fn match_group_by_truncation

Michael Shoemate

October 22, 2025

> 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Proves soundness of match_group_by_truncation in mod.rs at commit f5bb719 (outdated[1]).

1 Hoare Triple

Precondition

Compiler Verified

Types matching pseudocode.

Precondition

None

Function

```python
def match_group_by_truncation(
    plan: DslPlan, identifier: Expr
) -> Optional[Tuple[DslPlan, Truncation, Bound]]:
    if not isinstance(plan, DslPlan.GroupBy): #
        return None

    input = plan.input
    keys = plan.keys
    aggs = plan.aggs
    apply = plan.apply
    options = plan.options

    if apply is not None or options != GroupbyOptions.default():
        return None #

    ids, by = partition(lambda expr: expr == identifier, keys) #

    if not ids: #
        return None

    return (
        input,
        Truncation.GroupBy(keys, aggs),
        Bound(by=by, per_group=1, num_groups=None), #
```

---

[1]See new changes with `git diff f5bb719..5cc6975 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. □