# Assignment 1 – BigQuery Field Modes

## Introduction

In BigQuery, schema fields can be classified using field modes: REQUIRED, NULLABLE, and REPEATED. These define whether a field must have a value, can be null, or supports multiple values. Understanding these modes is crucial for designing effective schemas and handling nested or repeated data structures.

## 1. REQUIRED

• The field must always have a non-null value.  
• Insert fails if the field is missing or null.  
  
Example:  
CREATE TABLE raw.customers (  
 customer\_id STRING NOT NULL, -- equivalent to REQUIRED  
 name STRING, -- NULLABLE by default  
 email STRING -- NULLABLE by default  
);  
  
• 'customer\_id' must always be provided.

## 2. NULLABLE (default)

• Field can have null values.  
• You can insert rows without providing this field.  
  
Example:  
CREATE TABLE raw.orders (  
 order\_id STRING,  
 customer\_id STRING,  
 discount FLOAT64  
);  
  
• 'discount' can be NULL if not applicable.

## 3. REPEATED

• Field stores an array of values (0 or more).  
• Used for nested or repeated data structures (arrays, structs).  
  
Data modeling often uses:  
- STRUCT (nested data) — represents a single record with multiple fields.  
- ARRAY (repeated data) — represents a list of values.  
  
These can be used separately or together and are ideal for semi-structured or JSON-like data.

## STRUCT (Nesting)

A STRUCT is a single nested object (like a row inside a row).  
  
Table Definition:  
CREATE TABLE staging.users\_with\_struct (  
 user\_id STRING,  
 profile STRUCT<  
 name STRING,  
 age INT64,  
 location STRING>  
);  
  
Insert Example:  
INSERT INTO staging.users\_with\_struct (user\_id, profile)  
VALUES  
 ('U001', STRUCT('Alice', 30, 'NY')),  
 ('U002', STRUCT('Bob', 28, 'LA'));  
  
Query Example:  
SELECT user\_id, profile.name, profile.age FROM staging.users\_with\_struct;

## ARRAY (Repeated)

An ARRAY is a list of values of the same type.  
  
Table Definition:  
CREATE TABLE staging.users\_with\_array (  
 user\_id STRING,  
 favorite\_colors ARRAY<STRING>  
);  
  
Insert Example:  
INSERT INTO staging.users\_with\_array (user\_id, favorite\_colors)  
VALUES  
 ('U001', ['red', 'blue']),  
 ('U002', ['green']);  
  
Query with UNNEST:  
SELECT user\_id, color  
FROM staging.users\_with\_array,  
UNNEST(favorite\_colors) AS color;

## Comparison Table: Modes

| Mode | Nullable? | Multiple Values? | Use Case |  
|----------|-----------|------------------|--------------------------|  
| REQUIRED | ❌ | ❌ | Must always have a value |  
| NULLABLE | ✅ | ❌ | Optional field |  
| REPEATED | ✅ | ✅ | Array or repeated records|

## STRUCT + ARRAY (Repeated Nested Records)

Table Definition:  
CREATE TABLE staging.users\_with\_addresses (  
 user\_id STRING,  
 addresses ARRAY<STRUCT<  
 street STRING,  
 city STRING,  
 zip STRING>>  
);  
  
Insert Example:  
INSERT INTO staging.users\_with\_addresses (user\_id, addresses)  
VALUES  
 ('U001', [STRUCT('123 Main St', 'New York', '10001'),  
 STRUCT('456 Side St', 'Brooklyn', '11201')]);  
  
  
Query Example:  
SELECT user\_id, addr.city, addr.zip  
FROM staging.users\_with\_addresses,  
UNNEST(addresses) AS addr;

## Feature Comparison: STRUCT vs ARRAY

| Feature | STRUCT | ARRAY |  
|--------------|-----------------------------|-----------------------------|  
| Represents | A single nested record | A repeated list of values |  
| Use Case | Nesting related fields | Repeating a field (e.g., tags) |  
| Querying | Dot notation (e.g., field.sub) | Use UNNEST() to access elements |  
| Can combine? | Yes (ARRAY<STRUCT<...>>) | Yes (ARRAY<STRUCT<...>>) |

## References

• BigQuery Nested and Repeated Fields: https://cloud.google.com/bigquery/docs/nested-repeated#sql  
• Best Practices: https://cloud.google.com/bigquery/docs/best-practices-performance-nested