

# fn match\_group\_by\_truncation

Michael Shoemate

January 20, 2026

This proof resides in “**contrib**” because it has not completed the vetting process.

Proves soundness of `match_group_by_truncation` in `mod.rs` at commit `f5bb719` (outdated<sup>1</sup>).

## 1 Hoare Triple

### Precondition

#### Compiler Verified

Types matching pseudocode.

### Precondition

None

### Function

```
1 def match_group_by_truncation(
2     plan: DslPlan, identifier: Expr
3 ) -> Optional[Tuple[DslPlan, Truncation, Bound]]:
4     if not isinstance(plan, DslPlan.GroupBy): #
5         return None
6
7     input = plan.input
8     keys = plan.keys
9     aggs = plan.aggs
10    apply = plan.apply
11    options = plan.options
12
13    if apply is not None or options != GroupbyOptions.default():
14        return None #
15
16    ids, by = partition(lambda expr: expr == identifier, keys) #
17
18    if not ids: #
19        return None
20
21    return (
22        input,
23        Truncation.GroupBy(keys, aggs),
24        Bound(by=by, per_group=1, num_groups=None), #
```

<sup>1</sup>See new changes with `git diff f5bb719..b5efac04 rust/src/transformations/make_stable_lazyframe/truncate/matching/mod.rs`

```

25     )
26
27
28 # part of Rust's standard lib, included for readability, with hardcoded types
29 def partition(
30     predicate, iterable: Iterable[Expr]
31 ) -> Tuple[HashSet[Expr], HashSet[Expr]]:
32     true_set = set()
33     false_set = set()
34     for item in iterable:
35         if predicate(item):
36             true_set.add(item)
37         else:
38             false_set.add(item)
39     return true_set, false_set

```

## Postcondition

**Theorem 1.1** (Postcondition). For a given query plan and user identifier expression, if the query plan bounds row contributions per-identifier via a group by, returns a triple containing the input to the truncation, the truncation itself, and the per-id bound on user contribution.

*Proof.* Lines 4-14 check if the query plan is a simple group-by, and exits without a match if not.

A valid group by truncation is a group by where the grouping keys contain a user identifier. So line 16 splits the group-by keys into two sets, and line 18 rejects the match if the user identifier is not in the group-by keys.

The remaining grouping columns are then reflected in the bound on user contributions, on line 24.  $\square$