# Lab 10: Practice Querying Nested and Repeated Fields

## Working with Nested and Repeated Fields

You are given a reporting table that has nested and repeated rows. Each ein has it's name and corresponding expense and revenue accounts repeated. You will practice querying repeated rows through the use of `UNNEST( )`

### Step 1

Skim the below schema and note the repeated fields

# Table Details: irs_990_repeated

Schema | Details | Preview

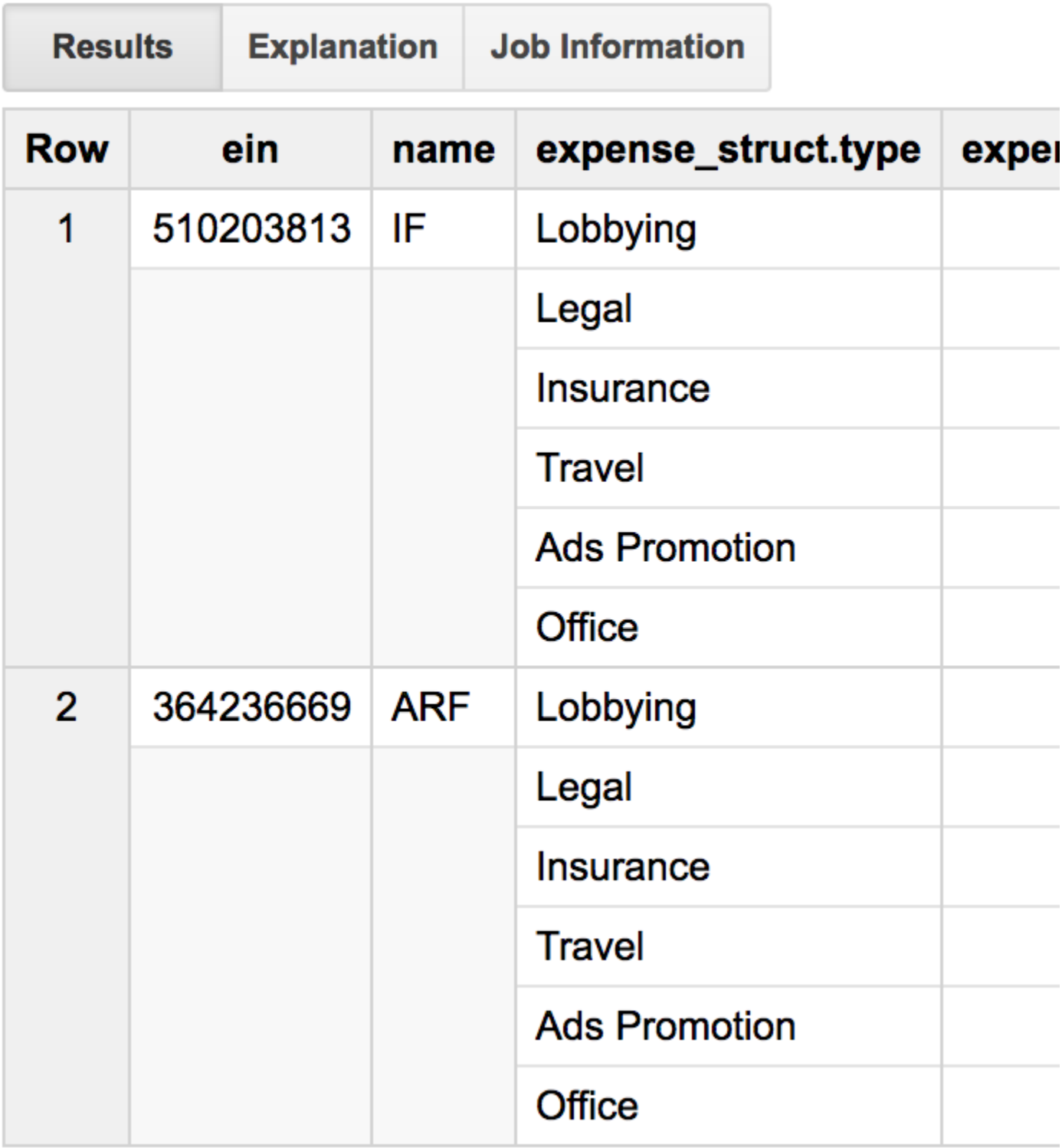
| | | | |
|---|---|---|---|
| **ein** | INTEGER | NULLABLE | Descr |
| **name** | STRING | NULLABLE | Descr |
| **expense_struct** | RECORD | REPEATED | Descr |
| expense_struct.**type** | STRING | NULLABLE | Descr |
| expense_struct.**amount** | INTEGER | NULLABLE | Descr |
| **revenue_struct** | RECORD | REPEATED | Descr |
| revenue_struct.**type** | STRING | NULLABLE | Descr |
| revenue_struct.**amount** | INTEGER | NULLABLE | Descr |

Step 2

Preview the Contents of the table by running the following query (or using the Web UI to find and Preview)

```
#standardSQL
SELECT * FROM `data-to-insights.irs_990.irs_990_repeated` LIMIT 10
```

The results should look similar to the below:



Results | Explanation | Job Information

| Row | ein | name | expense_struct.type | expe |
|---|---|---|---|---|
| 1 | 510203813 | IF | Lobbying | |
| | | | Legal | |
| | | | Insurance | |
| | | | Travel | |
| | | | Ads Promotion | |
| | | | Office | |
| 2 | 364236669 | ARF | Lobbying | |
| | | | Legal | |
| | | | Insurance | |
| | | | Travel | |
| | | | Ads Promotion | |
| | | | Office | |

Note: Using BigQuery with **Standard SQL** will automatically flatten the array content into rows (as shown above) compared to Legacy SQL will return an error stating you need to explicitly flatten.

### Step 3

Attempt to filter for all the **Legal expenses** by running the below query

```
#standardSQL
SELECT * FROM `data-to-insights.irs_990.irs_990_repeated`
WHERE expense_struct.type = 'Legal'
LIMIT 10
```

### Step 4

Confirm the error

**Error:** Cannot access field type on a value with type ARRAY<STRUCT<type STRING, amount INT64>>
We need to find another way to access those array elements.

### Step 5

Find the top 10 nonprofits that spent the most on legal expenses in the table. As a start, complete the below query by adding in the appropriate `UNNEST( )` and `WHERE` Clause filter on the expense type for 'Legal'. Hint: `UNNEST( )` typically follows the `FROM` much like a JOIN and should enclose a STRUCT.

```
#standardSQL
# Expenses by Category for each EIN
SELECT
  ein,
  expense
FROM `data-to-insights.irs_990.irs_990_repeated` n
CROSS JOIN expense_struct AS expense
ORDER BY expense.amount DESC
LIMIT 10
```

### Step 6

Compare against the below solution

```
#standardSQL
# Expenses by Category for each EIN
SELECT
  ein,
  expense
FROM `data-to-insights.irs_990.irs_990_repeated` n
CROSS JOIN UNNEST(n.expense_struct) AS expense
```

```
WHERE expense.type = 'Legal'
ORDER BY expense.amount DESC
LIMIT 10
```

### Step 7

Run the Query

### Step 8

Confirm the Result:

Results | Explanation | Job Information

| Row | ein | expense.type | expense.amount |
|---|---|---|---|
| 1 | 135598093 | Legal | 37124278 |
| 2 | 941196203 | Legal | 27291529 |
| 3 | 311626179 | Legal | 25097385 |
| 4 | 61242574 | Legal | 22771111 |
| 5 | 208295721 | Legal | 22270284 |
| 6 | 42103594 | Legal | 22062000 |
| 7 | 581954432 | Legal | 21756603 |
| 8 | 390833612 | Legal | 18268168 |
| 9 | 942854057 | Legal | 18224597 |
| 10 | 133298818 | Legal | 17976120 |

Note: The `CROSS JOIN` and `UNNEST( )` can be shorthanded as shown below. The `CROSS JOIN` becomes a comma and the `UNNEST( )` is optional as BigQuery assumes you want to unnest the n.expense_struct.

```
#standardSQL
# Expenses by Category for each EIN
SELECT
  ein,
  expense
```

```
FROM `data-to-insights.irs_990.irs_990_repeated` n, n.expense_struct AS expense
WHERE expense.type = 'Legal'
ORDER BY expense.amount DESC
LIMIT 10
```

# Congratulations!

You have completed the **Nested and Repeated Fields** lab

### Learning Review

Arrays contain a set of values in a single field
Use UNNEST() to work with data in arrays
STRUCTs are containers for data
BigQuery Repeated Fields are Arrays of Structs
BigQuery tables with repeated fields are essentially pre-joined tables with significant performance benefits

## References

- Standard SQL and Legacy SQL handle repeated fields in very different ways. Refer to the migration guide to see key differences in syntax.
- Standard SQL Working with Arrays

Provide Feedback on this Lab