

# Low-Level Design (LLD) – Warehouse Delivery Status Analysis

**Domain:** Logistics & Supply Chain

**Assessment Type:** Milestone

**Difficulty:** Medium–High | **Total Marks:** 20

**Concepts Covered:**

- DataFrame creation
  - Multi-column grouping
  - Date parsing and transformation
  - Filtering and data validation
  - Missing value handling
  - Data summarization and pivoting (unstack)
- 

## Problem Statement

You are working with a logistics company analyzing warehouse delivery performance to monitor operational efficiency, identify problem areas, and optimize delivery chains. Each record includes delivery ID, source warehouse, delivery date, and delivery status. Your goal is to extract insights, identify high-risk warehouses, and clean bad data using Pandas.

---

## Functions to Implement

---

### 1. Create Delivery DataFrame

```
def create_delivery_df(delivery_data: list) -> pd.DataFrame:
```

**Input:**

```
[  
    [1001, "WH1", "2024-06-01", "Delivered"],  
    [1002, "WH2", "2024-06-01", "Delayed"]  
]
```

**Output Columns:**

DeliveryID, Warehouse, Date, Status

---

## **2. Monthly Delivery Rate Per Warehouse**

```
def compute_monthly_delivery_rate(df: pd.DataFrame) -> pd.DataFrame:
```

### **Logic:**

- Extract Month (YYYY-MM) from Date
  - Count "Delivered" records per warehouse per month
  - Calculate delivery rate = (Delivered count / Total count) \* 100
  - Group by Warehouse + Month and compute rate
- 

## **3. Add Delay Flag**

```
def add_delay_flag(df: pd.DataFrame) -> pd.DataFrame:
```

### **Output:**

- Add a column IsDelayed = 1 if Status == "Delayed" else 0
- 

## **4. Warehouses with High Returns**

```
def high_returns(df: pd.DataFrame, threshold: int) -> pd.DataFrame:
```

### **Logic:**

- Group by Warehouse
  - Count where Status == "Returned"
  - Return those with count > threshold
  - Include columns: Warehouse, Return Count
- 

## **5. Warehouse Delivery Status Summary**

```
def warehouse_status_summary(df: pd.DataFrame) -> pd.DataFrame:
```

### **Logic:**

- Group by Warehouse and Status
  - Use .unstack(fill\_value=0)
  - Return counts of each status per warehouse
  - Columns: Warehouse, Delivered, Delayed, Returned
-

## 6. Clean Delivery Data

```
def clean_delivery_data(df: pd.DataFrame) -> pd.DataFrame:
```

### Logic:

- Drop rows where Status is not in ["Delivered", "Delayed", "Returned"]
  - Drop rows with null Status values
  - Return cleaned DataFrame
- 

### Test Case Matrix

TC ID	Description	Function	Marks
TC1	Create structured delivery DataFrame	create_delivery_df()	2.5
TC2	Compute monthly delivery rate per warehouse	compute_monthly_delivery_rate()	2.5
TC3	Add delay flag column	add_delay_flag()	2.5
TC4	Identify warehouses with frequent returns	high_returns()	2.5
TC5	Warehouse-wise status summary	warehouse_status_summary()	2.5
HTC1	Clean records with invalid status and nulls	clean_delivery_data()	2.5
HTC2	Handle warehouse with perfect delivery rate	compute_monthly_delivery_rate()	2.5
HTC3	Warehouse with only one type of status	warehouse_status_summary()	2.5

---

### Example Input & Output (TC1)

```
create_delivery_df([
    [1001, "WH1", "2024-06-01", "Delivered"],
    [1002, "WH2", "2024-06-01", "Delayed"]
])
```

### Expected Output:

DeliveryID | Warehouse | Date | Status

1001 | WH1 | 2024-06-01 | Delivered

1002 | WH2 | 2024-06-01 | Delayed

---

## **Example Input & Output (TC2)**

### **Input DataFrame:**

DeliveryID | Warehouse | Date | Status  
1001 | WH1 | 2024-06-01 | Delivered  
1002 | WH1 | 2024-06-02 | Returned  
1003 | WH2 | 2024-06-01 | Delayed

### **Expected Output:**

Warehouse | Month | Delivery Rate  
WH1 | 2024-06 | 50.0  
WH2 | 2024-06 | 0.0

---

## **Example Input & Output (TC3)**

### **Input DataFrame:**

DeliveryID | Warehouse | Date | Status  
1001 | WH1 | 2024-06-01 | Delivered  
1002 | WH2 | 2024-06-01 | Delayed

### **Expected Output (with new column):**

DeliveryID | Warehouse | Date | Status | IsDelayed  
1001 | WH1 | 2024-06-01 | Delivered | 0  
1002 | WH2 | 2024-06-01 | Delayed | 1

---

## **Example Input & Output (TC4)**

### **Input DataFrame:**

DeliveryID | Warehouse | Date | Status  
1001 | WH1 | 2024-06-01 | Returned  
1002 | WH1 | 2024-06-02 | Returned  
1003 | WH2 | 2024-06-03 | Delivered

```
high_returns(df, threshold=1)
```

**Expected Output:**

Warehouse | Return Count

WH1 | 2

---

**Example Input & Output (TC5)**

Input DataFrame:

DeliveryID	Warehouse	Date	Status
1001	WH1	2024-06-01	Delivered
1002	WH2	2024-06-01	Delayed
1003	WH1	2024-06-02	Returned

**Expected Output:**

Warehouse	Delivered	Delayed	Returned
WH1	1	0	1
WH2	0	1	0

---

**Visible vs Hidden Test Cases**

**VISIBLE TESTS (5 tests, 12.5 marks total):**

- Straightforward test cases with clean data and standard scenarios
- Students see these and must pass to earn visible marks
- Marks: 2.5 each

**HIDDEN TESTS (3 tests, 7.5 marks total):**

- Edge cases and data quality scenarios
  - Test data robustness: invalid statuses, null values, single-status warehouses
  - Students can see the logic pattern from visible tests but must handle edge cases
-

**Valid Delivery Status Values:**

- Delivered (successful completion)
- Delayed (late delivery)
- Returned (item returned/rejected)

**Invalid/Rejected values:**

- "Lost", "Pending", None, null values, or any other status

**Clean Data Requirements:**

- All rows must have valid Status in [Delivered, Delayed, Returned]
- No null/None values in Status column