Manufacturing Operations Analysis using SQL

Project Overview:

Analyzed 1,000 manufacturing jobs across five machines using SQL to identify production bottlenecks, failure patterns, and energy inefficiencies. The project focused on transforming raw operational data into actionable insights that improve efficiency, reduce costs, and enhance delivery performance.

Key Findings & Strategic Recommendation:

- 48.5% of jobs experienced delays
- Grinding is the main bottleneck (avg. 73 processing units)
- Machine M03 is the least energy-efficient (8.76 units/job)
- **Strategic Recommendation:** Optimize the Grinding process and perform preventive maintenance on Machine M03 to reduce energy costs and improve production timelines

---

1. Data Cleaning

Queries:

```sql
-- Remove records with missing critical identifiers (Job_ID or Machine_ID)

DELETE FROM hms_manufacturing_data
WHERE Job_ID IS NULL
   OR Machine_ID IS NULL;
```

```sql
-- Assign a default status to jobs with missing Job_Status

UPDATE hms_manufacturing_data
SET Job_Status = 'Unknown'
WHERE Job_Status IS NULL;


-- Standardize text fields to ensure consistent grouping and
analysis

UPDATE hms_manufacturing_data
SET Operation_Type = UPPER(LTRIM(RTRIM(Operation_Type))),
    Optimization_Category =
LTRIM(RTRIM(Optimization_Category));


-- Correct invalid timestamps where Actual_End occurs before
Actual_Start

UPDATE hms_manufacturing_data
SET Actual_End = NULL
WHERE Actual_End < Actual_Start;


-- Remove invalid negative values for energy consumption

UPDATE hms_manufacturing_data
SET Energy_Consumption = 0
WHERE Energy_Consumption < 0;


-- Remove invalid negative values for material usage

UPDATE hms_manufacturing_data
SET Material_Used = 0
WHERE Material_Used < 0;


-- Remove duplicate Job_ID records while keeping the most
recent entry

WITH CTE AS (
    SELECT Job_ID,
           ROW_NUMBER() OVER (PARTITION BY Job_ID ORDER BY
Actual_Start DESC) AS DuplicateCount
    FROM hms_manufacturing_data
)
```

```sql
DELETE FROM CTE
WHERE DuplicateCount > 1;
```

---

2. Verification

Query:

```sql
SELECT
    COUNT(*) as Total_Jobs,
    AVG(Processing_Time) as Avg_Time,
    SUM(CASE WHEN Actual_End IS NULL THEN 1 ELSE 0 END) as
Incomplete_Jobs
FROM hms_manufacturing_data;
```

Output :

Total_Jobs	Avg_Time	Incomplete_Jobs
1000	71	129

Insight:

The dataset contains 1000 jobs with an average processing time of 71, while 12.9% of jobs are incomplete, indicating potential operational and data quality issues.

Recommendation:

Incomplete jobs should be investigated and excluded from performance calculations, and average processing time should be analyzed separately for completed jobs to ensure accurate efficiency measurement.

---

3. Key KPIs

3.1 Total Jobs & Machines

```
SELECT
    COUNT(*) AS Total_Jobs,
    COUNT(DISTINCT Machine_ID) AS Total_Machines
FROM hms_manufacturing_data;
```

Output:

Total_Jobs	Total_Machines
1000	5

Insight:

Each machine has 200 jobs, which may put high pressure on the machines if their capacity is lower than this number.

Recommendation:

Review machine capacity and distribute jobs according to machine efficiency to avoid delays

---

3.2 Cycle Time per Machine

```
SELECT
    Machine_ID,
    AVG(Processing_Time) AS Avg_Processing_Time
FROM hms_manufacturing_data
GROUP BY Machine_ID

ORDER BY Avg_Processing_Time DESC;
```

Output:

Machine_ID	Avg_Processing_Time
M02	74
M04	71
M01	70
M05	70
M03	69

Insight:.

M02 is slightly slower than the other machines, which may create minor delays in overall production. Most machines have similar processing times, so workload can be mostly balanced.

Recommendation:.

Consider assigning fewer jobs to M02 or optimize its operation to reduce processing time. Monitor machines regularly to maintain balanced production.

3.3 Schedule Adherence & Delay Percentage

```
SELECT
    COUNT(*) AS Total_Jobs,
    SUM(CASE
        WHEN Actual_End > Scheduled_End THEN 1
        ELSE 0
    END) AS Delayed_Jobs
FROM hms_manufacturing_data;
```

Output:

Total_Jobs	Delayed_Jobs
1000	485

```
--Percentage

SELECT
```

```sql
    CAST(SUM(CASE WHEN Actual_End > Scheduled_End THEN 1 ELSE 
0 END) * 100.0
         / COUNT(*) AS DECIMAL(5,2)) AS Delay_Percentage
FROM hms_manufacturing_data;
```

Output:

Delay_Percentage
48.50

Insight:.

Nearly half of the jobs (48.5%) were completed later than scheduled, indicating a significant delay issue in the production process. This could impact delivery timelines and customer satisfaction.

Recommendation:

Investigate the root causes of delays, such as machine inefficiency, resource shortages, or scheduling issues. Consider optimizing workflows, balancing workloads, and performing preventive maintenance on slower machines.

3.4 Energy Consumption per Machine

```sql
SELECT
    Machine_ID,
    AVG(Energy_Consumption) AS Avg_Energy,
    SUM(Energy_Consumption) AS Total_Energy
FROM hms_manufacturing_data
GROUP BY Machine_ID

ORDER BY Total_Energy DESC;
```

Output:

Machine_ID	Avg_Energy	Total_Energy
M01	8.452547	1791.94
M02	8.423428	1768.92
M04	8.538492	1699.16
M05	8.455854	1631.98
M03	8.759892	1629.34

Insight:

M01 consumed the most total energy, likely because it processed more jobs or operated longer. M03 has the highest average energy per job, suggesting it is less energy-efficient. Overall, energy consumption is fairly similar across machines.

Recommendation:

Monitor machines like M03 for energy efficiency improvements. Consider scheduling high-energy jobs on more efficient machines (like M02 or M01 if workload allows) and perform maintenance to reduce excessive energy usage.

3.5 Energy Efficiency per Operation

```
SELECT
    Operation_Type,
    AVG(Energy_Consumption / NULLIF(Processing_Time,0)) AS
Energy_per_Minute
FROM hms_manufacturing_data
GROUP BY Operation_Type

ORDER BY Energy_per_Minute DESC;
```

Output:

Operation_Type	Energy_per_Minute
DRILLING	0.1562121454984
ADDITIVE	0.1503438688309
LATHE	0.1472314864183
MILLING	0.1461505498023
GRINDING	0.1433594498832

Insight:

Drilling consumes the most energy per minute, making it the least energy-efficient operation. Grinding is the most efficient in terms of energy usage per minute. This suggests that certain operations inherently consume more energy regardless of machine efficiency.

Recommendation:.

Focus on optimizing high-energy operations like Drilling and Additive. Consider scheduling these operations on machines with better energy efficiency or investing in energy-saving technologies for these specific processes.

---

4. Business & Analytical Questions

4.1 Bottleneck Analysis

Which operation types have the longest average processing time and are causing production bottlenecks?

```
SELECT
    Operation_Type,
    AVG(Processing_Time) AS Avg_Processing_Time,
    COUNT(*) AS Job_Count
FROM hms_manufacturing_data
GROUP BY Operation_Type

ORDER BY Avg_Processing_Time DESC;
```

Output:

Operation_Type	Avg_Processing_Time	Job_Count
GRINDING	73	208
LATHE	71	212
DRILLING	71	189
MILLING	70	201
ADDITIVE	70	190

Insight:

Grinding has the longest average processing time (73 units), making it the **main bottleneck** in production.

Lathe and Drilling also take relatively long (71 units) and contribute to delays.

Milling and Additive are faster and less likely to cause bottlenecks.

Recommendation:

Focus on **optimizing Grinding operations** to reduce delays—consider better scheduling or machine upgrades.

Review Lathe and Drilling operations to improve efficiency.

Balance workload by assigning jobs from slower operations to machines with higher capacity if possible.

4.2 Machine Performance

Which machines are underperforming based on average processing time and energy consumption?

```
SELECT
    Machine_ID,
    AVG(Processing_Time) AS Avg_Processing_Time,
    AVG(Energy_Consumption) AS Avg_Energy
FROM hms_manufacturing_data
```

```sql
GROUP BY Machine_ID
ORDER BY Avg_Processing_Time DESC, Avg_Energy DESC;
```

Output:

Machine_ID	Avg_Processing_Time	Avg_Energy
M02	74	8.423428
M04	71	8.538492
M05	70	8.455854
M01	70	8.452547
M03	69	8.759892

Insight:

M02 has the **longest processing time (74)**, making it a potential bottleneck, even though its energy consumption is relatively moderate.

M03 is the **most energy-intensive per job (8.76 units)** but is the fastest machine (69 units).

M04, M05, and M01 have balanced processing times and energy, but M04 uses slightly more energy.

Recommendation:

*Investigate M02 to **reduce processing time** (e.g., optimize operations, improve scheduling).*

*Monitor M03 for **energy efficiency**, consider maintenance or operational improvements.*

*Consider balancing jobs across machines to **optimize both time and energy usage**.*

4.3 Job Completion Status

```
SELECT
    Job_Status,
    COUNT(*) AS Job_Count
FROM hms_manufacturing_data

GROUP BY Job_Status;
```

Output:

Job_Status	Job_Count
Failed	129
Delayed	198
Completed	673

Insight:

Majority of jobs (673 out of 1000) are ***completed successfully*** *(~67%).*

Delayed jobs *account for 19.8%, and* ***failed jobs*** *are 12.9%.*

While most jobs finish, nearly ***1/3 of jobs face issues****, which could affect production efficiency and delivery timelines.*

Recommendation:

Investigate ***failed jobs*** *to identify causes (machine errors, material issues, operator errors).*

Optimize scheduling and processes to ***reduce delayed jobs****.*

Monitor KPIs regularly to ***improve overall production efficiency***

4.4 Optimization Impact

Which optimization categories are associated with better performance?

```
SELECT
    Optimization_Category,
    AVG(Processing_Time) AS Avg_Processing_Time,
    AVG(Energy_Consumption) AS Avg_Energy
FROM hms_manufacturing_data
GROUP BY Optimization_Category

ORDER BY Avg_Processing_Time ASC;
```

Output:

Optimization_Category	Avg_Processing_Time	Avg_Energy
Moderate Efficiency	68	11.405464
High Efficiency	69	5.586211
Low Efficiency	72	8.492292
Optimal Efficiency	89	2.461666

Insight:

High Efficiency gives the best balance of processing time (69) and energy (5.59). Optimal Efficiency saves energy (2.46) but is slow (89). Low Efficiency is slow and consumes more energy

Recommendation:

Use High Efficiency operations for balanced performance. Optimize Low Efficiency tasks.

4.5 High-Cost Jobs

Which individual jobs have exceptionally high processing time or energy consumption?

```
SELECT
    top 10 Job_ID,
    Machine_ID,
    Processing_Time,
    Energy_Consumption
FROM hms_manufacturing_data
ORDER BY Processing_Time DESC, Energy_Consumption DESC;
```

Output:

Job_ID	Machine_ID	Processing_Time	Energy_Consumption
J407	M02	120	11.91
J755	M04	120	10.96
J987	M02	120	10.87
J773	M05	120	6.96
J257	M02	120	5.09
J660	M01	120	3.14
J575	M05	120	2.21
J111	M01	119	13.26
J796	M01	119	11.52
J469	M04	119	9.17

Insight:

Some jobs, especially J407, J755, and J987, have **exceptionally high processing time (120)** and high energy consumption (>10). These may indicate **machine overload, material issues, or complex operations**.

Recommendation :

Review these jobs individually to **identify causes of high time or energy**. Consider rescheduling, splitting tasks, or optimizing machine settings.

Final Summary

This SQL-based analysis identifies production bottlenecks, inefficient machines, schedule delays, and energy-intensive operations. The insights and recommendations provide guidance for process optimization, cost reduction, and capacity planning.

Insights:

1. Total of 1000 jobs were processed by 5 machines, with 48.5% of jobs delayed, indicating moderate production efficiency.
2. Grinding, Lathe, and Drilling are the main bottlenecks due to longest average processing times (69–73 units).
3. Machine M03 consumes the most energy per job (8.76) while M02 has the longest processing time (74), highlighting inefficiencies.
4. High Efficiency optimization category balances processing time (69) and energy (5.59), while Optimal Efficiency minimizes energy but slows production (89).
5. Some individual jobs (e.g., J407, J755, J987) have exceptionally high processing time and energy consumption, indicating machine overload or complex operations.

Recommendations:

1. Focus on optimizing bottleneck operations (Grinding, Lathe, Drilling) to reduce delays.
2. Monitor and improve energy efficiency on high-consuming machines like M03.
3. Use High Efficiency optimization category for best balance of speed and energy.
4. Investigate failed and delayed jobs individually to identify causes and improve process reliability.
5. Reschedule or split high-processing-time jobs (e.g., J407, J755) and consider machine maintenance or upgrades.

Data Pipeline Automation:

Overview: To ensure that the manufacturing insights—such as the 48.5% delay rate and 12.9% failure rate —remain accurate over time, I developed a stored procedure to automate the data cleaning process. This replaces manual updates and standardizes all operational metrics.

SQL Implementation:

```
CREATE PROCEDURE sp_CleanManufacturingData
AS
BEGIN
    SET NOCOUNT ON;

    -- 1. Remove records with missing critical identifiers
[cite: 7]
    DELETE FROM hms_manufacturing_data
    WHERE Job_ID IS NULL OR Machine_ID IS NULL;
```

```sql
    -- 2. Assign a default status to jobs with missing
Job_Status [cite: 10]
    UPDATE hms_manufacturing_data
    SET Job_Status = 'Unknown'
    WHERE Job_Status IS NULL;

    -- 3. Standardize text fields for consistent grouping
[cite: 14]
    UPDATE hms_manufacturing_data
    SET Operation_Type = UPPER(LTRIM(RTRIM(Operation_Type))),
        Optimization_Category =
LTRIM(RTRIM(Optimization_Category));

    -- 4. Correct invalid timestamps (End before Start) [cite:
20]
    UPDATE hms_manufacturing_data
    SET Actual_End = NULL
    WHERE Actual_End < Actual_Start;

    -- 5. Remove invalid negative values for energy and
material [cite: 24, 28]
    UPDATE hms_manufacturing_data
    SET Energy_Consumption = 0 WHERE Energy_Consumption < 0;

    UPDATE hms_manufacturing_data
    SET Material_Used = 0 WHERE Material_Used < 0;

    -- 6. Remove duplicate Job_ID records [cite: 32]
    WITH CTE AS (
        SELECT Job_ID,
               ROW_NUMBER() OVER (PARTITION BY Job_ID ORDER
BY Actual_Start DESC) AS DuplicateCount
        FROM hms_manufacturing_data
    )
    DELETE FROM CTE WHERE DuplicateCount > 1;

    PRINT 'Manufacturing Data Pipeline: Cleaning Successful.';
END;

GO
```

Final Summary & Business Impact

This SQL-based analysis provides a comprehensive overview of manufacturing efficiency. By identifying that 48.5% of jobs are delayed and pinpointing Grinding as the primary bottleneck (73 units avg. time), the factory can now implement targeted improvements.

Key Outcomes:

- **Operational Readiness**: The automated cleaning pipeline ensures that future KPIs are built on accurate, standardized data.
- **Cost Reduction**: Focused maintenance on Machine M03 (the highest energy consumer) and Machine M02 (the slowest performer) will directly reduce overhead costs.
- **Strategic Growth**: By prioritizing the "High Efficiency" category, the production line can achieve an optimal balance between speed and energy consumption.