
- **Foreign Keys:**
 - Branch_ID → Branches
 - Lubricant_ID → Lubricants
- **Purpose:** Tracks quantity of each lubricant stocked at each branch.

Relationships:

Many-to-One with Branches

Many-to-One with Lubricants

Many-to-Many relationship between Branches and Lubricants (via this junction table)

6. Lubricants

Primary Key: Lubricant_ID

Purpose: Stores master data for lubricants, including name and unit cost.

Relationships:

One-to-Many with Branch_Lubricants (a lubricant can be stocked at many branches)

7. Stock_Alerts

No foreign keys explicitly shown, but logically connected via Fuel_Type

Purpose: Alerts system when fuel stock falls below the defined threshold.

Relationships

Many-to-One with Fuel_Stock or Branches (each alert is for one branch/fuel type)

8. Performance_Metrics

Foreign Key: Branch_ID → Branches

Purpose: Aggregates daily performance data (sales, costs, etc.) from Fuel_Stock, Daily_Sales_Report, and Expenditure for branch-level insights.

Relationships:

Many-to-One with Branches

One-to-One or Aggregated relationship with data from:

Fuel_Stock, Daily_Sales_Report, and Expenditure

4. Dataset Description Petrol Pump Database

No.	File Name	Description	Key Columns
1	Branches.csv	Records daily fuel stock quantities at each branch.	Branch_ID, Branch_Name, Location
2	Fuel_Stock.csv	Records daily fuel stock quantities at each branch.	Stock_ID, Branch_ID, Fuel_Type, Quantity, Stock_Date
3	Daily_Sales_Report.csv	Details of daily fuel sales transactions across branches.	Sale_ID, Branch_ID, Sale_Date, Fuel_Type, Quantity_Sold, Total_Sale_Amount
4	Expenditure.csv	Tracks expenses incurred by each branch with type and amount.	Expenditure_ID, Branch_ID, Expense_Date, Expense_Type, Amount
5	Lubricants.csv	Defines lubricant types and their unit costs.	Lubricant_ID, Lubricant_Name, Unit_Cost
6	Branch_Lubricants.csv	Mapping of lubricants stocked in each branch.	Branch_ID, Lubricant_ID, Quantity
7	Performance_Metrics.csv	Captures daily performance including sales, expenditure, and profit.	Metric_ID, Branch_ID, Report_Date, Daily_Sales, Daily_Expenditure, Profit
8	Stock_Alerts.csv	Alerts generated when	Alert_ID,

		fuel stock falls below a threshold.	Branch_ID, Fuel_Type, Current_Stock, Threshold, Alert_Date
--	--	-------------------------------------	--

Purpose: To detect fuel stock discrepancies across branches using data transformations in Informatica IICS, enabling early identification of wastage, theft and data entry errors.

4. Transformations

Scenario 1: Intelligent Reordering System for Lubricants.

- **Objective**

To implement an automated, data-driven lubricant reordering system by analyzing weekly sales trends and current stock, ensuring timely alerts and minimizing the risk of stockouts.

- **Tables Involved**

Table Name
Branch_Lubricants
Dally_Sales_Report
Lubricantss
Reorder_Alerts

- **Transformations Used in IICS**

Transformation	Purpose
Aggregator	Groups lubricant sales data by brand; calculates Weekly_Sales.
Expression	Computes Average_Consumption_Rate = Weekly_Sales / 7; calculates Days_Left = Current_Stock / Average_Consumption_Rate.
Filter	Filters records where Days_Left < 5.
Update Strategy	Inserts flagged records into the Reorder_Alerts table.

- **Business Use Cases**

Use Case	Value Provided
Timely lubricant restocking	Prevents service delays due to stockouts.
Automated reorder flagging	Reduces manual monitoring efforts.

Use Case	Value Provided
Trend-based stock management	Matches procurement to actual consumption behavior.

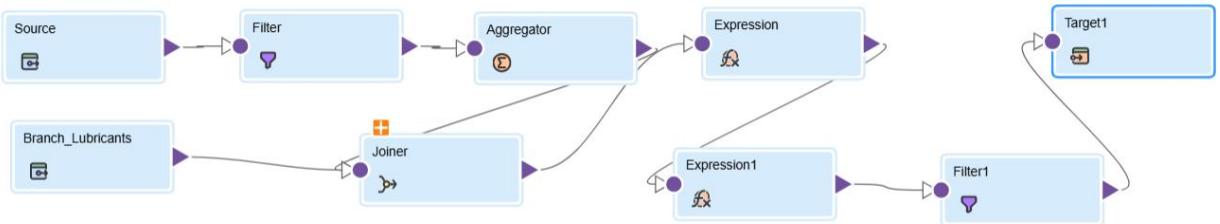


FIGURE 1: Automates lubricant reordering by analyzing sales, stock, and days left to trigger low-stock alerts.

A	B	C
Branch_ID	Lubricant_	Quantity
0	1	21
1	2	74
2	5	81
3	2	63
4	2	85

Scenario 2: Branch-Level Stock Alert

- **Objective**

To identify and classify low-stock situations at branch level using stock percentage metrics, ensuring early intervention and continuous fuel availability.

- **Tables Involved**

Table Name
Fuel_Stock
Branches
Stock_Alerts

- **Transformations Used in IICS**

Transformation	Purpose
Expression	Calculates Stock_Percent = (Closing_Stock / Opening_Stock) * 100.
Filter	Filters records where Stock_Percent < 10, indicating low stock.
Router	Classifies records into:

● Business Use Cases

Use Case	Value Provided
Early detection of low fuel stock	Prevents branch-level service disruption.
Categorized alerting system	Enables prioritized refueling actions.
Automated monitoring	Reduces manual stock checks and human error.



FIGURE 2: Identifies and classifies low fuel stock levels using stock percentage, routing alerts for early action.

A	B	C	D	E	F	G
Brand	Opening_Stock	Closing_Stock	Sales_Liter	Collection_Stock	Stock_Percent	
MSD 1	60088	2442	63543	224938	4.06	
MSD 2	70891	1881	47538	992286	2.65	
HSD 2	8504	295	166737	280717	3.47	

Scenario 3: Fuel Wastage Detection

Objective

To detect and prioritize fuel inventory discrepancies by comparing expected and recorded closing stock, enabling timely resolution of losses due to operational or human errors.

● Tables Involved

Table Name
Fuel_Stock
Dally_Sales_Report
Performance_Metrics (optional)

- **Transformations Used in IICS**

Transformation	Purpose
Expression	Calculates two key metrics: • Expected_Closing = Opening_Stock - Sales • Discrepancy = Expected_Closing - Closing_Stock This quantifies the difference between expected and actual stock.
Filter	Filters records where Discrepancy > 0, indicating a potential loss or inconsistency in fuel stock. Only mismatched records are passed to the next step.
Sorter	Sorts the filtered records in descending order of discrepancy to prioritize investigation of the most severe mismatches first.

- **Business Use Cases**

Use Case	Value Provided
Detecting hidden losses in fuel inventory	Flags discrepancies from leakage, theft, or manual error.
Prioritizing investigations	Helps identify branches with the largest stock mismatches.
Ensuring inventory integrity	Strengthens accuracy in stock tracking and reporting.

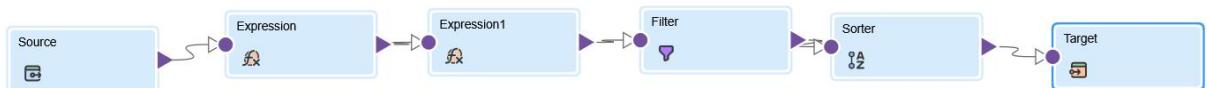


FIGURE 3: Detects and prioritizes fuel stock discrepancies by calculating, filtering, and sorting mismatch values.

A	B	C	D	E	F	G
BL_Brand	BL_Opening	BL_Closing	BL_Sales	BL_Collectio	BL_Expecte	Discrepancy
HSD 1	198159	43520	2071	102586	196088	152568
MSD 1	197559	53990	24187	361759	173372	119382
MSD 1	179553	16788	51777	122561	127776	110988
MSD 2	193619	8925	76258	759993	117361	108436
HSD 1	197658	81273	9591	867096	188067	106794
MSD 2	162153	12339	44298	662336	117855	105516
MSD 1	158806	15467	38811	499392	119995	104528
HSD 1	134051	31779	1874	873512	132177	100398
HSD 1	142435	26263	17721	45589	124714	98451

Scenario 4: Daily Net-Profit Calculation

- Objective

To automate the calculation of daily net profit for each branch by aggregating revenue and cost data, enabling accurate financial reporting and performance tracking.

- Tables Involved

Table Name
Sales_Data
Expenditure_Data
Daily_Sales_Report

- Transformations Used in IICS

Transformation Type	Purpose
Aggregator	-Aggregate sales data by date and branch to calculate Total_Revenue = Cash_Sale + Credit_Sale + HPCL + IDFC - Aggregate expenditure data by date and branch to calculate Total_Cost = Fixed_Cost + Variable_Cost
Expression	Calculate Net_Profit = Total_Revenue - Total_Cost for each branch on a daily basis

- Business Use Cases

Use Case	Description
Profitability Analysis	Track daily net profit per branch to identify profitable and underperforming locations
Financial Reporting	Automate daily sales and profit reporting to improve accuracy

Use Case	Description
Automation	and reduce manual efforts
Strategic Decision Support	Provide branch-wise profitability data to support budget allocation, cost control, and pricing strategies
Performance Comparison	Enable comparison of revenue, costs, and net profit across multiple branches over time

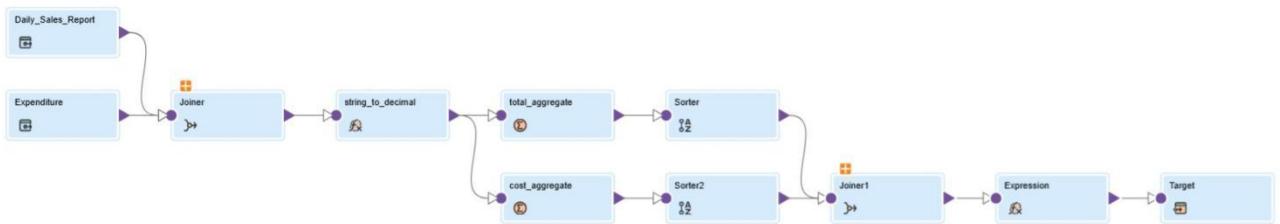


FIGURE 4: Automate daily net profit calculation per branch by aggregating revenue and cost data for precise financial reporting and performance insights.

A	B	C	D
total_cost	Date	Total_Revenue_Num	Net_Profit
25441.65	16-04-2025	89817.2	64375.55
30964.12	19-04-2025	99341.07	68376.95
7679.86	20-04-2025	29297.18	21617.32
32764.5	21-04-2025	72483.06	39718.56
27414.72	22-04-2025	109439.8	82025.08
14553.52	26-04-2025	22895.5	8341.98
3674.96	29-04-2025	28335.52	24660.56
12147.78	30-04-2025	33495.51	21347.73
10879.64	01-05-2025	77601.74	66722.1
40177.24	02-05-2025	142390.38	102213.1
8495.86	04-05-2025	34966.23	26470.37
28217.86	05-05-2025	35809.55	7591.69
21067.26	07-05-2025	63486	42418.74
6682.82	08-05-2025	30784.37	24101.55
8652.64	10-05-2025	8801.58	148.94
4442.6	11-05-2025	15506.97	11064.37
43768.64	13-05-2025	194318.96	150550.3

Scenario 5: Identifying Orders with Pending Payments

Objective

To automatically detect and flag orders with large payment mismatches by comparing sales data with payment collections, improving financial integrity and fraud detection.

● Tables Involved

Table Name	Description
Fuel_Stock	Contains daily fuel sales volume and pricing data (e.g., Sales_Liters, Avg_Price_Per_Liter)
Daily_Payment	Records daily collected payment amounts for fuel sales

Table Name	Description
Fraud_Report	Target table to store flagged transactions with significant revenue mismatches

● Transformations Used in IICS

Transformation Type	Purpose
Joiner	Joins Fuel_Stock and Daily_Payment on Date to align sales and payment data
Expression	Calculates Expected_Revenue and Revenue_Difference using: Expected_Revenue = Sales_Liters * Avg_Price_Per_Liter Revenue_Difference = ABS(Expected_Revenue - Collected_Amount)
Filter	Filters records where Revenue_Difference > 10,000 to identify suspicious transactions
Sorter	Sorts the filtered records in descending order of Revenue_Difference to prioritize review

● Business Use Cases

Use Case	Description
Fraud Detection	Identifies discrepancies between expected and actual revenue to catch potential fraud or manipulation
Payment Reconciliation	Helps in reconciling sales records with collected payments across branches
Audit and Compliance	Supports audits by highlighting financial mismatches for further investigation
Operational Oversight	Provides daily oversight to detect anomalies early and reduce revenue leakage

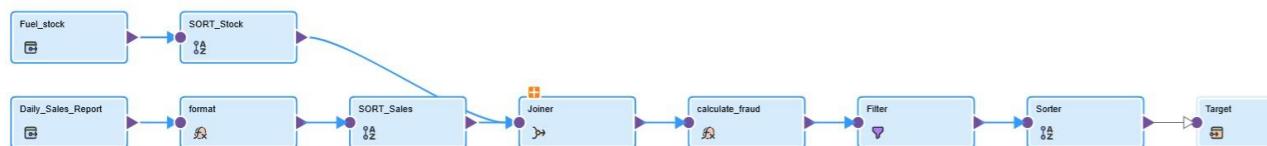


FIGURE 5: Automatically flag orders with major payment mismatches by comparing sales and collections, enhancing financial accuracy and fraud detection.

A	B	C	D	E	F	G	H	I
Sale_ID	Branch_ID	Fuel_Type	Quantity_S	Total_Sale	SaleDate	Price_Per_L	Revenue_D	Fraud_Flag
27	4	Diesel	446	19962.69	04/22/2025	1	1441	y
48	2	Petrol	132	13297.36	04/21/2025	1	2031	y
47	5	Petrol	177	9625.23	#####	1	3867	y

Scenario 6: Branch Performance Ranking

Objective

To calculate and rank branches based on revenue per litre of fuel sold, enabling identification of high-performing locations and improvement areas.

- Tables Involved

Table Name	Description
FUEL_STOCK	Contains daily fuel sales data per branch, including Sales_Liters and Collection
Branch_Performance_Report	Target table to store ranked branch performance data with Revenue_Per_Liter metric

- Transformations Used in IICS

Transformation Type	Purpose
Source	Reads raw data from the FUEL_STOCK table
Aggregator	Groups data by Branch_ID and calculates: - Total_Collection = SUM(Collection) - Total_Sales_Liters = SUM(Sales_Liters)
Expression	Calculates Revenue_Per_Liter = Total_Collection / Total_Sales_Liters
Rank	Ranks branches based on Revenue_Per_Liter in descending order to highlight top performers
Router	Segments branches into performance tiers (e.g., Top 10%, Next 20%, Others)
Target	Loads the ranked and categorized data into Branch_Performance_Report

- Business Use Cases

Use Case	Description
Performance Benchmarking	Identifies branches generating the most revenue per litre of fuel sold
Incentive Planning	Helps design reward programs for top-performing branches
Operational Efficiency Monitoring	Assesses revenue efficiency and flags underperforming branches

Use Case	Description
Strategic Resource Allocation	Guides management in deploying resources or support based on performance

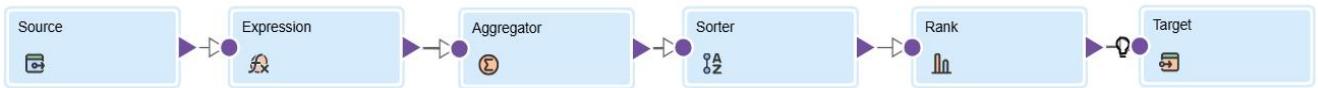


FIGURE 6: Rank branches by revenue per litre to identify top performers and target areas for improvement.

Branch_ID	Avg_Revenue_Per_Litre	Total_Sale_Amount	Quantity_Sold	Rank
3	105.34	47,702.69	453	1
5	91.91	18,749.94	204	2
2	50.04	19,113.35	382	3
1	24.41	5,881.74	241	4

Scenario 7: Daily Payment Spike Detection

- **Objective**

To detect and report abnormal payment spikes by analyzing deviations from historical patterns using Z-scores, enhancing operational transparency and fraud detection

- **Tables Involved**

Table Name	Description
SALES_DATA	Contains daily sales data per branch, including payment types: Cash, Credit, HPCL, IDFC
Payment_Spike_Report	Target table that stores records of abnormal spikes detected in payment types

- **Transformations Used in IICS**

Transformation Type	Purpose
Source	Reads daily payment data from SALES_DATA
Normalize	Unpivots columns (Cash, Credit, HPCL, IDFC) into individual rows with fields: Date, Payment_Type, Amount

Transformation Type	Purpose
Aggregator	Groups data by Payment_Type to compute: - Mean - Standard Deviation (StdDev) of Amount
Expression	Calculates Z_Score = (Amount - Mean) / StdDev to measure deviation from typical values
Filter	Filters records where Z_Score > 2, flagging days with abnormal spikes in payment amounts
Target	Loads the flagged records into Payment_Spike_Report for further review or alerting

● Business Use Cases

Use Case	Description
Anomaly Detection	Detect unusual payment behavior that could indicate errors or fraud
Peak Sales Monitoring	Identify days with significantly high activity in a specific payment mode
Financial Oversight	Helps finance teams investigate irregularities and understand sales trends
Business Event Tracking	Correlate spikes with promotions, events, or operational anomalies

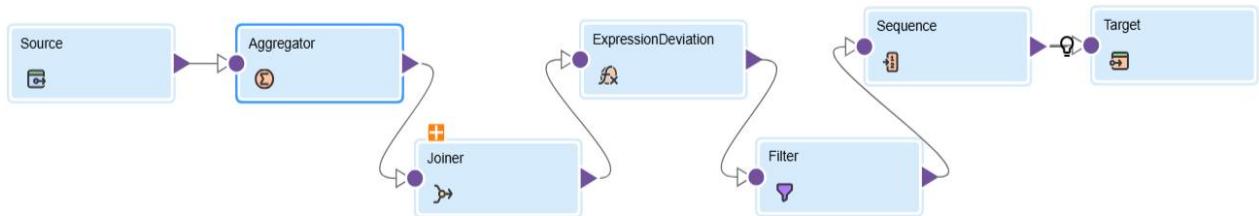


FIGURE 7: Detect abnormal payment spikes using Z-score analysis to boost transparency and uncover potential fraud.

A	B	C	D	E	F	G	H	I
Metric_ID	Branch_ID	Report_Dat	Daily_Sales	Daily_Exper	Profit	Avg_profit	Deviation(%)	Anomaly
1	1	07-05-2025	101843.94	34905.35		66938.59	6000	200% Yes
2	4	23-04-2025	94204.1	12064.47		82139.63	2000	-175% Yes

Scenario 8: Real-Time Budget Deviation Alerts

Objective

To automate detection of budget overruns by comparing actual daily costs against planned budgets and generate categorized alerts for timely financial intervention.

- **Tables Involved**

- **Transformations Used in IICS**

Transformation Type	Purpose
Source	Reads daily_Expenditure data from the source object, such as EXPENDITURE_DATA.
Expression	Adds a new field Alert_Flag using a conditional expression: Alert_Flag = IF(daily_Expenditure > 40000, 'YES', 'NO')
Target	Loads the transformed records, including the Alert_Flag, into the target object for storage or reporting.

- **Business Use Cases**

Use Case	Description
Budget Monitoring	Ensures daily costs remain within the allocated budget
Financial Risk Control	Triggers alerts when overspending occurs, allowing timely corrective action
Vendor Spend Management	Tracks expenditure deviations by vendor for contract and cost control
Alert-based Decision Support	Supports management in responding to cost spikes proactively



FIGURE 8: Automatically detect budget overruns and trigger categorized alerts for timely financial control.

Metric_ID	Branch_ID	Report_Date	Daily_Sale	Daily_Expense	Profit	Alert_flag
1	1 #####	101843.9	34905.35	66938.59	No	
2	2 #####	94204.1	12064.47	82139.63	No	
3	3 #####	63101.32	21521.17	41580.15	No	
4	4 #####	52886.94	35200.89	17686.05	No	
5	2 #####	65837.82	32033.56	33804.26	No	
6	5 #####	59320.92	49690.29	9630.63	Yes	
7	1 #####	144706.1	35908.39	108797.7	No	
8	2 #####	143265.3	21674.46	121590.9	No	

Scenario 9: Lubricant Inventory Upsert

Objective

To streamline lubricant inventory management by automating inserts and updates based on data comparisons, minimizing errors and manual effort while maintaining stock accuracy.

- Tables Involved

Table Name	Description
New_Lubricant_Data	Incoming lubricant records with details like Lubricant_ID, Lubricant_Name, Cost
Existing_Inventory	Current lubricant stock records with existing cost and identifiers
Updated_Inventory	Target table where inserts and updates are applied to maintain accurate inventory

- Transformations Used in IICS

Transformation Type	Purpose
Joiner	Joins New_Lubricant_Data with Existing_Inventory on Lubricant_ID or Lubricant_Name to identify matching records
Expression	Evaluates record status: - Is_New = ISNULL(Existing.Lubricant_ID) - Is_Updated = NOT ISNULL(Existing.Lubricant_ID) AND New.Cost != Existing.Cost - Is_Same = NOT ISNULL(Existing.Lubricant_ID) AND New.Cost = Existing.Cost
Router	Routes records into three flows: - New records for insertion - Updated records for update - Same records ignored

Transformation Type	Purpose
Update Strategy	Applies data operations: - Inserts flagged as DD_INSERT - Updates flagged as DD_UPDATE - Ignores unchanged records from further processing
Target	Loads processed records into Updated_Inventory to maintain accurate lubricant stock

● Business Use Cases

Use Case	Description
Automated Inventory Updates	Ensures new lubricants are added, changed costs are updated, and unchanged records are skipped
Data Consistency and Accuracy	Prevents redundant data writes, reducing errors and improving data integrity
Efficient Stock Management	Facilitates timely and precise lubricant stock updates supporting procurement and pricing strategies
Operational Efficiency	Reduces manual inventory checks and accelerates update cycles

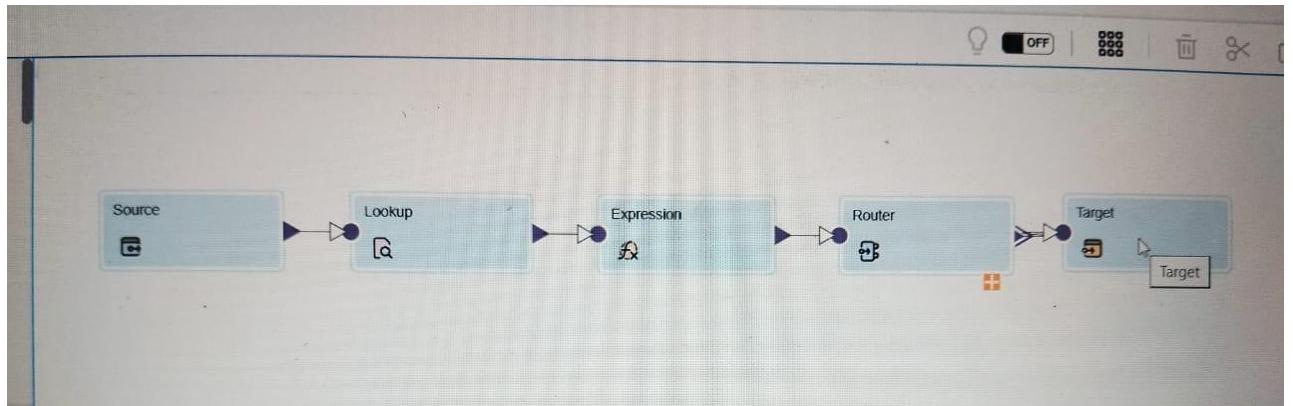


FIGURE 9: Automate lubricant inventory updates to reduce errors, cut manual effort, and ensure stock accuracy.

Branch_ID	Lubricant_Quantity	Old_cost	New_cost	Action
1	1 Lube_1	100	150	155.82 Update Price
2	2 Lube_2	80	135	141 Update Price
3	5 Lube_5	60	330	336.2 Update Price

Scenario 10: Branch-wise Stock Summary Report

Objective

To generate a consolidated stock summary for each branch using aggregation of fuel and lubricant data, enabling efficient monitoring and reporting of inventory trends.

- **Tables Involved**

Table Name
FUEL_STOCK
LUBRICANT_STOCK
Branch_Stock_Report

- **Transformations Used in IICS**

Transformation Type	Purpose
Source	Reads fuel and lubricant stock data from FUEL_STOCK and LUBRICANT_STOCK tables
Aggregator	Groups data by Branch_ID and calculates: $\begin{aligned} \text{Total_Opening_Stock} &= \text{SUM}(\text{Opening_Stock}) \\ \text{Total_Sales} &= \text{SUM}(\text{Sales}) \\ \text{Total_Closing_Stock} &= \text{SUM}(\text{Closing_Stock}) \\ \text{Total_Lubricants} &= \text{SUM}(\text{Lubricant_Quantity}) \end{aligned}$
Target	Loads the summarized data into Branch_Stock_Report for reporting

- **Business Use Cases**

Use Case	Description
Daily/Weekly Branch Reporting	Generates branch-level stock summaries to monitor fuel and lubricant usage trends
Inventory Monitoring	Tracks opening, closing stock, and sales for performance analysis
Supply Chain Planning	Supports restocking decisions and procurement planning
Branch Performance Comparison	Enables comparison of stock movement across branches

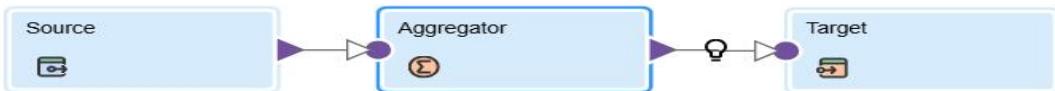


FIGURE 10: Consolidated stock summary per branch using aggregated fuel and lubricant data for efficient inventory monitoring and trend reporting.

Branch_ID	Total_Stock
1	4500
2	3000
3	5200
4	2000
5	4800

Conclusion:

This project used **Informatica IICS** to detect and analyze fuel stock discrepancies, financial fraud, stock alerts, and branch performance. Through automated data transformations, we gained key insights like potential theft, delayed deliveries, profit tracking, and customer segmentation—supporting **data-driven decision making** in the fuel management domain.

Appendix

Tool Used: Informatica Intelligent Cloud Services (IICS)

Data Sources: Fuel stock, sales, expenses, lubricants, customer orders

Transformations: Aggregator, Expression, Filter, Lookup, Sorter, Joiner, Router

Key Scenarios:

Fuel wastage detection

Reorder alerts for lubricants

Revenue discrepancy detection

Branch performance ranking

Daily net profit & sales report generation.

Roles and Responsibilities:

Team Member	Responsibility
Pragatilaxmi Itigowni	Reordering Lubricants, Branch-Level Stock Alter, Fuel Wastage Detection
Narendra Kaliwal	Daily Net-Profit Calculation, Identifying Orders with Pending Payments, Branch Performance Ranking
Khushi Chalageri	Branch Performance Ranking, Daily Payment Spike Detection, Real-Time Budget Deviation Alerts
Apeksha Desai	Lubricant Inventory Upsert, Trigger an Alert Flag When Daily Expenditure Crosses the Threshold of ₹40,000