**Workaround Dataverse Trigger for When Two Specific Fields Are Changed Together**

You're absolutely right — the **"When a row is modified"** trigger in **Power Automate (Dataverse)** doesn't support directly filtering flows to run **only when specific combinations of fields change** (i.e., *Field A AND Field B* changed together). It triggers when **any** of the selected columns change (logical **OR**), not **AND**.

Here’s how to work around this limitation using conditional logic in your flow:

**✅ Step-by-Step Workaround: Trigger When Two Specific Fields Are Changed Together**

**🧩 Assumptions**

* Dataverse Table: MyTable
* Fields: fieldA, fieldB
* You want the flow to run **only when both fieldA AND fieldB are changed** in a single modification.

**1. Trigger: When a row is modified (Dataverse)**

* Table name: MyTable
* Scope: Choose based on your needs.
* **Select columns**: fieldA, fieldB (optional for performance — not mandatory).

**2. Get the Previous Version of the Record**

Since the trigger gives you only the **new values**, you’ll need the **previous values** to compare what actually changed.

**Option A: Use a "Get row" action**

* Use a **Get row by ID**, referencing the same row (using Row ID from the trigger).
* But this gives the **current** version again — so you may need versioning logic.

**Option B: Use Audit Logs (Complex)**

If auditing is enabled in Dataverse, you could extract historical values — but this is **not straightforward** in Power Automate directly.

**✔ Best Simple Workaround: Store Previous Values in Shadow Fields**

Create two shadow fields, e.g., prevFieldA and prevFieldB, and update them at the end of each flow run.

Then:

**3. Condition: Compare Old and New Values**

Add a **condition block** like this:

(fieldA != prevFieldA) AND (fieldB != prevFieldB)

Use expressions like:

and(

not(equals(triggerOutputs()?['body/fieldA'], triggerOutputs()?['body/prevFieldA'])),

not(equals(triggerOutputs()?['body/fieldB'], triggerOutputs()?['body/prevFieldB']))

)

This ensures the flow proceeds **only if both values changed**.

**4. Do Your Logic**

Only execute your desired logic (e.g., send email, update status, etc.) **within the If true branch** of the condition above.

**5. Update the Shadow Fields**

Update prevFieldA and prevFieldB with the new values of fieldA and fieldB respectively, so next time you'll have the old values for comparison.

**✅ Benefits of This Method**

* Works within Power Automate’s limitations.
* Doesn’t rely on auditing or complex triggers.
* Gives precise control over field-change logic.

Would you like a sample flow template or step-by-step screenshots for implementation?