



GENETRIX

SALESFORCE MARKETING CLOUD • JOURNEY BUILDER FIX

Journey Builder Sent to All Records Despite “Evaluate New Records Only”

The setting does exactly what it says — but a SQL Query Activity set to Overwrite destroys the previous-run snapshot it depends on, so every record looks new. Here is the root cause and the incremental-SQL fix.

by **Genetrix Technology**

Published June 10, 2026 • genetrix.tech/blogs

Journey Builder Sent to All Records Despite “Evaluate New Records Only”

By Genetrix Marketing • Published June 10,

2026 • <https://genetrix.tech/blogs/journey-builder-sent-to-all-records-despite-evaluate-new-records-only-heres-why/>

The setting does exactly what it says — but a SQL Query Activity set to Overwrite destroys the previous-run snapshot it depends on, so every record looks new. Here is the root cause and the incremental-SQL fix.

The setup looked correct. The Journey was configured with a daily Automation Studio entry source, the SQL query was pulling only recent purchases, and Contact Evaluation was set to “Evaluate new records only.” When the Journey activated, it sent to every contact already in the Data Extension — including people who had made purchases weeks ago and should never have received that email.

This is one of those Journey Builder behaviors that seems like a bug but is actually documented platform behavior that almost nobody reads carefully enough before going live. We have seen it catch teams off guard repeatedly. The setting does exactly what it says — but only under specific conditions that most setups accidentally violate.

What “Evaluate New Records Only” Actually Does

When Journey Builder runs against a Data Extension entry source on a schedule, “Evaluate new records only” tells it to compare the current state of the DE against a snapshot of the DE from the previous run. Only records that did not exist in that previous snapshot are treated as new and injected into the Journey.

That comparison depends entirely on the DE having a stable state between runs. Journey Builder needs to look at what was there before and what has been added since. If something removes that historical state — if the DE’s previous contents are wiped before the comparison can happen — Journey Builder has nothing to compare against. Every record in the DE looks new because as far as it can tell, all of them just appeared.

Root cause: When the SQL Query Activity feeding the entry DE is set to Overwrite, it deletes and replaces the entire DE contents on every run. Journey Builder’s “new records” comparison depends on the previous DE state existing. Overwrite destroys that state before Journey Builder can use it. The result is that every record in the DE is treated as new on every run, regardless of what the Contact Evaluation setting says.

The platform is not ignoring your setting. It is doing exactly what you configured. The problem is that the SQL action upstream is eliminating the condition that makes the setting work.

The Exact Scenario That Causes This

The typical broken setup looks like this. An Automation Studio automation runs daily. It contains a SQL Query Activity that selects all contacts who made a purchase, with the query action set to Overwrite on the target DE. Journey Builder is then triggered from that automation, using the same DE as the entry source, with Contact Evaluation set to “Evaluate new records only.”

Broken setup

```
SQL Query runs daily
-> Action: Overwrite DE (wipes all history)
-> Journey evaluates 'new records only'
-> No prior state to compare. All records look new.
=> Sends to EVERY record in the DE
```

Correct setup

```
SQL Query selects only today's new records
-> Action: Overwrite DE (only new records written)
-> Journey set to 'Evaluate all records'
-> DE only contains today's new data. All eligible.
=> Sends only to genuinely new records
```

The Fix

The cleanest solution is to change where the filtering happens. Instead of putting all historical records into the DE and relying on Journey Builder to determine which ones are new, you build the incremental logic into the SQL query itself. The DE then only ever contains the records you actually want the Journey to process on that specific run. With that in place, you switch Journey Builder to “Evaluate all records” — because every record in the DE is already a new one by definition.

Automation Studio SQL – Incremental daily new purchases

```
/* Pull only contacts who made a new purchase in the last 24 hours.
DE action: Overwrite.
Journey Contact Evaluation: Evaluate all records.
Every record written here is a genuinely new purchaser. */

SELECT
    p.SubscriberKey,
    p.EmailAddress,
    p.FirstName,
    p.PurchaseDate,
    p.ProductName,
    p.OrderValue

FROM Purchase_Events p

WHERE p.PurchaseDate >= DATEADD(day, -1, GETDATE())
    AND p.PurchaseDate < GETDATE()
```

What about the Append approach? You can use Append instead of Overwrite and keep “Evaluate new records only” — but this requires careful management. The DE will grow indefinitely, and if the SQL ever inserts a duplicate SubscriberKey, Journey Builder’s behavior depends on whether you have a primary key constraint. For high-volume orgs running daily, Overwrite with incremental SQL is cleaner and less likely to cause downstream problems.

The First Run Problem

There is one additional edge case worth knowing about. Even with the correct setup, the very first run after you activate a Journey has no previous snapshot to compare against regardless of what approach you use. SFMC handles this with a specific setting that appears during activation: you can choose whether the first run processes all records currently in the DE or skips them entirely and only picks up records from the next run onward.

If your DE already has records in it at activation time and you choose “all records” for the first run, those contacts will enter the Journey. This is expected behavior and is actually documented — but it surprises people who

assume “evaluate new records only” protects them at activation too. It does not. Clear the DE before activating, or choose “new records only” for the initial run, if you do not want the existing population to enter.

How to Verify Your Setup Is Correct

- Open the SQL Query Activity in your automation and check the Data Action. If it says Overwrite, your SQL must only select records you want to process today — not all historical records.
- If your SQL selects all historical records with Overwrite, switch to incremental SQL with a date filter. Change Journey Contact Evaluation to “Evaluate all records.”
- If you want to keep Overwrite and “Evaluate new records only,” your SQL must only ever write records that did not exist in the DE before the current run.
- Before activating any Journey against a pre-populated DE, clear the DE first if you do not want existing records to enter on the first run.
- After your fix, run the automation once manually and check the Journey’s entry count against the DE record count.
- Check your Journey’s entry history in Journey Analytics after the first few scheduled runs to confirm the daily entry counts look right.

Frequently Asked Questions

Does this behavior only affect SQL Query Activities?

No. The same issue applies to Filter Activities. The ‘Evaluate new records only’ option cannot be used when overwriting with a SQL Activity or when using a Filter Activity. In those cases, it functions the same as selecting ‘Evaluate all records.’ If you are using a Filter Activity that rebuilds the DE from scratch each run, the same root cause applies and the same fix is needed.

Can I use Append instead of Overwrite to keep the setting?

Yes, with caveats. If you use Append, Journey Builder can compare the current DE state against its previous snapshot because the old records are still there. Only genuinely new rows get picked up. The risk is that the DE grows indefinitely over time, and you need to ensure your SQL is not inserting duplicate SubscriberKeys on each run. For most daily recurring journeys we recommend incremental SQL with Overwrite because it is more predictable and easier to maintain at scale.

We already sent to the wrong contacts. What do we do now?

First, stop the Journey if it is still running. Then assess the scope: query your journey’s sending data to identify which contacts received the email unexpectedly. Whether you need to send a correction email depends on your brand guidelines and the nature of the content. For the fix going forward, implement incremental SQL as described above, clear the DE, and reactivate.

What if I need historical records in the DE for other journey logic but still only inject new ones?

Use two Data Extensions. One is the full historical DE you use for decision split lookups or personalization data. The second is the daily entry DE that only ever contains today’s new records, populated by a separate incremental SQL query. Use the entry DE as the Journey’s entry source with “Evaluate all records,” and reference the full historical DE via Contact Data or attribute lookups within the Journey canvas.

Journey Builder Behaving Unexpectedly?

SFMC’s Journey Builder has several behaviors like this one that are documented but not obvious until something goes wrong in production. Genetrix reviews Journey configurations, automation setups, and data architecture for teams who want to catch these before go-live, not after.

Talk to Genetrix »