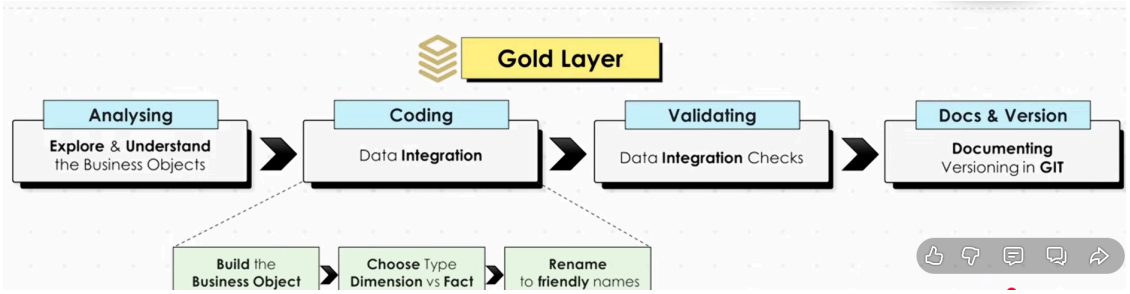
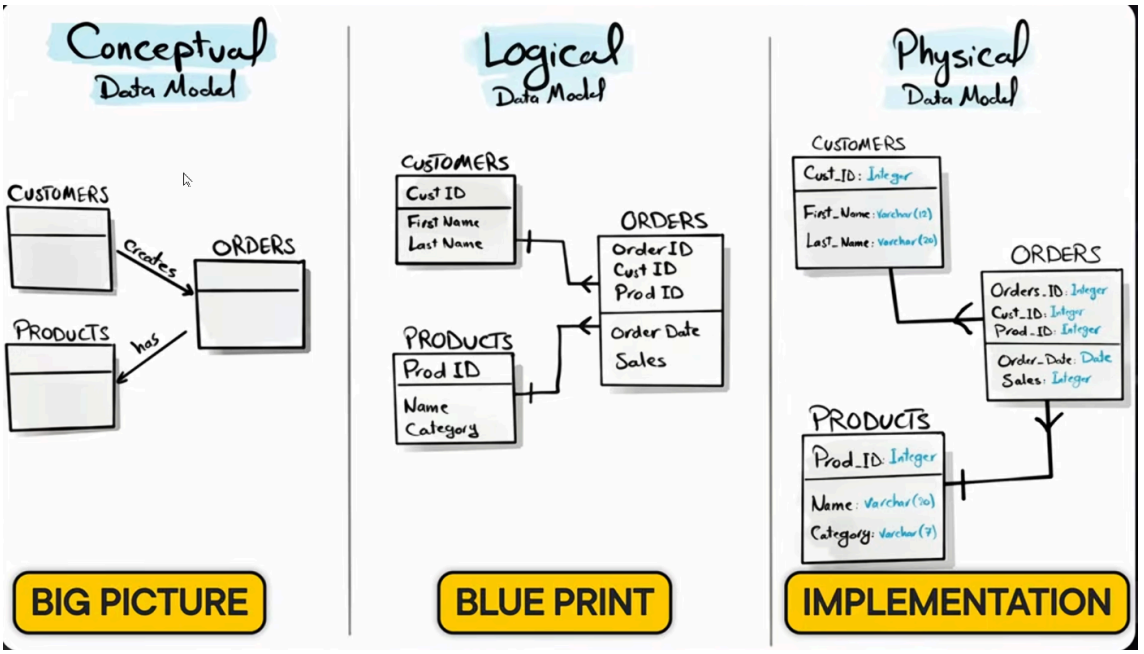
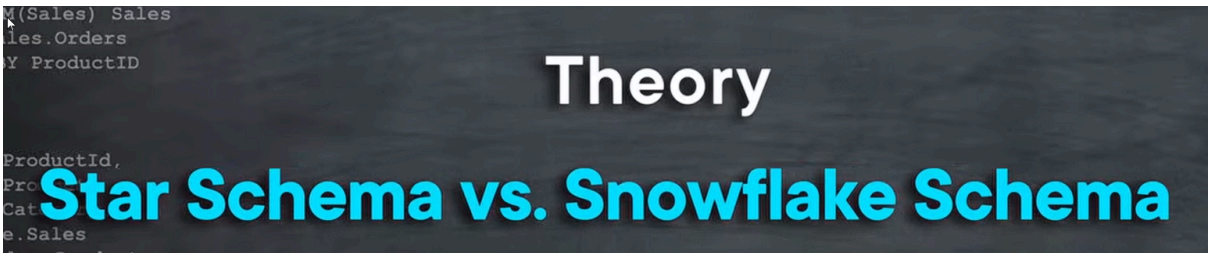
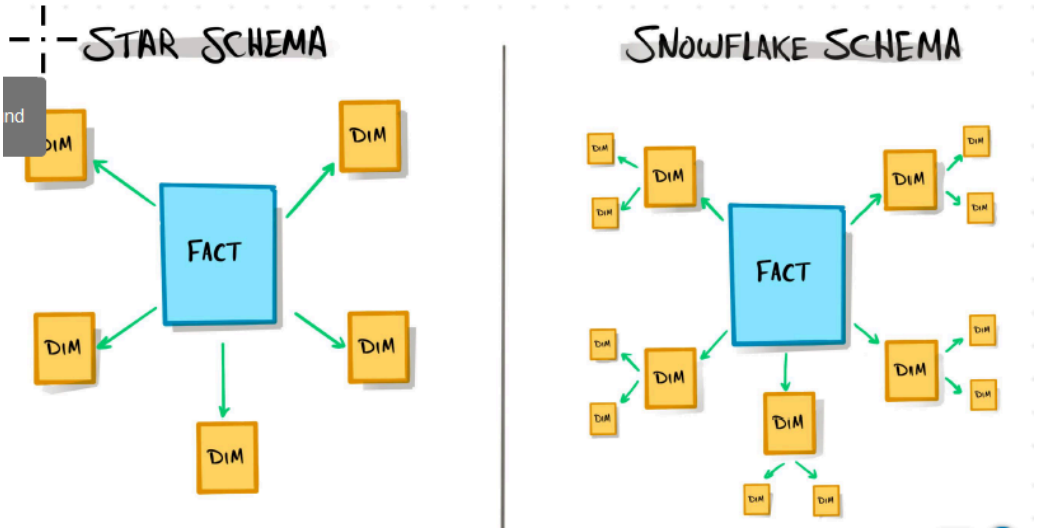


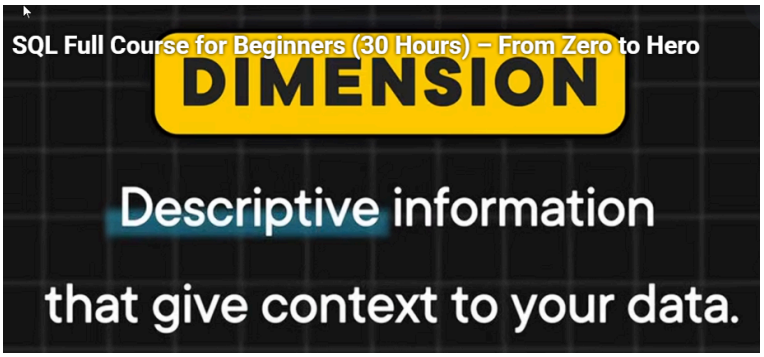


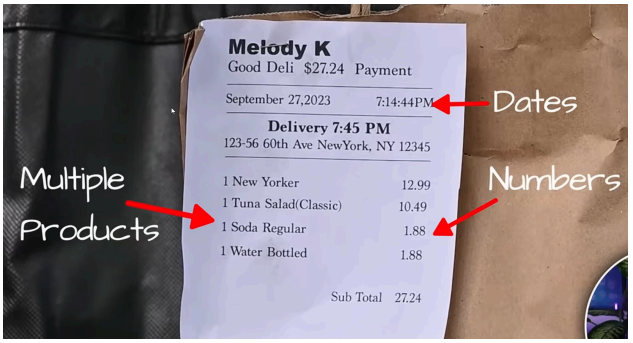
nosotros vamos a hacer el logical data model

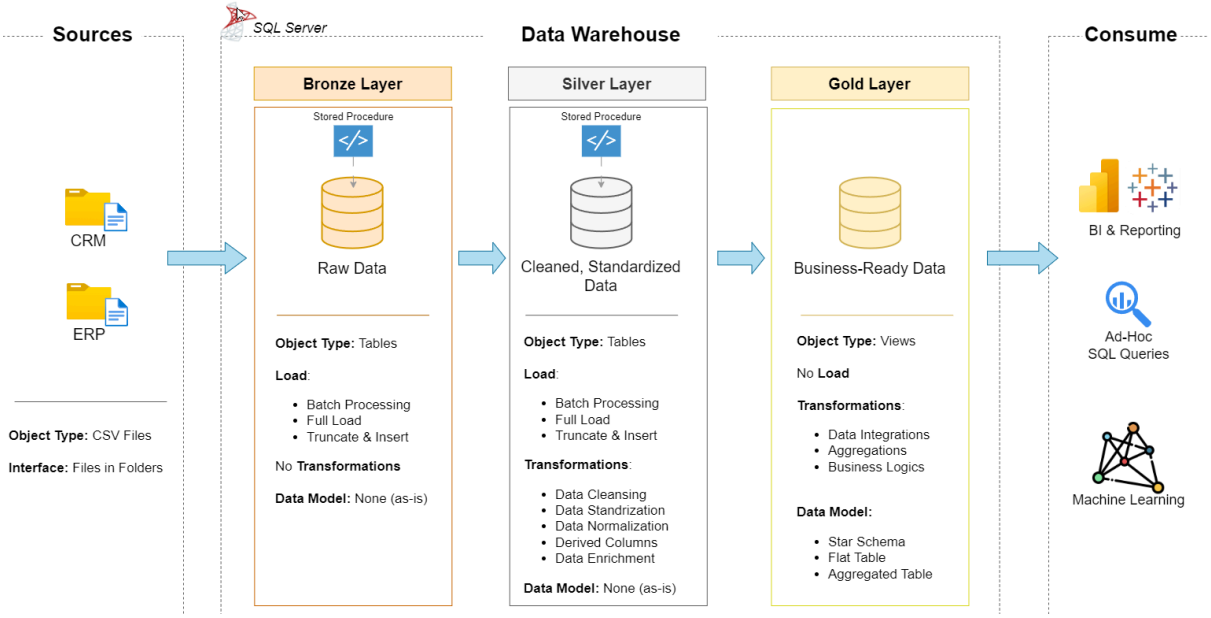


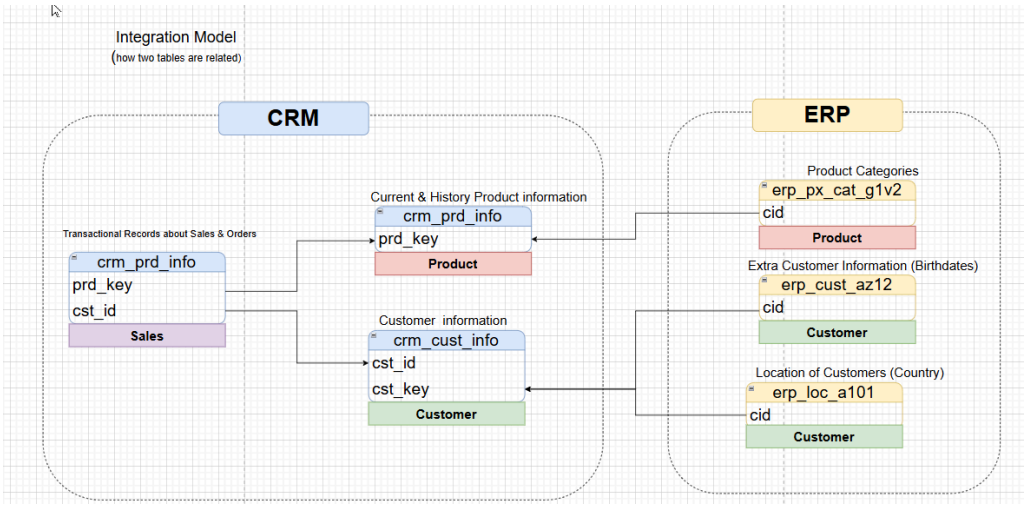


We are going to use star schema



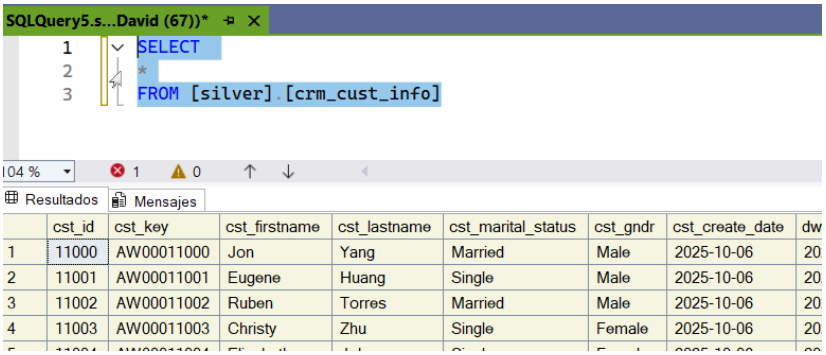






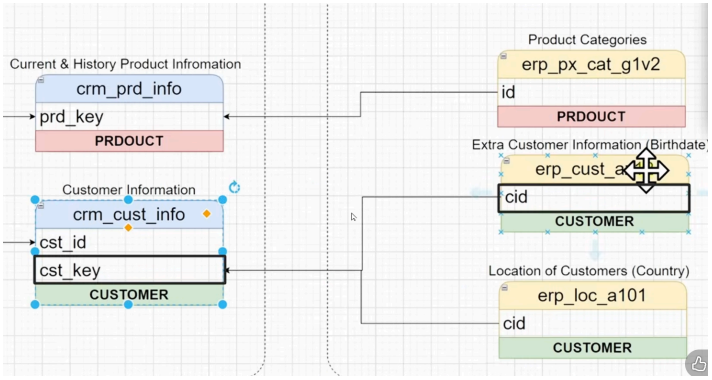
Let's start with the customer table

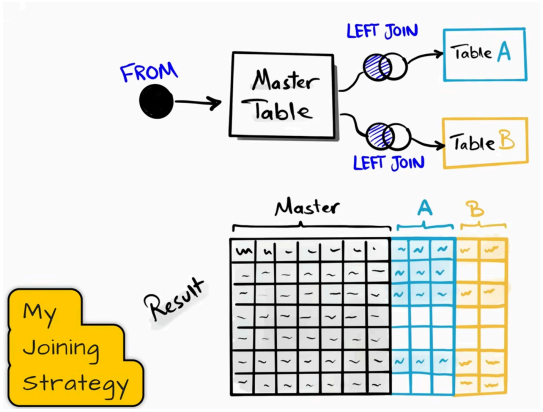
Now we're going to collect data from the Silver Layer. The silver is already clean.



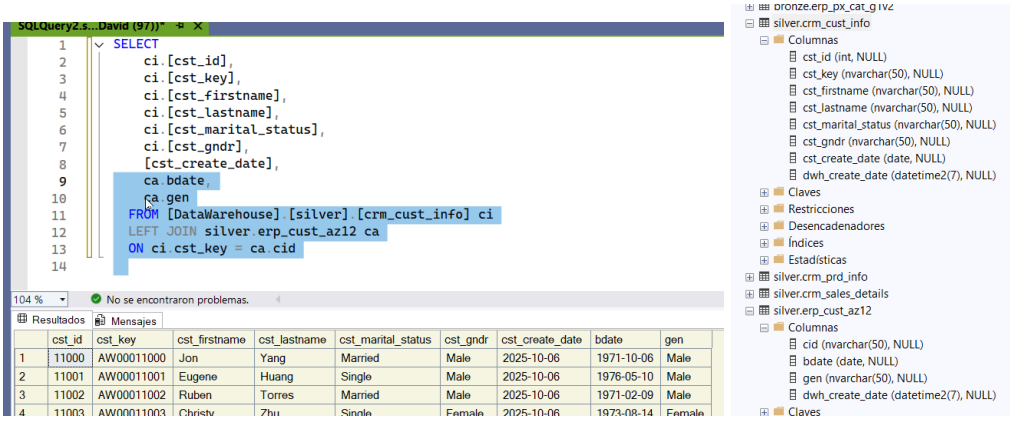
	cst_id	cst_key	cst_firstname	cst_lastname	cst_marital_status	cst_gndr	cst_create_date	dw
1	11000	AW00011000	Jon	Yang	Married	Male	2025-10-06	20
2	11001	AW00011001	Eugene	Huang	Single	Male	2025-10-06	20
3	11002	AW00011002	Ruben	Torres	Married	Male	2025-10-06	20
4	11003	AW00011003	Christy	Zhu	Single	Female	2025-10-06	20

Let's choose the columns that will be displayed in the golden layer.
I'll give the table an alias.

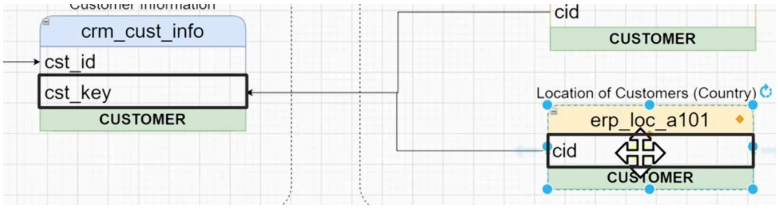




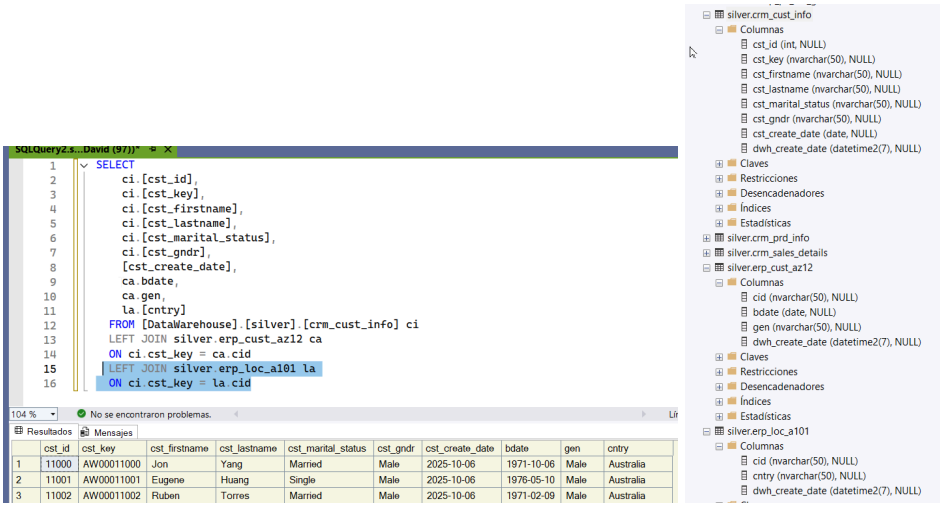
We need to join the data with the CID and the customer key. Avoid the inner join.



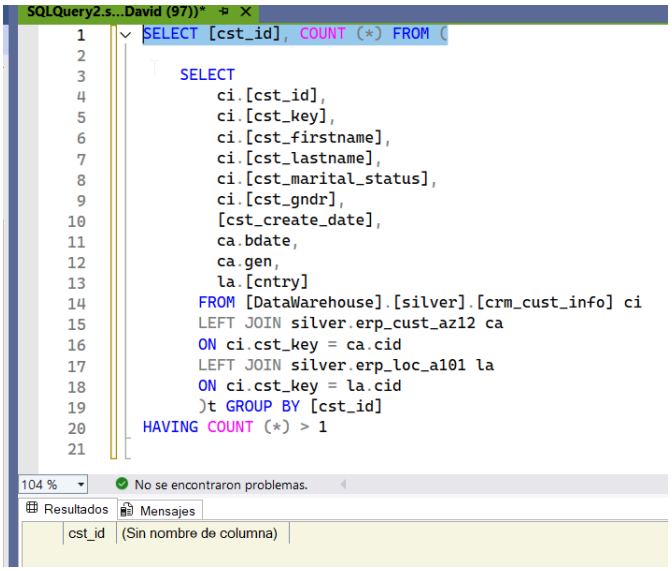
we join these two tables.



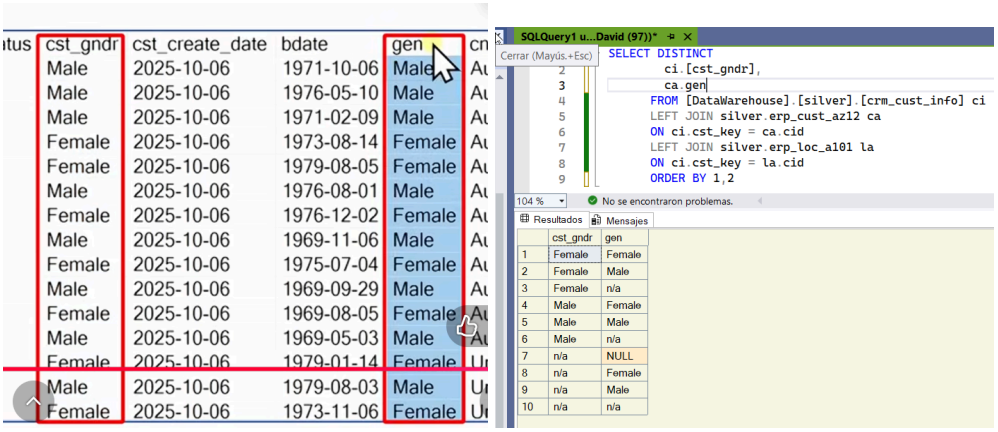
we connect to the other location information table with key



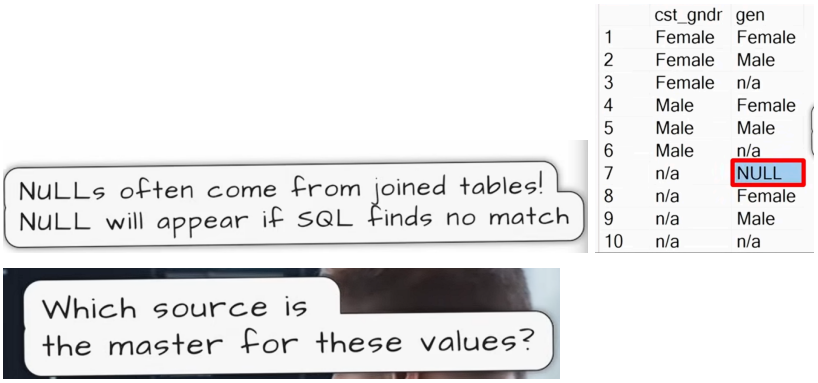
Let's check if there are duplicates in the query using group by



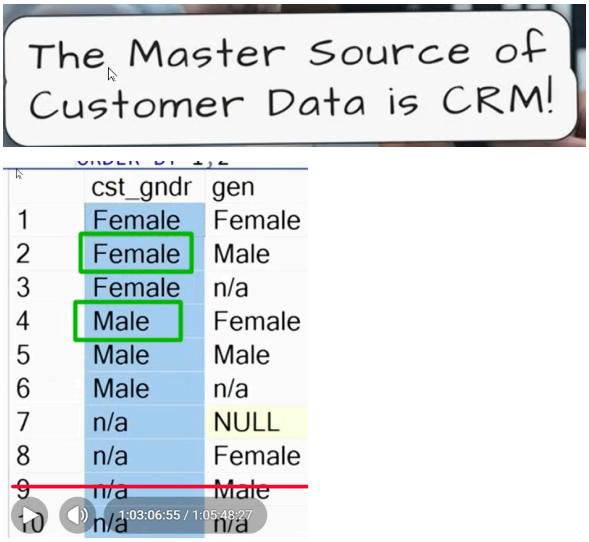
This query is used to see if we have duplicates in the PK.
This means that after joining all these tables, there is no problem.
We have a problem because we have two sources of gender information. One comes from ci.cst_gndr and the other from ca.gen



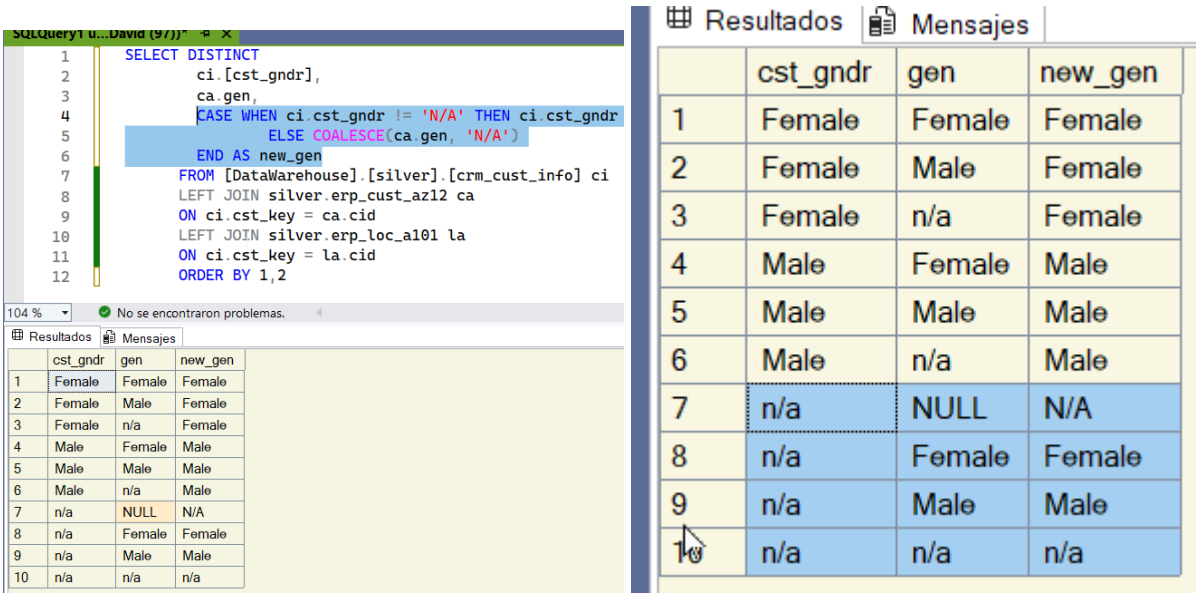
We need to do data integration. We'll remove all the information and leave those two columns. We'll leave only DISTINCT.
We have all the scenarios.



The NUll comes from joining tables.
That means there's a customer table and a crm table that aren't available in the other tables.



this source system is the master. -- CRM is the MAster for gender info



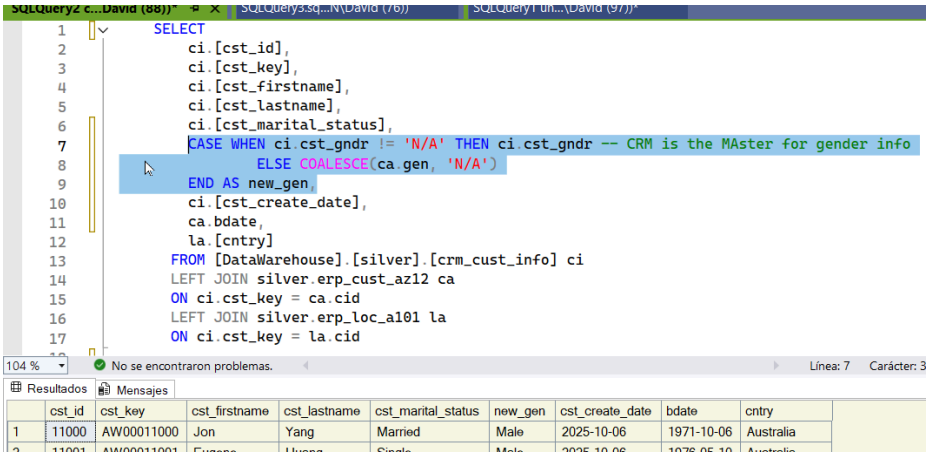
```sql
SELECT DISTINCT
        ci.[cst_gndr],
        ca.gen,
        CASE WHEN ci.cst_gndr != 'N/A' THEN ci.cst_gndr
                ELSE COALESCE(ca.gen, 'N/A')
        END AS new_gen
      FROM [DataWarehouse].[silver].[crm_cust_info] ci
      LEFT JOIN silver.erp_cust_az12 ca
      ON ci.cst_key = ca.cid
      LEFT JOIN silver.erp_loc_a101 la
      ON ci.cst_key = la.cid
      ORDER BY 1,2
```

104 % — No se encontraron problemas.

Resultados | Mensajes

	cst_gndr	gen	new_gen
1	Female	Female	Female
2	Female	Male	Female
3	Female	n/a	Female
4	Male	Female	Male
5	Male	Male	Male
6	Male	n/a	Male
7	n/a	NULL	N/A
8	n/a	Female	Female
9	n/a	Male	Male
10	n/a	n/a	n/a

If we have data from the CRM system we have to use it is not equal this means its not available (this means we dont have a value female or male) then go and used it. Otherwise that means if its not available then go and use the information fro mnthe second table. We have to deal with the NULL we use the COALESCE if this is a null then N/A. So when we have a null it will say NA.
Since we do not have information in the first table, we use the second table. We have a beautiful new column where we integrate 2 source systems in one.



```sql
SELECT
    ci.[cst_id],
    ci.[cst_key],
    ci.[cst_firstname],
    ci.[cst_lastname],
    ci.[cst_marital_status],
    CASE WHEN ci.cst_gndr != 'N/A' THEN ci.cst_gndr -- CRM is the MAster for gender info
            ELSE COALESCE(ca.gen, 'N/A')
    END AS new_gen,
    ci.[cst_create_date],
    ca.bdate,
    la.[cntry]
  FROM [DataWarehouse].[silver].[crm_cust_info] ci
  LEFT JOIN silver.erp_cust_az12 ca
  ON ci.cst_key = ca.cid
  LEFT JOIN silver.erp_loc_a101 la
  ON ci.cst_key = la.cid
```

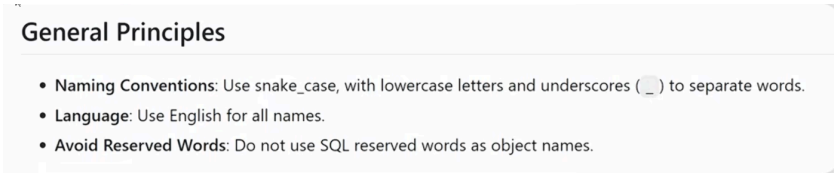
104 % — No se encontraron problemas. Línea: 7 Carácter: 3

Resultados | Mensajes

	cst_id	cst_key	cst_firstname	cst_lastname	cst_marital_status	new_gen	cst_create_date	bdate	cntry
1	11000	AW00011000	Jon	Yang	Married	Male	2025-10-06	1971-10-06	Australia

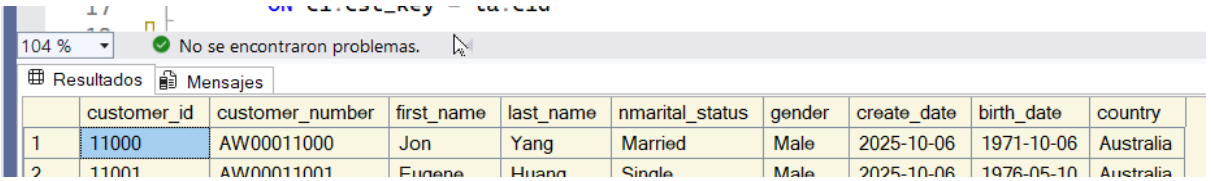
I joined those two tables with the when statement and then replaced it in the main query.
We have no duplicates. We took three tables and put them in one object.

Now we are going to give it friendly names.



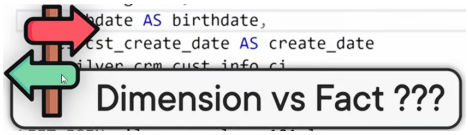
General Principles

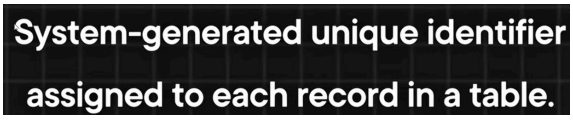
- **Naming Conventions**: Use snake_case, with lowercase letters and underscores (_) to separate words.
- **Language**: Use English for all names.
- **Avoid Reserved Words**: Do not use SQL reserved words as object names.



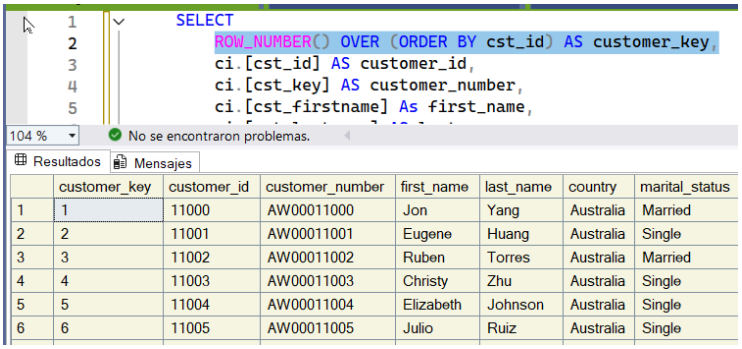
	customer_id	customer_number	first_name	last_name	nmarital_status	gender	create_date	birth_date	country
1	11000	AW00011000	Jon	Yang	Married	Male	2025-10-06	1971-10-06	Australia
2	11001	AW00011001	Eugene	Huang	Single	Male	2025-10-06	1976-05-10	Australia

We changed the names to more friendly ones





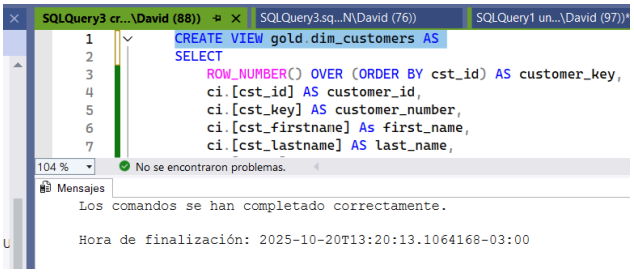
we are going to use a windows function ROW

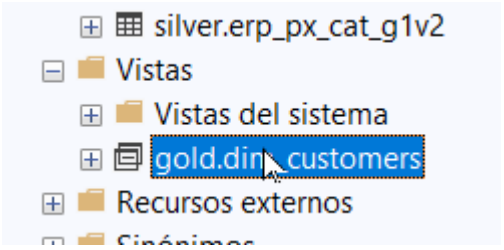


	customer_key	customer_id	customer_number	first_name	last_name	country	marital_status
1	1	11000	AW00011000	Jon	Yang	Australia	Married
2	2	11001	AW00011001	Eugene	Huang	Australia	Single
3	3	11002	AW00011002	Ruben	Torres	Australia	Married
4	4	11003	AW00011003	Christy	Zhu	Australia	Single
5	5	11004	AW00011004	Elizabeth	Johnson	Australia	Single
6	6	11005	AW00011005	Julio	Ruiz	Australia	Single

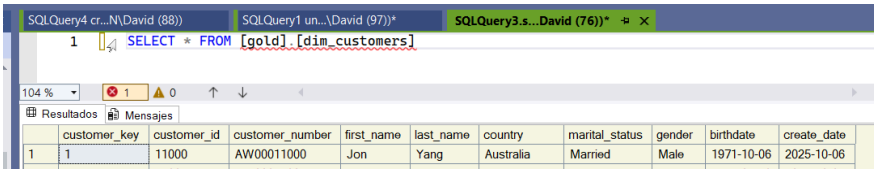
now we can use this key to connect the data model.

Now we need to create the object





our first object



	customer_key	customer_id	customer_number	first_name	last_name	country	marital_status	gender	birthdate	create_date
1	1	11000	AW00011000	Jon	Yang	Australia	Married	Male	1971-10-06	2025-10-06

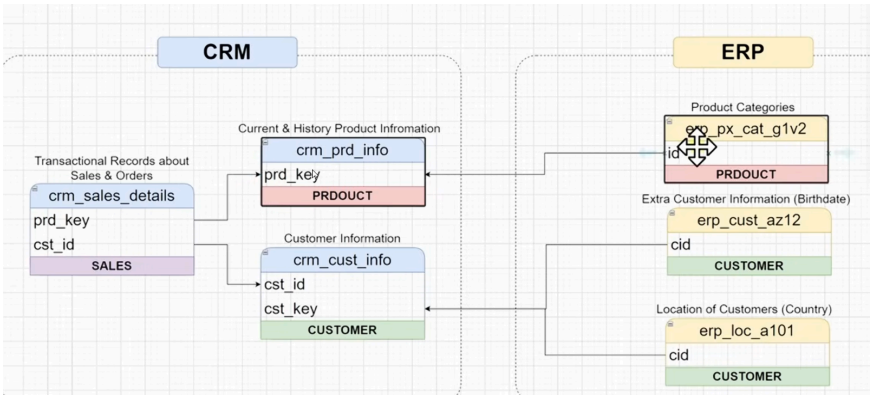
```
SELECT DISTINCT gender
FROM [gold].[dim_customers]
```

104 % — No se encontraron problemas.

Resultados | Mensajes

	gender
1	n/a
2	Male
3	Female

Now we have to build the 2nd object



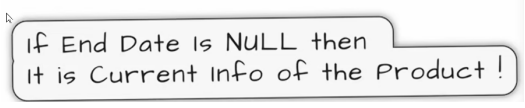

```
SELECT
pn.[prd_id],
pn.[cat_id],
pn.[prd_key],
pn.[prd_nm],
pn.[prd_cost],
pn.[prd_line],
pn.[prd_start_dt],
pn.[prd_end_dt]
FROM [silver].[crm_prd_info] pn
```

104 % — No se encontraron problemas.

Resultados | Mensajes

	prd_id	cat_id	prd_key	prd_nm	prd_cost	prd_line	prd_start_dt	prd_end_dt
1	478	AC_BC	BC-M005	Mountain Bottle Cage	4	Mountain	2013-07-01	NULL
2	479	AC_BC	BC-R205	Road Bottle Cage	3	Road	2013-07-01	NULL
3	477	AC_BC	WB-H098	Water Bottle - 30 oz.	2	Other Sales	2013-07-01	NULL

this obj contains historical info as well as current information. We are not using the primary key we are using the product key. We need to filter the historical data and stay with only current data.



if the end date is null thern that means is the current data

8	215	AC_HE	HL-U509	Sport-100 Helmet- Black	12	Other Sales	2011-07-01	2012-06-30
9	216	AC_HE	HL-U509	Sport-100 Helmet- Black	14	Other Sales	2012-07-01	2013-06-30
10	217	AC_HE	HL-U509	Sport-100 Helmet- Black	13	Other Sales	2013-07-01	NULL

we have the same pd but the first two we have a data but in the last we have null bcs is the current info. This means that it is open and its not closed yet.If we want only the current information its simple.

```sql
SELECT
pn.[prd_id],
pn.[cat_id],
pn.[prd_key],
pn.[prd_nm],
pn.[prd_cost],
pn.[prd_line],
pn.[prd_start_dt],
pn.[prd_end_dt]
FROM [silver].[crm_prd_info] pn
WHERE [prd_end_dt] IS NULL
```

No se encontraron problemas.

Resultados | Mensajes

	prd_id	cat_id	prd_key	prd_nm	prd_cost	prd_line	prd_start_dt	prd_end_dt
1	478	AC_BC	BC-M005	Mountain Bottle Cage	4	Mountain	2013-07-01	NULL
2	479	AC_BC	BC-R205	Road Bottle Cage	3	Road	2013-07-01	NULL
3	477	AC_BC	WB-H098	Water Bottle - 30 oz.	2	Other Sales	2013-07-01	NULL
4	483	AC_BR	RA-H123	Hitch Rack - 4-Bike	45	Other Sales	2013-07-01	NULL

```sql
SELECT
pn.[prd_id],
pn.[cat_id],
pn.[prd_key],
pn.[prd_nm],
pn.[prd_cost],
pn.[prd_line],
pn.[prd_start_dt],
pn.[prd_end_dt]
FROM [silver].[crm_prd_info] pn
WHERE [prd_end_dt] IS NULL -- Filter out all historical data
```

No se encontraron problemas.

Resultados | Mensajes

	prd_id	cat_id	prd_key	prd_nm	prd_cost	prd_line	prd_start_dt	prd_end_dt
1	478	AC_BC	BC-M005	Mountain Bottle Cage	4	Mountain	2013-07-01	NULL
2	479	AC_BC	BC-R205	Road Bottle Cage	3	Road	2013-07-01	NULL

you will only have the current product. Now we dont need the end date bcs its all null
now the next step is joined the the product categories. from the erp. We are going to use the id
you will only have the current product. Now we dont need the end date bcs its all null
now the next step is joined the the product categories. from the erp. We are going to use the id

```sql
SELECT
    pn.[prd_id],
    pn.[cat_id],
    pn.[prd_key],
    pn.[prd_nm],
    pn.[prd_cost],
    pn.[prd_line],
    pn.[prd_start_dt],
    pc.cat,
    pc.[subcat],
    pc.[maintenance]
FROM [silver].[crm_prd_info] pn
LEFT JOIN [silver].[erp_px_cat_g1v2] pc
ON pn.cat_id = pc.id
WHERE [prd_end_dt] IS NULL -- Filter out all historical data
```

No se encontraron problemas.

Resultados | Mensajes

	prd_id	cat_id	prd_key	prd_nm	prd_cost	prd_line	prd_start_dt	cat	subcat	maintenance
1	478	AC_BC	BC-M005	Mountain Bottle Cage	4	Mountain	2013-07-01	Accessories	Bottles and Cages	No
2	479	AC_BC	BC-R205	Road Bottle Cage	3	Road	2013-07-01	Accessories	Bottles and Cages	No
3	477	AC_BC	WB-H098	Water Bottle - 30 oz.	2	Other Sales	2013-07-01	Accessories	Bottles and Cages	No

now we joined the tables

Resultados | Mensajes

	prd_id	cat_id	prd_key	prd_nm	prd_cost	prd_line	prd_start_dt
1	478	AC_BC	BC-M005	Mountain Bottle Cage	4	Mountain	2013-07-01
2	479	AC_BC	BC-R205	Road Bottle Cage	3	Road	2013-07-01
3	477	AC_BC	WB-H098	Water Bottle - 30 oz.	2	Other Sales	2013-07-01

cat	subcat	maintenance
Accessories	Bottles and Cages	No
Accessories	Bottles and Cages	No
Accessories	Bottles and Cages	No

first table and second table. Now we have collected all the product information from the 2 systems. Now we need to check the quality of this results.

we want to make sure the product key is unique bcs we are going to use it later in order to join it with the sales.

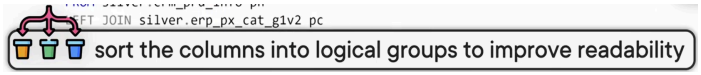
```sql
SELECT [prd_key], COUNT(*) FROM (
SELECT
    pn.[prd_id],
    pn.[cat_id],
    pn.[prd_key],
    pn.[prd_nm],
    pn.[prd_cost],
    pn.[prd_line],
    pn.[prd_start_dt],
    pc.cat,
    pc.[subcat],
    pc.[maintenance]
FROM [silver].[crm_prd_info] pn
LEFT JOIN [silver].[erp_px_cat_g1v2] pc
ON pn.cat_id = pc.id
WHERE [prd_end_dt] IS NULL -- Filter out
)t GROUP BY [prd_key]
HAVING COUNT(*)>1
```

104 % No se encontraron problemas.

Resultados Mensajes

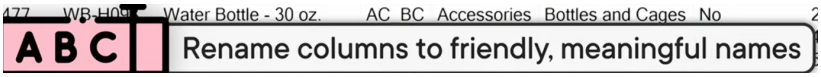
prd_key	(Sin nombre de columna)

We check that the PRD key is not duplicated. We don't have any duplicates.




```sql
WHERE prd_end_dt IS NULL --Filter out all historical dat
```

prd_id	prd_key	prd_nm	cat_id	cat	subcat	maintenance	prd_cost	prd_line	prd_start_dt
478	BC-M005	Mountain Bottle Cage	AC_BC	Accessories	Bottles and Cages	No	4	Mountain	2013-07-01
479	BC-R205	Road Bottle Cage	AC_BC	Accessories	Bottles and Cages	No	3	Road	2013-07-01
477	WB-H098	Water Bottle - 30 oz.	AC_BC	Accessories	Bottles and Cages	No	2	Other Sales	2013-07-01
483	RA-H123	Hitch Rack - 4-Bike	AC_BR	Accessories	Bike Racks	Yes	45	Other Sales	2013-07-01



because this is a dimensio we need to create a PK we are going to use the wx fx row nm

```sql
SELECT
    ROW_NUMBER () OVER (ORDER BY pn.[prd_start_dt], pn.[prd_key] ) AS product_key,
    pn.[prd_id] AS product_id,
    pn.[prd_key] AS product_number,
    pn.[prd_nm] AS product_name,
    pn.[cat_id] AS category_id,
    pc.cat AS category,
    pc.[subcat] AS subcategory,
    pc.[maintenance],
    pn.[prd_cost] AS cost,
    pn.[prd_line] AS product_line,
    pn.[prd_start_dt] AS start_date

FROM [silver].[crm_prd_info] pn
LEFT JOIN [silver].[erp_px_cat_g1v2] pc
ON pn.cat_id = pc.id
WHERE [prd_end_dt] IS NULL -- Filter out all historical data
```

104 % No se encontraron problemas.

Resultados Mensajes

	product_key	product_id	product_number	product_name	category_id	category	subcategory	maintenance
1	1	210	FR-R92B-58	HL Road Frame - Black- 58	CO_RF	Components	Road Frames	Yes
2	2	211	FR-R92R-58	HL Road Frame - Red- 58	CO_RF	Components	Road Frames	Yes
3	3	348	BK-M82B-38	Mountain-100 Black- 38	BI_MB	Bikes	Mountain Bikes	Yes
4	4	349	BK-M82B-42	Mountain-100 Black- 42	BI_MB	Bikes	Mountain Bikes	Yes

now we have generated it a pk for each product. And we are going to use it in order to connect to our data model,

Now latest build the view

```sql
CREATE VIEW gold.dim_products AS
SELECT
    ROW_NUMBER () OVER (ORDER BY pn.[prd_start_dt],
    pn.[prd_id] AS product_id,
    pn.[                    number,
    pn.[                    ame,
    pn.[cat_id] AS category_id,
    pc.cat AS category,
    pc.[subcat] AS subcategory,
    pc.[maintenance],
    pn.[prd_cost] AS cost,
    pn.[prd_line] AS product_line,
    pn.[prd_start_dt] AS start_date

FROM [silver].[crm_prd_info] pn
LEFT JOIN [silver].[erp_px_cat_g1v2] pc
ON pn.cat_id = pc.id
WHERE [prd_end_dt] IS NULL -- Filter out all histor
```

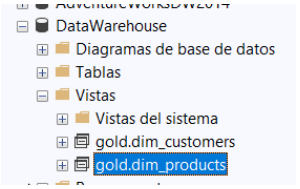
columna prd_id(int, null)

No se encontraron problemas.

Mensajes

```
Los comandos se han completado correctamente.

Hora de finalización: 2025-10-20T14:04:30.9445452-03:00
```



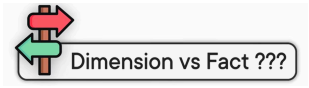
We have created our object. Gold layer 2nd dimension products

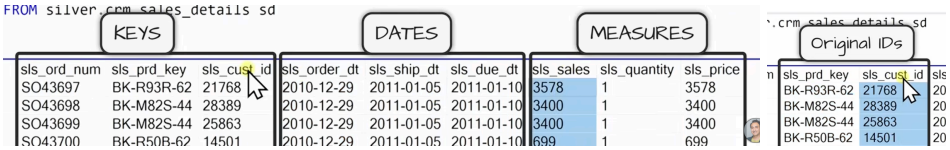

```sql
SELECT
sd.[sls_ord_num],
sd.[sls_prd_key],
sd.[sls_cust_id],
sd.[sls_order_dt],
sd.[sls_ship_dt],
sd.[sls_due_dt],
sd.[sls_sales],
sd.[sls_quantity],
sd.[sls_price]
FROM [silver].[crm_sales_details] sd
```

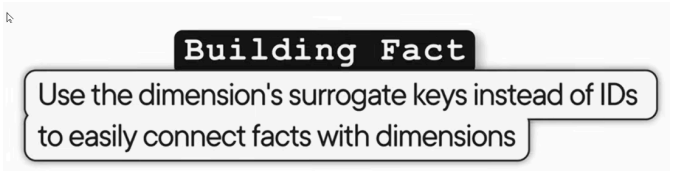
No se encontraron problemas.

Resultados | Mensajes

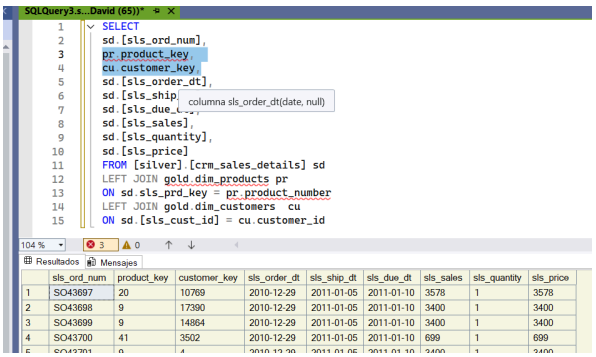
	sls_ord_num	sls_prd_key	sls_cust_id	sls_order_dt	sls_ship_dt	sls_due_dt	sls_sales	sls_quantity	sls_price
1	SO43697	BK-R93R-62	21768	2010-12-29	2011-01-05	2011-01-10	3578	1	3578
2	SO43698	BK-M82S-44	28389	2010-12-29	2011-01-05	2011-01-10	3400	1	3400

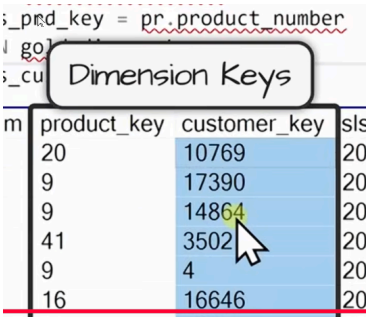






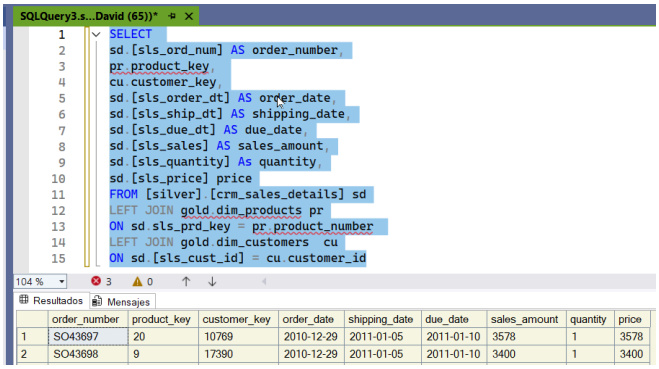
we are going to join both dimensions in order to
the prd key and the customer id those information comes from the source system we are going to replace those 2 information with the surrogate key that we have generated it.
DATA LOOK UP
in the silver layer we don't have any subrogate keys we have in the golden layer.
the only information that we need from the dimension is the key

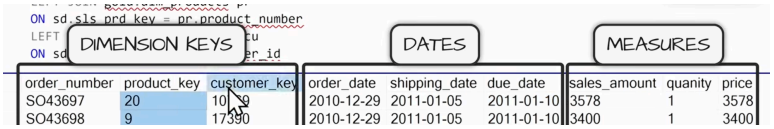




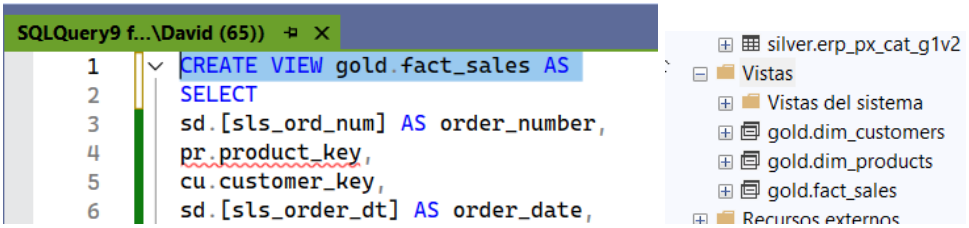
In this way we join the tables with the surrogate key that we created and delete the ones that come by default.

In our fact table we have the 2 keys from the dimension ans this will help us to connect the data model to connect the fact with the dimensions. Now we give them friends and names

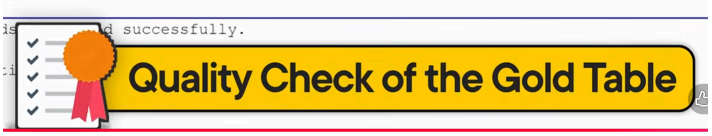


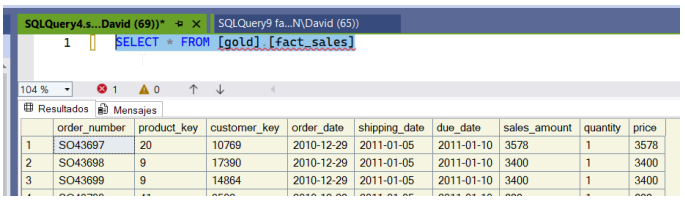


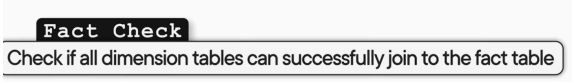
they have friendly names.

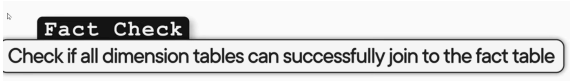


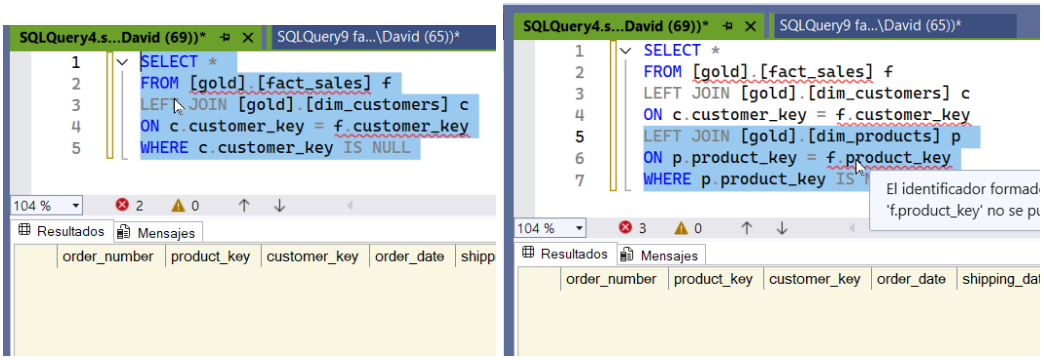
now we have 3 object in the gold layer. 2 dimension and 1 fact



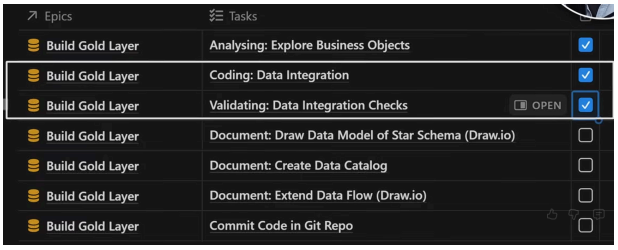


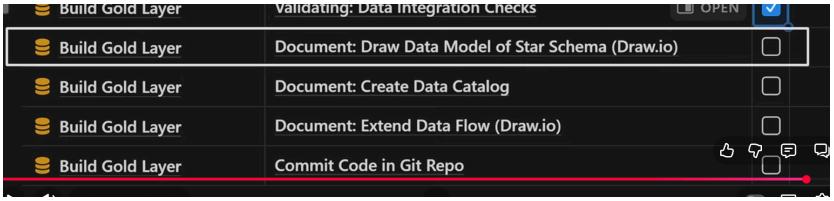


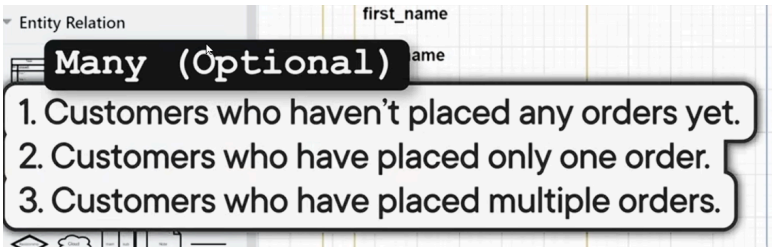


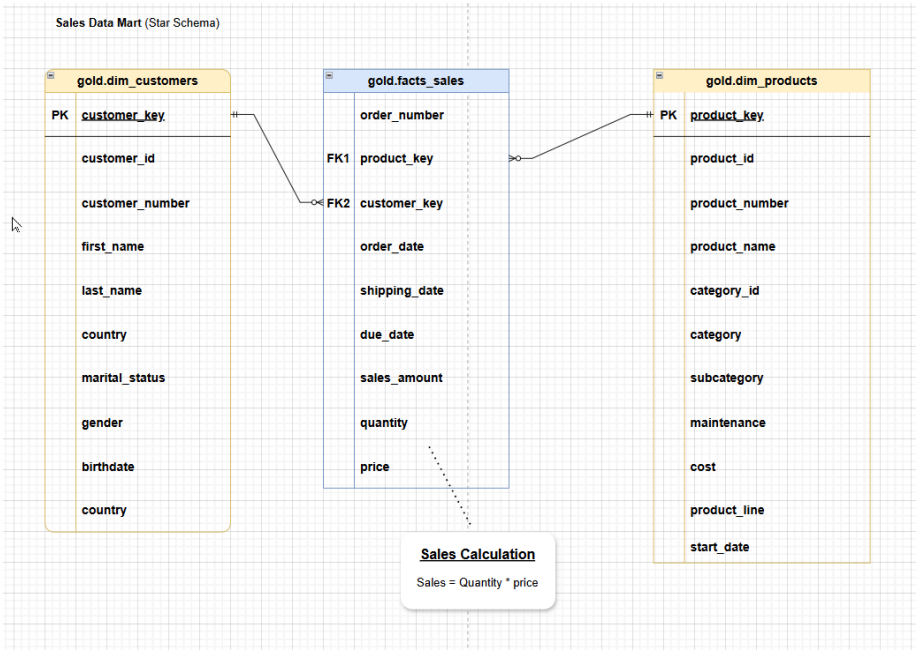


everything is matching perfectly.



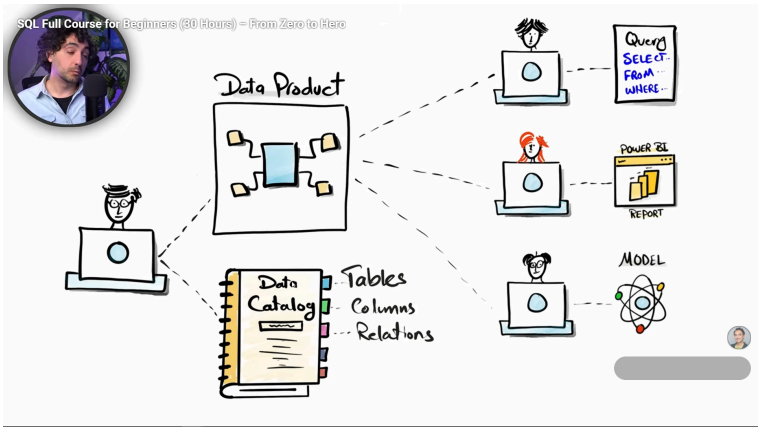




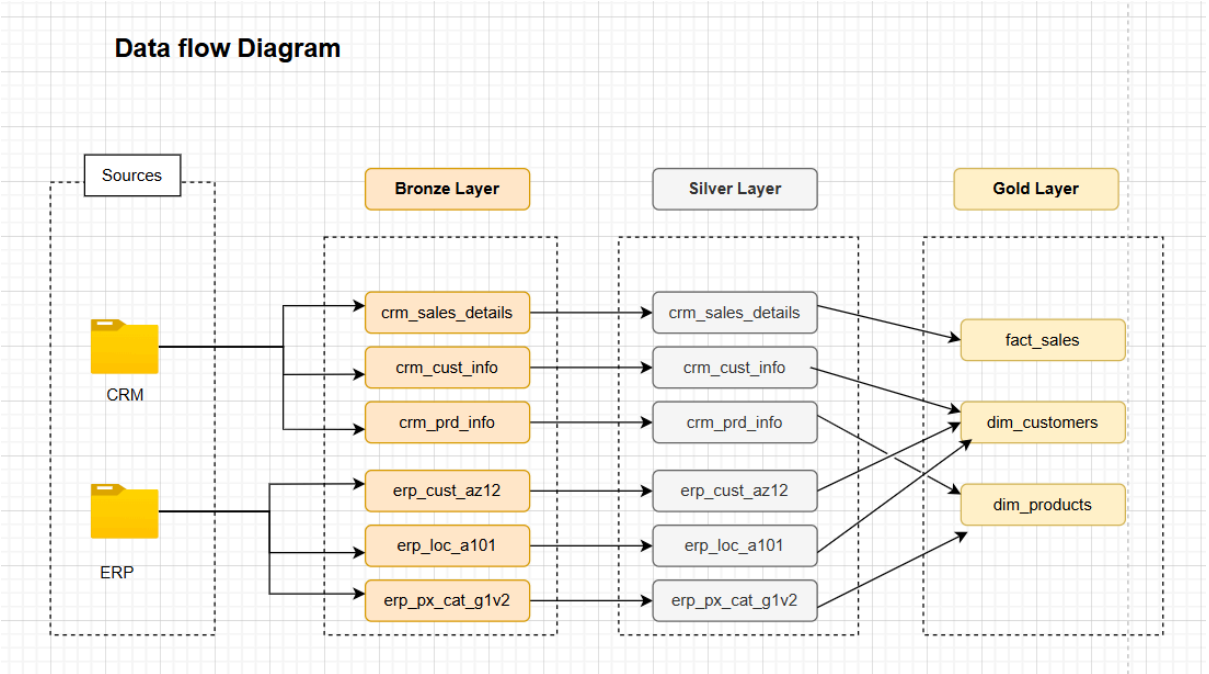

Sales Data Mart (Star Schema)
gold.dim_customers
PK customer_key
customer_id
customer_number
first_name
last_name
country
marital_status
gender
birthdate
country
gold.facts_sales
order_number
FK1 product_key
FK2 customer_key
order_date
shipping_date
due_date
sales_amount
quantity
price
gold.dim_products
PK product_key
product_id
product_number
product_name
category_id
category
subcategory
maintenance
cost
product_line
start_date
Sales Calculation
Sales = Quantity * price


Build Gold Layer
Document: Draw Data Model of Star Schema (Draw.io)
Build Gold Layer
Document: Create Data Catalog


Build Gold Layer
Data Catalog


SQL Full Course for Beginners (30 Hours) – From Zero to Hero
Data Product
Query
SELECT
FROM
WHERE
POWER BI
REPORT
MODEL
Data Catalog
Tables
Columns
Relations


Build Gold Layer
Data Flow


Data flow Diagram
Sources
CRM
ERP
Bronze Layer
crm_sales_details
crm_cust_info
crm_prd_info
erp_cust_az12
erp_loc_a101
erp_px_cat_g1v2
Silver Layer
crm_sales_details
crm_cust_info
crm_prd_info
erp_cust_az12
erp_loc_a101
erp_px_cat_g1v2
Gold Layer
fact_sales
dim_customers
dim_products

The end.