

Dear support

We are encountering an issue with the latest version (3.2.0) of the Spanner client library for Python where the StreamedResultSet is missing an element of an array. The problem seems to be located in the `_merge_array`-function of the file `google->cloud->spanner_v1->streamed` [1]. Our code calls the `execute_sql` on a transaction (provided by the library) which returns a StreamedResultSet. The stream that is consumed by this ResultSet when looping over it rows can contain multiple chunks. If that is indeed the case, the pending chunk (of the previous response) will be merged with the first value of the new response as it is possible that the previous response contained only part of a column at the end [2]. This part of the code can cause (and is causing in our case) an element to be removed from the array column.

The latest occurrence of this issue was caused by a chunked stream that split an array column in two halves. The column that was being read contained the follow value: `[NULL, NULL, NULL, '5352537', '5352538', '5352539', ...]` and the first chunk of the response contained the first 3 elements `[NULL, NULL, NULL]` and the second chunk had the remaining of the list: `['5352537', '5352538', '5352539', ...]`

The `_merge_chunk`-function will call the `merge_by_type`-function which will use the `_merge_array` as the type of the column is `ARRAY<STRING>`. The latter function will first execute some basic checks and sanity checks and will then execute the following code:

```
first = rhs.pop(0) # -> value: '5352537'
if first.HasField("null_value"): # can't merge
    lhs.append(first)
else:
    last = lhs.pop() # -> value: NULL
    try:
        merged = _merge_by_type(last, first, element_type) # -> return value: '5352537'
    except Unmergeable:
        lhs.append(last)
        lhs.append(first)
    else:
        lhs.append(merged)
return Value(list_value=ListValue(values=(lhs + rhs)))
```

The variables `lhs` and `rhs` are assigned to `[NULL, NULL, NULL]` and `['5352537', '5352538', '5352539', ...]` respectively at the beginning of this code block. Stepping through this code shows us that `first` will be assigned to `'5352537'`, `last` to `NULL` and `element_type` will be `STRING` as the column type is `ARRAY<STRING>`. The `_merge_by_type`-call will thus use the `_merge_string` function which will return a new Value of which the string value is equal to `last.string_value + first.string_value`. As `last` is a `NULL`, the string value will be the empty string and the returned value will just be the same as the variable `first`. At this point, the third element of the input variable `lhs` is completely removed from the array.

We are not familiar with this part of the library but it looks strange to us that you would want to merge two elements of an array unless it is possible that the values in an array are also chunked.

Kind regards