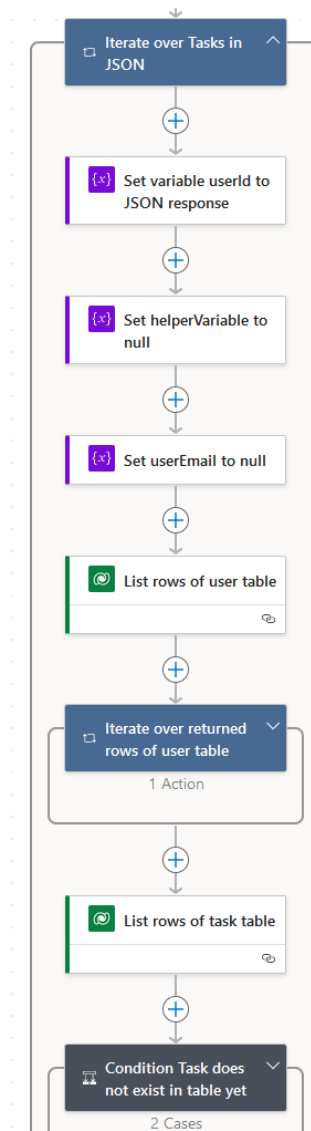


Technical Process (detailed) of flow "Get tasks from Valuestreamer to Planner"
This page shows the process of the flow with all details.

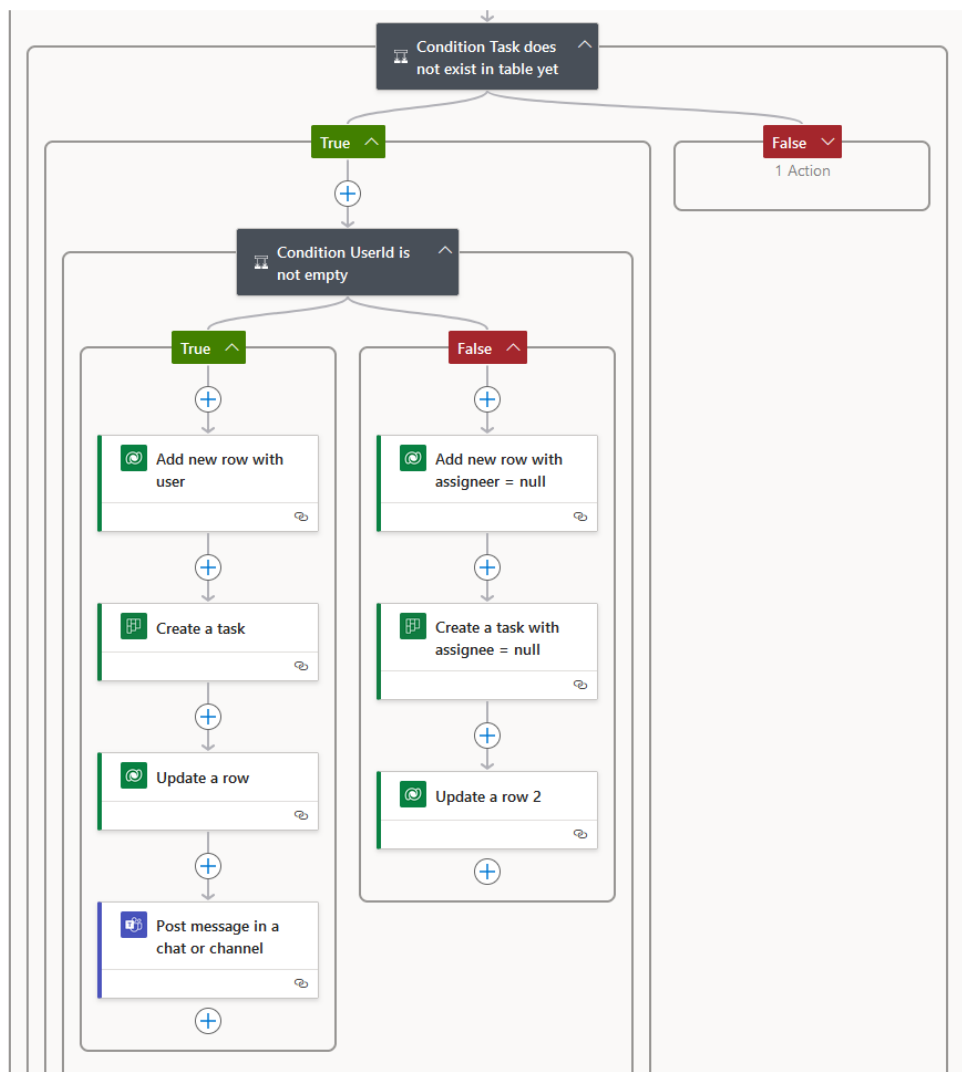
Process Overview

Step	Screenshot
1-7	The screenshot displays a vertical sequence of actions in a Power Automate flow: Manually trigger a flow: The starting action, represented by a blue icon with a hand cursor. HTTP request to GET all tasks from VS Team: An action with a green icon, preceded by a join connector (a circle with a plus sign). Parse JSON for tasks: An action with a purple icon, preceded by a join connector. Initialize variable userId: An action with a purple icon, preceded by a join connector. Initialize variable userEmail: An action with a purple icon, preceded by a join connector. Initialize HelpGetPlannerUserId: An action with a purple icon, preceded by a join connector. Iterate over Tasks in JSON: A loop action with a dark blue header and a light blue body, preceded by a join connector. The loop body is labeled "7 Actions". Each action is connected to the next by a downward arrow, and each is preceded by a join connector, indicating a sequential process where each step waits for the previous one to complete.

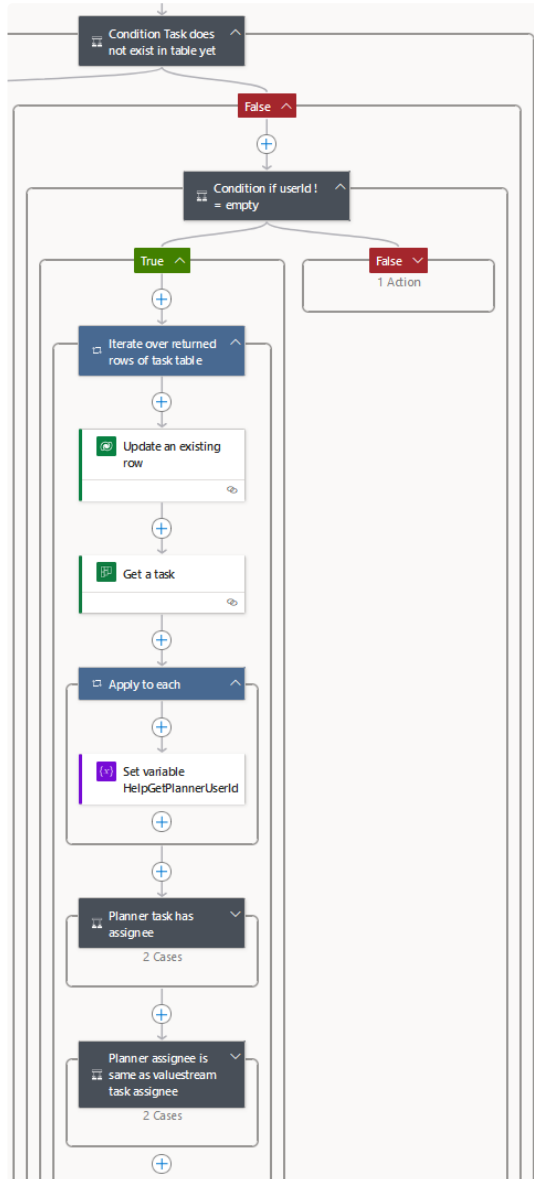
8-15



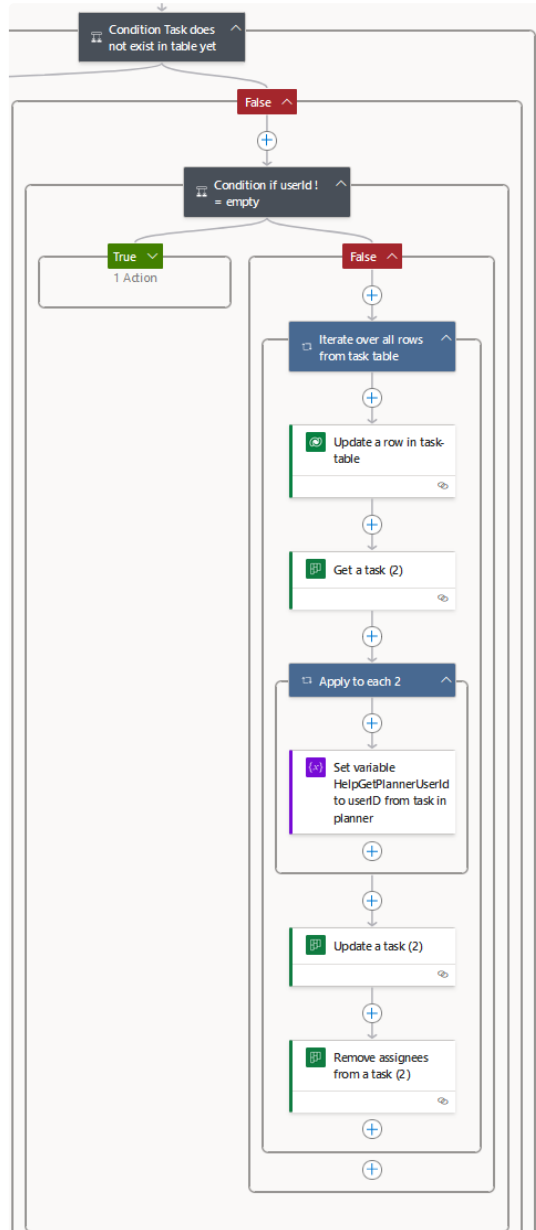
15-15.1.1.4 /
15.1.2.3



15-15.2.1.6.1.1 /
15.2.1.6.2.3



15-15.2.2.6



Step by step

Step	Description	Power Automate Function:	Variables, Parameters, Formula
		Connector, Display-Name	
1.	Flow is triggered.	Trigger: No Connector,	-

		Manually Trigger a flow																					
2.	Request data from Valuestreamer containing the tasks from a certain team.	HTTP: HTTP, HTTP request to GET all tasks from VS Team	Parameters <table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>Uri:</td><td>/api/exchange/tasks</td></tr><tr><td>Method:</td><td>Get</td></tr><tr><td colspan="2">Headers:</td></tr><tr><td>Authorization:</td><td>Basic YOURTOKEN</td></tr><tr><td>Accept:</td><td>application/json</td></tr><tr><td colspan="2">Queries:</td></tr><tr><td>include:</td><td>All</td></tr><tr><td>team:</td><td>YOURTEAM-ID</td></tr><tr><td>progress:</td><td>CREATED,PLANNED,INPROGRESS</td></tr></tbody></table>	Key	Value	Uri:	/api/exchange/tasks	Method:	Get	Headers:		Authorization:	Basic YOURTOKEN	Accept:	application/json	Queries:		include:	All	team:	YOURTEAM-ID	progress:	CREATED,PLANNED,INPROGRESS
Key	Value																						
Uri:	/api/exchange/tasks																						
Method:	Get																						
Headers:																							
Authorization:	Basic YOURTOKEN																						
Accept:	application/json																						
Queries:																							
include:	All																						
team:	YOURTEAM-ID																						
progress:	CREATED,PLANNED,INPROGRESS																						
3.	Parse the response of step 2 in a JSON-Object.	Parse JSON: Data Operations, Parse JSON for tasks	Parameter Content: body('HTTP_request_to_GET_all_tasks_from_VS_Team') JSON-Schema <pre>{ "type": "object", "properties": { "page": { "type": "object", "properties": { "page": { "type": "integer" }, "elementsOnPage": { "type": "integer" } } } } }</pre>																				

```
    },
    "elementsPerPage": {
      "type": "integer"
    },
    "totalPages": {
      "type": "integer"
    },
    "totalElements": {
      "type": "integer"
    }
  }
},
"entries": {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "id": {
        "type": "string"
      },
      "category": {
        "type": "string"
      },
      "createdByTeam": {
        "type": "string"
      },
      "label": {
        "type": [
          "string",
          "null"
        ]
      },
      "priority": {
        "type": "boolean"
      },
      "progress": {
        "type": "string"
      }
    }
  }
}
```

```
    },
    "responsiblePerson":
{
    "type": [
        "string",
        "null"
    ]
},
    "responsibleTeam": {
        "type": "string"
    },
    "status": {
        "type": "integer"
    },
    "targetEffort": {
        "type": "number"
    },
    "targetEndDate": {
        "type": "string"
    },
    "title": {
        "type": "string"
    }
},
    "required": [
        "id",
        "createdByTeam",
        "priority",
        "progress",
        "responsibleTeam",
        "responsiblePerson",
        "targetEffort",
        "targetEndDate",
        "title"
    ]
}
}
```

			} }						
4.	Create new variable called userID with datatype String and no value.	Initialize Variable: Variables, Initialize Variable userID	Parameter <table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>Name:</td><td>userID</td></tr><tr><td>Type:</td><td>String</td></tr></tbody></table>	Key	Value	Name:	userID	Type:	String
Key	Value								
Name:	userID								
Type:	String								
5.	Create new variable called userEmail with datatype String and no value.	Initialize Variable: Variables, Initialize Variable userEmail	Parameter <table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>Name:</td><td>userEmail</td></tr><tr><td>Type:</td><td>String</td></tr></tbody></table>	Key	Value	Name:	userEmail	Type:	String
Key	Value								
Name:	userEmail								
Type:	String								
6.	Create new variable called HelpGetPlannerUserId with datatype String and no value.	Initialize Variable: Variables, HelpGetPlannerU serID	Parameter <table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>Name:</td><td>HelpGetPlannerUserId</td></tr><tr><td>Type:</td><td>String</td></tr></tbody></table>	Key	Value	Name:	HelpGetPlannerUserId	Type:	String
Key	Value								
Name:	HelpGetPlannerUserId								
Type:	String								
7.	For-each-Loop: Iterate over all returned tasks from VS, using the JSON-Object from step 3.	For each: Control, For each	Iterate over outputs('Parse_JSON_ for_tasks')? ['body']? ['entries']"						
8.	Set variable UserID to Assignee from current task.	Set variable: Variables, Set variable userId to JSON response	Parameter <table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>Name:</td><td>userId</td></tr><tr><td>Value:</td><td>items('For_each')? ['responsiblePerson']</td></tr></tbody></table>	Key	Value	Name:	userId	Value:	items('For_each')? ['responsiblePerson']
Key	Value								
Name:	userId								
Value:	items('For_each')? ['responsiblePerson']								
9.	Set variable HelpGetPlannerUserId to	Set variable: Variables,	Parameter <table><thead><tr><th>Key</th><th>Value</th></tr></thead></table>	Key	Value				
Key	Value								

	null (to delete values from previous iterations)	Set helperVariable to null	Name: HelpGetPlannerUserId Value: null						
10.	Set variable userEmail to null (to delete values from previous iterations)	Set variable: Variables, Set userEmail to null	Parameter <table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>Name:</td><td>userEmail</td></tr><tr><td>Value:</td><td>null</td></tr></tbody></table>	Key	Value	Name:	userEmail	Value:	null
Key	Value								
Name:	userEmail								
Value:	null								
11.	List rows from User-Table that belong to the current UserID (the variable was set in step 8)	List rows Microsoft Dataverse, List rows of user table	Parameter <table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>Table Name:</td><td>YourUserTable</td></tr><tr><td>Filter Rows:</td><td>userIDinTabel eq 'variables('userID')'</td></tr></tbody></table>	Key	Value	Table Name:	YourUserTable	Filter Rows:	userIDinTabel eq 'variables('userID')'
Key	Value								
Table Name:	YourUserTable								
Filter Rows:	userIDinTabel eq 'variables('userID')'								
12.	For-each-Loop: Iterate over all returned rows from step 11. In step 11 only one row should be returned, because each user exists only once in the table. However PowerAutomate does not know this and therefore the For-Each-Loop is needed.	For each: Control Iterate over returned rows of user table	Iterate over outputs('List_rows_of_user_table')?['body/value']						
13.	Compare current UserID from step 8 to userId from the returned rows in step 11. If equal, then task from Valuestreamer has an assignee and that assignee exists in user table (true). If true: Go to step 13.1	Condition: Control Condition userId is equal to userId of user table	Condition variables('userId') is equal to items('Iterate_over_returned_rows_of_user_table')?['cr4b3_uuid']						

	If false: Go to step 13.2.1								
13.1.1	For-each-Loop: Iterate over all returned rows from step 11.	For each: Control Iterate over returned rows of user table (2)	▽ Iterate over <pre>outputs('List_rows_of_user_table')? ['body/value']</pre>						
13.1.2	Set variable userEmail to email of current iteration of step 13.1.1	Set variable: Variables, Set variable userEmail	▽ Parameter <table><tr><th>Key</th><th>Value</th></tr><tr><td>Name:</td><td>userEmail</td></tr><tr><td>Value:</td><td><pre>items('Iterate_over_returned_rows_of_user_table_(2)')? ['cr4b3_email']</pre></td></tr></table>	Key	Value	Name:	userEmail	Value:	<pre>items('Iterate_over_returned_rows_of_user_table_(2)')? ['cr4b3_email']</pre>
Key	Value								
Name:	userEmail								
Value:	<pre>items('Iterate_over_returned_rows_of_user_table_(2)')? ['cr4b3_email']</pre>								
13.2.1	Set variable userEmail to null.	Set variable: Variables, Set variable userEmail to null	▽ Parameter <table><tr><th>Key</th><th>Value</th></tr><tr><td>Name:</td><td>userEmail</td></tr><tr><td>Value:</td><td>null</td></tr></table>	Key	Value	Name:	userEmail	Value:	null
Key	Value								
Name:	userEmail								
Value:	null								
14.	List rows from Task-Table that belong to the current UserID (the variable UserID was set in step 8)	List rows Microsoft Dataverse, List rows of task table	▽ Parameter <table><tr><th>Key</th><th>Value</th></tr><tr><td>Table Name:</td><td>YourTaskTable</td></tr><tr><td>Filter Rows:</td><td>userIDinTable eq 'variables('userID')'</td></tr></table>	Key	Value	Table Name:	YourTaskTable	Filter Rows:	userIDinTable eq 'variables('userID')'
Key	Value								
Table Name:	YourTaskTable								
Filter Rows:	userIDinTable eq 'variables('userID')'								
15.	Compare the amount of rows returned in step 14 to 0.	For each: Control, Condition Task does not exist in	▽ Condition <pre>length(outputs('List_rows_of_task_table')?['body/value'])</pre>						

	If the amount is 0, then the task does not exist in the task table (true). If the amount is 1 or higher, then the task does exist in the task table (false). If true: Go to step 15.1.1 If false: Go to step 15.2.1	table yet	equals 0
15.1.1	Compare the value of userID (set in step 8) to 0. If the value is not 0, then it is not empty and the task does have an assignee (true). If the value is 0, then it is empty and the task does not have an assignee (false). If true: Go to step 15.1.1.1 If false: Go to step 15.1.1.2	For each: Control, Condition UserId is not empty	Condition length(variables('userID')) is not equal to 0
15.1.1.1	Current situation: Case 1: Task only exists in Valuestreamer Task in Valuestreamer is assignend Task has to be created in task table. Therefore a new row is added, with the values of the current task of the iteration of step 7	Add a new row: Microsoft Dataverse, Add new row with user	Parameters Key Value Table Name: YourTaskTable UUID: items('Iterate_over_Tasks_in_JS_ON')?['id'] Assignee: variables('userID') AzureEntraID_UserMail: variables('userEmail')

			DueDate: items('Iterate_over_Tasks_in_JS_ON')?['targetEndDate'] Progress: items('Iterate_over_Tasks_in_JS_ON')?['progress'] Title: items('Iterate_over_Tasks_in_JS_ON')?['title']												
15.1.1.2	Task has to be created in Planner.	Create a task: Planner, Create a task	Parameters <table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>Group Id:</td><td>YourGroupId</td></tr><tr><td>Plan Id:</td><td>YourPlanId</td></tr><tr><td>Title:</td><td>Valuestreamer: items('Iterate_over_Tasks_in_JS_ON')?['title']</td></tr><tr><td>Assigned User Ids:</td><td>variables('userEmail')</td></tr><tr><td>DueDateTime:</td><td>items('Iterate_over_Tasks_in_JS_ON')?['targetEndDate']</td></tr></tbody></table>	Key	Value	Group Id:	YourGroupId	Plan Id:	YourPlanId	Title:	Valuestreamer: items('Iterate_over_Tasks_in_JS_ON')?['title']	Assigned User Ids:	variables('userEmail')	DueDateTime:	items('Iterate_over_Tasks_in_JS_ON')?['targetEndDate']
Key	Value														
Group Id:	YourGroupId														
Plan Id:	YourPlanId														
Title:	Valuestreamer: items('Iterate_over_Tasks_in_JS_ON')?['title']														
Assigned User Ids:	variables('userEmail')														
DueDateTime:	items('Iterate_over_Tasks_in_JS_ON')?['targetEndDate']														
15.1.1.3	ID from newly created Planner task (step 15.1.1.2) has to be written into task table.	Update a row: Microsoft Dataverse, Update a row	Parameter <table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>Table Name:</td><td>YourTaskTable</td></tr><tr><td>Row ID:</td><td>body('Add_new_row_with_assignee')? ['YourTaskTable_task_sid']</td></tr></tbody></table>	Key	Value	Table Name:	YourTaskTable	Row ID:	body('Add_new_row_with_assignee')? ['YourTaskTable_task_sid']						
Key	Value														
Table Name:	YourTaskTable														
Row ID:	body('Add_new_row_with_assignee')? ['YourTaskTable_task_sid']														
15.1.1.4	Inform user via teams that a new task has been	Post message in a chat or	Parameter <table><thead><tr><th>Key</th><th>Value</th></tr></thead></table>	Key	Value										
Key	Value														

created in planner and assigned to him.

End of iteration - jump back to step 7

channel:
Microsoft Teams,
Post message in a chat or channel

Post As: Flow bot
Post in: Chat with Flow bot
Recipient: variable('userEmail')

Message

Test Nachricht - API Anbindung ValueStreamer

Aufgabe

items('Iterate_over_Tasks_in_JSON')?['title']} aus dem ValueStreamer wurde dir zugewiesen!

DueDate:

@{items('Iterate_over_Tasks_in_JSON')?['targetEndDate']}

und Effort:

@{items('Iterate_over_Tasks_in_JSON')?['targetEffort']} Stunden

15.1.2.1

Current Situation: Case 2: Task only exists in Valuestreamer Task in Valuestreamer is NOT assignend

Task has to be created in task table. Therefore a new row is added, with the values of the current task of the iteration of step 7. The task has no assignee, therefore the user remains empty.

Add a new row:
Microsoft Dataverse,
Add new row with assignee = null

Parameters

Key	Value
-----	-------

Table Name:	YourTaskTable
-------------	---------------

UUID:

```
items('Iterate_over_Tasks_in_JSON')?['id']
```

DueDate:

```
items('Iterate_over_Tasks_in_JSON')?['targetEndDate']
```

			Progress: items('Iterate_over_Tasks_in_JSON')?['progress'] Title: items('Iterate_over_Tasks_in_JSON')?['title']
15.1.2.2	Task has to be created in Planner.	Create a task: Planner, Create a task with assignee = null	Parameters Key Value Group Id: YourGroupId Plan Id: YourPlanId Title: Valuestreamer: items('Iterate_over_Tasks_in_JSON')?['title'] DueDateTime: items('Iterate_over_Tasks_in_JSON')?['targetEndDate']
15.1.2.3	ID from newly created Planner task (step 15.1.2.2) has to be written into task table. End of iteration - jump back to step 7.	Update a row: Microsoft Dataverse, Update a row 2	Parameter Key Value Table Name: YourTaskTable Row ID: body('Add_new_row_with_assignee=_null')?['YourTaskTable_task_sid']

15.2.1	Compare the value of userID (set in step 8) to 0. If the value is not 0, then it is not empty and the task does have an assignee (true). If the value is 0, then it is empty and the task does not have an assignee (false). If true: Go to step 15.2.1.1 If false: Go to step 15.2.2.1	Condition: Control, Condition UserId is not empty	Condition <code>length(variables('userId'))</code> is not equal to 0						
15.2.1.1	Situation: Case 3: Task exists in Valuestreamer and Planner. Task is assigned in Valuestreamer For-each-Loop: Iterate over returned rows in task-table from step 14.	For each: Control, Iterate over returned rows of task table	Iterate over <code>outputs('List_rows_of_task_table')?['body/value']</code>						
15.2.1.2	Task has to be updated in task table.	Update a row: Microsoft Dataverse, Updating an existing row	Parameters <table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>Table Name:</td><td>YourTaskTable</td></tr><tr><td>UUID:</td><td><code>items('Iterate_over_Tasks_in_JSON')?['id']</code></td></tr></tbody></table>	Key	Value	Table Name:	YourTaskTable	UUID:	<code>items('Iterate_over_Tasks_in_JSON')?['id']</code>
Key	Value								
Table Name:	YourTaskTable								
UUID:	<code>items('Iterate_over_Tasks_in_JSON')?['id']</code>								

			DueDate: <pre>items('Iterate_over_Tasks_in_JSON')? ['targetEndDate']</pre> Progress: <pre>items('Iterate_over_Tasks_in_JSON')? ['progress']</pre> Title: <pre>items('Iterate_over_Tasks_in_JSON')? ['title']</pre>						
15.2.1.3	Task table has to be synchronized with tasks in Planner.	Get a task: Microsoft Planner, Get a task	Parameter <table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>Task Id:</td><td><pre>body('Update_an_existing_row')? ['cr4b3_plannerid']</pre></td></tr></tbody></table>	Key	Value	Task Id:	<pre>body('Update_an_existing_row')? ['cr4b3_plannerid']</pre>		
Key	Value								
Task Id:	<pre>body('Update_an_existing_row')? ['cr4b3_plannerid']</pre>								
15.2.1.4	Set variable HelpGetPlannerUserId to userId of task in Planner (Loop will be created automatically, when value is set to userID)	Set Variable: Variables, Set variable HelpGetPlannerUserId	Parameter <table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>Name:</td><td>HelpGetPlannerUserId</td></tr><tr><td>Value:</td><td><pre>items('Apply_to_each')?['userId']</pre></td></tr></tbody></table>	Key	Value	Name:	HelpGetPlannerUserId	Value:	<pre>items('Apply_to_each')?['userId']</pre>
Key	Value								
Name:	HelpGetPlannerUserId								
Value:	<pre>items('Apply_to_each')?['userId']</pre>								
15.2.1.5	Compare the value of HelpGetPlannerUserId with 0. If greater than 0, HelpGetPlannerUserId is	Condition: Control, Planner task has assignee	Condition <pre>length(variables('HelpGetPlannerUserId')) is greater than 0</pre>						

	not empty (true). If not greater, HelpGetPlannerUserId is empty (false). If true: Go to step 15.2.1.5.1.1 If false: Go to step 15.2.1.5.2.1								
15.2.1.5.1.1	Get User profile from Office365-App	Get user profile (V2): Office 365 Users, Get user profile (V2)	Parameter <table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>User (UPN):</td><td>variables('HelpGetPlannerUserId')</td></tr></tbody></table>	Key	Value	User (UPN):	variables('HelpGetPlannerUserId')		
Key	Value								
User (UPN):	variables('HelpGetPlannerUserId')								
15.2.1.5.1.2	Set variable HelpGetPlannerUserId to mail from returned value of step 15.2.1.5.1.1	Set variable: Variables, Set variable HelpGetPlannerUserId (2)	Parameter <table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>Name:</td><td>HelpGetPlannerUserId</td></tr><tr><td>Value:</td><td> outputs('Get_user_profile_(V2)')?['body/mail'] </td></tr></tbody></table>	Key	Value	Name:	HelpGetPlannerUserId	Value:	outputs('Get_user_profile_(V2)')?['body/mail']
Key	Value								
Name:	HelpGetPlannerUserId								
Value:	outputs('Get_user_profile_(V2)')?['body/mail']								
15.2.1.5.2.1	Set variable HelpGetPlannerUserId to null.	Set variable: Variables, Set variable HelpGetPlannerUserId (3)	Parameter <table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>Name:</td><td>HelpGetPlannerUserId</td></tr><tr><td>Value:</td><td>null</td></tr></tbody></table>	Key	Value	Name:	HelpGetPlannerUserId	Value:	null
Key	Value								
Name:	HelpGetPlannerUserId								
Value:	null								
15.2.1.6	Compare the value of userEmail to HelpGetPlannerUserId.	Condition: Control,	Condition						

	If equal, then both tasks have the same assignee (true). If not equal, then assignees are different and the assignee in planner has to be changed (false). If true: Go to step 15.2.1.6.1.1 If false: Go to step 15.2.1.6.2.1	Planner assignee is same as valuestreamer task assignee	<pre>variables('userEmail') is equal to variables('HelpGetPlannerUserId')</pre>										
15.2.1.6.1.1	Update task in planner with values from valuestreamer. End of iteration - jump back to step 7.	Update a task: Microsoft Planner, Update a task	Parameter <table><tr><th>Key</th><th>Value</th></tr><tr><td>Task Id:</td><td><pre>body('Update_an_existing_row')?['cr4b3_plannerid']</pre></td></tr><tr><td>Title:</td><td><pre>outputs('Update_an_existing_row')?['body/cr4b3_title']</pre></td></tr><tr><td>DueDateTime:</td><td><pre>outputs('Update_an_existing_row')?['body/cr4b3_duedate']</pre></td></tr><tr><td>Progress:</td><td><pre>outputs('Update_an_existing_row')?['body/cr4b3_progress']</pre></td></tr></table>	Key	Value	Task Id:	<pre>body('Update_an_existing_row')?['cr4b3_plannerid']</pre>	Title:	<pre>outputs('Update_an_existing_row')?['body/cr4b3_title']</pre>	DueDateTime:	<pre>outputs('Update_an_existing_row')?['body/cr4b3_duedate']</pre>	Progress:	<pre>outputs('Update_an_existing_row')?['body/cr4b3_progress']</pre>
Key	Value												
Task Id:	<pre>body('Update_an_existing_row')?['cr4b3_plannerid']</pre>												
Title:	<pre>outputs('Update_an_existing_row')?['body/cr4b3_title']</pre>												
DueDateTime:	<pre>outputs('Update_an_existing_row')?['body/cr4b3_duedate']</pre>												
Progress:	<pre>outputs('Update_an_existing_row')?['body/cr4b3_progress']</pre>												

15.2.1.6.2.1	Update task in planner to add a new assignee.	Update a task: Microsoft Planner, Update a task and adding a new assignee	▼ Parameter <table><tr><th>Key</th><th>Value</th></tr><tr><td>Task Id:</td><td><code>body('Update_an_existing_row')?['cr4b3_plannerid']</code></td></tr><tr><td>Title:</td><td><code>outputs('Update_an_existing_row')?['body/cr4b3_title']</code></td></tr><tr><td>DueDateTime:</td><td><code>outputs('Update_an_existing_row')?['body/cr4b3_duedate']</code></td></tr><tr><td>Progress:</td><td><code>outputs('Update_an_existing_row')?['body/cr4b3_progress']</code></td></tr><tr><td>Add Assigned Users:</td><td><code>variables('userEmail')</code></td></tr></table>	Key	Value	Task Id:	<code>body('Update_an_existing_row')?['cr4b3_plannerid']</code>	Title:	<code>outputs('Update_an_existing_row')?['body/cr4b3_title']</code>	DueDateTime:	<code>outputs('Update_an_existing_row')?['body/cr4b3_duedate']</code>	Progress:	<code>outputs('Update_an_existing_row')?['body/cr4b3_progress']</code>	Add Assigned Users:	<code>variables('userEmail')</code>
Key	Value														
Task Id:	<code>body('Update_an_existing_row')?['cr4b3_plannerid']</code>														
Title:	<code>outputs('Update_an_existing_row')?['body/cr4b3_title']</code>														
DueDateTime:	<code>outputs('Update_an_existing_row')?['body/cr4b3_duedate']</code>														
Progress:	<code>outputs('Update_an_existing_row')?['body/cr4b3_progress']</code>														
Add Assigned Users:	<code>variables('userEmail')</code>														
15.2.1.6.2.2	Remove old assignees from task. This has to be done by an extra action. The action “Update task in planner” also has the option to remove an assignee - however this did not work	Remove old assignee from a task: Microsoft Planner, Remove assingees from a task	▼ Parameter <table><tr><th>Key</th><th>Value</th></tr><tr><td>Task Id:</td><td><code>outputs('Update_an_existing_row')?['body/cr4b3_plannerid']</code></td></tr></table>	Key	Value	Task Id:	<code>outputs('Update_an_existing_row')?['body/cr4b3_plannerid']</code>								
Key	Value														
Task Id:	<code>outputs('Update_an_existing_row')?['body/cr4b3_plannerid']</code>														

	while programming this flow (Jan2024)		Remove Assigned Users: <code>variables('HelpGetPlannerUserId')</code>								
15.2.1.6.2.3	Inform user via teams that a task has been assigned to him. End of iteration - jump back to step 7	Post message in a chat or channel: Microsoft Teams, Post message in a chat or channel	Parameter <table><tr><th>Key</th><th>Value</th></tr><tr><td>Post As:</td><td>Flow bot</td></tr><tr><td>Post in:</td><td>Chat with Flow bot</td></tr><tr><td>Recipient:</td><td><code>variable('userEmail')</code></td></tr></table> Message Test Nachricht - API Anbindung ValueStreamer Aufgabe items('Iterate_over_Tasks_in_JSON')?['title']} aus dem ValueStreamer wurde dir zugewiesen! DueDate: @{items('Iterate_over_Tasks_in_JSON')?['targetEndDate']} und Effort: @{items('Iterate_over_Tasks_in_JSON')?['targetEffort']} Stunden	Key	Value	Post As:	Flow bot	Post in:	Chat with Flow bot	Recipient:	<code>variable('userEmail')</code>
Key	Value										
Post As:	Flow bot										
Post in:	Chat with Flow bot										
Recipient:	<code>variable('userEmail')</code>										
15.2.2.1	Situation: Case 4: Task exists in Valuestreamer and Planner. Task is NOT assigned in Valuestreamer	For each: Control, Iterate over all rows from task table	Iterate <code>outputs('List_rows_of_task_table')?['body/value']</code>								

	For-Each-Loop: Iterate over all returned tasks from step 14.		
15.2.2.2	Update a row in task table.	Update a row: Microsoft Dataverse Update a row in task-table	Parameter Key Value Row Id: item()? ['cr4b3_vs_ejotpoc_tasksid'] Title: items('Iterate_over_Tasks_in_JSON')? ['title'] DueDate: items('Iterate_over_Tasks_in_JSON')? ['targetEndDate'] Progress: items('Iterate_over_Tasks_in_JSON')? ['progress']
15.2.2.3	Get task from Planner.	Get a task: Microsoft Planner, Get a task (2)	Parameter Task id: outputs('Update_a_row_in_task-table')? ['body/cr4b3_plannerid']
15.2.2.4	Set variable HelpGetPlannerUserId to userID from task in planner	Set Variable: Variables,	Parameter Key Value

	(step 15.2.2.3). (Creates automatically a loop).	Set variable HelpGetPlannerUserId to userID from task in planner	Name: HelpGetPlannerUserId Value: items('Apply_to_each_2')?['userId']										
15.2.2.5	Update task in planner.	Update a task: Microsoft Dataverse, Update a task (2)	Parameter <table><tr><th>Key</th><th>Value</th></tr><tr><td>Task</td><td>id: outputs('Update_a_row_in_task-table')?['body/cr4b3_planner_id']</td></tr><tr><td>Title:</td><td>outputs('Update_a_row_in_task-table')?['body/cr4b3_title']</td></tr><tr><td>DueDateTime:</td><td>items('Iterate_over_all_rows_from_task_table')?['cr4b3_duedate']</td></tr><tr><td>Progress:</td><td>outputs('Update_a_row_in_task-table')?['body/cr4b3_progress']</td></tr></table>	Key	Value	Task	id: outputs('Update_a_row_in_task-table')?['body/cr4b3_planner_id']	Title:	outputs('Update_a_row_in_task-table')?['body/cr4b3_title']	DueDateTime:	items('Iterate_over_all_rows_from_task_table')?['cr4b3_duedate']	Progress:	outputs('Update_a_row_in_task-table')?['body/cr4b3_progress']
Key	Value												
Task	id: outputs('Update_a_row_in_task-table')?['body/cr4b3_planner_id']												
Title:	outputs('Update_a_row_in_task-table')?['body/cr4b3_title']												
DueDateTime:	items('Iterate_over_all_rows_from_task_table')?['cr4b3_duedate']												
Progress:	outputs('Update_a_row_in_task-table')?['body/cr4b3_progress']												

3.

← Get tasks from ValueStreamer

 Parse JSON for tasks ⋮ <

Parameters Settings Code View About

Content *

 Body × fx

Schema * `body("HTTP_request_to_GET_all_tasks_from_VS_Team")`

```
{
  "type": "object",
  "properties": {
    "page": {
      "type": "object",
      "properties": {
        "page": {
          "type": "integer"
        },
        "elementsOnPage": {
          "type": "integer"
        },
        "elementsPerPage": {
          "type": "integer"
        },
        "totalPages": {
          "type": "integer"
        },
        "totalElements": {
          "type": "integer"
        }
      }
    }
  }
}
```

[Use sample payload to generate schema](#)

7.

 Iterate over Tasks in JSON ⋮ <

Parameters Settings Code View About

Select An Output From Previous Steps *

 Body ent... × fx

`outputs('Parse_JSON_for_tasks')['body']['entries']`

8.

← Get tasks from ValueStreamer

 Set variable userId to JSON response ⋮ <

Parameters Settings Code View About

Name *

userId ⌵

Value *

 responsi... × fx

`items('Iterate_over_Tasks_in_JSON')['responsiblePerson']`

11.

← Get tasks from ValueStream

 List rows of user table ⋮ <

Parameters Settings Code View Testing About

Table Name *

Name of your usertable ▼

Advanced parameters

Showing 1 of 8 ▼ Show all Clear all

Filter Rows

cr4b3_uuid eq {x} userId × fx ×

Enter an OData style filter expression to limit which rows are listed

13.

← Get tasks from ValueStream

 Condition userId is equal to userId of user table ⋮ <

Parameters Settings Code View About

Condition Expression *

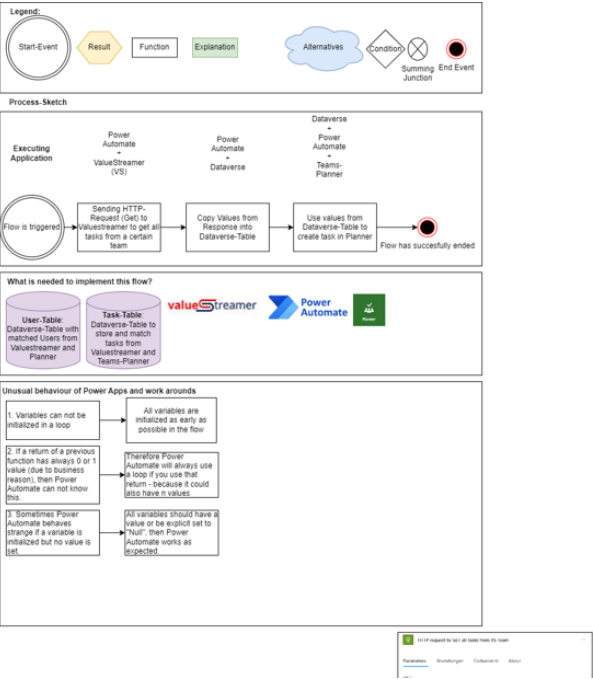
Provide the values to compare and select the operator to use.

AND ▼

☐ {x} userId × is equal to fx {x} cr4b3_uuid × ⋯

+ New item ▼

items('Iterate_over_returned_rows_of_user_table')['cr4b3_uuid']



Draw.IO-File with Diagram-data:

