

Defining the JSON data in BQ

```
CREATE OR REPLACE TABLE `project_xpto.dataset_xpto.table_xpto` (  
  json_data STRING  
);
```

```
INSERT INTO `project_xpto.dataset_xpto.table_xpto` (json_data)  
VALUES (  
  '{  
    "name": "Alice",  
    "age": 35,  
    "married": false,  
    "children": [  
      '{  
        "name": "Bob",  
        "age": 10,  
        "school": {  
          "name": "Springfield Elementary",  
          "grade": "5th"  
        }  
      },  
      '{  
        "name": "Charlie",  
        "age": 8,  
        "school": {  
          "name": "Springfield Elementary",  
          "grade": "3rd"  
        }  
      }  
    ],  
    "address": {  
      "street": "Maple Avenue",  
      "number": 456,  
      "city": "Springfield",  
      "state": "IL",  
      "zip_code": "62704"  
    },  
    "phone": [  
      '{  
        "type": "home",  
        "number": "555-1234"  
      }  
    ]  
  }'
```

```

    },'
    {'
      "type": "mobile",'
      "number": "555-5678"'
    },'
  ],'
  "employment": {'
    "status": "employed",'
    "position": "Software Engineer",'
    "company": {'
      "name": "Tech Innovations",'
      "address": {'
        "street": "Tech Park",'
        "number": 789,'
        "city": "Springfield",'
        "state": "IL",'
        "zip_code": "62701"'
      }
    }
  }
}
}
}
);

```

Working with JSON in BigQuery

```

WITH base_people_data AS (
SELECT

PARSE_JSON(json_data) AS json_data_parsed

FROM `treinamentos-dados.treinamento_bq.tb_people_bq`

),

structured_base_people_data AS (

SELECT

--first level

STRUCT (

```

```

JSON_EXTRACT_SCALAR(json_data_parsed, '$.name') AS
name,
JSON_EXTRACT_SCALAR(json_data_parsed, '$.age') AS
age,
JSON_EXTRACT_SCALAR(json_data_parsed, '$.married') AS
married
) AS person,

--second level for children
STRUCT(

JSON_EXTRACT_SCALAR(children, '$.name') AS
name,
JSON_EXTRACT_SCALAR(children, '$.age') AS
age

) AS children,

-- second level for address

STRUCT(
JSON_EXTRACT_SCALAR(json_data_parsed.address, '$.street') AS
street,
JSON_EXTRACT_SCALAR(json_data_parsed.address, '$.number') AS
number,
JSON_EXTRACT_SCALAR(json_data_parsed.address, '$.city') AS
city,
JSON_EXTRACT_SCALAR(json_data_parsed.address, '$.state') AS
state,
JSON_EXTRACT_SCALAR(json_data_parsed.address, '$.zip_code') AS
zip_code
) AS address,

--second level for phone
STRUCT(
JSON_EXTRACT_SCALAR(phone, '$.type')
AS type,
JSON_EXTRACT_SCALAR(phone, '$.number')
AS number
) as phone,

--second level for employment
STRUCT (
JSON_EXTRACT_SCALAR(json_data_parsed.employment, '$.status')
AS status,

```

```

    JSON_EXTRACT_SCALAR(json_data_parsed.employment, '$.position')
AS position,
    JSON_EXTRACT_SCALAR(json_data_parsed.employment.company, '$.name')
AS name

) AS employment,

--third level for company employment
STRUCT (
JSON_EXTRACT_SCALAR(json_data_parsed.employment.company.address, '$.street') AS
street,
JSON_EXTRACT_SCALAR(json_data_parsed.employment.company.address, '$.number') AS
number,
JSON_EXTRACT_SCALAR(json_data_parsed.employment.company.address, '$.city') AS
city,
JSON_EXTRACT_SCALAR(json_data_parsed.employment.company.address, '$.state') AS
state,
JSON_EXTRACT_SCALAR(json_data_parsed.employment.company.address, '$.zip_code') AS
zip_code
) as company_adress

FROM base_people_data AS bpd
    LEFT JOIN UNNEST(JSON_EXTRACT_ARRAY(json_data_parsed, '$.phone')) AS
phone
    LEFT JOIN UNNEST(JSON_EXTRACT_ARRAY(json_data_parsed, '$.children')) AS
children
),

base_distinct as (
SELECT
*,
ROW_NUMBER() OVER (PARTITION BY children.name ORDER BY children.age DESC) AS
row_num

FROM structured_base_people_data

)

SELECT * FROM base_distinct WHERE row_num=1

```

Expected result

```
1 WITH base_people_data AS (  
2 SELECT  
3  
4 PARSE_JSON(json_data) AS json_data_parsed  
5  
6 FROM `treinamentos-dados.treinamento_bq.tb_people_bq`  
7  
8 )  
9  
10 structured_base_people_data AS (  
11
```

Query results [SAVE RESULTS](#) [EXPLORE DATA](#)

Line	person.name	person.age	person.married	children.name	children.age	address.street	address.number	address.city	address.state
1	Alice	35	false	Bob	10	Maple Avenue	456	Springfield	IL
2	Alice	35	false	Charlie	8	Maple Avenue	456	Springfield	IL

suits [SAVE RESULTS](#) [EXPLORE DATA](#)

employment.status	employment.position	employment.name	company.address.street	company.address.number	company.address.city	company.address.state	company.address.ZIP code	row_num
employed	Software Engineer	Tech Innovations	Tech Park	789	Springfield	IL	62701	1
employed	Software Engineer	Tech Innovations	Tech Park	789	Springfield	IL	62701	1