Google Bigquery - Structs and Arrays Simplified

Author – Nilesh Khandalkar

Structs – Where Type = Record is a Structs
Array – Where Type = Record and Mode = Repeated, it's a Array.

Let me illustrate both with an example from a table I created for taxi_trip.

In the below Schema for **taxi_trip**, notice we have attribute **event** as the record type and mode is repeated so this is an array. We have another attribute **pickup** as the record type but mode here is not repeated, it's a Structs

taxi_trip — QUERY TABLE

Field name	Type	Mode	Policy tags	Description
order_id	STRING	NULLABLE		
service_type	STRING	NULLABLE		
Payment_Method	STRING	NULLABLE		
event	RECORD	REPEATED		
event. status	STRING	NULLABLE		
event. time	TIMESTAMP	NULLABLE		
pickup	RECORD	NULLABLE		
pickup. start_loc	STRING	NULLABLE		
pickup. drop_loc	STRING	NULLABLE		
total_dist	FLOAT	NULLABLE		

INSERT

Now let's insert values into this table, not that we have 2 attributes with Record type, out of which 1 is mode repeated.

```
INSERT INTO
  `Test_Project.Nil_Test.taxi_trip` (order_id,
    service_type,
    payment_method,
    event,
    pickup,
    total_dist)
VALUES
  ("A001","OLA PRIME","DEBIT CARD",[("Picked",
      TIMESTAMP('2020-05-05 01:00:00')),("dropped",
      TIMESTAMP('2020-05-05 02:00:00'))],("VASHI",
      "BANDRA"),25.5)
```

See how we have inserted into the table taxi_trip – Order_id, Service_type, Payment_method is simple insert that we do in SQL, for **event** (it's a record with repeated mode) we can have multiple line items within. We use it here to specify event_status and event_time as shown in above schema for taxi_trip
This has to be inserted in square brackets [] and each event has to be in circular brackets()
[("Picked", TIMESTAMP('2020-05-05 01:00:00')),("dropped",TIMESTAMP('2020-05-05 02:00:00'))]
You can keep adding more line items as required separated by comma.

For pickup (record type but not repeated) we have start_loc and drop_loc, we simply mention it in circular brackets separated by comma
("VASHI", "BANDRA")

Now let's see the output.

```
select * from `Test_Project.Nil_Test.taxi_trip`
```

Query complete (0.2 sec elapsed, 169 B processed)

Job information | Results | JSON | Execution details

Row	order_id	service_type	Payment_Method	event.status	event.time	pickup.start_loc	pickup.drop_loc	total_dist
1	A001	OLA PRIME	DEBIT CARD	Picked	2020-05-05 01:00:00 UTC	VASHI	BANDRA	25.5
				dropped	2020-05-05 02:00:00 UTC			
2	A002	OLA MINI	DEBIT CARD	Picked	2020-05-05 03:00:00 UTC	BANDRA	VASHI	25.5
				dropped	2020-05-05 04:00:00 UTC			

Notice that for each order_id we have multiple record for event as it is repeated.

JSON Format O/p -


```
[
 {
  "order_id": "A001",
  "service_type": "OLA PRIME",
  "Payment_Method": "DEBIT CARD",
  "event": [
   {
    "status": "Picked",
    "time": "2020-05-05 01:00:00 UTC"
   },
   {
    "status": "dropped",
    "time": "2020-05-05 02:00:00 UTC"
   }
  ],
  "pickup": {
   "start_loc": "VASHI",
   "drop_loc": "BANDRA"
  },
  "total_dist": "25.5"
 },
 {
  "order_id": "A002",
  "service_type": "OLA MINI",
  "Payment_Method": "DEBIT CARD",
  "event": [
   {
    "status": "Picked",
    "time": "2020-05-05 03:00:00 UTC"
   },
   {
    "status": "dropped",
    "time": "2020-05-05 04:00:00 UTC"
   }
  ],
  "pickup": {
   "start_loc": "BANDRA",
   "drop_loc": "VASHI"
  },
  "total_dist": "25.5"
 }
]
```


Select

Now, to select only specific attributes which are of type Record (here for this record type, the mode is not repeated). So we are able to select it directly.

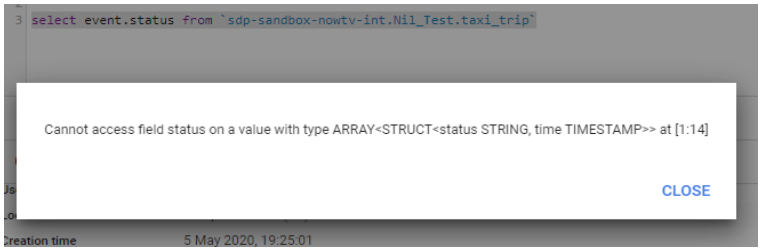
select pickup.start_loc,pickup.drop_loc from `Test_Project.Nil_Test.taxi_trip`

Query complete (0.2 sec elapsed, 30 B processed)

Job information | Results | JSON | Execution details

Row	start_loc	drop_loc
1	VASHI	BANDRA
2	BANDRA	VASHI

But if we try to select the same for event which is again record but mode is repeated, it gives error.



So, here we have to use UNNEST function.
select p.status,p.time from `Test_Project.Nil_Test.taxi_trip` ,unnest(event) as p

Query complete (0.3 sec elapsed, 66 B processed)

Job information | Results | JSON | Execution details

Row	status	time
1	Picked	2020-05-05 01:00:00 UTC
2	dropped	2020-05-05 02:00:00 UTC
3	Picked	2020-05-05 03:00:00 UTC
4	dropped	2020-05-05 04:00:00 UTC

Consider the below schema for Taxi_Trip_1, pickup struct has different datatype attributes and that's possible as it is in the case of arrays.

Taxi_Trip_1 QUERY TABLE COPY

Schema | Details | Preview

Field name	Type	Mode	Policy tags	Description
order_id	STRING	NULLABLE		
service_type	STRING	NULLABLE		
Payment_Method	STRING	NULLABLE		
pickup	RECORD	NULLABLE		
pickup. startloc	STRING	NULLABLE		
pickup. droploc	STRING	NULLABLE		
pickup. totalkms	INTEGER	NULLABLE		

So to insert, its –

```
INSERT INTO
  `Test_Project.Nil_Test.Taxi_Trip_1` (order_id,
    service_type,
    payment_method,
    pickup)
VALUES
  ('A001','Prime','Debit',('Vashi',
      'Bandra',
      25))
```

Again, here to select pickup it's pretty straight forward.

select pickup from `Test_Project.Nil_Test.Taxi_Trip_1`

Row	pickup.startloc	pickup.droploc	pickup.totalkms
1	Vashi	Bandra	25

select pickup.startloc from `Test_Project.Nil_Test.Taxi_Trip_1`

Row	startloc
1	Vashi

Now let's go one level down, we have the below table which has 2 level of structs

Field name	Type	Mode	Policy tag
orderid	STRING	NULLABLE	
passengerdet	RECORD	NULLABLE	
passengerdet. pickupinfo	RECORD	NULLABLE	
passengerdet.pickupinfo. startloc	STRING	NULLABLE	
passengerdet.pickupinfo. starttime	TIMESTAMP	NULLABLE	
passengerdet. dropinfo	RECORD	NULLABLE	
passengerdet.dropinfo. droploc	STRING	NULLABLE	
passengerdet.dropinfo. droptime	TIMESTAMP	NULLABLE	

To insert into this table, we have to write the below sql –
insert into `sdp-sandbox-nowtv-int.Nil_Test.taxi_trip_4` (orderid,passengerdet) values ('a001',struct(struct('vashi',current_timestamp()),struct('bandra',current_timestamp())))

Note here, we have used the keyword struct to form the insert sql as it is 2 level of structs.
O/p –

Row	orderid	passengerdet.pickupinfo.startloc	passengerdet.pickupinfo.starttime	passengerdet.dropinfo.droploc	passengerdet.dropinfo.droptime
1	a001	vashi	2020-05-27 12:22:29.744713 UTC	bandra	2020-05-27 12:22:29.744713 UTC

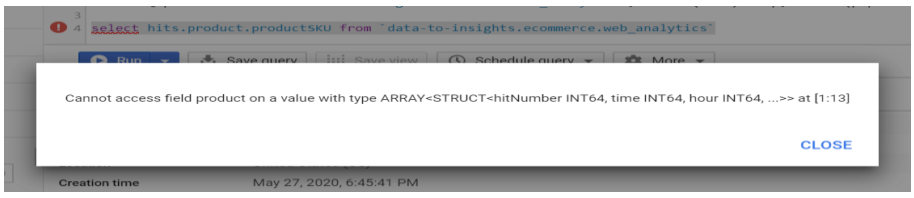
To select specific col which is an struct, we can write

select passengerdet.pickupinfo.startloc from `sdp-sandbox-nowtv-int.Nil_Test.taxi_trip_4`

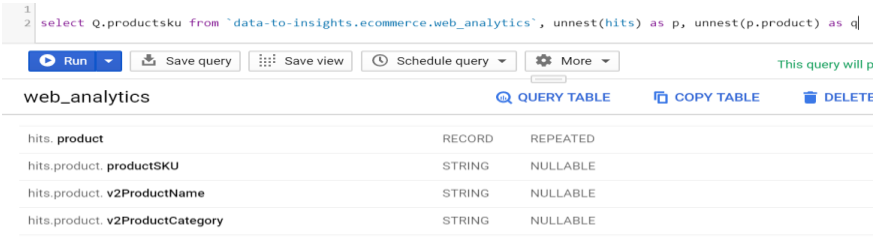
When to UNNEST and when to not

Now comes some interesting past, when to unnest and when to not. See below example, hits which is a parent column and is of type record with mode repeated, so simply selecting like below hits.product.productsku won't work and will give error.

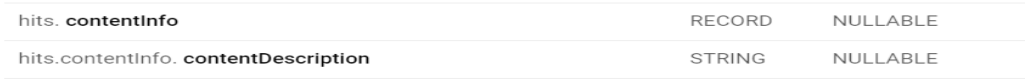
hits	RECORD	REPEATED



So, here we have to make use of UNNEST, since it's a 2 level array we unnest it twice (twice because parent hits and then nested product both are mode repeated), once for hits and then for product. Then we select the productsku as mentioned below.



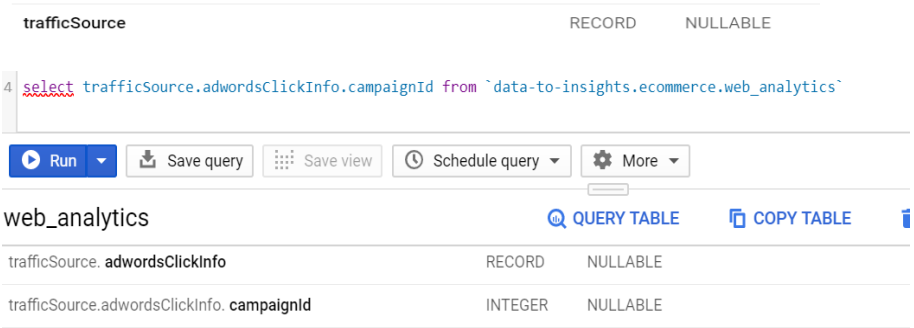
If inside hits, we have nested column as Record and mode nullable, still we wont be able to query directly as the parent column hits is mode repeated.



select hits.contentInfo.contentDescription from `data-to-insights.ecommerce.web_analytics` wont work and give error

Here we have to unnest at one level (because here parent hit is mode repeated but the nested contentinfo is mode nullable) then query
select p.contentInfo.contentDescription from `data-to-insights.ecommerce.web_analytics`, unnest(hits) as p will work

But in below example trafficsource which is a parent column is of type record and mode nullable (not repeated), so this will allow to query directly and UNNEST is not required.



So basically, points to consider are -

Scenario1	Parent Column: Type-Record Mode-Nullable	can query directly
Scenario2	Parent Column: Type-Record Mode-Nullable Nested Column: Type-Record Mode-Nullable	can query directly
Scenario3	Parent Column: Type-Record Mode-Repeated	UNNEST Parent

Scenario4	Parent Column: Type-Record Mode-Repeated Nested Column: Type-Record Mode-Repeated	UNNEST Parent and UNNEST Nested
Scenario5	Parent Column: Type-Record Mode-Repeated Nested Column: Type-Record Mode-Nullable	UNNEST Parent

You can check **data-to-insights.ecommerce.web_analytics** which is a public dataset for schema details in which they have used all types of Structs and Arrays.

Please reach out to me nileshk611@gmail.com for any clarification.