**Google Bigquery - Structs and Arrays Simplified**

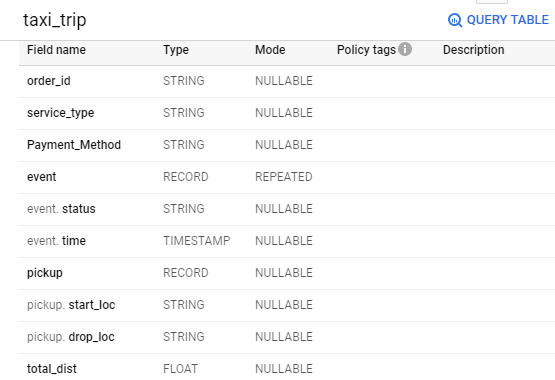
**Author – Nilesh Khandalkar**

**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



**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'))]

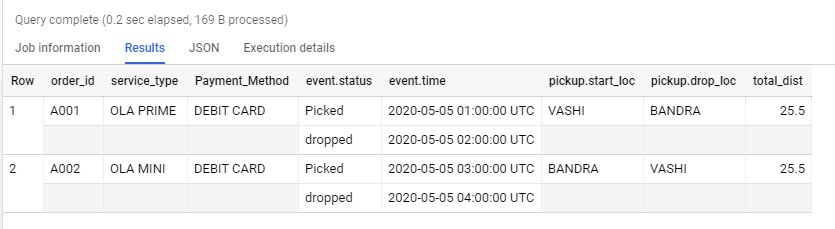
You can keep adding more line items as required separated by comma.

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`



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"

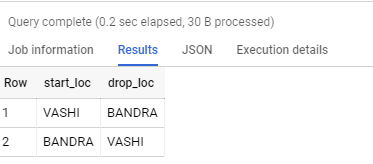
}

]

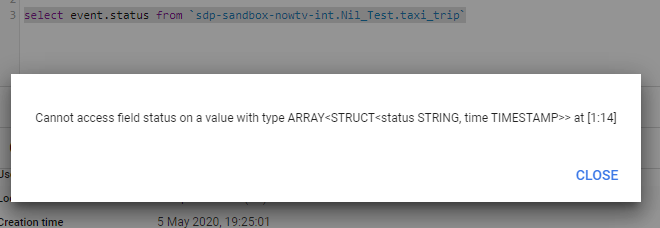
**Select**

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`

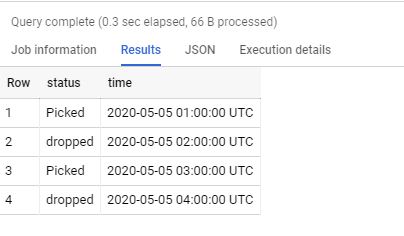


But if we try to select the same for event which is again record but mode is repeated, it gives error.

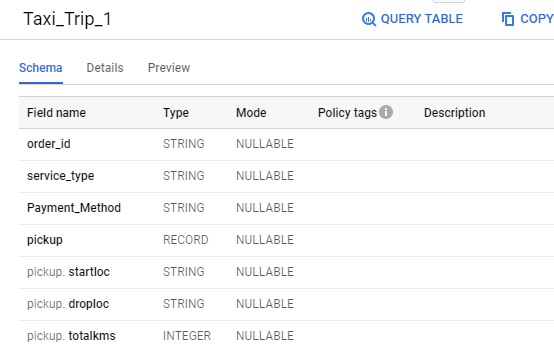


So, here we have to use UNNEST function.

select p.status,p.time from `Test\_Project.Nil\_Test.taxi\_trip`,unnest(event) as p



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.



So to insert, its –

INSERT INTO

`Test\_Project.Nil\_Test.Taxi\_Trip\_1` (order\_id,

service\_type,

payment\_method,

pickup)

VALUES

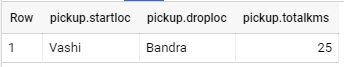
('A001','Prime','Debit',('Vashi',

'Bandra',

25))

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

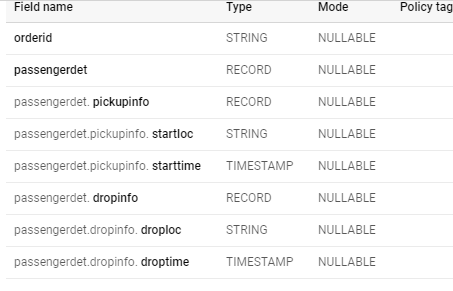
select pickup from `Test\_Project.Nil\_Test.Taxi\_Trip\_1`



select pickup.startloc from `Test\_Project.Nil\_Test.Taxi\_Trip\_1`



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



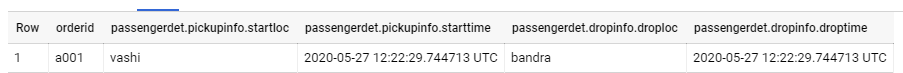
To insert into this table, we have to write the below sql –

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 –



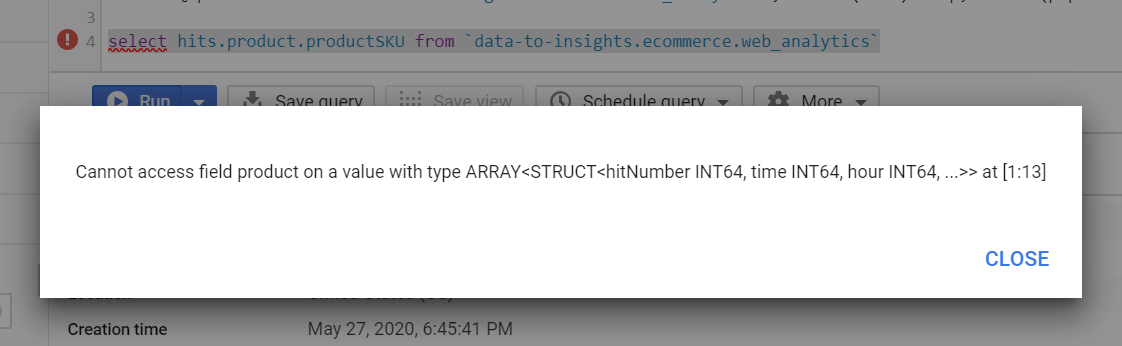
To select specific col which is an struct, we can write

select passengerdet.pickupinfo.startloc from `sdp-sandbox-nowtv-int.Nil\_Test.taxi\_trip\_4`

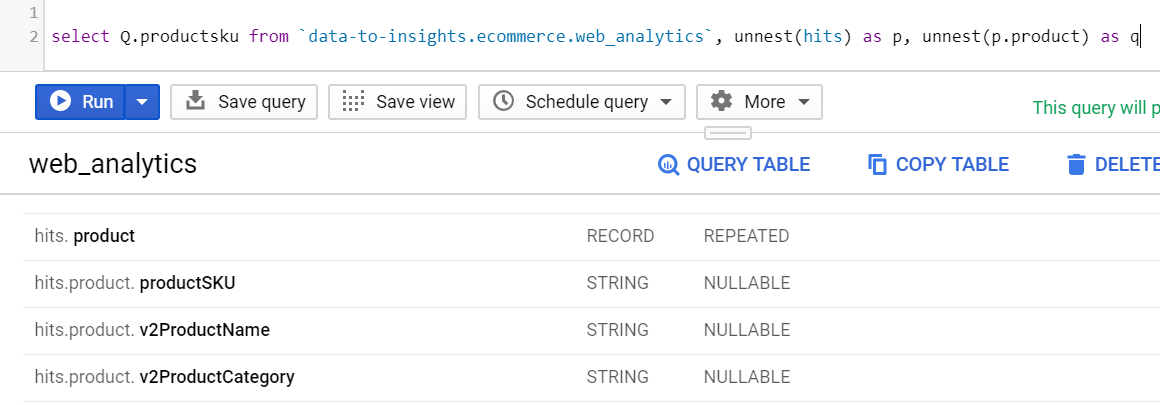
When to UNNEST and when to not

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.

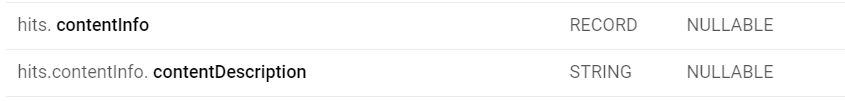




So, here we have to make use of UNNEST, since it’s a 2 level array we unnest it twice (twice because parent hits and then nested product both are mode repeated), once for hits and then for product. Then we select the productsku as mentioned below.



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.



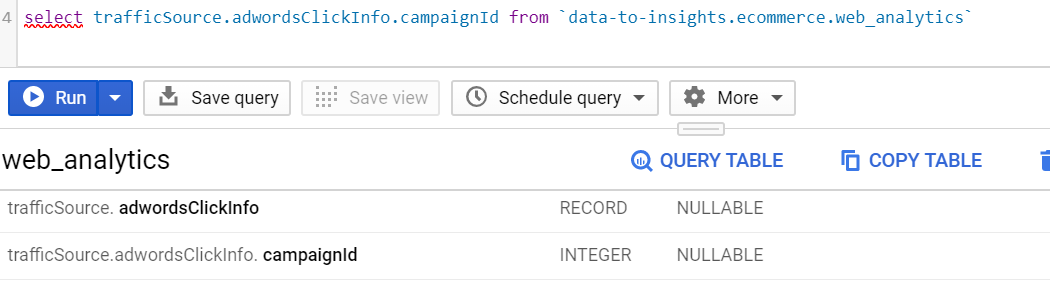
select hits.contentInfo.contentDescription from `data-to-insights.ecommerce.web\_analytics` wont work and give error

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 [nilesh.khandalkar@accenture.com](mailto:nilesh.khandalkar@accenture.com) for any clarification.