

# Handling JSON → Medallion (Bronze → Silver → Gold) in Fabric Warehouse

## Raw Data:

- Validating the Raw Data

The screenshot shows the Object Explorer pane on the left with a tree view of databases and tables. The SQLQuery3 window on the right displays a query to select all from the dbo.Json\_Bronze table, showing 12 rows of JSON data. The results are presented in a grid with columns: RowId, SourceSystem, SourceFile, IngestedAt, and RecordDetails.

RowId	SourceSystem	SourceFile	IngestedAt	RecordDetails
1	CRM	crm_2026_02_16_nested.json	2026-02-16 14:01:58.437367	{ "customerId": 123, "name": "Rakesh", "address": { "city": "Bengaluru", "zip": "560001", "geo": { "lat": 12.9716, "lon": 77.5946 } }, "tags": [ "vip" ] }
2	CRM	crm_bad.json	2026-02-16 14:01:58.437367	{ "bad-json": true }
3	CRM	crm_2026_02_16.json	2026-02-16 14:01:58.437367	{ "customerId": 124, "name": "Ananya", "address": { "city": "Mumbai", "zip": "400001", "geo": { "lat": 19.076, "lon": 72.8777 } }, "tags": [ "new" ] }
4	CRM	crm_2026_02_16.json	2026-02-16 14:01:58.437367	{ "customerId": 125, "name": "Vikram", "address": { "city": "Delhi", "zip": "110001", "tags": [ "vip" ], "orders": [ { "orderId": 9004, "amount": 5400 } ] }
5	CRM	crm_2026_02_16.json	2026-02-16 14:01:58.437367	{ "customerId": 126, "name": "Neha", "address": { "city": "Pune", "zip": "411001", "tags": [ "newsletter" ], "orders": [ { "orderId": 9005, "amount": 0.0, "shipping": 99 } ] }
6	CRM	crm_2026_02_16.json	2026-02-16 14:01:58.437367	{ "customerId": 127, "name": "Suresh", "address": { "city": "Chennai", "zip": "600001", "tags": [ "orderId": 9006, "amount": 0.0, "shipping": 99 } }
7	CRM	crm_2026_02_16.json	2026-02-16 14:01:58.437367	{ "customerId": 128, "name": "Meera", "address": { "city": "Hyderabad", "tags": [ "loyalty", "vip" ], "orders": [ { "orderId": 9007, "amount": 799.0, "shipping": 99 } ] }
8	CRM	crm_2026_02_16.json	2026-02-16 14:01:58.437367	{ "customerId": 129, "name": "Arun", "address": { "city": "Kolkata", "zip": "700001", "tags": [ "new", "promo" ] }
9	CRM	crm_2026_02_16.json	2026-02-16 14:01:58.437367	{ "customerId": 130, "name": "Priya", "address": { "city": "Ahmedabad", "zip": "380001", "tags": [ "newsletter", "promo" ], "orders": [ { "orderId": 9008, "amount": 600.0, "shipping": 99 } ] }
10	CRM	crm_bad_trailing_comma.json	2026-02-16 14:01:58.437367	{ "customerId": 131, "name": "Kiran", "address": { "city": "Jaipur", "zip": "302001", "tags": [ "vip" ], "orders": [ { "orderId": 9011, "amount": 600.0, "shipping": 99 } ] }
11	CRM	crm_2026_02_16.json	2026-02-16 14:01:58.437367	{ "customerId": 132, "name": "Nitin", "address": { "city": "Lucknow", "zip": "226001", "geo": { "lat": 26.8467, "lon": 80.9462 } }, "tags": [ "vip" ], "orders": [ { "orderId": 9012, "amount": 226001.0, "geo": { "lat": 26.8467, "lon": 80.9462 } } ] }
12	CRM	crm_2026_02_16.json	2026-02-16 14:01:58.437367	{ "customerId": 133, "name": "Divya", "address": { "city": "Bengaluru", "zip": "560103", "geo": { "lat": 12.9352, "lon": 77.6245 } }, "tags": [ "vip" ], "orders": [ { "orderId": 9013, "amount": 560103.0, "geo": { "lat": 12.9352, "lon": 77.6245 } } ] }

The screenshot shows the Object Explorer pane on the left with a tree view of databases and tables. The SQLQuery3 window on the right displays a query to select top 50 rows from the dbo.Json\_Bronze table, filtering for rows where ISJSON(RecordDetails) is not equal to 1 or RecordDetails is null. The results are presented in a grid with columns: RowId, SourceFile, and RecordDetails.

RowId	SourceFile	RecordDetails
1	crm_2026_02_16_nested.json	{ "customerId": 123, "name": "Rakesh", "address": { "city": "Bengaluru", "zip": "560001", "geo": { "lat": 12.9716, "lon": 77.5946 } }, "tags": [ "vip" ] }
2	crm_bad.json	{ "bad-json": true }
3	crm_2026_02_16.json	{ "customerId": 124, "name": "Ananya", "address": { "city": "Mumbai", "zip": "400001", "geo": { "lat": 19.076, "lon": 72.8777 } }, "tags": [ "new" ] }
4	crm_2026_02_16.json	{ "customerId": 125, "name": "Vikram", "address": { "city": "Delhi", "zip": "110001", "tags": [ "vip" ], "orders": [ { "orderId": 9004, "amount": 5400 } ] }
5	crm_2026_02_16.json	{ "customerId": 126, "name": "Neha", "address": { "city": "Pune", "zip": "411001", "tags": [ "newsletter" ], "orders": [ { "orderId": 9005, "amount": 0.0, "shipping": 99 } ] }
6	crm_2026_02_16.json	{ "customerId": 127, "name": "Suresh", "address": { "city": "Chennai", "zip": "600001", "tags": [ "orderId": 9006, "amount": 0.0, "shipping": 99 } ] }
7	crm_2026_02_16.json	{ "customerId": 128, "name": "Meera", "address": { "city": "Hyderabad", "tags": [ "loyalty", "vip" ], "orders": [ { "orderId": 9007, "amount": 799.0, "shipping": 99 } ] }
8	crm_2026_02_16.json	{ "customerId": 129, "name": "Arun", "address": { "city": "Kolkata", "zip": "700001", "tags": [ "new", "promo" ] } }
9	crm_2026_02_16.json	{ "customerId": 130, "name": "Priya", "address": { "city": "Ahmedabad", "zip": "380001", "tags": [ "newsletter", "promo" ], "orders": [ { "orderId": 9008, "amount": 600.0, "shipping": 99 } ] } }
10	crm_bad_trailing_comma.json	{ "customerId": 131, "name": "Kiran", "address": { "city": "Jaipur", "zip": "302001", "tags": [ "vip" ], "orders": [ { "orderId": 9011, "amount": 600.0, "shipping": 99 } ] } }
11	crm_2026_02_16.json	{ "customerId": 132, "name": "Nitin", "address": { "city": "Lucknow", "zip": "226001", "geo": { "lat": 26.8467, "lon": 80.9462 } }, "tags": [ "vip" ], "orders": [ { "orderId": 9012, "amount": 226001.0, "geo": { "lat": 26.8467, "lon": 80.9462 } } ] }
12	crm_2026_02_16.json	{ "customerId": 133, "name": "Divya", "address": { "city": "Bengaluru", "zip": "560103", "geo": { "lat": 12.9352, "lon": 77.6245 } }, "tags": [ "vip" ], "orders": [ { "orderId": 9013, "amount": 560103.0, "geo": { "lat": 12.9352, "lon": 77.6245 } } ] }

- Bad JSON

The screenshot shows the Object Explorer pane on the left with a tree view of databases and tables. The SQLQuery3 window on the right displays a query to select top 50 rows from the dbo.Json\_Bronze table, filtering for rows where ISJSON(RecordDetails) is not equal to 1 or RecordDetails is null. The results are presented in a grid with columns: RowId, SourceFile, and RecordDetails.

RowId	SourceFile	RecordDetails
1	crm_bad.json	{ "bad-json": true }
2	crm_bad_trailing_comma.json	{ "customerId": 131, "name": "Kiran", "address": { "city": "Jaipur", "zip": "302001", "tags": [ "vip" ], "orders": [ { "orderId": 9011, "amount": 600.0, "shipping": 99 } ] } }

## • JSON Validation

The screenshot shows two windows from the SQL Server Management Studio (SSMS) interface.

**Object Explorer:** Shows the database structure for "My workspace (SQL Server 12.0.2000.8 - rakshar@microsoft.com)". It includes several databases like Warehouses and SQL Analytics Endpoints, Dataflows, and demo databases. Under the "dbo" schema, there are tables for JsonNestedArrayStage, JsonParquetStage, Json\_Bronze, Json\_Quarantine, and Json\_Silver, along with views.

**SQL Query Editor:** A query window titled "SQLQuery3.s....com (178)\*". The query uses JSON functions to validate rows in the "RecordDetails" column of the "Json\_Bronze" table. It checks for the existence of specific keys: ".custId", ".name", ".address.city", and ".orders". The results show 10 rows, all of which pass validation (HasCustomerId = 1, HasName = 1, HasCity = 1, HasOrders = 1).

**Object Explorer (Bottom):** Similar to the top one, showing the same database structure. The "Json\_Quarantine" table is highlighted with a red box in the "Tables (filtered)" section under the "dbo" schema.

**SQL Query Editor (Bottom):** A query window titled "SQLQuery3.s....com (178)\*". The query selects all columns from the "Json\_Quarantine" table. The results show two rows where the "ValidationNote" column indicates "Invalid JSON" for both entries due to bad JSON syntax.

## • Loading and Validation in Silver Layer

The screenshot shows two windows from the SSMS interface.

**Object Explorer:** Shows the database structure for "My workspace (SQL Server 12.0.2000.8 - rakshar@microsoft.com)". It includes the same set of databases and tables as the previous screenshot, with the "Json\_Quarantine" table highlighted.

**SQL Query Editor:** A query window titled "SQLQuery3.s....com (178)\*". The query inserts invalid JSON data into the "Json\_Silver" table. It then performs a self-join to compare the "RowId", "SourceSystem", "SourceFile", and "IngestedAt" columns between the "Json\_Quarantine" and "Json\_Silver" tables. The results show two rows from the "Json\_Quarantine" table being inserted into the "Json\_Silver" table, with validation notes indicating they are invalid JSON.

**Object Explorer (Bottom):** Shows the database structure for "My workspace (SQL Server 12.0.2000.8 - rakshar@microsoft.com)". The "Json\_Silver" table is highlighted with a red box in the "Tables (filtered)" section under the "dbo" schema.

**SQL Query Editor (Bottom):** A query window titled "SQLQuery3.s....com (178)\*". The query selects all columns from the "Json\_Silver" table. The results show 12 rows of valid customer data, with the "ValidationNote" column indicating "OK" for all entries.

- Gold Layer with Flattened Data

Object Explorer

SQL Server 12.0.2000.8 - rakshar@microsoft.com

Customer\_Gold

Tables

CustomerAddress\_Gold  
Customer\_Gold  
OrderCoupon\_Gold  
OrderItemAttr\_Gold  
OrderItem\_Gold  
OrderMeta\_Gold  
OrderShipping\_Gold  
Order\_Gold

Results

```

1   SELECT * FROM Customer_Gold
2   SELECT * FROM Order_Gold
3   SELECT * FROM OrderItem_Gold

```

custId	custName	city
128	Meera	Hyderabad
125	Vikram	Delhi
132	Nitin	Lucknow
124	Ananya	Mumbai
126	Neha	Pune
123	Rakesh	Bengaluru
127	Suresh	Chennai
129	Arjun	Kolkata
130	Priya	Ahmedabad
133	Diya	Bengaluru

custId	orderId	amount
123	9001	2500.5
123	9002	999
124	9003	120.75
125	9004	5400
127	9005	0
128	9006	799.99
128	9007	199.5
129	9008	350
130	9009	1499
130	9010	250
132	9012	999.99
133	9013	3250.25

custId	orderId	sku	qty
123	9001	A1	2
123	9001	B9	1
123	9002	C3	5
124	9003	D4	1
125	9004	X1	1
125	9004	X2	2
125	9004	X3	3
127	9005	FREE1	1
128	9006	M1	2
128	9007	M2	1
128	9007	M3	1
130	9009	P1	1

Object Explorer

SQL Server 12.0.2000.8 - rakshar@microsoft.com

Customer\_Gold

Tables

CustomerAddress\_Gold  
Customer\_Gold  
OrderCoupon\_Gold  
OrderItemAttr\_Gold  
OrderItem\_Gold  
OrderMeta\_Gold  
OrderShipping\_Gold  
Order\_Gold

Results

```

6
7
8   attrs": {"color": "yellow", "size": "M"}, {"sku": "M3", "qty": 10}
9

```

custId	city	zip	geo_lat	geo_lon
129	Kolkata	700001	NULL	NULL
130	Ahmedabad	380001	NULL	NULL
124	Mumbai	400001	19.076	72.8777
127	Chennai	600001	NULL	NULL
133	Bengaluru	560103	12.935	77.6245
125	Delhi	110001	NULL	NULL
128	Hyderabad	NULL	NULL	NULL
132	Lucknow	226001	26.8467	80.9462

custId	orderId	method	eta
123	9001	EXPRESS	2026-02-20
123	9002	STANDARD	2026-02-25
124	9003	STANDARD	2026-02-21
125	9004	EXPRESS	2026-02-19
127	9005	STANDARD	2026-02-22
128	9006	STANDARD	2026-02-23
128	9007	STANDARD	2026-02-23
129	9008	STANDARD	2026-02-24

custId	orderId	coupon
123	9001	WELCOME
124	9003	NEW5
128	9006	LOYAL20
129	9008	PROMO
130	9009	FEB10
130	9010	WELCOME
132	9012	NEW5
133	9013	LOYAL20

custId	orderId	meta_key	meta_val
123	9001	channel	web
123	9001	campaign	FEB-2026
123	9002	channel	store
124	9003	channel	app
125	9004	channel	web
125	9004	campaign	VIP
127	9005	channel	store
128	9006	channel	web

custId	orderId	sku	color	size
123	9001	A1	black	XL
123	9001	B9	white	S
123	9002	C3	black	S
124	9003	D4	white	S
125	9004	X1	black	S
125	9004	X2	white	S
125	9004	X3	black	S
127	9005	FREE1	black	S
128	9006	M1	black	XL
128	9007	M2	white	S
128	9007	M3	black	S
130	9009	P1	white	S

- Views for Reporting

Object Explorer

Connect ▾ X

My workspace (SQL Server 12.0.2000.8 - rakshar@microsoft.com)

- Warehouses and SQL Analytics Endpoints
- AzFabricSQL
- AzureEmissionInsightDWH
- DataflowsStagingLakehouse
- DataflowsStagingWarehouse
- DemoWarehouseLearn
- healthcare1\_msft\_admin
- healthcare1\_msft\_bronze
- healthcare1\_msft\_silver
- IoTEvHubLsBronze
- OpenMirroredDatabase
- OyoDWH
- rsdemo
- Schemas
  - INFORMATION\_SCHEMA
  - dbo
    - Tables
    - Views
    - vw\_Gold\_Customer360**
    - vw\_Gold\_OrderItem\_Fact**
    - vw\_Gold\_SalesByCity**
    - Functions
    - Stored Procedures
  - queryinsights
  - sys
- Security
- rsdemolh
- Schemas
  - INFORMATION\_SCHEMA
  - dbo
    - Tables
    - Views
    - Functions
    - Stored Procedures
  - queryinsights
  - sys

```
SQLQuery3.sql...com (178)* X
1   SELECT * FROM dbo.vw_Gold_OrderItem_Fact
2   SELECT * FROM vw_Gold_Customer360
3   SELECT * FROM dbo.vw_Gold_SalesByCity
```

100 % X 1 A 0 ↑ ↓

Results Messages

custId	custName	city	orderId	orderAmount	sku	qty
1	128	Meera	Hyderabad	9007	199.5	M3
2	125	Vikram	Delhi	9004	5400	X3
3	132	Nitin	Lucknow	9012	999.99	N1
4	124	Ananya	Mumbai	9003	120.75	D4
5	123	Rakesh	Bengaluru	9002	999	C3
6	127	Suresh	Chennai	9005	0	FREE1
7	130	Priya	Ahmedabad	9010	250	P3
8	133	Divya	Bengaluru	9013	3250.25	DV2
9	128	Meera	Hyderabad	9007	199.5	M2
10	128	Meera	Hyderabad	9006	799.99	M1
11	125	Vikram	Delhi	9004	5400	X2
12	125	Vikram	Delhi	9004	5400	X1

custId	custName	city	orderCount	totalSpend	avgOrderValue
1	128	Meera	Hyderabad	2	999.49
2	125	Vikram	Delhi	1	5400
3	132	Nitin	Lucknow	1	999.99
4	124	Ananya	Mumbai	1	120.75
5	126	Neha	Pune	0	NULL
6	123	Rakesh	Bengaluru	2	3499.5
7	127	Suresh	Chennai	1	0
8	129	Arjun	Kolkata	1	350
9	130	Priya	Ahmedabad	2	1749
10	133	Divya	Bengaluru	1	3250.25

city	customerCount	orderCount	totalRevenue	avgOrderValue
1	Chennai	1	1	0
2	Hyderabad	1	2	999.49
3	Ahmedabad	1	2	1749
4	Pune	1	0	NULL
5	Bengaluru	2	3	6749.75
6	Mumbai	1	1	120.75
7	Kolkata	1	1	350
8	Delhi	1	1	5400
9	Lucknow	1	1	999.99