Project Deliverable

Part 2

1) Set the role and warehouse as accountadmin and compute_wh. Created the database and staging schema within the database

```
-- Set the role as Account admin to use in this worksheet --
USE ROLE accountadmin;

-- Set the warehouse to use in this worksheet --
USE WAREHOUSE compute_wh;

-- Creating a Star-Schema Database for our project --
CREATE OR REPLACE DATABASE DE_Project;

-- Creating a Staging Schema --
CREATE OR REPLACE SCHEMA DE_Project.stage_data;
```

status
Schema STAGE_DATA successfully created.

Created 3 dimension and a fact table within the staging schema to clean and transform the data

Syntax for Customer dimension table:

```
-- Creating dimension and fact tables within a stage_data schema--

-- Creating Customer dimension table --
CREATE OR REPLACE TABLE DE_Project.stage_data.Customer_Dim(
Customer_ID NUMBER(19,0) PRIMARY KEY,
First_Name VARCHAR(500),
Last_Name VARCHAR(500),
Email VARCHAR(500),
Phone VARCHAR(500),
Cust_Address VARCHAR(1000),
Cust_State VARCHAR(100),
Cust_City VARCHAR(100),
Cust_zip DECIMAL(10)
);
```

status
Table CUSTOMER_DIM successfully created.

Syntax for Product dimension table:

```
-- Creating Product dimension table --
CREATE OR REPLACE TABLE DE_Project.stage_data.Product_Dim(
Product_Id NUMBER(19,0) PRIMARY KEY,
Brand VARCHAR(200),
Name VARCHAR(200),
Description VARCHAR(1000),
Price DECIMAL(10,2),
MSRP DECIMAL(10,2),
Cost DECIMAL(10,2)
);
```

status
Table PRODUCT_DIM successfully created.

Syntax for Orders Fact table:

```
-- Creating an Orders fact table --
CREATE OR REPLACE TABLE DE_Project.stage_data.Orders_Fact(
Order_Id Number(19,0) PRIMARY KEY,
Order_Amt DECIMAL(20,2),
Order_Qty DECIMAL(20),
Customer_Id Number(19,0),
Order_Date TIMESTAMP_TZ,
FOREIGN KEY(Customer_Id) REFERENCES Customer_Dim(Customer_Id)
);
```

Results | Chart

status
Table ORDERS_FACT successfully created.

Syntax for Orders details dimension table:

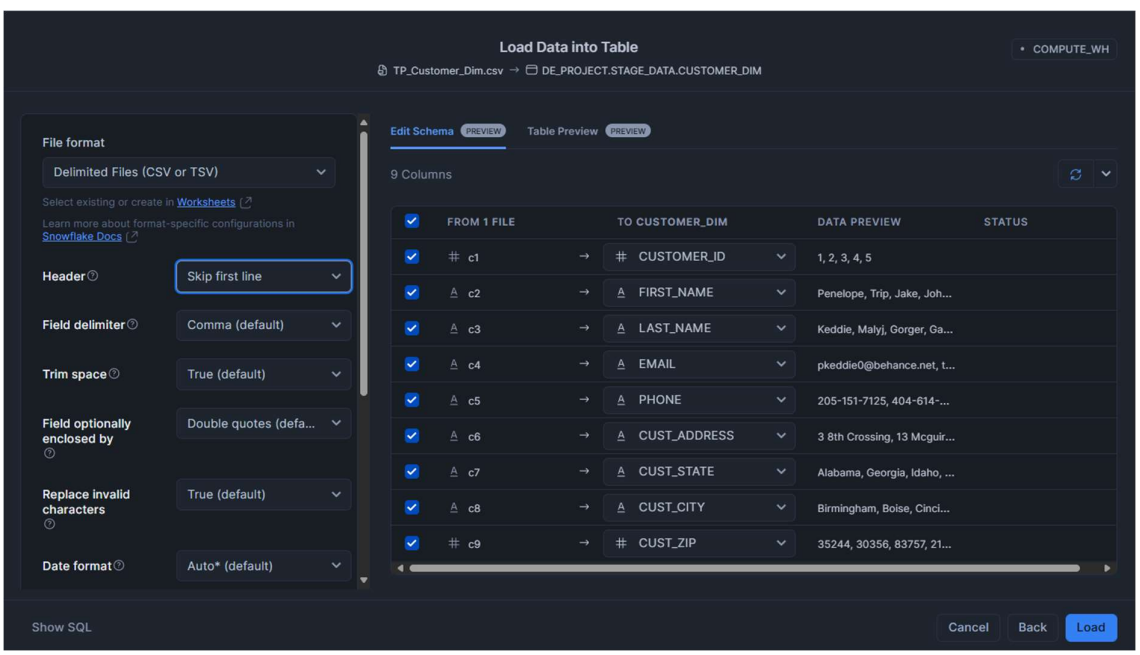
```
-- Creating an Order Detail dimension table --
CREATE OR REPLACE TABLE DE_Project.stage_data.Order_detail_dim(
Order_detail_Id NUMBER(19,0) PRIMARY KEY,
Order_Id NUMBER(19,0),
Product_Id NUMBER(19,0),
FOREIGN KEY(Order_Id) REFERENCES Orders_Fact(Order_Id),
FOREIGN KEY(Product_Id) REFERENCES Product_Dim(Product_Id)
);
```

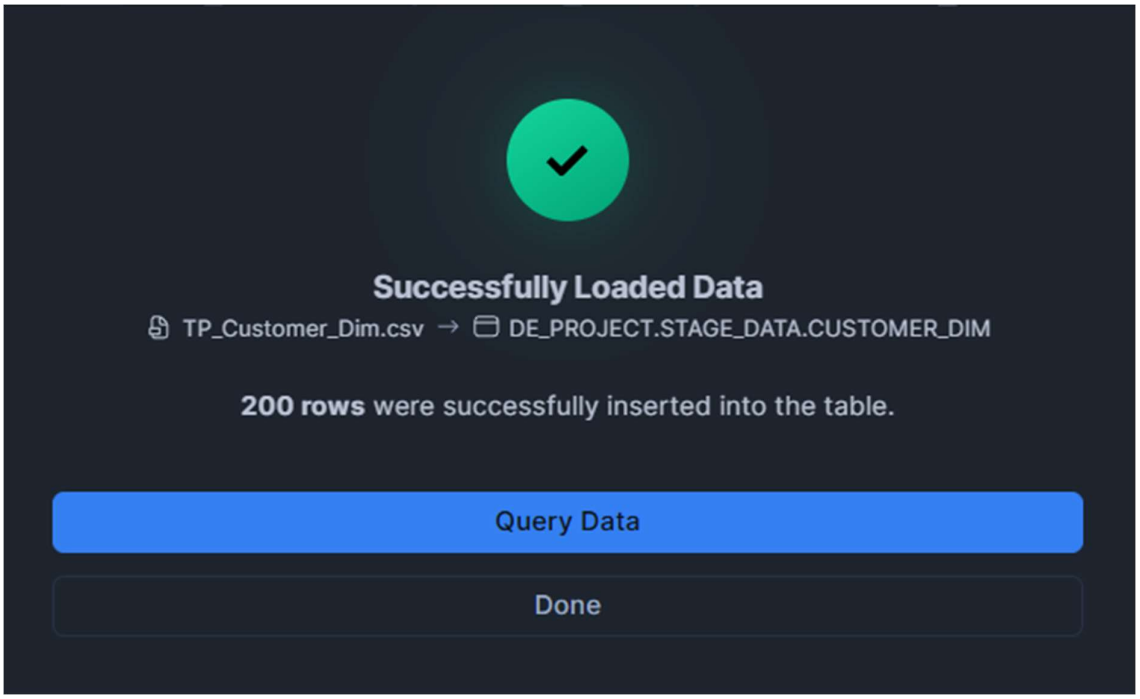
Results | Chart

status
Table ORDER_DETAIL_DIM successfully created.

Query Details
Query duration 303ms

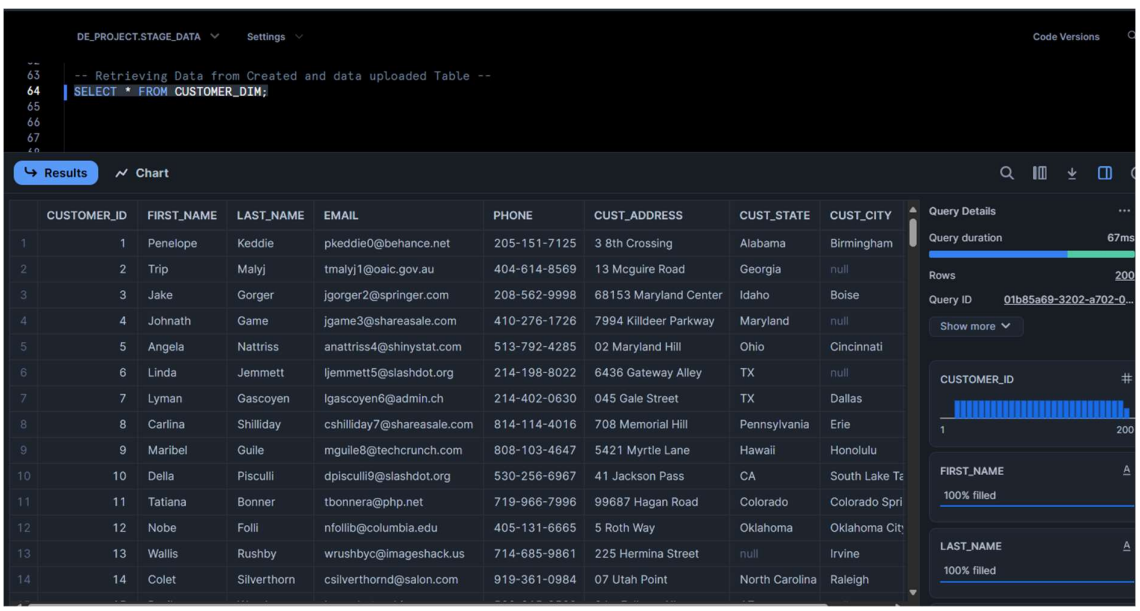
2)





Above snapshots shows the successful upload of customer data into customer dimension table.

Similarly, I've uploaded data for other tables as well



	CUSTOMER_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE	CUST_ADDRESS	CUST_STATE	CUST_CITY
1	1	Penelope	Keddie	pkeddie0@behance.net	205-151-7125	3 8th Crossing	Alabama	Birmingham
2	2	Trip	Malyj	tmalyj1@oaic.gov.au	404-614-8569	13 Mcguire Road	Georgia	null
3	3	Jake	Gorger	jgorger2@springer.com	208-562-9998	68153 Maryland Center	Idaho	Boise
4	4	Johnath	Game	jgame3@shareasale.com	410-276-1726	7994 Killdeer Parkway	Maryland	null
5	5	Angela	Nattriss	anattriss4@shinystat.com	513-792-4285	02 Maryland Hill	Ohio	Cincinnati
6	6	Linda	Jemmett	ljemmett5@slashdot.org	214-198-8022	6436 Gateway Alley	TX	null
7	7	Lyman	Gascoyen	lgascoyen6@admin.ch	214-402-0630	045 Gale Street	TX	Dallas
8	8	Carlina	Shilliday	cshilliday7@shareasale.com	814-114-4016	708 Memorial Hill	Pennsylvania	Erie
9	9	Maribel	Guile	mguile8@techcrunch.com	808-103-4647	5421 Myrtle Lane	Hawaii	Honolulu
10	10	Della	Pisculli	dpisculli9@slashdot.org	530-256-6967	41 Jackson Pass	CA	South Lake Ta
11	11	Tatiana	Bonner	tbonnera@php.net	719-966-7996	99687 Hagan Road	Colorado	Colorado Spri
12	12	Nobe	Folli	nfollib@columbia.edu	405-131-6665	5 Roth Way	Oklahoma	Oklahoma City
13	13	Wallis	Rushby	wrushbyc@imageshack.us	714-685-9861	225 Hermina Street	null	Irvine
14	14	Colet	Silverthorn	csilverthornd@salon.com	919-361-0984	07 Utah Point	North Carolina	Raleigh

```sql
-- Retrieving Data from Created and data uploaded Table --
SELECT * FROM Orders_Fact;
```

Results | Chart

	ORDER_ID	ORDER_AMT	ORDER_QTY	CUSTOMER_ID	ORDER_DATE
1	1	543.50	3	20	2022-07-14 15:43:22.000 +0000
2	2	138.62	4	156	2022-09-25 08:25:31.000 +0000
3	3	421.68	1	6	2023-07-22 02:54:07.000 +0000
4	4	197.81	3	153	2022-03-20 00:00:00.000 -0700
5	5	742.61	3	89	2023-10-05 17:16:24.000 +0000
6	6	832.69	1	135	2022-03-10 22:21:33.000 +0000
7	7	363.86	5	54	2023-04-19 01:44:14.000 +0000
8	8	371.74	3	125	2022-04-19 04:37:36.000 +0000
9	9	784.62	1	196	2023-02-01 15:28:41.000 +0000
10	10	631.18	1	167	2022-12-04 13:41:21.000 +0000
11	11	79.39	4	96	2022-02-16 08:14:18.000 +0000
12	12	597.73	1	149	2023-02-07 22:56:40.000 +0000
13	13	115.31	2	157	2022-07-17 05:40:32.000 +0000

```sql
-- Retrieving Data from Created and data uploaded Table --
SELECT * FROM PRODUCT_DIM;
```

Results | Chart

	PRODUCT_ID	BRAND	NAME	DESCRIPTION	PRICE	MSRP	COST
1	1	TechWave	SpeedBoost Wireless Charging Pad	Fast - efficient - wireless power.	39.65	44.41	14.67
2	2	TechWave	PixelGrip Gaming Mouse	Precision control for gamers	26.55	29.74	9.82
3	3	TechWave	SoundSync Bluetooth Earbuds	Seamless audio freedom anytime	69.99	78.39	25.90
4	4	TechWave	UltraView HD Webcam	Crystal-clear video conferencing	32.42	36.31	12.00
5	5	TechWave	TurboDrive External SSD	Lightning-fast storage on-the-go	110.99	124.31	41.07
6	6	EcoEra	GreenGrow Indoor Herb Garden	Sustainable - fresh home gardening	209.99	235.19	77.70
7	7	EcoEra	PureFlow Bamboo Water Bottle	Eco-friendly hydration companion	29.99	33.59	11.10
8	8	EcoEra	EarthGuard Plant-Based Cleaner	Natural - powerful cleaning solution	17.98	20.14	6.65
9	9	EcoEra	EcoCycle Recycled Laptop Sleeve	Protect your tech sustainably	45.68	51.16	16.90
10	10	EcoEra	BioBlend Organic Protein Powder	Clean - plant-powered nutrition boost	89.99	100.79	33.30
11	11	LuxeLife	GlamGlow LED Vanity Mirror	Illuminate your beauty routine	199.98	223.98	73.99
12	12	LuxeLife	SilkSoothe Sleep Mask Set	Luxurious - restful sleep experience	45.99	51.51	17.02
13	13	LuxeLife	PurePlush Faux Fur Throw	Cozy up in ultimate comfort	87.66	98.18	32.43

```sql
-- Retrieving Data from Created and data uploaded Table --
SELECT * FROM ORDER_DETAIL_DIM;
```

Results | Chart

	ORDER_DETAIL_ID	ORDER_ID	PRODUCT_ID
1	1	213	16
2	2	60	6
3	3	188	12
4	4	229	16
5	5	121	12
6	6	115	13
7	7	39	3
8	8	127	1
9	9	115	7
10	10	159	20
11	11	50	13
12	12	53	3
13	13	118	16

Using Select statements, above four snapshots show the file data successfully uploaded to the respective dimension and fact tables

3)

Initial Data Exploration, Cleaning and Transforming Data:

Customer Dimension Table:

```sql
-- No of records in customer_dim table --
SELECT COUNT(*) AS Total_Number_of_Customers FROM CUSTOMER_DIM;
```

Results | Chart

	TOTAL_NUMBER_OF_CUSTOMERS
1	200

Using below script, I've identified that there are no duplicates in customer table

```sql
-- Checking any duplicate names, email or phone --
SELECT First_Name, Last_Name, Email, Phone, COUNT(*) AS No_Of_Duplicates
FROM CUSTOMER_DIM
GROUP BY First_Name, Last_Name, Email, Phone
HAVING COUNT(*) > 1;
```

Results | Chart

FIRST_NAME	LAST_NAME	EMAIL	PHONE	NO_OF_DUPLICATES

Query produced no results

```sql
-- Chcking Null values in each column --

SELECT COUNT(*) AS Total_Records, COUNT(First_Name) AS First_Name_Count , COUNT(Last_Name) AS Last_Name_Count, COUNT(Email) AS Email_Count, COUNT(Phone) AS Phone_Count,
COUNT(Cust_Address) AS Address_Count, COUNT(Cust_State) AS State_Count, COUNT(Cust_City) AS City_Count,
COUNT(Cust_Zip) AS Zip_Count
FROM CUSTOMER_DIM;
```

Results | Chart

	TOTAL_RECORDS	FIRST_NAME_COUNT	LAST_NAME_COUNT	EMAIL_COUNT	PHONE_COUNT	ADDRESS_COUNT	STATE_COUNT	CITY_COUNT	ZIP_COUNT
1	200	200	200	200	191	191	168	167	170

Query Details
Query duration 21ms

Above SQL query shows that there are null values in Phone, Address, State, City, Zip columns of customer table

```sql
-- Handling Null values in Customer_dim table --
UPDATE Customer_Dim SET Phone = '111-111-1111'
WHERE Phone IS NULL;

UPDATE Customer_Dim SET Cust_Address = 'NA'
WHERE Cust_Address IS NULL;

UPDATE Customer_Dim SET Cust_State = 'NA'
WHERE Cust_State IS NULL;

UPDATE Customer_Dim SET Cust_City = 'NA'
WHERE Cust_City IS NULL;

UPDATE Customer_Dim SET Cust_Zip = 00000
WHERE Cust_Zip IS NULL;
```

Results | Chart

	number of rows updated	number of multi-joined rows updated
1	30	0

Handled null values in customer table by updating it with default values like ‘111-111-111’, ‘N/A’ and 00000 for respective columns

```sql
-- Chcking Null values in each column --

SELECT COUNT(*) AS Total_Records, COUNT(First_Name) AS First_Name_Count , COUNT(Last_Name) AS Last_Name_Count, COUNT(Email) AS Email_Count, COUNT(Phone) AS Phone_Count, COUNT(Cust_Address) AS
Address_Count, COUNT(Cust_State) AS State_Count, COUNT(Cust_City) AS City_Count,
COUNT(Cust_Zip) AS Zip_Count
FROM CUSTOMER_DIM;
```

Results | Chart

	TOTAL_RECORDS	FIRST_NAME_COUNT	LAST_NAME_COUNT	EMAIL_COUNT	PHONE_COUNT	ADDRESS_COUNT	STATE_COUNT	CITY_COUNT	ZIP_COUNT
1	200	200	200	200	200	200	200	200	200

Query Details
Query duration 116ms

Above query shows that null values are handled accurately

```sql
SELECT * FROM CUSTOMER_DIM;
```

Results | Chart

	CUSTOMER_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE	CUST_ADDRESS	CUST_STATE	CUST_CITY	CUST_ZIP
180	163	Grenville	Sawley	gsawley4i@buzzfeed.com	415-717-3505	588 Lindbergh Avenue	CA	San Francisco	0
181	30	Remy	Stebbings	rstebbingst@hhs.gov	111-111-1111	62782 Northland Junction	Ohio	Cincinnati	0
182	176	Vallie	Gurney	vgurney4v@ebay.com	405-688-4159	5145 Springs Court	Oklahoma	Oklahoma City	0
183	69	Misti	Nyssens	mnyssens1w@1und1.de	417-598-5542	NA	Missouri	Springfield	0
184	94	Brucie	Napper	bnapper2l@theguardian.com	218-604-5228	NA	Minnesota	Duluth	0
185	196	Giralda	Davers	gdavers5f@quantcast.com	312-559-6283	2 Graedel Park	Illinois	Chicago	0
186	46	Laraine	Crepin	lcrepin19@sbwire.com	415-144-6926	993 Nova Road	NA	San Francisco	0
187	68	Augustine	Hattiff	ahattiff1v@linkedin.com	901-880-1243	3362 Clyde Gallagher Place	NA	Memphis	0
188	95	Noby	Elwel	nelwel2m@princeton.edu	605-117-8019	660 Pine View Park	NA	Sioux Falls	0
189	144	Cecil	Cyson	ccyson3z@php.net	217-372-1447	4 Blaine Alley	NA	Springfield	0
190	192	Belia	Duckels	bduckels5b@canalblog.com	469-146-4261	54444 Arkansas Terrace	NA	Plano	0
191	193	Donall	Kimbly	dkimbly5c@nymag.com	702-451-4726	NA	NA	Las Vegas	0
192	60	Ferne	Rosencwaig	frosencwaig1n@bloomberg.com	704-651-4326	189 Packers Circle	North Carolina	NA	0
193	76	Ebba	Laver	elaver23@usgs.gov	917-515-8153	7 Sommers Court	New York	NA	0

```sql
UPDATE Customer_Dim SET Cust_Zip = 11111
WHERE Cust_Zip = 0;
```

Results | Chart

	number of rows updated	number of multi-joined rows updated
1	30	0

```sql
SELECT * FROM CUSTOMER_DIM;
```

Results | Chart

	CUSTOMER_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE	CUST_ADDRESS	CUST_STATE	CUST_CITY	CUST_ZIP
184	94	Brucie	Napper	bnapper2l@theguardian.com	218-604-5228	NA	Minnesota	Duluth	11111
185	196	Giralda	Davers	gdavers5f@quantcast.com	312-559-6283	2 Graedel Park	Illinois	Chicago	11111
186	46	Laraine	Crepin	lcrepin19@sbwire.com	415-144-6926	993 Nova Road	NA	San Francisco	11111
187	68	Augustine	Hattiff	ahattiff1v@linkedin.com	901-880-1243	3362 Clyde Gallagher Place	NA	Memphis	11111
188	95	Noby	Elwel	nelwel2m@princeton.edu	605-117-8019	660 Pine View Park	NA	Sioux Falls	11111
189	144	Cecil	Cyson	ccyson3z@php.net	217-372-1447	4 Blaine Alley	NA	Springfield	11111
190	192	Belia	Duckels	bduckels5b@canalblog.com	469-146-4261	54444 Arkansas Terrace	NA	Plano	11111
191	193	Donall	Kimbly	dkimbly5c@nymag.com	702-451-4726	NA	NA	Las Vegas	11111
192	60	Ferne	Rosencwaig	frosencwaig1n@bloomberg.com	704-651-4326	189 Packers Circle	North Carolina	NA	11111
193	76	Ebba	Laver	elaver23@usgs.gov	917-515-8153	7 Sommers Court	New York	NA	11111
194	82	Dallis	Kristof	dkristof29@youku.com	972-189-7328	323 Florence Center	TX	NA	11111
195	109	Marni	Hearnah	mhearnah30@sourceforge.net	315-289-5823	14 Clove Street	New York	NA	11111
196	159	Stirling	Horsfield	shorsfield4e@myspace.com	918-968-2025	NA	Oklahoma	NA	11111
197	35	Hendrik	Hubery	hhuberyy@devhub.com	336-692-5772	66274 Kropf Place	NA	NA	11111
198	100	Tyler	Erasmus	terasmus2r@cbslocal.com	727-425-1723	NA	NA	NA	11111

However, I have updated null values with 0000 for customer zip column but it got reflected as only 0 in the column, so to make the column and table more consistency, I've updated it with the value 1111.

Order Detail Dimension Table:

No null values are found in this table.

```sql
-- -- Cleaning and Transforming Order Detail Dimension Tables --

-- Checking for missing values --
SELECT COUNT(Order_detail_id) AS Total_Count, COUNT(Order_id) AS Order_id_Count, COUNT(Product_Id) AS Product_id_Count
FROM ORDER_DETAIL_DIM;
```

Results | Chart

	TOTAL_COUNT	ORDER_ID_COUNT	PRODUCT_ID_COUNT
1	300	300	300

Product Dimension Table:

```sql
-- -- Cleaning and Transforming Product Dimension Tables --

-- Checking for missing values --
SELECT COUNT(PRODUCT_ID) AS Total_Count, COUNT(Brand) AS Brand_Count, COUNT(Name) AS Name_Count, COUNT(Description) AS Description_Count, COUNT(Price) AS Price_Count, COUNT(MSRP) AS Msrp_Count,
COUNT(Cost) AS Cost_Count
FROM PRODUCT_DIM;
```

Results | Chart

	TOTAL_COUNT	BRAND_COUNT	NAME_COUNT	DESCRIPTION_COUNT	PRICE_COUNT	MSRP_COUNT	COST_COUNT
1	20	20	20	20	20	20	20

Query Details
Query duration 102ms

```sql
-- Checking For duplicate Product names --
SELECT NAME, COUNT(NAME) AS Duplicates
FROM PRODUCT_DIM
GROUP BY NAME
HAVING COUNT(NAME) > 1;
```

Results | Chart

NAME	DUPLICATES

Above query shows that there no duplicates and null values in product table.

```sql
-- Maintaining Consistency in Product_Dim Table --
UPDATE Product_Dim
SET Description = 'Fast -  efficient -  wireless power'
WHERE Product_ID = 1;
```

Results | Chart

	number of rows updated	number of multi-joined rows updated
1	1	0

I've noticed an inconsistency in description column of product table, where a value was ending with a full stop, so I have updated the value to maintain consistency and data integrity

Order Facts Table:

```sql
-- Cleaning and Transforming Order Facts Tables --

-- Checking Null Values --
SELECT COUNT(Order_Id) AS Total_Count, COUNT(Order_Amt) AS Order_Amt_Count, COUNT(Order_Qty) AS Order_Qty_Count, COUNT(Customer_ID) AS Customer_Id_Count,
COUNT(Order_Date) AS Order_Date_Count
FROM ORDERS_FACT;
```

Results | Chart

	TOTAL_COUNT	ORDER_AMT_COUNT	ORDER_QTY_COUNT	CUSTOMER_ID_COUNT	ORDER_DATE_COUNT
1	250	250	250	250	250

Above query shows that there no null values in order facts table

```sql
-- Transforming Order_Date Format from Timestamp to Date for Comphresiveness --

ALTER TABLE ORDERS_FACT
ADD COLUMN New_Order_Date DATE;

UPDATE ORDERS_FACT
SET New_Order_Date = CAST(Order_Date AS DATE);

ALTER TABLE ORDERS_FACT
ADD COLUMN New_Order_Time STRING;

UPDATE ORDERS_FACT
SET New_Order_Time = TO_CHAR(ORDER_DATE, 'HH24:MI:SS');
```

Results | Chart

	number of rows updated	number of multi-joined rows updated
1	250	0

Results | Chart

	ORDER_ID	ORDER_AMT	ORDER_QTY	CUSTOMER_ID	ORDER_DATE	NEW_ORDER_DATE	NEW_ORDER_TIME
1	1	543.50	3	20	2022-07-14 15:43:22.000 +0000	2022-07-14	15:43:22
2	2	138.62	4	156	2022-09-25 08:25:31.000 +0000	2022-09-25	08:25:31
3	3	421.68	1	6	2023-07-22 02:54:07.000 +0000	2023-07-22	02:54:07
4	4	197.81	3	153	2022-03-20 00:00:00.000 -0700	2022-03-20	00:00:00
5	5	742.61	3	89	2023-10-05 17:16:24.000 +0000	2023-10-05	17:16:24
6	6	832.69	1	135	2022-03-10 22:21:33.000 +0000	2022-03-10	22:21:33
7	7	363.86	5	54	2023-04-19 01:44:14.000 +0000	2023-04-19	01:44:14
8	8	371.74	3	125	2022-04-19 04:37:36.000 +0000	2022-04-19	04:37:36
9	9	784.62	1	196	2023-02-01 15:28:41.000 +0000	2023-02-01	15:28:41
10	10	631.18	1	167	2022-12-04 13:41:21.000 +0000	2022-12-04	13:41:21
11	11	79.39	4	96	2022-02-16 08:14:18.000 +0000	2022-02-16	08:14:18

Using the snapshots above, I created two new columns, New order date and New order time, as part of a transformation of the Order_date column, which contains timestamp data. By extracting and separating the date and time values from Order_date, I populated these new columns with distinct date and time values. This approach enhances clarity and usability for downstream users, enabling easier analysis and a more intuitive understanding of order data without needing to manipulate timestamp information directly.

```sql
-- Minimum and Maximum Order Quantity and Order Date --
SELECT MIN(ORDER_QTY) AS Minimum_qty, MAX(ORDER_QTY) AS Maximum_qty, MIN(NEW_ORDER_DATE) AS Min_Date, MAX(NEW_ORDER_DATE) AS Max_Date
FROM ORDERS_FACT;
```

Results | Chart

	MINIMUM_QTY	MAXIMUM_QTY	MIN_DATE	MAX_DATE
1	1	6	2022-01-11	2023-12-30

```sql
SELECT MIN(ORDER_AMT) , MAX(ORDER_AMT) FROM ORDERS_FACT;
```

Results | Chart

	MIN(ORDER_AMT)	MAX(ORDER_AMT)
1	3.19	998.96

Using above code to check if there's any negative or zero values in order amount, order quantity and order date

```sql
-- Adding a new column as 'order year' to perform group by for downstream users--
ALTER TABLE ORDERS_FACT
ADD COLUMN Order_Year DECIMAL(20);

UPDATE ORDERS_FACT
SET Order_Year = YEAR(NEW_ORDER_DATE);
```

Results | Chart

	number of rows updated	number of multi-joined rows updated
1	250	0

Extracted the order year from the transformed timestamp column i.e. New order date

```sql
SELECT * FROM ORDERS_FACT;
```

Results | Chart

	ORDER_ID	ORDER_AMT	ORDER_QTY	CUSTOMER_ID	ORDER_DATE	NEW_ORDER_DATE	NEW_ORDER_TIME	ORDER_YEAR
1	1	543.50	3	20	2022-07-14 15:43:22.000 +0000	2022-07-14	15:43:22	2022
2	2	138.62	4	156	2022-09-25 08:25:31.000 +0000	2022-09-25	08:25:31	2022
3	3	421.68	1	6	2023-07-22 02:54:07.000 +0000	2023-07-22	02:54:07	2023
4	4	197.81	3	153	2022-03-20 00:00:00.000 -0700	2022-03-20	00:00:00	2022
5	5	742.61	3	89	2023-10-05 17:16:24.000 +0000	2023-10-05	17:16:24	2023
6	6	832.69	1	135	2022-03-10 22:21:33.000 +0000	2022-03-10	22:21:33	2022
7	7	363.86	5	54	2023-04-19 01:44:14.000 +0000	2023-04-19	01:44:14	2023
8	8	371.74	3	125	2022-04-19 04:37:36.000 +0000	2022-04-19	04:37:36	2022
9	9	784.62	1	196	2023-02-01 15:28:41.000 +0000	2023-02-01	15:28:41	2023
10	10	631.18	1	167	2022-12-04 13:41:21.000 +0000	2022-12-04	13:41:21	2022
11	11	79.39	4	96	2022-02-16 08:14:18.000 +0000	2022-02-16	08:14:18	2022
12	12	597.73	1	149	2023-02-07 22:56:40.000 +0000	2023-02-07	22:56:40	2023
13	13	115.31	2	157	2022-07-17 05:40:32.000 +0000	2022-07-17	05:40:32	2022
14	14	666.38	3	156	2022-04-12 02:57:26.000 +0000	2022-04-12	02:57:26	2022
15	15	930.54	4	94	2022-01-31 09:47:38.000 +0000	2022-01-31	09:47:38	2022
16	16	632.85	5	152	2022-05-18 09:06:20.000 +0000	2022-05-18	09:06:20	2022
17	17	510.10	6	148	2022-03-03 13:11:48.000 +0000	2022-03-03	13:11:48	2022
18	18	50.91	6	12	2023-04-20 14:56:23.000 +0000	2023-04-20	14:56:23	2023

Above queried data shows us the transformed order facts table.

4) Created a production schema

```sql
-- Creating a production schema--
CREATE OR REPLACE SCHEMA DE_PROJECT.Prod_Data;
```

Results | Chart

	status
1	Schema PROD_DATA successfully created.

5) Created Order, Order Detail, Product and Customer Table in the Snowflake Production Schema.

```sql
-- Creating Dimension and Fact Tables in production schema --

-- Creating Customer Dimension table --
CREATE OR REPLACE TABLE DE_PROJECT.PROD_DATA.PROD_CUSTOMER_DIM(
Customer_ID NUMBER(19,0) PRIMARY KEY,
First_Name VARCHAR(500),
Last_Name VARCHAR(500),
Email VARCHAR(500),
Phone VARCHAR(500),
Cust_Address VARCHAR(1000),
Cust_State VARCHAR(100),
Cust_City VARCHAR(100),
Cust_zip DECIMAL(10),
Updated_Date TIMESTAMP_TZ
);
```

Results | Chart

status
Table PROD_CUSTOMER_DIM successfully created.

```sql
-- Creating Dimension and Fact Tables in production schema --

-- Creating Product Dimension table --
CREATE OR REPLACE TABLE DE_PROJECT.PROD_DATA.PROD_PRODUCT_DIM(
Product_ID NUMBER(19,0) PRIMARY KEY,
Brand VARCHAR(200),
Name VARCHAR(200),
Description VARCHAR(1000),
Price DECIMAL(10,2),
MSRP DECIMAL(10,2),
Cost DECIMAL(10,2),
Updated_Datetime TIMESTAMP_TZ
);
```

Results | Chart

status
Table PROD_PRODUCT_DIM successfully created.

```sql
-- Creating Dimension and Fact Tables in production schema --

-- Creating Orders Fact table --
CREATE OR REPLACE TABLE DE_PROJECT.PROD_DATA.PROD_ORDERS_FACT(
Order_Id NUMBER(19,0) PRIMARY KEY,
Order_Amt DECIMAL(20,2),
Order_Qty DECIMAL(20),
Customer_Id NUMBER(19,0),
Order_Date DATE,
Order_Year DECIMAL(20),
Updated_Datetime TIMESTAMP_TZ,
FOREIGN KEY (Customer_Id) REFERENCES DE_PROJECT.PROD_DATA.PROD_CUSTOMER_DIM(Customer_Id)
);
```

Results | Chart

status
Table PROD_ORDERS_FACT successfully created.

```sql
-- Creating Dimension and Fact Tables in production schema --

-- Creating Orders detail dimension table --
CREATE OR REPLACE TABLE DE_PROJECT.PROD_DATA.PROD_ORDER_DETAIL_DIM(
Order_detail_Id NUMBER(19,0) PRIMARY KEY,
Order_Id NUMBER(19,0),
Product_Id NUMBER(19,0),
Updated_Datetime TIMESTAMP_TZ,
FOREIGN KEY (Order_Id) REFERENCES DE_PROJECT.PROD_DATA.PROD_ORDERS_FACT(Order_Id),
FOREIGN KEY (Product_Id) REFERENCES DE_PROJECT.PROD_DATA.PROD_PRODUCT_DIM(Product_Id)
);
```

Results | Chart

status
Table PROD_ORDER_DETAIL_DIM successfully created.

6) Inserted cleaned and transformed data from staging tables to the prod tables

```sql
-- Inserting records into Production schema from Stage schema --

-- Inserting into Production Customer table--
INSERT INTO DE_PROJECT.PROD_DATA.PROD_CUSTOMER_DIM
SELECT *, CURRENT_TIMESTAMP FROM STAGE_DATA.CUSTOMER_DIM;
```

Results | Chart

	number of rows inserted
1	200

```sql
-- Inserting records into Production schema from Stage schema --

-- Inserting into Production Products table--
INSERT INTO DE_PROJECT.PROD_DATA.PROD_PRODUCT_DIM
SELECT *, CURRENT_TIMESTAMP FROM STAGE_DATA.PRODUCT_DIM;
```

Results | Chart

	number of rows inserted
1	20

```sql
-- Inserting records into Production schema from Stage schema --

-- Inserting into Production Orders Fact table--
INSERT INTO DE_PROJECT.PROD_DATA.PROD_ORDERS_FACT
SELECT ORDER_ID, ORDER_AMT, ORDER_QTY, CUSTOMER_ID, NEW_ORDER_DATE, ORDER_YEAR, CURRENT_TIMESTAMP FROM STAGE_DATA.ORDERS_FACT;
```

Results | Chart

	number of rows inserted
1	250

```sql
-- Inserting records into Production schema from Stage schema --

-- Inserting into Production Order details table--
INSERT INTO DE_PROJECT.PROD_DATA.PROD_ORDER_DETAIL_DIM
SELECT * , CURRENT_TIMESTAMP FROM STAGE_DATA.ORDER_DETAIL_DIM;
```

Results | Chart

	number of rows inserted
1	300

7) Checked my production tables if all the data were loaded correctly or not

```
SELECT * FROM DE_PROJECT.PROD_DATA.PROD_CUSTOMER_DIM;
```

Results | Chart

	CUSTOMER_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE	CUST_ADDRESS	CUST_STATE	CUST_CITY	CUST_ZIP	UPDATED_DATE
1	1	Penelope	Keddie	pkeddie0@behance.net	205-151-7125	3 8th Crossing	Alabama	Birmingham	35244	2024-11-13 18:02:24.717 -0800
2	3	Jake	Gorger	jgorger2@springer.com	208-562-9998	68153 Maryland Center	Idaho	Boise	83757	2024-11-13 18:02:24.717 -0800
3	5	Angela	Nattriss	anattriss4@shinystat.com	513-792-4285	02 Maryland Hill	Ohio	Cincinnati	45249	2024-11-13 18:02:24.717 -0800
4	7	Lyman	Gascoyen	lgascoyen6@admin.ch	214-402-0630	045 Gale Street	TX	Dallas	75277	2024-11-13 18:02:24.717 -0800
5	8	Carlina	Shilliday	cshilliday7@shareasale.com	814-114-4016	708 Memorial Hill	Pennsylvania	Erie	16510	2024-11-13 18:02:24.717 -0800
6	9	Maribel	Guile	mguile8@techcrunch.com	808-103-4647	5421 Myrtle Lane	Hawaii	Honolulu	96820	2024-11-13 18:02:24.717 -0800
7	10	Della	Pisculli	dpisculli9@slashdot.org	530-256-6967	41 Jackson Pass	CA	South Lake Tahoe	96154	2024-11-13 18:02:24.717 -0800
8	11	Tatiana	Bonner	tbonnera@php.net	719-966-7996	99687 Hagan Road	Colorado	Colorado Springs	80920	2024-11-13 18:02:24.717 -0800
9	12	Nobe	Folli	nfollib@columbia.edu	405-131-6665	5 Roth Way	Oklahoma	Oklahoma City	73135	2024-11-13 18:02:24.717 -0800

```
SELECT * FROM DE_PROJECT.PROD_DATA.PROD_PRODUCT_DIM;
```

Results | Chart

	PRODUCT_ID	BRAND	NAME	DESCRIPTION	PRICE	MSRP	COST	UPDATED_DATETIME
1	1	TechWave	SpeedBoost Wireless Charging Pad	Fast - efficient - wireless power	39.65	44.41	14.67	2024-11-13 18:07:24.032 -0800
2	2	TechWave	PixelGrip Gaming Mouse	Precision control for gamers	26.55	29.74	9.82	2024-11-13 18:07:24.032 -0800
3	3	TechWave	SoundSync Bluetooth Earbuds	Seamless audio freedom anytime	69.99	78.39	25.90	2024-11-13 18:07:24.032 -0800
4	4	TechWave	UltraView HD Webcam	Crystal-clear video conferencing	32.42	36.31	12.00	2024-11-13 18:07:24.032 -0800
5	5	TechWave	TurboDrive External SSD	Lightning-fast storage on-the-go	110.99	124.31	41.07	2024-11-13 18:07:24.032 -0800
6	6	EcoEra	GreenGrow Indoor Herb Garden	Sustainable - fresh home gardening	209.99	235.19	77.70	2024-11-13 18:07:24.032 -0800
7	7	EcoEra	PureFlow Bamboo Water Bottle	Eco-friendly hydration companion	29.99	33.59	11.10	2024-11-13 18:07:24.032 -0800
8	8	EcoEra	EarthGuard Plant-Based Cleaner	Natural - powerful cleaning solution	17.98	20.14	6.65	2024-11-13 18:07:24.032 -0800
9	9	EcoEra	EcoCycle Recycled Laptop Sleeve	Protect your tech sustainably	45.68	51.16	16.90	2024-11-13 18:07:24.032 -0800

```
SELECT * FROM DE_PROJECT.PROD_DATA.PROD_ORDERS_FACT;
```

Results | Chart

	ORDER_ID	ORDER_AMT	ORDER_QTY	CUSTOMER_ID	ORDER_DATE	ORDER_YEAR	UPDATED_DATETIME
1	1	543.50	3	20	2022-07-14	2022	2024-11-13 18:15:11.724 -0800
2	2	138.62	4	156	2022-09-25	2022	2024-11-13 18:15:11.724 -0800
3	3	421.68	1	6	2023-07-22	2023	2024-11-13 18:15:11.724 -0800
4	4	197.81	3	153	2022-03-20	2022	2024-11-13 18:15:11.724 -0800
5	5	742.61	3	89	2023-10-05	2023	2024-11-13 18:15:11.724 -0800
6	6	832.69	1	135	2022-03-10	2022	2024-11-13 18:15:11.724 -0800
7	7	363.86	5	54	2023-04-19	2023	2024-11-13 18:15:11.724 -0800
8	8	371.74	3	125	2022-04-19	2022	2024-11-13 18:15:11.724 -0800
9	9	784.62	1	196	2023-02-01	2023	2024-11-13 18:15:11.724 -0800
10	10	631.18	1	167	2022-12-04	2022	2024-11-13 18:15:11.724 -0800

```
SELECT * FROM DE_PROJECT.PROD_DATA.PROD_ORDER_DETAIL_DIM;
```

Results | Chart

	ORDER_DETAIL_ID	ORDER_ID	PRODUCT_ID	UPDATED_DATETIME
1	1	213	16	2024-11-13 18:20:05.275 -0800
2	2	60	6	2024-11-13 18:20:05.275 -0800
3	3	188	12	2024-11-13 18:20:05.275 -0800
4	4	229	16	2024-11-13 18:20:05.275 -0800
5	5	121	12	2024-11-13 18:20:05.275 -0800
6	6	115	13	2024-11-13 18:20:05.275 -0800
7	7	39	3	2024-11-13 18:20:05.275 -0800
8	8	127	1	2024-11-13 18:20:05.275 -0800
9	9	115	7	2024-11-13 18:20:05.275 -0800
10	10	159	20	2024-11-13 18:20:05.275 -0800
11	11	50	13	2024-11-13 18:20:05.275 -0800
12	12	53	3	2024-11-13 18:20:05.275 -0800
13	13	118	16	2024-11-13 18:20:05.275 -0800
14	14	214	1	2024-11-13 18:20:05.275 -0800
15	15	172	6	2024-11-13 18:20:05.275 -0800
16	16	161	11	2024-11-13 18:20:05.275 -0800

The returned results confirm that all cleaned and transformed data from the staging tables have been accurately loaded into the production tables. This successful migration ensures that the data is now fully prepared and optimized for downstream users, who can leverage the reliable, production-ready dataset for analysis, reporting, and decision-making. The structured and accessible data in the production tables will streamline workflows and support seamless data integration across various business functions.

8) Successfully connected and configured between Snowflake and Tableau to see the Database, schemas and Tables.