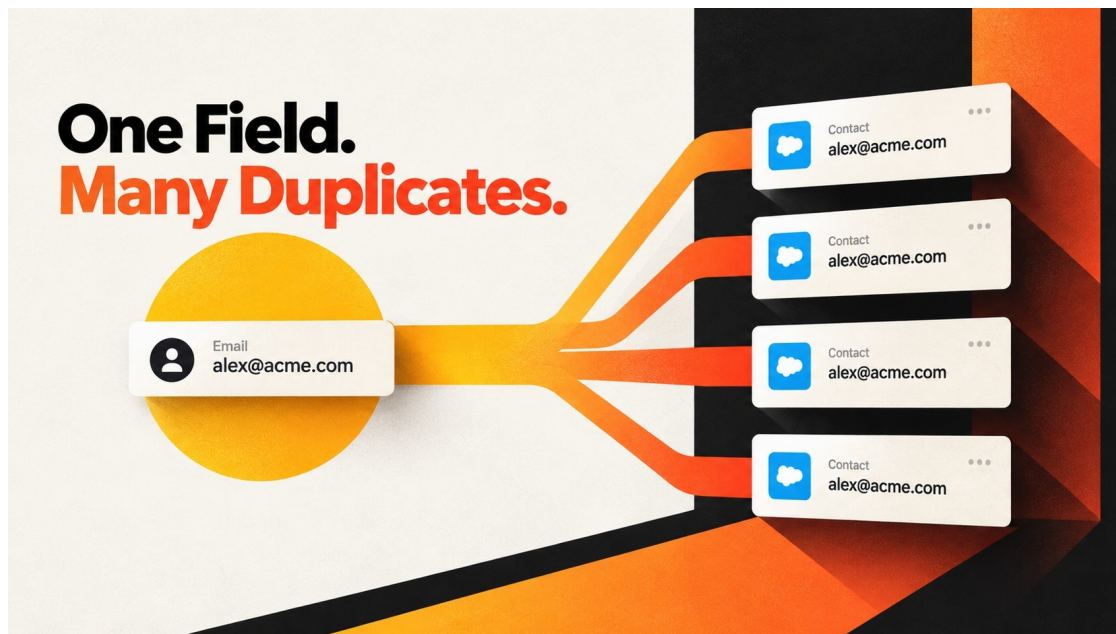


Single-Field Matching Creates Duplicates

Smart Lookup Data Loader | TwinStack Solutions

Blog 3 of 10



Duplicates rarely arrive in a batch. They arrive one row at a time, over months, from imports that looked like they worked.

How the failure happens

An upsert needs a rule for deciding whether a row is an update or an insert. Match on a single field and that rule is only as reliable as that one value.

Take email. It looks like a natural key for Contacts. Then you meet the real data:

- The same person appears as `j . chen@acme . com` and `j chen@acme . com`.
- A shared inbox like `accounts@acme . com` covers four people.
- A contact changed employer and kept the record but changed the address.
- Somebody typed a trailing space.

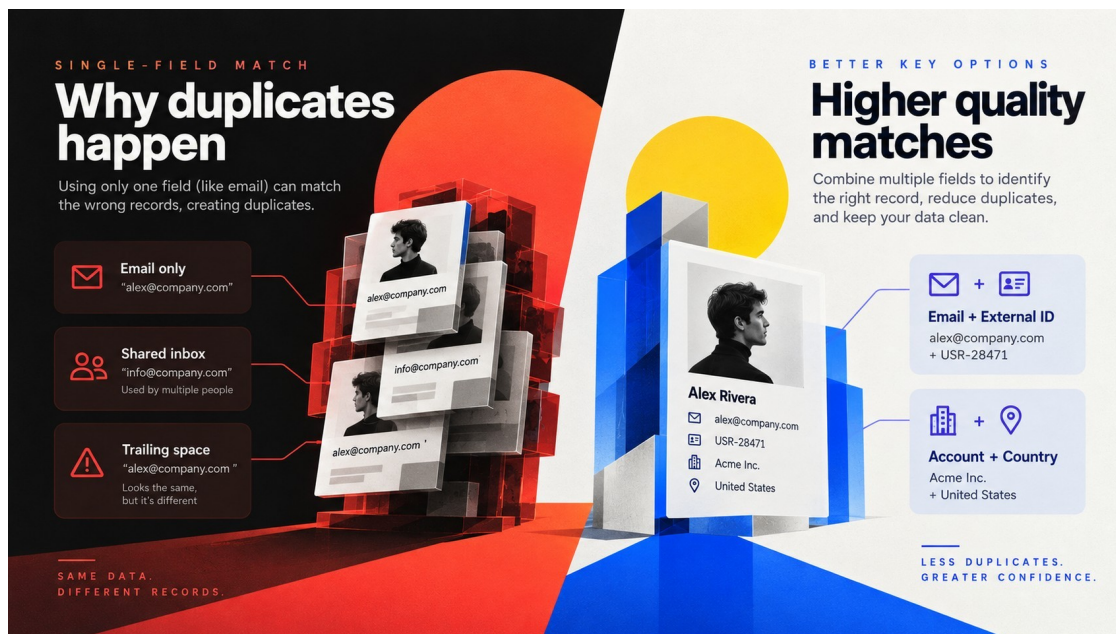
Every one of those creates a new record rather than updating the existing one, and the import reports complete success. Nothing failed. The loader did exactly what you told it to do.

Account name is worse. “Acme Corp”, “Acme Corporation”, “ACME Corp.” and “Acme Corp Ltd” are four records to a matching engine and one company to everybody else.

Why you find out late

A duplicate does not break anything immediately. It sits there accumulating: an activity logged against one copy, an opportunity against the other, a support case against a third. Reporting looks fine because the totals still add up. The problem surfaces when someone notices a customer’s history is split, which is usually months later and usually in front of that customer.

By then, merging is expensive, because the records have genuinely different children attached.



Why single-field matching creates duplicates, and what a better key looks like.

Choosing a better key

Use an external ID where one exists. If the source system has its own identifier, put it in a custom External ID field and match on that. It was designed to be unique and it does not change when a person’s job title does.

Match on a combination. Email plus external code, or account name plus billing country, or last name plus account. Two fields agreeing is a much stronger signal than one.

Normalise before you match. Trim whitespace, standardise case, strip punctuation from names. Five minutes in the source file prevents a lot of near-misses.

Be specific about people versus roles. If a mailbox is shared, it is not a person identifier, and matching Contacts on it will merge four humans into one record or split one into four.

A useful test before you commit

Take a hundred rows from the middle of your file, not the top. The top of a file is usually the cleanest part, because it is what whoever built it was looking at while they built it.

Run your intended match rule against those hundred rows and check two things: how many matched more than one existing record, and how many matched none. Both numbers should be near zero. If either is not, your key is wrong, and you have found out for the cost of a hundred rows.

What multi-field matching changes

Smart Lookup Data Loader lets you choose a combination of fields as the match rule rather than a single one, and it does this before anything is written. You set the rule, review what it will do, then run.

That does not remove the need to think about your data. It removes the situation where a single ambiguous column silently becomes your definition of a unique record.

SEO metadata

Field	Value
Slug	salesforce-duplicate-records-match-fields
Target keyword	salesforce upsert duplicate records
Secondary keywords	salesforce duplicate accounts import, salesforce match key, upsert external id salesforce
Meta description	Most duplicate records created during a Salesforce import come from matching on one field. Here is how to choose a match key that holds up.
Type	Data quality

Smart Lookup Data Loader is a free, native Salesforce managed package by TwinStack Solutions. Install it from the AppExchange, or read more at twinstack.net/products/smart-lookup-data-loader