Week 3 Task:

Submitted By Freedom Urom Unya

Task:

Define key metrics for task management efficiency.

Gather user feedback for frontend refinements.

Manage timelines and sprint planning

To measure **task management efficiency**, we need metrics that evaluate **productivity, timeliness, workload balance, and automation effectiveness**.

Here’s a structured **data model** for tracking **task activity**, which includes entities for tasks, users, statuses, and activity logs.

**Entities and Attributes**

**1. USERS (USERS)**

Tracks individuals assigned to tasks.

* user\_id (PK) - Unique identifier for each user.
* name - Full name of the user.
* email - Contact email.
* role - User role (e.g., Admin, Manager, Employee).
* created\_at - Timestamp of when the user was added.

**2. TASKS (TASKS)**

Stores task details.

* task\_id (PK) - Unique identifier for each task.
* title - Brief description of the task.
* description - Detailed task information.
* assigned\_to (FK → users.user\_id) - User assigned to the task.
* status (FK → task\_status.status\_id) - Current task status.
* priority - Task priority (Low, Medium, High).
* due\_date - Deadline for completion.
* created\_at - Timestamp when the task was created.
* updated\_at - Last modification timestamp.

**3. Task Status (task\_status)**

Defines task progress stages.

* status\_id (PK) - Unique identifier.
* status\_name - Status label (e.g., To Do, In Progress, Completed, On Hold).

**4. Task Activity Log (task\_activity)**

Logs all task-related actions.

* activity\_id (PK) - Unique identifier.
* task\_id (FK → tasks.task\_id) - Task affected.
* user\_id (FK → users.user\_id) - User who made the change.
* action - Description of the action (Created, Updated, Completed, etc.).
* old\_value - Previous value (if applicable).
* new\_value - New value (if applicable).
* timestamp - Time of action.

**Relationships**

* **Users ↔ Tasks** (One-to-Many): A user can be assigned multiple tasks.
* **Tasks ↔ Status** (Many-to-One): A task has one status at a time.
* **Tasks ↔ Activity Log** (One-to-Many): Each task has multiple activity logs.

To generate **reports on user behavior and trends**, we can use the **task activity data** to analyze productivity, performance, and workflow efficiency. Below are some key reports you can generate:

**1. USER PRODUCTIVITY REPORT**

* **Purpose:** Tracks how many tasks a user completes within a given time frame.
* **Key Metrics:**
  + Total tasks assigned
  + Completed tasks
  + Average task completion time
  + Tasks overdue vs. completed on time
* **SQL Query Example:**

SELECT

u.user\_id,

u.name,

COUNT(t.task\_id) AS total\_tasks,

SUM(CASE WHEN t.status\_id = (SELECT status\_id FROM task\_status WHERE status\_name = 'Completed') THEN 1 ELSE 0 END) AS completed\_tasks,

AVG(DATEDIFF(t.updated\_at, t.created\_at)) AS avg\_completion\_time

FROM users u

LEFT JOIN tasks t ON u.user\_id = t.assigned\_to

GROUP BY u.user\_id, u.name;

**2. Task Completion Trends Report**

* **Purpose:** Identifies how task completion rates fluctuate over time.
* **Key Metrics:**
  + Weekly/monthly completed tasks
  + Task completion trends over time
  + Peak productivity hours
* **SQL Query Example:**

SELECT

DATE\_FORMAT(updated\_at, '%Y-%m') AS month,

COUNT(task\_id) AS completed\_tasks

FROM tasks

WHERE status\_id = (SELECT status\_id FROM task\_status WHERE status\_name = 'Completed')

GROUP BY month

ORDER BY month ASC;

**3. Issue Resolution Time Report**

* **Purpose:** Measures how long it takes to fix bugs/issues.
* **Key Metrics:**
  + Average issue resolution time
  + Open vs. closed issues
  + High-severity issue resolution time
* **SQL Query Example:**

SELECT

i.severity,

AVG(DATEDIFF(t.updated\_at, t.created\_at)) AS avg\_resolution\_time,

COUNT(i.issue\_id) AS total\_issues

FROM issues i

JOIN tasks t ON i.task\_id = t.task\_id

WHERE t.status\_id = (SELECT status\_id FROM task\_status WHERE status\_name = 'Completed')

GROUP BY i.severity;

**4. User Engagement Report**

* **Purpose:** Identifies how active users are in the system.
* **Key Metrics:**
  + Number of logins per user
  + Number of tasks updated per user
  + Comments and discussions on tasks
* **SQL Query Example:**

SELECT

ta.user\_id,

u.name,

COUNT(ta.activity\_id) AS total\_actions

FROM task\_activity ta

JOIN users u ON ta.user\_id = u.user\_id

GROUP BY ta.user\_id, u.name

ORDER BY total\_actions DESC;

**5. Workflow Automation Effectiveness Report**

* **Purpose:** Measures the impact of automated workflows on task completion.
* **Key Metrics:**
  + Percentage of tasks auto-assigned
  + Approval vs. rejection rates
  + Time saved through automation
* **SQL Query Example:**

SELECT

w.task\_type,

COUNT(w.rule\_id) AS total\_automations,

SUM(CASE WHEN t.status\_id = (SELECT status\_id FROM task\_status WHERE status\_name = 'Completed') THEN 1 ELSE 0 END) AS completed\_automations

FROM workflow\_rules w

LEFT JOIN tasks t ON w.task\_type = t.task\_type

GROUP BY w.task\_type;

**Visualizations (Charts & Graphs)**

To make reports more useful, consider **dashboard visualizations**:

📊 **Bar Charts** – User productivity, Issue resolution time  
📈 **Line Charts** – Task completion trends  
📊 **Pie Charts** – Task status distribution

**Software Development & Workflow Automation** use case:

**Productivity & Performance**

✅ Developer performance (tasks completed per sprint)  
✅ Task backlog analysis (pending vs. completed)  
✅ Sprint velocity (average tasks completed per sprint)

**Issue Tracking & Resolution**

✅ Bug fix turnaround time  
✅ High-severity vs. low-severity issue trends  
✅ Most frequent bug categories

**Workflow Automation Impact**

✅ Effectiveness of automated task assignments  
✅ Average approval/rejection time  
✅ Reduction in manual task workload

**Team Collaboration & Engagement**

✅ User activity levels (who is most/least active?)  
✅ Most-discussed tasks (high-comment tasks)  
✅ Meeting deadlines (on-time vs. overdue tasks)

**User Interaction Analysis for Usability Improvements**

To improve usability, we need to analyze how users interact with the system. Here’s a structured approach:

**1. Key Metrics for Usability Analysis**

🔹 **Task Completion Rate** – How often users complete tasks successfully.  
🔹 **Time on Task** – How long users take to complete tasks.  
🔹 **Error Rate** – How often users fail or abandon tasks.  
🔹 **Navigation Patterns** – Identify friction points in workflows.  
🔹 **Feature Usage** – Which features are used the most/least.  
🔹 **Automation Effectiveness** – Are automated workflows reducing manual effort?

**2. Data Sources & SQL Queries**

**A. Task Completion Rate**

* **Purpose:** Measure how many tasks assigned to users are completed.
* **Query:**

sql

CopyEdit

SELECT

u.user\_id,

u.name,

COUNT(t.task\_id) AS total\_tasks,

SUM(CASE WHEN t.status\_id = (SELECT status\_id FROM task\_status WHERE status\_name = 'Completed') THEN 1 ELSE 0 END) AS completed\_tasks,

(SUM(CASE WHEN t.status\_id = (SELECT status\_id FROM task\_status WHERE status\_name = 'Completed') THEN 1 ELSE 0 END) \* 100.0) / COUNT(t.task\_id) AS completion\_rate

FROM users u

LEFT JOIN tasks t ON u.user\_id = t.assigned\_to

GROUP BY u.user\_id, u.name;

* **Usability Insight:** If completion rates are low, tasks may be too complex or the UI could be unclear.

**B. Time on Task**

* **Purpose:** Identify bottlenecks where users spend too much time.
* **Query:**

sql

CopyEdit

SELECT

task\_id,

assigned\_to,

TIMESTAMPDIFF(MINUTE, created\_at, updated\_at) AS time\_spent\_minutes

FROM tasks

WHERE status\_id = (SELECT status\_id FROM task\_status WHERE status\_name = 'Completed');

* **Usability Insight:** If users take too long, the UI might be unintuitive or workflows inefficient.

**C. Error & Abandonment Rate**

* **Purpose:** Track incomplete tasks and workflow failures.
* **Query:**

sql

CopyEdit

SELECT

assigned\_to,

COUNT(task\_id) AS total\_tasks,

SUM(CASE WHEN status\_id = (SELECT status\_id FROM task\_status WHERE status\_name = 'Blocked') THEN 1 ELSE 0 END) AS abandoned\_tasks,

(SUM(CASE WHEN status\_id = (SELECT status\_id FROM task\_status WHERE status\_name = 'Blocked') THEN 1 ELSE 0 END) \* 100.0) / COUNT(task\_id) AS abandonment\_rate

FROM tasks

GROUP BY assigned\_to;

* **Usability Insight:** If many tasks are abandoned, users may be facing difficulties or system errors.

**D. Navigation Patterns & Bottlenecks**

* **Purpose:** Identify steps where users struggle in workflows.
* **Query:**

sql

CopyEdit

SELECT

ta.task\_id,

ta.user\_id,

ta.action,

ta.timestamp

FROM task\_activity ta

WHERE action IN ('Task Viewed', 'Task Updated', 'Task Abandoned')

ORDER BY task\_id, timestamp;

* **Usability Insight:** If users repeatedly **view but don’t update** a task, the interface might be unclear.

**E. Feature Usage Analysis**

* **Purpose:** Identify which features users engage with the most.
* **Query:**

sql

CopyEdit

SELECT

action,

COUNT(activity\_id) AS usage\_count

FROM task\_activity

GROUP BY action

ORDER BY usage\_count DESC;

* **Usability Insight:** If some features are rarely used, they might need better UI placement or clearer instructions.

**F. Automation Effectiveness**

* **Purpose:** Determine whether workflow automation is reducing manual work.
* **Query:**

sql

CopyEdit

SELECT

w.task\_type,

COUNT(w.rule\_id) AS total\_automations,

COUNT(t.task\_id) AS affected\_tasks,

(COUNT(t.task\_id) \* 100.0) / COUNT(w.rule\_id) AS automation\_effectiveness

FROM workflow\_rules w

LEFT JOIN tasks t ON w.task\_type = t.task\_type

GROUP BY w.task\_type;

* **Usability Insight:** If automation effectiveness is low, rules may not be configured optimally.

**3. Visualization & Action Plan**

📊 **Dashboard Suggestions:**  
✅ **Bar Chart:** Task completion rate per user  
✅ **Heatmap:** High-abandonment areas in workflows  
✅ **Pie Chart:** Feature usage distribution  
✅ **Line Chart:** Time-on-task trends over time

🎯 **Actionable Improvements:**

* **Improve UI flow** for tasks with high abandonment.
* **Enhance onboarding/tutorials** for underused features.
* **Refine automation rules** if manual work remains high.
* **Optimize task assignment** if some users struggle with completion.