ETL Course:  
  
Data Classification and Categories

|  |  |
| --- | --- |
| Information | Data Serving a certain context and purpose |
| Transactional Data | Describes an Event or Transaction |
| Master Data | Data that is consistently shared across an entire organization |
| Reference Data | Structures and standardizes |
| MetaData |  |
|  |  |

**Data Quality Dimension:** a. Consistency. B. Completeness c. Uniqueness

d. Accuracy e. validity f. Timeliness

Completeness: null value

Consistency:

Uniqueness:

Data Quality Repository:

Data Quality Component:   
Data Profiling🡪Data Quality Analysis: validating->Data Cleansing🡪 Matching, Eliminating errors, Duplicates, Inconsistencies ->Data Quality Monitoring: Reporting and Monitoring

Data Profiling: 1. Metadata Discovery (Data Types, length, precision, nullable, Keywords, Tag, Relationship (PM Key, Foreign Key,) 2. Descriptive Methods: count, Mean, Max, Min, Distribution. 3. Validation Levels: Sch ema, Entity, Attribute

Data Quality Tools: Gartner Magic Quadrant for Data Quality Toos(Gartner 2017)

DQ Tools: Most recognized tool in the Market: Informatica Data Quality, Talend Open Studio for Data Quality, IBM InfoSphere, Quality Stage and Information Analyzer, SAS Data Quality, Oracle, Others

1. Verify Data Completeness: No Null Value
2. Verify Data Uniqueness: Each Customer has unique ID to identify each row customer when customers have same name or address or zip code.

Customer will be assigned to Regional Manager, so Manager ID should be unique. Order: each order should be unique that can be retrieved by Order ID, Product: Each produ-+ct should be identified by Unique ID

**Data Quality Views VS. Tables:**

**--**Regression Testing and automated ETL testing requires an Ad-Hoc view of data.

--A dynamic approach to ETL testing.

--Views do not require manual INSERT, DELETE, UPDATE operations on original tables but it returns data per query execution.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Source System | Staging Area | Data Warehouse | DATA Mart | End User A |
| Source A |  | No one has access | -🡪 Ready to use for End User | End User **B** |
| Source B |  |  |  | End User C |
| Source C |  |  |  |  |

**We need to create Ad-hoc View of Data to verify following:**

**Build the Views to verify Data Quality:**

**--Data Profiling.**

**--Data Quality Uniqueness.**

**--Data Quality Completeness.**

**--Data Quality Validity.**

**--Data Quality Consistency.**

**--Data Quality Integrity.**

**Data Quality Completeness:** It identify if a primary key or Foreign Key or Composite Key is Null in a table.

Verify OrderID or OrderDate is Null value or not.

Select \* From VW\_Order+-//\*

1. **a. To verify the Completeness, without Null value-DQ view – Completeness**

CREATE TABLE public."STG\_Orders"

(

"RowID" numeric,

"OrderID" numeric,

"OrderDate" date,

"OrderPriority" character varying,

"OrderQuantity" numeric,

"Sales" numeric,

"Discount" numeric,

"ShipMode" character varying,

"Profit" numeric,

"UnitPrice" numeric,

"ShippingCost" numeric,

"CustomerName" character varying,

"ProductName" character varying,

"ShipDate" date

)

1. **b. To confirm if RowID is Null value and then count them.**

**---**'STG\_Orders' AS table

create or replace view dq\_completeness as

select

'STG\_Orders' AS table,

**'RowID'** AS attribute,

sum(case WHEN "RowID" IS NULL THEN 1 ELSE 0 END) AS count\_null\_records

from public."STG\_Orders"

union all

select

'STG\_Orders',

**'OrderID'** As attribute,

sum(case WHEN "OrderID" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Orders"

union all

select

'STG\_Orders',

**'OrderDate'**,

sum(case WHEN "OrderDate" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Orders"

union all

select

'STG\_Orders',

**'OrderPriority'**,

sum(case WHEN "OrderPriority" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Orders"

union all

select

'STG\_Orders',

**'OrderQuantity'**,

sum(case WHEN "OrderQuantity" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Orders"

union all

select

'STG\_Orders',

**'Sales'**,

sum(case WHEN "Sales" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Orders"

union all

select

'STG\_Orders',

**'Discount'**,

sum(case WHEN "Discount" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Orders"

union all

select

'STG\_Orders',

**'ShipMode'**,

sum(case WHEN "ShipMode" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Orders"

union all

select

'STG\_Orders',

**'Profit'**,

sum(case WHEN "Profit" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Orders"

union all

select

'STG\_Orders',

**'UnitPrice'**,

sum(case WHEN "UnitPrice" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Orders"

union all

select

'STG\_Orders',

**'ShippingCost'**,

sum(case WHEN "ShippingCost" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Orders"

union all

select

'STG\_Orders',

**'CustomerName',**

sum(case WHEN "CustomerName" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Orders"

union all

select

'STG\_Orders',

**'ProductName'**,

sum(case WHEN "ProductName" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Orders"

union all

select

'STG\_Orders',

**'ShipDate'**,

sum(case WHEN "ShipDate" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Orders"

union all

**--'STG\_Returns' As Table**

select

'STG\_Returns',

**'OrderID'**,

sum(case WHEN "OrderID" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Returns"

union all

select

'STG\_Returns',

'Status',

sum(case WHEN "Status" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Returns"

union all

select

'STG\_Products',

'ProductName',

sum(case WHEN "ProductName" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Products"

union all

select

'STG\_Products',

'ProductCategory',

sum(case WHEN "ProductCategory" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Products"

union all

select

'STG\_Products',

**'ProductSubCategory'**,

sum(case WHEN "ProductSubCategory" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Products"

union all

select

'STG\_Products',

'ProductContainer',

sum(case WHEN "ProductContainer" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Products"

union all

select

'STG\_Products',

'ProductBaseMargin',

sum(case WHEN "ProductBaseMargin" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Products"

union all

select

'STG\_RegionManagers',

'Region',

sum(case WHEN "Region" IS NULL THEN 1 ELSE 0 END)

from public."STG\_RegionManagers"

union all

select

'STG\_RegionManagers',

'Manager',

sum(case WHEN "Manager" IS NULL THEN 1 ELSE 0 END)

from public."STG\_RegionManagers"

union all

select

'STG\_Customers',

'CustomerName',

sum(case WHEN "CustomerName" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Customers"

union all

select

'STG\_Customers',

'Province',

sum(case WHEN "Province" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Customers"

union all

select

'STG\_Customers',

'Region',

sum(case WHEN "Region" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Customers"

union all

select

'STG\_Customers',

'CustomerSegment',

sum(case WHEN "CustomerSegment" IS NULL THEN 1 ELSE 0 END)

from public."STG\_Customers"

union all

select

'DWH\_Customers',

'CustomerName',

sum(case WHEN "CustomerName" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Customers"

union all

select

'DWH\_Customers',

'Province',

sum(case WHEN "Province" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Customers"

union all

select

'DWH\_Customers',

'Region',

sum(case WHEN "Region" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Customers"

union all

select

'DWH\_Customers',

'CustomerSegment',

sum(case WHEN "CustomerSegment" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Customers"

union all

select

'DWH\_Orders',

'RowID',

sum(case WHEN "RowID" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Orders"

union all

select

'DWH\_Orders',

'OrderID',

sum(case WHEN "OrderID" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Orders"

union all

select

'DWH\_Orders',

'OrderDate',

sum(case WHEN "OrderDate" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Orders"

union all

select

'DWH\_Orders',

'OrderPriority',

sum(case WHEN "OrderPriority" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Orders"

union all

select

'DWH\_Orders',

'OrderQuantity',

sum(case WHEN "OrderQuantity" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Orders"

union all

select

'DWH\_Orders',

'Sales',

sum(case WHEN "Sales" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Orders"

union all

select

'DWH\_Orders',

'Discount',

sum(case WHEN "Discount" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Orders"

union all

select

'DWH\_Orders',

'ShipMode',

sum(case WHEN "ShipMode" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Orders"

union all

select

'DWH\_Orders',

'Profit',

sum(case WHEN "Profit" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Orders"

union all

select

'DWH\_Orders',

'UnitPrice',

sum(case WHEN "UnitPrice" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Orders"

union all

select

'DWH\_Orders',

'ShippingCost',

sum(case WHEN "ShippingCost" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Orders"

union all

select

'DWH\_Orders',

'CustomerName',

sum(case WHEN "CustomerName" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Orders"

union all

select

'DWH\_Orders',

'ProductName',

sum(case WHEN "ProductName" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Orders"

union all

select

'DWH\_Orders',

'ShipDate',

sum(case WHEN "ShipDate" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Orders"

union all

select

'DWH\_Products',

'ProductName',

sum(case WHEN "ProductName" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Products"

union all

select

'DWH\_Products',

'ProductCategory',

sum(case WHEN "ProductCategory" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Products"

union all

select

'DWH\_Products',

'ProductSubCategory',

sum(case WHEN "ProductSubCategory" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Products"

union all

select

'DWH\_Products',

'ProductContainer',

sum(case WHEN "ProductContainer" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Products"

union all

select

'DWH\_Products',

'ProductBaseMargin',

sum(case WHEN "ProductBaseMargin" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Products"

union all

select

'DWH\_RegionManagers',

'Region',

sum(case WHEN "Region" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_RegionManagers"

union all

select

'DWH\_RegionManagers',

'Manager',

sum(case WHEN "Manager" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_RegionManagers"

union all

select

'DWH\_Returns',

'OrderID',

sum(case WHEN "OrderID" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Returns"

union all

select

'DWH\_Returns',

'Status',

sum(case WHEN "Status" IS NULL THEN 1 ELSE 0 END)

from public."DWH\_Returns"

union all

select

'DM\_CustomerRegionalSales',

'CustomerName',

sum(case WHEN "CustomerName" IS NULL THEN 1 ELSE 0 END)

from public."DM\_CustomerRegionalSales"

union all

select

'DM\_CustomerRegionalSales',

'Province',

sum(case WHEN "Province" IS NULL THEN 1 ELSE 0 END)

from public."DM\_CustomerRegionalSales"

union all

select

'DM\_CustomerRegionalSales',

'Region',

sum(case WHEN "Region" IS NULL THEN 1 ELSE 0 END)

from public."DM\_CustomerRegionalSales"

union all

select

'DM\_CustomerRegionalSales',

'CustomerSegment',

sum(case WHEN "CustomerSegment" IS NULL THEN 1 ELSE 0 END)

from public."DM\_CustomerRegionalSales"

union all

select

'DM\_CustomerRegionalSales',

'OrderID',

sum(case WHEN "OrderID" IS NULL THEN 1 ELSE 0 END)

from public."DM\_CustomerRegionalSales"

union all

select

'DM\_CustomerRegionalSales',

'OrderDate',

sum(case WHEN "OrderDate" IS NULL THEN 1 ELSE 0 END)

from public."DM\_CustomerRegionalSales"

union all

select

'DM\_CustomerRegionalSales',

'Sales',

sum(case WHEN "Sales" IS NULL THEN 1 ELSE 0 END)

from public."DM\_CustomerRegionalSales"

;

Select Count(\*) From public.”STG\_Order”

Select distinct “Order\_ID”, Count (\*) From public.”STG\_Order”

Group By “Order\_ID”

Having Count(\*)>1

**------- 2. a. Data Uniqueness.** For instance “Order\_ID” can’t can be duplicated since it is a Primary Key. It has to be unique.  
  
--DQ view - Uniqueness

1. create or replace view dq\_uniqueness as
2. select
3. 'STG\_Orders' AS table,
4. 'OrderID' AS unique\_constraint,
5. count("OrderID") AS count\_records,
6. count(distinct "OrderID") AS count\_distinct\_records,
7. count("OrderID") - count(distinct "OrderID") AS count\_duplicates
8. from public."STG\_Orders"
9. union all
10. select
11. 'STG\_Returns',
12. 'OrderID',
13. count("OrderID"),
14. count(distinct "OrderID"),
15. count("OrderID") - count(distinct "OrderID")
16. from public."STG\_Returns"
17. union all
18. select
19. 'STG\_Products',
20. 'ProductName',
21. count("ProductName"),
22. count(distinct "ProductName"),
23. count("ProductName") - count(distinct "ProductName")
24. from public."STG\_Products"
25. union all
26. select
27. 'STG\_RegionManagers',
28. 'Manager',
29. count("Manager"),
30. count(distinct "Manager"),
31. count("Manager") - count(distinct "Manager")
32. from public."STG\_RegionManagers"
33. union all
34. select
35. 'STG\_Customers',
36. 'CustomerName',
37. count("CustomerName"),
38. count(distinct "CustomerName"),
39. count("CustomerName") - count(distinct "CustomerName")
40. from public."STG\_Customers"
41. union all
42. select
43. 'DWH\_Customers',
44. 'CustomerName',
45. count("CustomerName"),
46. count(distinct "CustomerName"),
47. count("CustomerName") - count(distinct "CustomerName")
48. from public."DWH\_Customers"
49. union all
50. select
51. 'DWH\_Orders',
52. 'OrderID',
53. count("OrderID"),
54. count(distinct "OrderID"),
55. count("OrderID") - count(distinct "OrderID")
56. from public."DWH\_Orders"
57. union all
58. select
59. 'DWH\_Products',
60. 'ProductName',
61. count("ProductName"),
62. count(distinct "ProductName"),
63. count("ProductName") - count(distinct "ProductName")
64. from public."DWH\_Products"
65. union all
66. select
67. 'DWH\_RegionManagers',
68. 'Manager',
69. count("Manager"),
70. count(distinct "Manager"),
71. count("Manager") - count(distinct "Manager")
72. from public."DWH\_RegionManagers"
73. union all
74. select
75. 'DWH\_Returns',
76. 'OrderID',
77. count("OrderID"),
78. count(distinct "OrderID"),
79. count("OrderID") - count(distinct "OrderID")
80. from public."DWH\_Returns"
81. union all
82. select
83. 'DM\_CustomerRegionalSales',
84. 'OrderID',
85. count("OrderID"),
86. count(distinct "OrderID"),
87. count("OrderID") - count(distinct "OrderID")
88. from public."DM\_CustomerRegionalSales"
89. ;

**-----------3. a. Data Validity:**

Select \* From public.”STG\_Orders  
Where “Order\_Quantity” <=0

Select \* From public.”STG\_Orders  
Where “OrderID” <=0

Select \* From public.”STG\_Orders”

Where “Sales” <=0

Select \* From public.”STG\_Orders”

Where “ShipDate” <=”OrderDate”

--DQ view - Validity

create or replace view dq\_validity as

select

'STG\_Orders' AS table,

'RowID' AS validity\_constraint,

'> 1' AS validity\_rule,

sum(case WHEN "RowID" > 0 THEN 1 ELSE 0 END) AS count\_valid\_records,

sum(case WHEN "RowID" > 0 THEN 0 ELSE 1 END) AS count\_invalid\_records

from public."STG\_Orders"

union all

select

'STG\_Orders',

'OrderID',

'> 1',

sum(case WHEN "OrderID" > 0 THEN 1 ELSE 0 END),

sum(case WHEN "OrderID" > 0 THEN 0 ELSE 1 END)

from public."STG\_Orders"

union all

select

'STG\_Orders',

'OrderDate',

'> 2000-01-01',

sum(case WHEN "OrderDate" > to\_date('2000-01-01','YYYY-MM-DD') THEN 1 ELSE 0 END),

sum(case WHEN "OrderDate" > to\_date('2000-01-01','YYYY-MM-DD') THEN 0 ELSE 1 END)

from public."STG\_Orders"

union all

select

'STG\_Orders',

'OrderQuantity',

'> 0',

sum(case WHEN "OrderQuantity" > 0 THEN 1 ELSE 0 END),

sum(case WHEN "OrderQuantity" > 0 THEN 0 ELSE 1 END)

from public."STG\_Orders"

union all

select

'STG\_Orders',

'Sales',

'> 0',

sum(case WHEN "Sales" > 0 THEN 1 ELSE 0 END),

sum(case WHEN "Sales" > 0 THEN 0 ELSE 1 END)

from public."STG\_Orders"

union all

select

'STG\_Orders',

'Discount',

'>= 0',

sum(case WHEN "Discount" >= 0 THEN 1 ELSE 0 END),

sum(case WHEN "Discount" >= 0 THEN 0 ELSE 1 END)

from public."STG\_Orders"

union all

select

'STG\_Orders',

'Discount',

'< 1',

sum(case WHEN "Discount" < 1 THEN 1 ELSE 0 END),

sum(case WHEN "Discount" < 1 THEN 0 ELSE 1 END)

from public."STG\_Orders"

union all

select

'STG\_Orders',

'UnitPrice',

'> 0',

sum(case WHEN "UnitPrice" > 0 THEN 1 ELSE 0 END),

sum(case WHEN "UnitPrice" > 0 THEN 0 ELSE 1 END)

from public."STG\_Orders"

union all

select

'STG\_Orders',

'ShippingCost',

'> 0',

sum(case WHEN "ShippingCost" > 0 THEN 1 ELSE 0 END),

sum(case WHEN "ShippingCost" > 0 THEN 0 ELSE 1 END)

from public."STG\_Orders"

union all

select

'STG\_Orders',

'ShipDate',

'> $OrderDate',

sum(case WHEN "ShipDate" >= "OrderDate" THEN 1 ELSE 0 END),

sum(case WHEN "ShipDate" >= "OrderDate" THEN 0 ELSE 1 END)

from public."STG\_Orders"

;

**-----4. Data Quality Consistency:**

Data Category or Data Quality that retain same degree of consistent across the database.

Select distinct(“OrderPriority”), count (\*) From public.”STG\_Orders”

Group By “OrderPriority”;

Select distinct(“ShipMode”), count (\*) From public.”STG\_Orders”

Group By “ShipMode”;

Result: Critical----🡪1608

High---------🡪1768

Low---------🡪1720  
 Medium---🡪1631

Not Specified->1672

--DQ view - Consistency

create or replace view dq\_consistency as

select

distinct

"OrderPriority" AS distinct\_values,

count(\*) as number\_records,

'STG\_Orders' AS table,

'OrderPriority' AS column

from public."STG\_Orders"

group by "OrderPriority"

union all

select

distinct

"ShipMode",

count(\*),

'STG\_Orders',

'ShipMode'

from public."STG\_Orders"

group by "ShipMode"

union all

select

distinct

"Status",

count(\*),

'STG\_Returns',

'Status'

from public."STG\_Returns"

group by "Status"

union all

select

distinct

"Province",

count(\*),

'STG\_Customers',

'Province'

from public."STG\_Customers"

group by "Province"

union all

select

distinct

"Region",

count(\*),

'STG\_Customers',

'Region'

from public."STG\_Customers"

group by "Region"

union all

select

distinct

"CustomerSegment",

count(\*),

'STG\_Customers',

'CustomerSegment'

from public."STG\_Customers"

group by "CustomerSegment"

union all

select

distinct

"Region",

count(\*),

'STG\_RegionManagers',

'Region'

from public."STG\_RegionManagers"

group by "Region"

union all

select

distinct

"ProductCategory",

count(\*),

'STG\_Products',

'ProductCategory'

from public."STG\_Products"

group by "ProductCategory"

union all

select

distinct

"ProductSubCategory",

count(\*),

'STG\_Products',

'ProductSubCategory'

from public."STG\_Products"

group by "ProductSubCategory"

**------5--Data Integrity Validation:**

Verify the Data from Datamart and target Data if any discrepancy exist between them. Verifying Source and Target data if no difference.  
  
Select “OrderID” From public.”DM\_CustomerRegionalSales”

Except ( Select public.“OrderID” From “STG\_Orders” a  
 INNER JOIN public. DM\_CustomerRegionalSales b on a.“OrderID”=b.”Ord

--DQ view - Integrity (data lineage)

create or replace view dq\_integrity as

select

'STG\_Orders' AS table,

'OrderID' AS column,

(SELECT "OrderID"::text FROM

public."DWH\_Orders"

EXCEPT --equivalent to minus in Oracle

(SELECT a."OrderID"::text

FROM public."STG\_Orders" a

INNER JOIN public."DWH\_Orders" b

on a."OrderID" = b."OrderID"

)

) AS integrity\_mismatch

union all

select

'STG\_Orders',

'OrderDate',

(SELECT "OrderDate"::text FROM

public."DWH\_Orders"

EXCEPT --equivalent to minus in Oracle

(SELECT a."OrderDate"::text

FROM public."STG\_Orders" a

INNER JOIN public."DWH\_Orders" b

on a."OrderID" = b."OrderID"

)

)

union all

select

'STG\_Orders',

'OrderPriority',

(SELECT "OrderPriority"::text FROM

public."DWH\_Orders"

EXCEPT --equivalent to minus in Oracle

(SELECT a."OrderPriority"::text

FROM public."STG\_Orders" a

INNER JOIN public."DWH\_Orders" b

on a."OrderID" = b."OrderID"

)

)

union all

select

'STG\_Orders',

'OrderQuantity',

(SELECT "OrderQuantity"::text FROM

public."DWH\_Orders"

EXCEPT --equivalent to minus in Oracle

(SELECT a."OrderQuantity"::text

FROM public."STG\_Orders" a

INNER JOIN public."DWH\_Orders" b

on a."OrderID" = b."OrderID"

)

)

union all

select

'STG\_Orders',

'Sales',

(SELECT "Sales"::text FROM

public."DWH\_Orders"

EXCEPT --equivalent to minus in Oracle

(SELECT a."Sales"::text

FROM public."STG\_Orders" a

INNER JOIN public."DWH\_Orders" b

on a."OrderID" = b."OrderID"

)

)

union all

select

'STG\_Orders',

'Discount',

(SELECT "Discount"::text FROM

public."DWH\_Orders"

EXCEPT --equivalent to minus in Oracle

(SELECT a."Discount"::text

FROM public."STG\_Orders" a

INNER JOIN public."DWH\_Orders" b

on a."OrderID" = b."OrderID"

)

)

union all

select

'STG\_Orders',

'ShipMode',

(SELECT "ShipMode"::text FROM

public."DWH\_Orders"

EXCEPT --equivalent to minus in Oracle

(SELECT a."ShipMode"::text

FROM public."STG\_Orders" a

INNER JOIN public."DWH\_Orders" b

on a."OrderID" = b."OrderID"

)

)

union all

select

'STG\_Orders',

'Profit',

(SELECT "Profit"::text FROM

public."DWH\_Orders"

EXCEPT --equivalent to minus in Oracle

(SELECT a."Profit"::text

FROM public."STG\_Orders" a

INNER JOIN public."DWH\_Orders" b

on a."OrderID" = b."OrderID"

)

)

union all

select

'STG\_Orders',

'UnitPrice',

(SELECT "UnitPrice"::text FROM

public."DWH\_Orders"

EXCEPT --equivalent to minus in Oracle

(SELECT a."UnitPrice"::text

FROM public."STG\_Orders" a

INNER JOIN public."DWH\_Orders" b

on a."OrderID" = b."OrderID"

)

)

union all

select

'STG\_Orders',

'ShippingCost',

(SELECT "ShippingCost"::text FROM

public."DWH\_Orders"

EXCEPT --equivalent to minus in Oracle

(SELECT a."ShippingCost"::text

FROM public."STG\_Orders" a

INNER JOIN public."DWH\_Orders" b

on a."OrderID" = b."OrderID"

)

)

union all

select

'STG\_Orders',

'CustomerName',

(SELECT "CustomerName"::text FROM

public."DWH\_Orders"

EXCEPT --equivalent to minus in Oracle

(SELECT a."CustomerName"::text

FROM public."STG\_Orders" a

INNER JOIN public."DWH\_Orders" b

on a."OrderID" = b."OrderID"

)

)

union all

select

'STG\_Orders',

'ProductName',

(SELECT "ProductName"::text FROM

public."DWH\_Orders"

EXCEPT --equivalent to minus in Oracle

(SELECT a."ProductName"::text

FROM public."STG\_Orders" a

INNER JOIN public."DWH\_Orders" b

on a."OrderID" = b."OrderID"

)

)

union all

select

'STG\_Orders',

'ShipDate',

(SELECT "ShipDate"::text FROM

public."DWH\_Orders"

EXCEPT --equivalent to minus in Oracle

(SELECT a."ShipDate"::text

FROM public."STG\_Orders" a

INNER JOIN public."DWH\_Orders" b

on a."OrderID" = b."OrderID"

)

)

;

-----------7. Data Profiling

Select \* From pg\_stat\_all\_tables;

SELECT \* FROM information\_schema.columns

WHERE table\_name like '%STG%' or table\_name like '%DWH%' or table\_name like '%DM%'

--View - Profiling\_count\_records

create or replace view data\_profiling\_count\_records as

SELECT schemaname as schema, relname as table, n\_tup\_ins as count FROM pg\_stat\_all\_tables

WHERE relname like '%STG%' or relname like '%DWH%' or relname like '%DM%'

;

--View - Profiling\_information\_schema

create or replace view data\_profiling\_information\_schema as

SELECT \* FROM information\_schema.columns

WHERE table\_name like '%STG%' or table\_name like '%DWH%' or table\_name like '%DM%' ;

-----8. Dashboard Monitoring :