# **Neo4j Work:** Graph Design, Adding Data, Cypher Queries

**Prepared by:** Mert OLÇAMAN

**Week:** 2&3

**Domain:** Industrial Manufacturing Analytics

**patika.dev**

**NewMind AI Bootcamp**

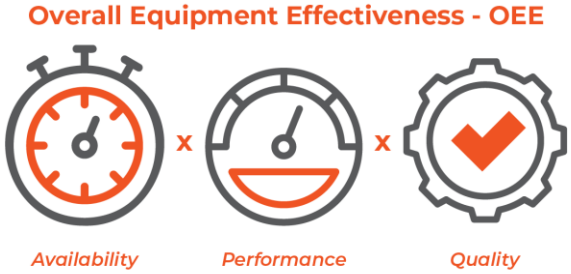
# INTRODUCTION

In this work, the data was generated by using the library called fake through Python. Operator and operation data was created seperately and groupped together afterwards, and some records can be seen in the table below. Additionally, the sample cypher query to create the nodes and relationships was provided into python and used loop to generate the all data code. Each type of cypher query to create this graph database can be seen in the separate text files in github directory. At first, constraints were given. Then, each node representation was created. At the end, relationships were specified. Each of the text file was run by different query input box, since the dataset is pretty long. Otherwise, I would get error while running all at the same time.

| EmployeeID | Name | Gender | DateOfBirth | OperationID | Part | Process | Shift | Machine | Date | Duration | OEE | Quality |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Employee1 | Danielle Johnson | M | 17/01/1983 | Operation1 | Piston | Milling | Day | M2 | 26/04/2025 | 64 | 0.68 | Medium |
| Employee1 | Danielle Johnson | M | 17/01/1983 | Operation19 | Piston | Assembly | Morning | M3 | 30/04/2025 | 38 | 0.73 | Medium |
| Employee1 | Danielle Johnson | M | 17/01/1983 | Operation25 | Gearbox | Drilling | Morning | M1 | 25/04/2025 | 31 | 0.61 | Medium |
| Employee1 | Danielle Johnson | M | 17/01/1983 | Operation32 | Shaft | Drilling | Morning | M1 | 29/04/2025 | 85 | 0.77 | Medium |
| Employee1 | Danielle Johnson | M | 17/01/1983 | Operation41 | Bearing | Painting | Night | M4 | 25/04/2025 | 179 | 0.63 | Low |
| Employee1 | Danielle Johnson | M | 17/01/1983 | Operation60 | Valve | Milling | Day | M2 | 25/04/2025 | 31 | 0.83 | Medium |
| Employee1 | Danielle Johnson | M | 17/01/1983 | Operation78 | Piston | Painting | Morning | M4 | 29/04/2025 | 171 | 0.7 | Medium |
| Employee1 | Danielle Johnson | M | 17/01/1983 | Operation81 | Bearing | Painting | Night | M4 | 28/04/2025 | 126 | 0.63 | Medium |
| Employee1 | Danielle Johnson | M | 17/01/1983 | Operation106 | Valve | Milling | Morning | M2 | 29/04/2025 | 176 | 0.79 | High |
| Employee1 | Danielle Johnson | M | 17/01/1983 | Operation169 | Shaft | Assembly | Morning | M3 | 30/04/2025 | 106 | 0.88 | Low |
| Employee1 | Danielle Johnson | M | 17/01/1983 | Operation170 | Gearbox | Milling | Day | M2 | 24/04/2025 | 146 | 0.68 | Medium |
| Employee1 | Danielle Johnson | M | 17/01/1983 | Operation171 | Valve | Milling | Morning | M2 | 25/04/2025 | 40 | 0.75 | Low |
| Employee1 | Danielle Johnson | M | 17/01/1983 | Operation234 | Shaft | Milling | Day | M2 | 30/04/2025 | 89 | 0.82 | Low |
| Employee2 | John Taylor | M | 22/04/1971 | Operation3 | Bearing | Painting | Morning | M4 | 26/04/2025 | 176 | 0.73 | High |
| Employee2 | John Taylor | M | 22/04/1971 | Operation33 | Piston | Milling | Morning | M2 | 28/04/2025 | 178 | 0.69 | Medium |
| Employee2 | John Taylor | M | 22/04/1971 | Operation36 | Bearing | Painting | Night | M4 | 25/04/2025 | 45 | 0.94 | Low |
| Employee2 | John Taylor | M | 22/04/1971 | Operation77 | Piston | Milling | Day | M2 | 27/04/2025 | 165 | 0.8 | High |
| Employee2 | John Taylor | M | 22/04/1971 | Operation115 | Valve | Milling | Night | M2 | 30/04/2025 | 65 | 0.9 | Low |
| Employee2 | John Taylor | M | 22/04/1971 | Operation125 | Gearbox | Painting | Day | M4 | 24/04/2025 | 60 | 0.64 | Medium |

In this work, how the manufacturing data in a factory can be used was showed. As a summarizon of the table; 5 parts are produced (piston, gearbox, shaft, bearing, valve) in 3 shifts (morning, day, night) by 20 operators using 4 different machines, each machine is used for different processes (milling, drilling, assembly, painting). Besides that, each part is checked after the production and the quality is specified as low, medium, and high. To understand the process better, duration, and oee values were also collected. Only one-month-period data was investigated.

**What is OEE?**



OEE stands for Overall Equipment Effectiveness, a parameter checked to figure out how effective the production goes. It can be thought as a combination of 3 different parameters, which makes the keeping up with the overall effectiveness easier.

$$OEE = Availability\ x\ Performance\ x\ Quality$$

$$Availability = \frac{Operating\ Time}{Total\ Spent\ Time}$$

$$Performance = \frac{Ideal\ Cycle\ Time\ x\ Total\ Pieces}{Operating\ Time}$$

$$Quality = \frac{Good\ Pieces}{Total\ Pieces}$$

The products of the terms (availability, performance, and quality) gives the value of OEE.

While calculating the value of availability, operating time is divided by total spent time (the stopping time is included). Therefore, that how much time is used effectively can be found.

As for performance, the expected time to produce a part (ideal cycle time) is multiplied by total producted parts to find the total expected spent time during operation. Then, it is divided by the operating time, not included the stops. The less time spent to produce a part, the higher value of performance.

Quality gives the good product rate - in other words, non-scrapping rate.

So, how the production process is effective can be checked thanks to the multiplication of these 3 values. If OEE value is less than expected, we can get a clue which part has a problem, which makes understanding the focusing area easier.

# GRAPH DESIGN

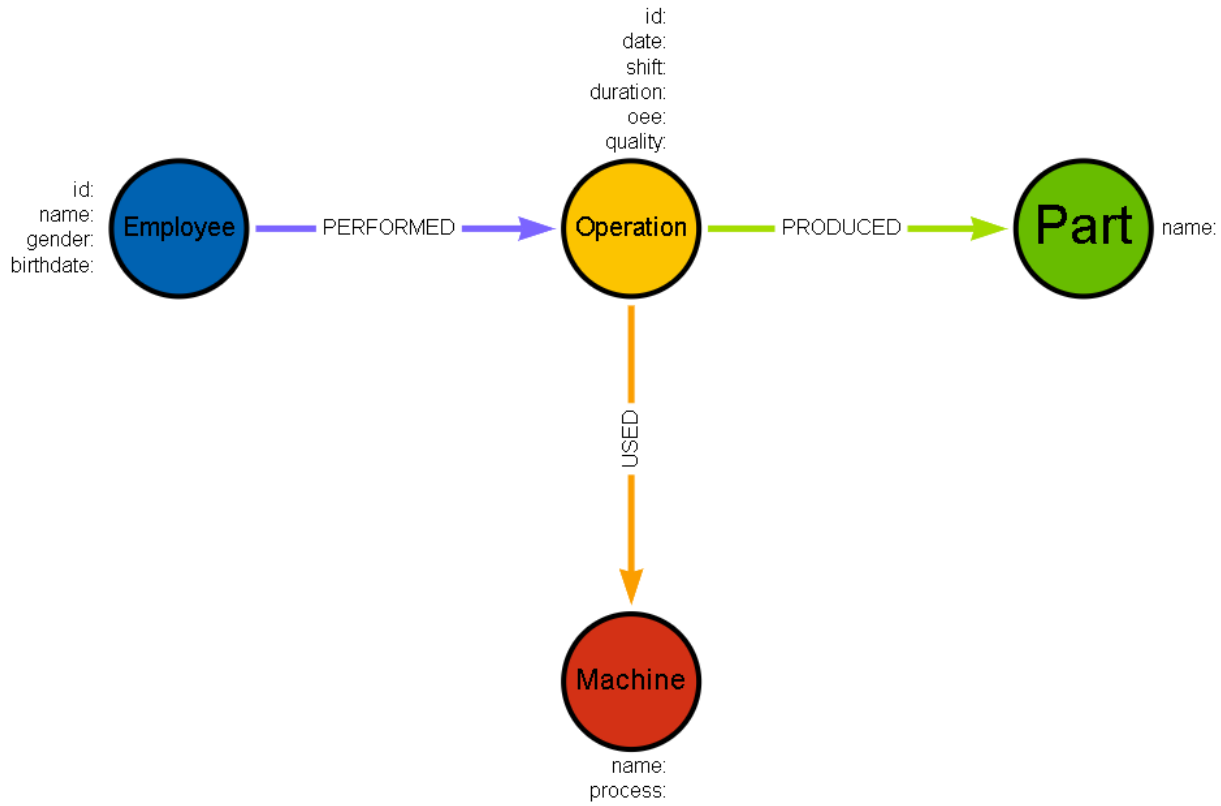
Graph design should contain all of the columns in the table. Each node representation can be seen below, divided into the categories related to the most similar ones.

Employee information was stored in the same node to keep all relevant information in the same node. There is no need of having any other types of information from other columns.

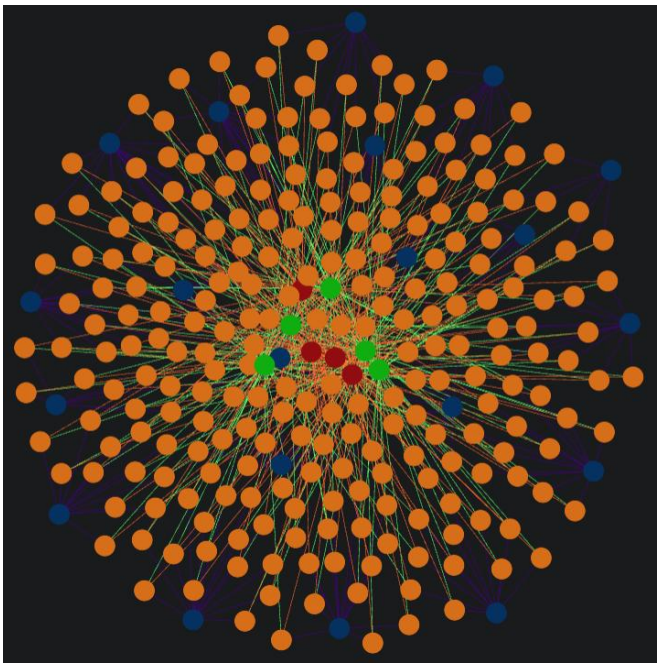
Operation has all information about the performed by operators such as date, shift, duration, oee, and quality. Part and machines were seperated since there are only limited number of them and they would have repeated, if they had been added as properties in the operation. Also, part and machine were represented as nodes to have more flexible queries to investigate each seperately.

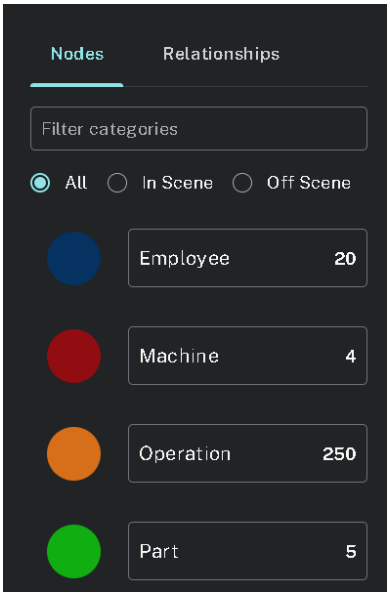
The reason why shifts were not added as separate nodes is that the complexity of the model was reduced. Still, different shift can be called by queries, but when there is a need of detailed investigation by each shift, refactoring can be applied on the model by taking each shift as a separate node.

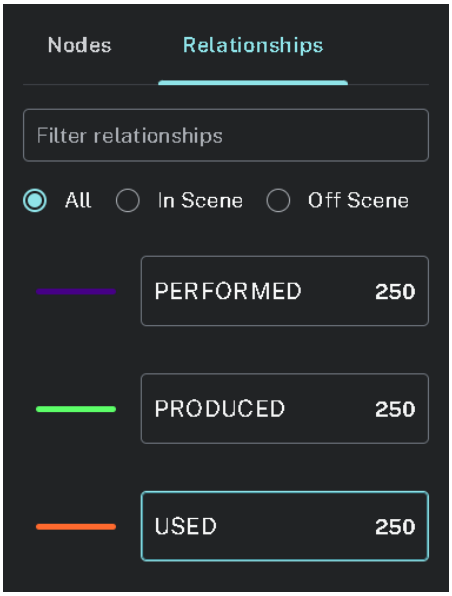
**Employee:** id, name, gender, birthdate
**Operation:** id, date, shift, duration, oee, quality
**Part:** name
**Machine:** name, process




Also, the relationships between the nodes were specified as performed, produced, and machine. These terms were chosen to make the node connections meaningful in terms of semantic.



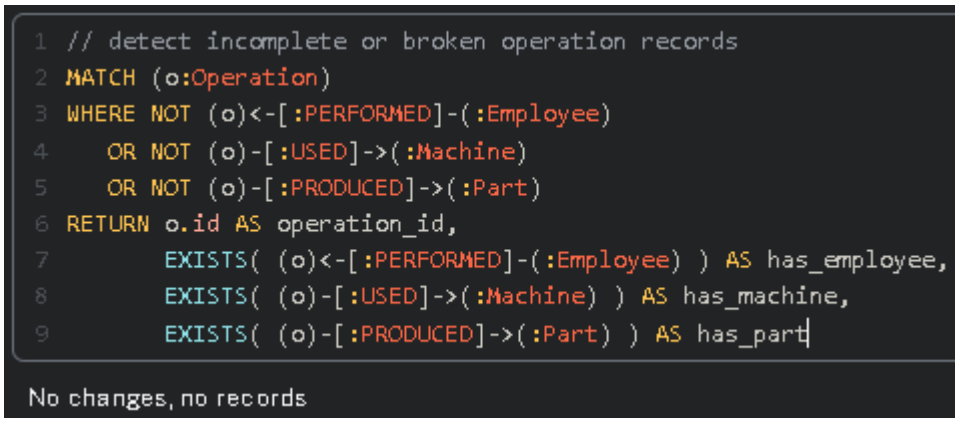



The total number of nodes and relationships can be seen in the images above. There are 20 employees, 4 machines, 5 parts, and 250 operations - which makes the number of each relationship 250 due to the design.

# TEST

Before starting the queries, it should be checked that if there is an error.



```
// detect incomplete or broken operation records
MATCH (o:Operation)
WHERE NOT (o)<-[:PERFORMED]-(:Employee)
   OR NOT (o)-[:USED]->(:Machine)
   OR NOT (o)-[:PRODUCED]->(:Part)
RETURN o.id AS operation_id,
       EXISTS( (o)<-[:PERFORMED]-(:Employee) ) AS has_employee,
       EXISTS( (o)-[:USED]->(:Machine) ) AS has_machine,
       EXISTS( (o)-[:PRODUCED]->(:Part) ) AS has_part
```

No changes, no records

According to the cypher query above, there is no record found, so all nodes were connected to each other properly.

# QUESTIONS

***Before starting, each question related to the query was specified on top of the query.***

```
//Which operators achieved the lowest and highest average OEE across all machines (with processes)?
MATCH (e:Employee)-[:PERFORMED]->(o:Operation)-[:USED]->(m:Machine)
WITH
  m.name AS machine, m.process AS process, e.name AS employee, avg(o.oee) AS avg_oee
ORDER BY machine, avg_oee ASC
WITH
  machine,
  process,
  collect({employee: employee, avg_oee: avg_oee}) AS statics
RETURN
  machine,
  process,
  statics[0].employee AS lowest_oee_employee,
  apoc.number.format(statics[0].avg_oee, "#0.00") AS lowest_avg_oee,
  statics[-1].employee AS highest_oee_employee,
  apoc.number.format(statics[-1].avg_oee, "#0.00") AS highest_avg_oee

```

Table RAW

| | machine | process | lowest_oee_employee | lowest_avg_oee | highest_oee_employee | highest_avg_oee |
|---|---|---|---|---|---|---|
| 1 | "M1" | "Drilling" | "Jeffery Wagner" | "0.65" | "Francisco Kelly" | "0.86" |
| 2 | "M2" | "Milling" | "Francisco Kelly" | "0.63" | "Brittany Johnson" | "0.93" |
| 3 | "M3" | "Assembly" | "Jason Gallagher" | "0.70" | "Melissa Delacruz" | "0.91" |
| 4 | "M4" | "Painting" | "Danielle Johnson" | "0.65" | "Robert Cole" | "0.89" |

The table highlights the operators with the lowest and highest average OEE per machine and associated process. Notably, Francisco Kelly shows inconsistent performance - having the lowest OEE on Milling (M2) but the highest on Drilling (M1) - suggesting that operator efficiency may be process-dependent. This insight can guide optimized task assignments based on individual strengths.

```
//Which employees have performed the most operations with the majority gender (top 5)?
MATCH (e:Employee)-[:PERFORMED]->(o:Operation)
RETURN e.name as employee, e.gender as gender, COUNT(e.name) as count
ORDER BY count DESC
LIMIT 5
```

Table RAW

| | employee | gender | count |
|---|---|---|---|
| 1 | "Amanda Dudley" | "M" | 19 |
| 2 | "Lisa Smith" | "M" | 18 |
| 3 | "Christopher Davis" | "M" | 17 |
| 4 | "Erica Mcclain" | "F" | 15 |
| 5 | "Helen Peterson" | "F" | 15 |

The table highlights the top 5 employees who performed the most operations, with counts ranging from 15 to 19. This indicates a relatively small gap in workload distribution among the top performers, which may reflect a fairly balanced task assignment system in the factory.

```
//What is the average operation duration per operator, and how does it vary by shift?
MATCH (e:Employee)-[:PERFORMED]->(o:Operation)
WITH e.name as employee, avg(o.duration) as avg_duration, o.shift as shift
ORDER BY avg_duration ASC
WITH shift,
    collect({
      employee: employee, avg_duration: avg_duration
    }) as statics
RETURN shift, statics[0].employee as employee, statics[0].avg_duration as avg_duration
```

Table RAW

| | shift | employee | avg_duration |
|---|---|---|---|
| 1 | "Night" | "Donna Mejia" | 67.0 |
| 2 | "Morning" | "Robert Cole" | 71.0 |
| 3 | "Day" | "Helen Peterson" | 74.5 |

The table shows the operator with the shortest average operation duration for each shift. The Night shift has the most efficient performer with an average of 67.0 minutes, followed by Morning (71.0) and Day (74.5).

```
//Find the min, max, and average OEE values for shifts?
MATCH (o:Operation)
RETURN o.shift as shift, apoc.number.format(min(o.oee),"#0.00") as min_OEE,
apoc.number.format(max(o.oee),"#0.00") as max_OEE, apoc.number.format(avg(o.oee),"#0.00") as average_OEE
```

Table RAW

| | shift | min_OEE | max_OEE | average_OEE |
|---|---|---|---|---|
| 1 | "Day" | "0.60" | "0.95" | "0.78" |
| 2 | "Morning" | "0.61" | "0.95" | "0.76" |
| 3 | "Night" | "0.61" | "0.95" | "0.77" |

The table compares OEE performance across shifts, showing very similar min (0.60-0.61) and max (0.95) values for all. However, the Day shift stands out slightly with the highest average OEE (0.78), suggesting marginally better overall efficiency compared to Morning (0.76) and Night (0.77), which doesn't change the overall result though.

```
// Which machines and process are used most frequently in operations?
MATCH (m:Machine)<-[:USED]-(o:Operation)
RETURN m.name as machine, m.process as process, COUNT(m.name) as count
ORDER BY count DESC
```

Table RAW

| | machine | process | count |
|---|---|---|---|
| 1 | "M4" | "Painting" | 70 |
| 2 | "M1" | "Drilling" | 64 |
| 3 | "M3" | "Assembly" | 59 |
| 4 | "M2" | "Milling" | 57 |

The table shows that Painting (M4) is the most frequently used process with 70 operations, followed by Drilling (M1) and Assembly (M3). This suggests that M4 may be a production bottleneck or a critical step in the workflow, which could benefit from efficiency optimization or capacity scaling.

```
//What is the average OEE by machine process type?
MATCH (o:Operation)-[:USED]->(m:Machine)
RETURN m.process as process_name, m.name as machine, apoc.number.format(avg(o.oee),"#0.00") as average_OEE
ORDER BY average_OEE DESC
```

Table RAW

| | process_name | machine | average_OEE |
|---|---|---|---|
| 1 | "Assembly" | "M3" | "0.78" |
| 2 | "Drilling" | "M1" | "0.77" |
| 3 | "Painting" | "M4" | "0.77" |
| 4 | "Milling" | "M2" | "0.76" |

The table ranks processes by their average OEE, showing that Assembly (M3) leads with 0.78, while Milling (M2) trails slightly at 0.76. Though the differences are small, this suggests that Assembly is the most efficient process, while Milling may benefit from performance improvements or closer monitoring.

```
// Which part types take the longest average operation time to produce?
MATCH (o:Operation)-[]->(p:Part)
RETURN p.name as part, avg(o.duration) as avg_duration
ORDER BY avg_duration DESC
```

Table RAW

| | part | avg_duration |
|---|---|---|
| 1 | "Shaft" | 115.61363636363636 |
| 2 | "Gearbox" | 106.9148936170213 |
| 3 | "Bearing" | 104.60869565217394 |
| 4 | "Valve" | 99.31372549019609 |
| 5 | "Piston" | 92.17741935483873 |

The table shows that Shaft takes the longest average time to produce at 115.61 units, followed by Gearbox and Bearing. These parts may involve more complex operations or precision, indicating potential areas to optimize production time or investigate process inefficiencies. Shaft has the longest average production time, while Piston, though quicker, still takes significant time compared to other parts. To prevent production bottlenecks, the simultaneous manufacturing of time-intensive parts like Shaft and Piston should be carefully scheduled and balanced across resources.

```
// Which operators are frequently involved in operations with low quality outcomes?
MATCH (e:Employee)-[:PERFORMED]->(o:Operation {quality:"Low"})
RETURN e.name as employee, e.gender as gender, COUNT(o.quality) as low_quality_count
ORDER BY low_quality_count DESC
LIMIT 5
```

Table RAW

| | employee | gender | low_quality_count |
|---|---|---|---|
| 1 | "Lisa Smith" | "M" | 8 |
| 2 | "Amanda Dudley" | "M" | 8 |
| 3 | "Brittany Johnson" | "M" | 6 |
| 4 | "Jason Gallagher" | "M" | 5 |
| 5 | "Barbara Bush" | "M" | 5 |

The table identifies the top 5 operators most frequently involved in low-quality outcomes, with Lisa Smith and Amanda Dudley each linked to 8 low-quality operations. This highlights a need for further investigation or targeted training to improve quality performance and reduce production defects.

```cypher
// What are the parts and machine which have higher than 50% low quality, and which machines were they produced by?
MATCH (p:Part)<-[:PRODUCED]-(o:Operation)-[:USED]->(m:Machine)
WITH
  m.name AS machine,
  p.name AS part,
  COUNT(*) AS total,
  COUNT(CASE WHEN o.quality = 'Low' THEN 1 END) * 1.0 / COUNT(*) * 100 AS low_rate,
  COUNT(CASE WHEN o.quality = 'Medium' THEN 1 END) * 1.0 / COUNT(*) * 100 AS medium_rate,
  COUNT(CASE WHEN o.quality = 'High' THEN 1 END) * 1.0 / COUNT(*) * 100 AS high_rate
WHERE low_rate > 50
RETURN
  machine, part, total,
  ROUND(low_rate, 2) AS low_quality_rate,
  ROUND(medium_rate, 2) AS medium_quality_rate,
  ROUND(high_rate, 2) AS high_quality_rate
ORDER BY low_quality_rate DESC

```

Table RAW

| | machine | part | total | low_quality_rate | medium_quality_rate | high_quality_rate |
|---|---|---|---|---|---|---|
| 1 | "M3" | "Bearing" | 12 | 75.0 | 25.0 | 0.0 |
| 2 | "M1" | "Bearing" | 10 | 60.0 | 40.0 | 0.0 |

The results show that Bearings produced on M3 and M1 have critically high low-quality rates - 75% and 60% respectively - with 0% high-quality output. This strongly suggests a quality issue specific to Bearing production, especially on M3, and indicates an urgent need for process review, maintenance, or operator retraining to reduce defect rates.

# SUMMARY

1) Operator performance appears to be closely tied to the process type. For instance, Francisco Kelly excels in Drilling but underperforms in Milling. If skilled operators for a specific part aren't present during a shift, it can lead to poor production quality. To prevent this, either more operators should be trained for those critical parts, or experienced employees with the potential to perform well should be hired and allocated accordingly.
2) Painting (Machine M4) is used more than any other process, so if production delays are occurring, this could be a bottleneck. It may be time to look into workload balancing or increasing capacity here.
3) Shaft and Gearbox parts take significantly longer to produce, and even Piston, though faster, still requires careful scheduling. Running these long-duration parts simultaneously could easily slow down the line.
4) A red flag comes from Bearing production, especially on Machines M1 and M3, which have extremely high rates of low-quality output and no high-quality products at all. This is a strong indicator that something in the process - or the machine itself - needs urgent review.
5) Some operators, such as Lisa Smith and Amanda Dudley, are repeatedly linked to low-quality outcomes. This doesn’t necessarily mean poor performance, but it does suggest the need for additional support, guidance, or reassignment.