// Completed projects → Status\_Completed

MATCH (p:project)

WHERE toLower(p.Status) = "completed"

REMOVE p:project

SET p:Status\_Completed

// Ongoing or Planned projects → Status\_On\_Track

MATCH (p:project)

WHERE toLower(p.Status) IN ["ongoing", "planned"]

REMOVE p:project

SET p:Status\_On\_Track

// On hold or Canceled projects → Status\_Delay

MATCH (p:project)

WHERE toLower(p.Status) IN ["on hold", "canceled"]

REMOVE p:project

SET p:Status\_Delay

// Completed tasks → Completed

MATCH (t:tasks)

WHERE toLower(t.Task\_Status) = "completed"

REMOVE t:tasks

SET t:Completed

// In progress tasks → Inprogress

MATCH (t:tasks)

WHERE toLower(t.Task\_Status) = "in progress"

REMOVE t:tasks

SET t:Inprogress

// Pending tasks → To\_Do

MATCH (t:tasks)

WHERE toLower(t.Task\_Status) = "pending"

REMOVE t:tasks

SET t:To\_Do

// Remove old status properties for clean up (Optional)

MATCH (n)

REMOVE n.Status, n.Task\_Status

Phue

1. Identify Team Member Roles with Highest or Lowest Performance.

MATCH (e:employee)-[:WORKS\_AS]->(r:role)<-[:WORKS\_AS]-(e2:employee)-[:HAS]->(k:KPI)

RETURN

r.role\_name AS Role,

ROUND(AVG(k.Employee\_Rating), 2) AS Avg\_Rating,

COUNT(k) AS Rated\_Employees

ORDER BY Avg\_Rating DESC

2. Team Member Workload by Severity

MATCH (e:employee)-[:DO]->(t)

WHERE t:To\_Do OR t:Inprogress OR t:Completed

RETURN

e.FirstName + ' ' + e.LastName AS Employee,

t.Severity AS Task\_Priority,

COUNT(t) AS Task\_Count,

SUM(t.Estimated\_Time) AS Total\_Estimated\_Hours

ORDER BY Employee, Task\_Priority;

3. Project Manager Checks Workload for Project Members