We are checking data. [Transformation]

→ 1. Table.

cust_info.

checking for duplicates.

```
Select
cust_id,
count(*)
From bronze.crm_cust_info
Group By cust_id.
Having count(*) > 1 or cust_id IS NULL
```

Gives the info of duplicates in the record.

Rank the duplicates.

```
Select
*
From (
Select
*,
ROW_NUMBER() OVER (Partition By cust_id ORDER BY cust_create_date
                   DESC) as flag_last
From bronze.crm_cust_info
~~Where cust_~~
) t where flag_last = 1
```

We ensure the record without any duplicates

The Table of strings columns.

checking for the unwanted spaces.

Remove the leading & tailing spaces.

```
TRIM (cst_firstname) as cst_firstname
                        cast_name.
```

we will Know if there any

```
Select
cst_firstname
From bronze.crm_cust_info where cst_firstname != TRIM(cst_f
```

2. In gender colum we see there are abbreviation for Female as (F) and Male as (M)., NULL as UNKNOWN. (n/a)

So create a rule to give it's full meaning. Female & Male.

```
case when UPPER(TRIM(cst_gndr)) = 'F' THEN 'Female'
     WHEN UPPER(TRIM(cst_gndr)) = 'M' THEN 'Male'
     ELSE 'n/a'
```

3. Do the same for malital status.

S - Single
M - married
NULL - n/a.

Now we are inserting all the transformed data into.

silver schema (table) silver.crm_cust_info.

```
Insert INTO silver.crm_cust_info.(
```

Remove Duplicates.
Ensure only one record per entity by identifying & retaining most relavent row.

Remove unwanted spaces.

Removes unnecessary spaces to ensure data consistency and uniformity across all records.

Data Normalization & standardization

Maps coded values to meaniful. user friendly descriptions

Handling Missing data

Fill in the blanks by adding a default values.

Product table.

1. Check for duplicates.

2. we need sub categories. from prod_key column.

Prod_Key ⇒ Replace (

Position, How many.

SUBSTRING (Prod_key, 1, 5), '-', '_') as cat_id,

Select prod_id.

Check with other table where it's a Foreign key
where (substring) NOT IN.
(Select distinct id from bronze. erp_px_cat_g1v2)

2nd Substring.

dynamic. untill last

Replace (substring (Prod_key, 7, Length ()), '-', '_') as prod_key.

where
(Select sls_prod_key from bronze. crm_sales_details)

3. In Prod_cost there are some null values so

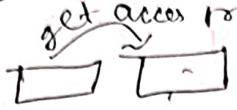
use ISNULL (prod_cost) as prod_cost

4. Check for Invalid Date Orders.

Select *
from bronze. crm_prod_info
where prd_end_dt < prd_start_dt

Lead & Lag

to access the next record.



LEAD (prd_start_dt) OVER (PARTITION BY prd_key ORDER BY Prd_start_dt)-1 AS prd_end_dt_test.

Quality checks.

1. Check for Nulls or Duplicates in pr key

```
Select prd_id.
count (*)
From ——
  Group by prd_id.
  Having count(*) > 1 or prd_id is null.
```

2. Check for unwanted spa.

Sales table.

- Unwanted spaces.

where sls_ord_num != Trim(sls_ord_num)

→ we connect this prod_key && sls_cust_id.

check Integrity of sls_prd_key {

Where sls_cust_id NOT IN (select cust_id from Silver.crm_cust_info).

Source -

Challenging here in this table is.

we have integers as date.

we have to convert them to date.

1. check for 0s and Null

Select

NULL IF (sls_order_dt, 0) sls_order_dt

From bronze.crm_sales_details

where sls_order_dt <= 0 or len

(sls_order_dt != 8 or sls_order_dt > 20500101 or <)

1 2 3 4 5 6 7 8.
2010 12 29

if len (8) < issue.

Check Outliers by validating the boundries of the data range.

---

fix it with

Case when sls_order_dt = 0 or LEN (sls_order_dt) != 8

Then NULL

Else cast(cast (sls_order_dt as varchar) as Date)

End sls_order_dt

Check for Invaid Date of Orders.

Select

*

from bronze.crm_sales_details

where sls_order_dt > sales_ship_dt or sls_order_dt >

sls_due_dt

Business Rules:

Σ Sales = Quantity * price

Neg, zero, nulls, are not allowed.

Rules

If Sales is negative, zero or Null, derive it using Quantity & price.

If price is zero or null calc using Sales & Quant

if price is -ve convert t +ve value.

```
Select Distinct
  sls_sales,
  sls_quantity,
  sls_price
  from bronze.crm_sales_details
   where sls_sales != sls_quant * sls_price
    or sls_sales IS Null or sls_quant IS Null or sls_price is
                                                  Null
    or sls_sales <=0 or sls_quant <=0 or sls_pr <=0

Case when sls_sales IS Null or sls_sales <=0 or sls_sales != 
                  sls_quant * ABS(sls_price)
      Then  sls_quant * ABS(sls_price)
End sls_sales

Case when sls_price <=0 . or sls_price IS Null
      Then    sls_sales / ISNULL (sls_quant)
End sls_price.
```

Data Modeling:

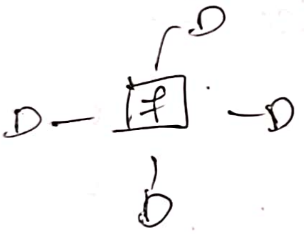
Raw data	logical Data model

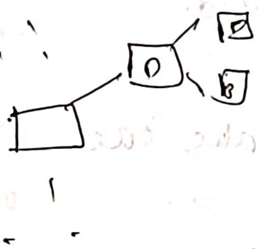
3 different ways to create DM

Conceptual DM: what are entities | Logical: differe wh & relat which are PK connection | Physical DM: dT, lines of d

Big Picture | Blue Point | IMPlementation

Star or snowflake Schem





Simple & Easy
Big Dimension.

More complex
Large Dimens...

Dimension
Descriptive inf
gives context to your data
who what where

Fact:
Quntitative inf
events
How much.
How many?

Gold layer:

Views
Data integration.
Agg:
Business logic & Rules

Star Schema:
Agg object
Flat Table

We have 2 Gender colm when we joined the tables. across customers.

So.

Data Integration

```
Select
  ci.cust_gndr,
  ca.gen,
  case when ci.cust_gend != 'n/a' Then ci.cust.gndr
       Else ca.gen COALESCE (ca.gen, 'n/a')
  End as gender.
  from Silver.crm_cust_info ci,
  left join Silver.erp_cust_azl2 ca.
  on ci.cust_key = ca.cid.
```

now change name as Snake case.

Surrogate key

System - generated unique identifies assigned to each record in a table.

DDL - based generation

- Query - based using (Row_no)

Row_Number () Over (order By cust_id) as customer_key

Create obj

Create view gold.dim_customers as

Select distinct gender from gold.dim_custo

Fact table

Sales transaction table.

Data look up → joining tables with surrogate key

Building facts

use dim's surrogate key instead of IDs to easily connect facts with dimensions

Data schema model

May.

Many 1-many

1. Cust who haven't placed any order yet
2. have. one —
3. have Multiple.

Data catalog

To understand the data better.