Lab 6. Task overdue

Author: Serge Luca aka "Doctor Flow"

Learning objective: Controls, manipulation Excel, conditions, Date & time, expressions

Duration: 20 minutes

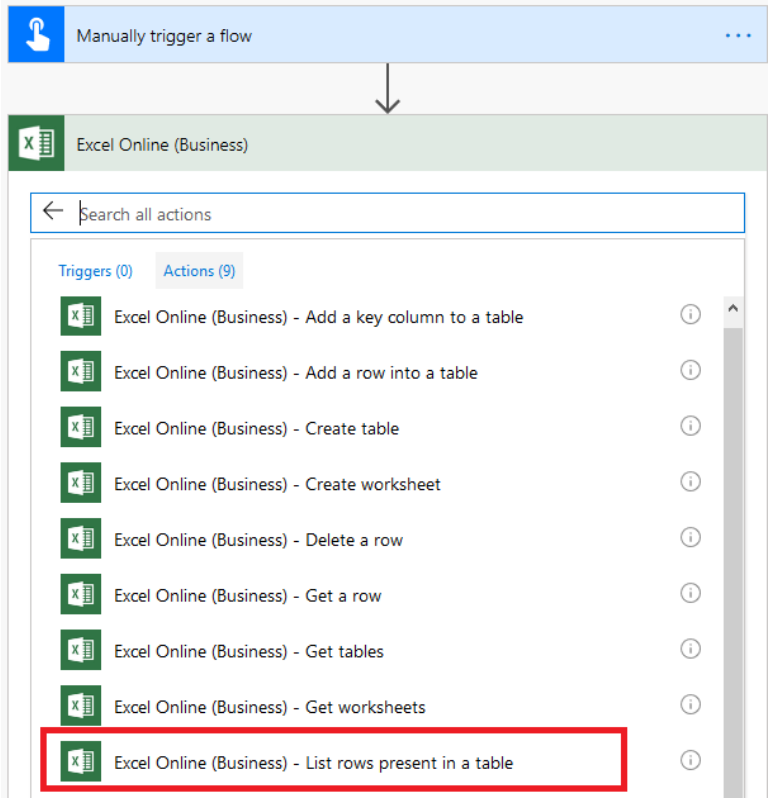
Scenario: We have an Excel document with a set of tasks, where some of these tasks are overdue. You will create a Flow that will find all overdue tasks and will send a report of these tasks.

Tasks:

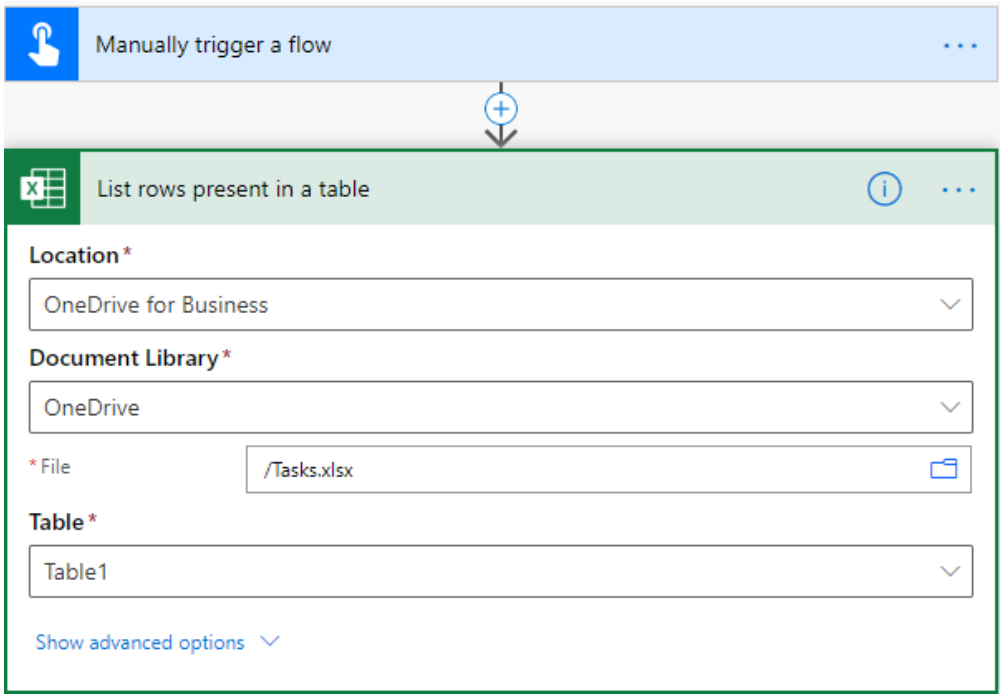
1. Create a new Excel file in your OneDrive for Business that looks like this (use the same columns). Before adding data in the Deadline column, make sure it is in **Text** format (otherwise your flows will crash!).

Task	Status	in charge	Deadline
Feed the cat	Started	user1@dynamicsjuly.onmicrosoft.com	1/1/2018
Call jon	Not Started	user1@dynamicsjuly.onmicrosoft.com	2/4/2018
Patch sql server	Started	user1@dynamicsjuly.onmicrosoft.com	7/6/2018
Call mum	Started	user1@dynamicsjuly.onmicrosoft.com	6/7/2018
Buy fruits	Started	user1@dynamicsjuly.onmicrosoft.com	8/8/2018
Call Kent	Not Started	user1@dynamicsjuly.onmicrosoft.com	8/8/2018
Buy a Porsche	Done	user1@dynamicsjuly.onmicrosoft.com	1/9/2018

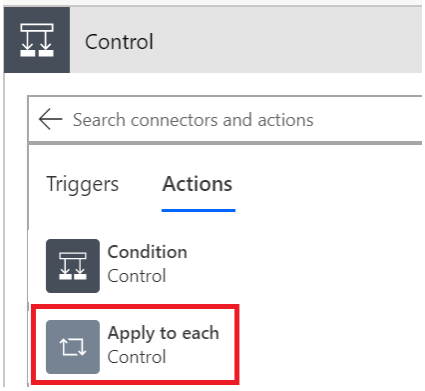
2. Name it **Tasks.xlsx**
3. Change the value of the **in-charge column** to your e-mail address and adjust some deadline value.
4. We want to write a Flow that will loop through all tasks, and that will check if the task is overdue. To do so, **create a new Instant Flow from blank** and use a **Manually trigger a flow** as a trigger**.** Name it **my Overdue Tasks.**
5. The Flow needs to connect to the **Tasks.xlsx** file, so add an **Excel Online (Business) – List present in a table** action (Not OneDrive !!!) as illustrated below:



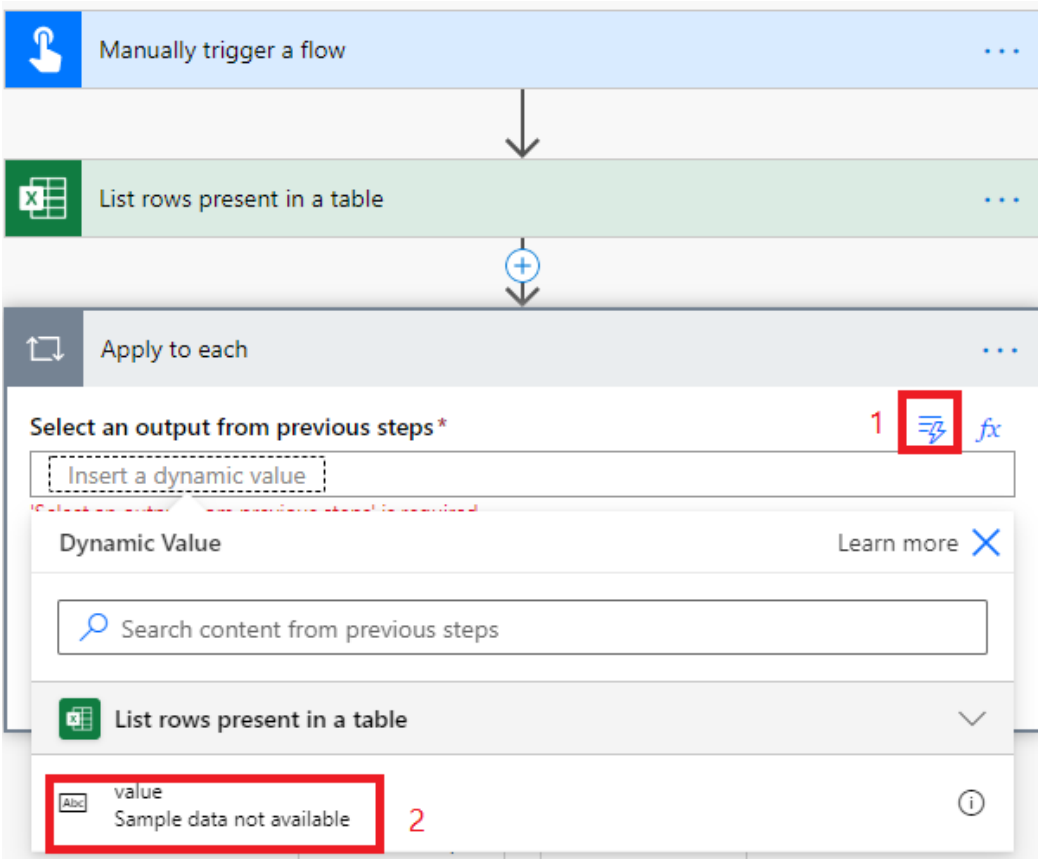
6. Set the action properties, as illustrated in the next picture:

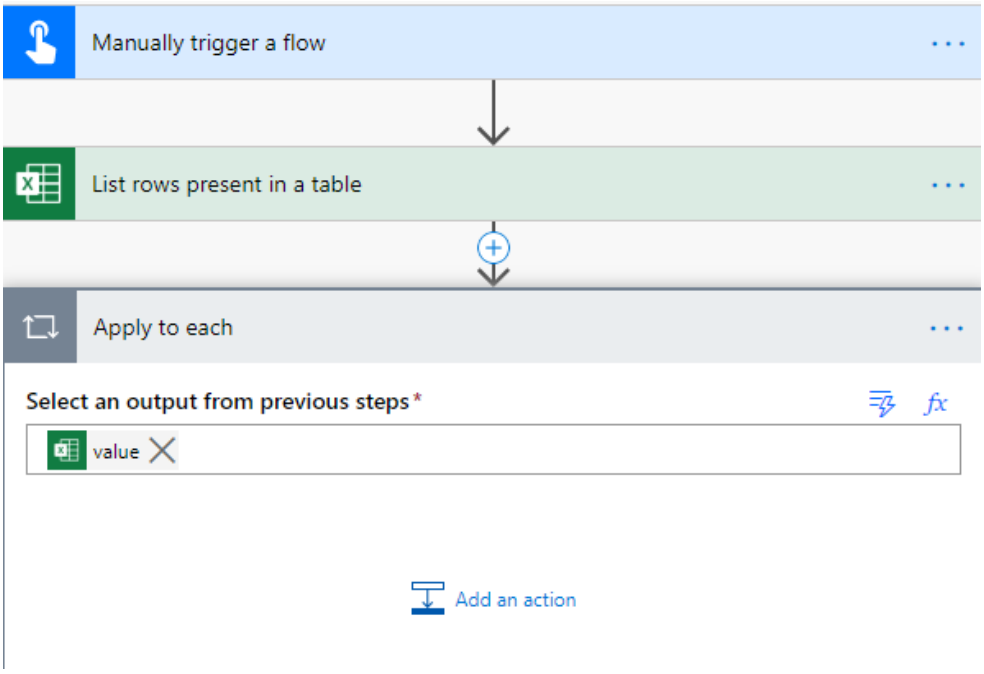


7. Let's loop through all tasks, so we need to add an **Apply to each**:

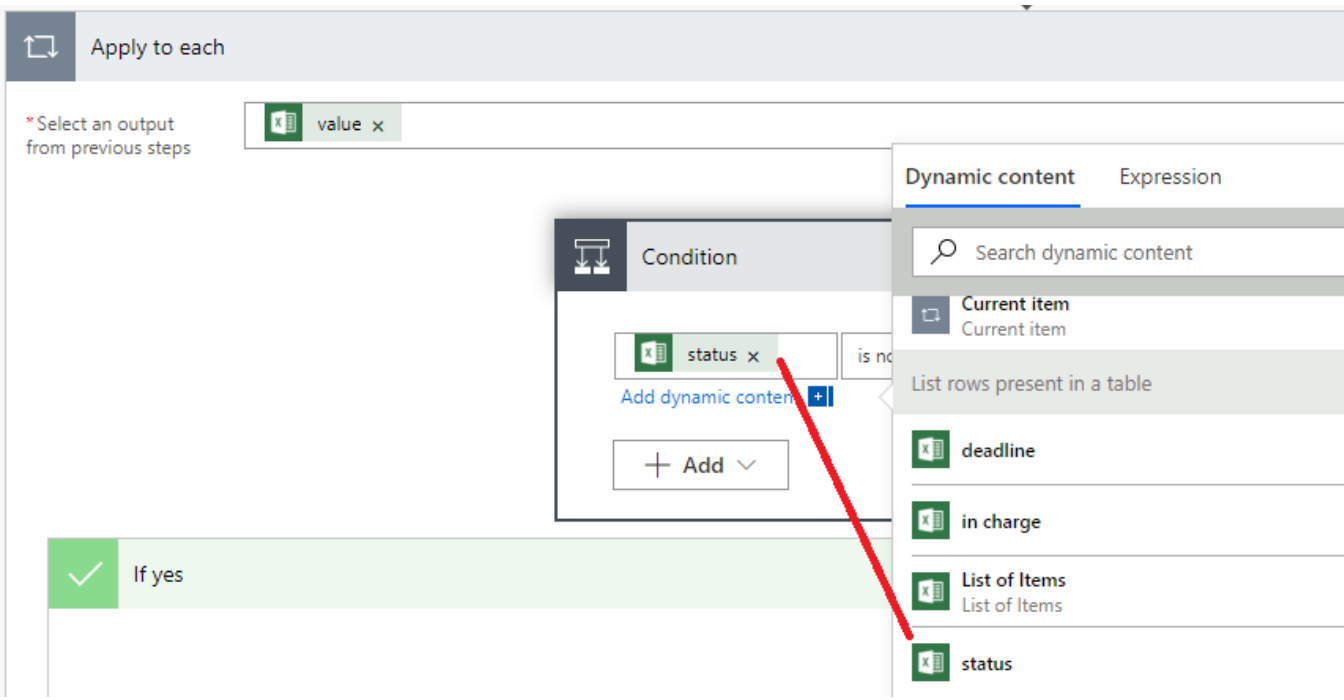


8. Select the add a **Dynamic value** in the **Apply to each** action:





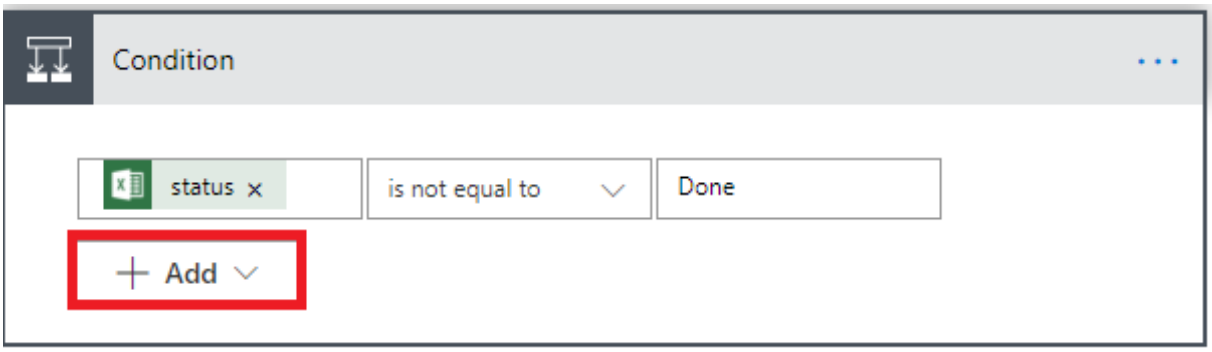
9. Add a **Condition** to filter the task where **Status** is not equal to **Done:**



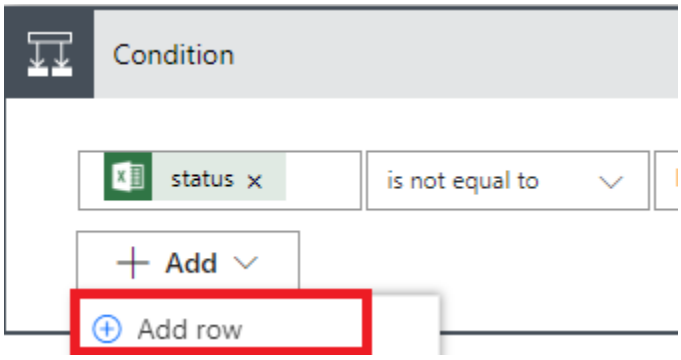
Make sure the operator used is “is not equal”.

10. Add a new sub condition that will check if the task is overdue:

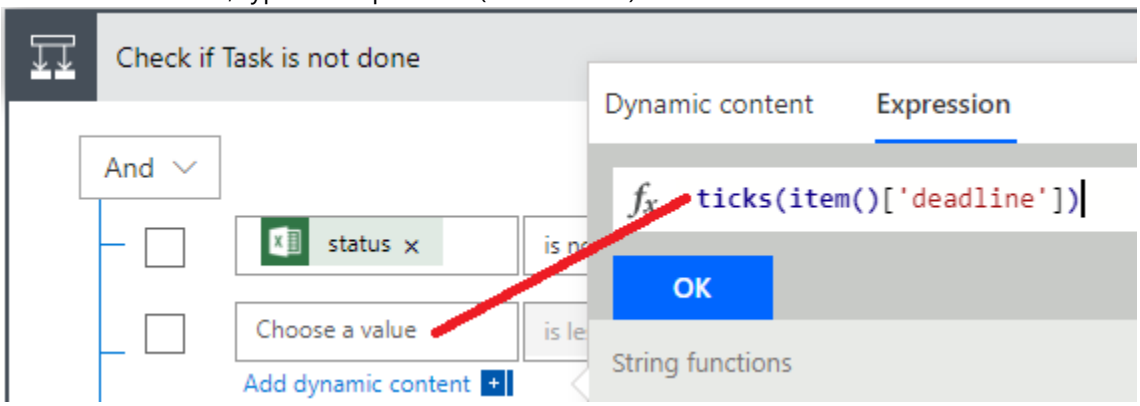
    a) Click Add:



b) Select Add row:



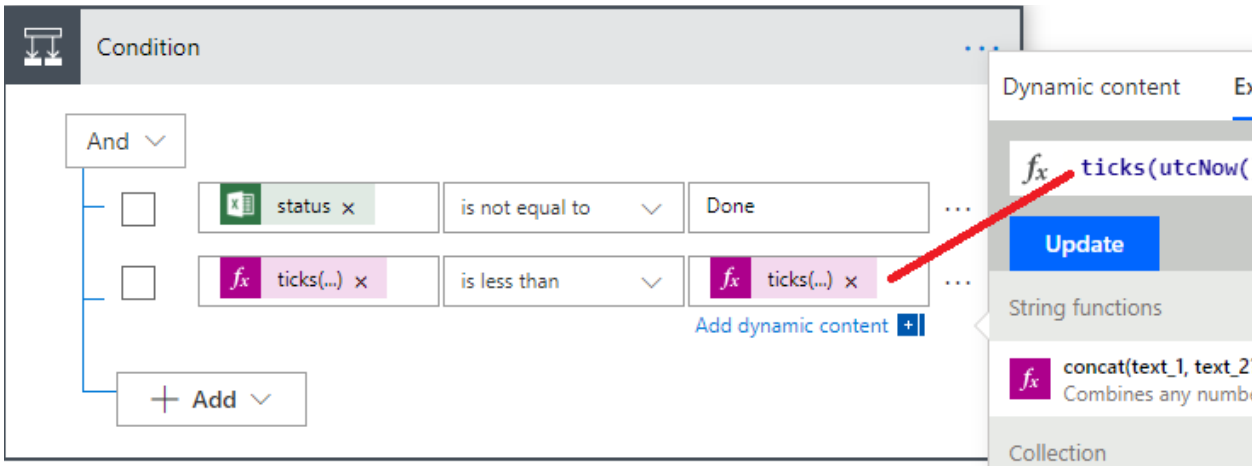
c) In "Choose a value", type the expression (and click Ok):



Since a timestamp is in string format, the **ticks** expression returns the number of ticks (100 nanosecond intervals) since 1st January 1601. By using ticks, we can compare two different timestamp values.

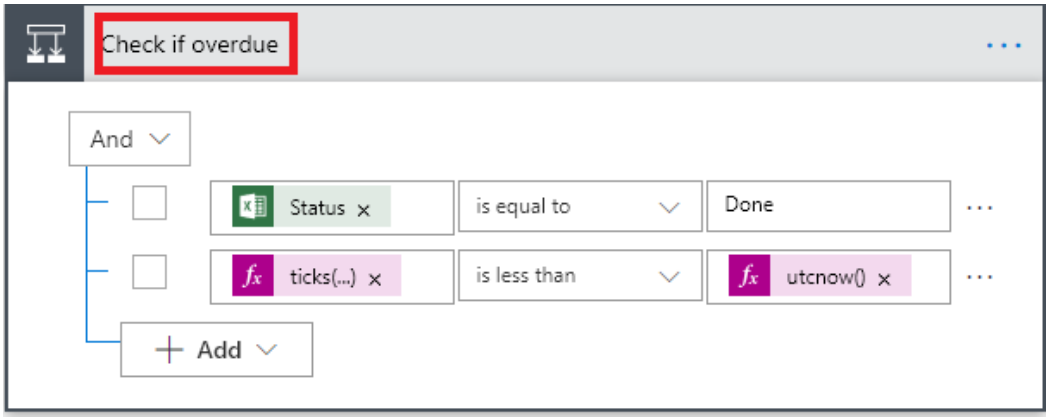
d) Select the operator is less than

e) Type the following expression (and click Ok):

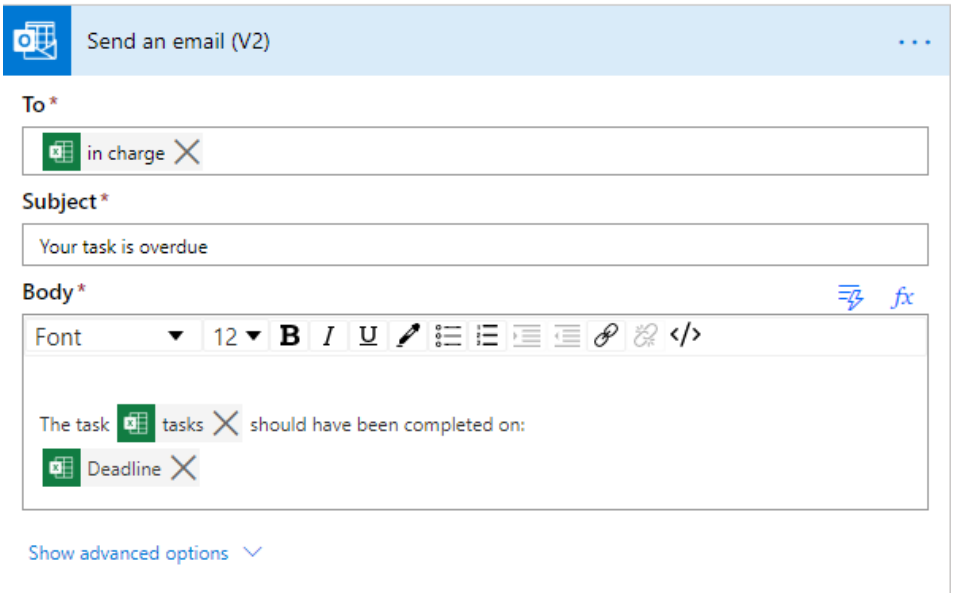


11. In the **If yes branch**, add a new **condition** where we will check if the due date is overdue:

12. Rename the condition to **Check if overdue** to provide your Flow with more clarity:



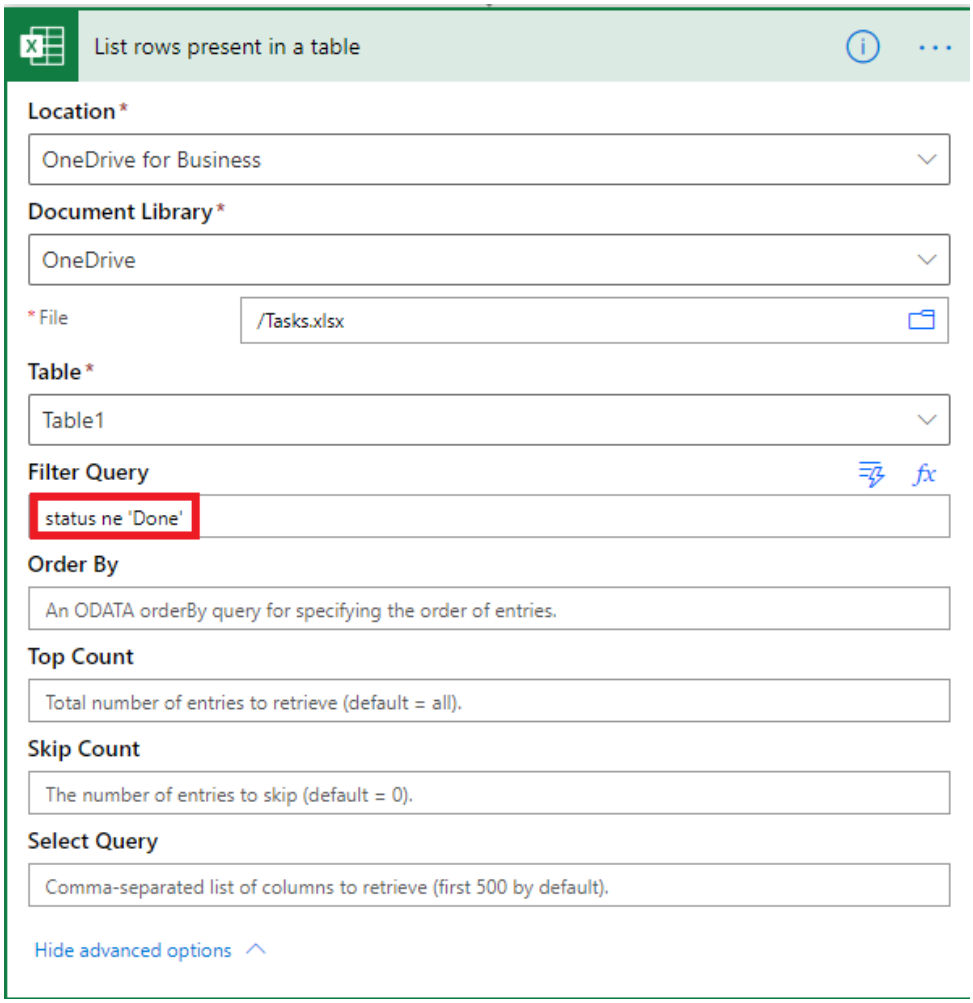
13. **Send an email (V2)** to the person in charge of this task and fill in the e-mail properties as following:
    a. The **To** field should get the Excel in charge value
    b. The **Subject** should be: "your task is overdue."
    c. The **Body** should be:



Note: If you cannot find the excel fields, click on **See more** link.

14. Run the Flow and check your e-mail.

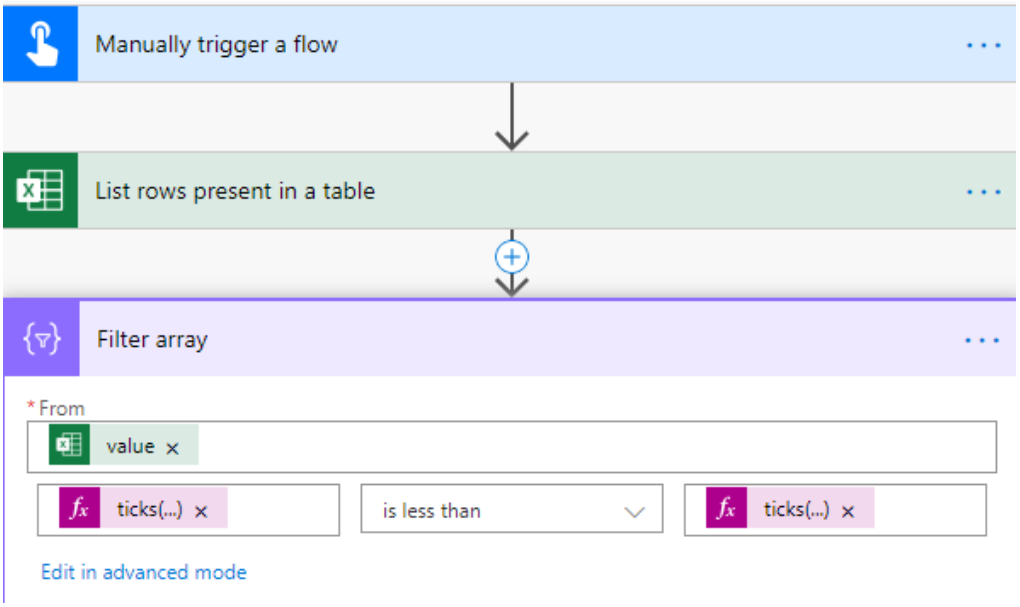
Note: instead of filtering the Tasks with the Status Done in a condition, you can also keep it more straightforward by using an OData Filter (click on **Show advanced options**) in the List rows present in a **table action**:



Using OData Filter (Filter query) is often considered as a good practice because the data are filtered on server-side and that can make you Flow running faster by using fewer resources.

In some cases, filtering data in the Flow itself cannot be avoided. In this case, you can use the Filter Array action instead of adding conditions.

Try the following code:


Manually trigger a flow
List rows present in a table
Filter array
* From
value
ticks(...)
is less than
ticks(...)
Edit in advanced mode