Data Ware House Project

Name: Hassan Ashfaq

Roll No: 19I-1708

Section: B

Project Overview:

We have to Building and Analyzing Near Real-time Data Warehouse Prototype for METRO. To mimic the near real-time Data Warehouse using 10,000 Transaction from METRO Against 100 products present in the Master Data. In Our Database we need Normalize data, but in case of Data Warehouse we need de-normalize data. So we can Analyze our any type of business.

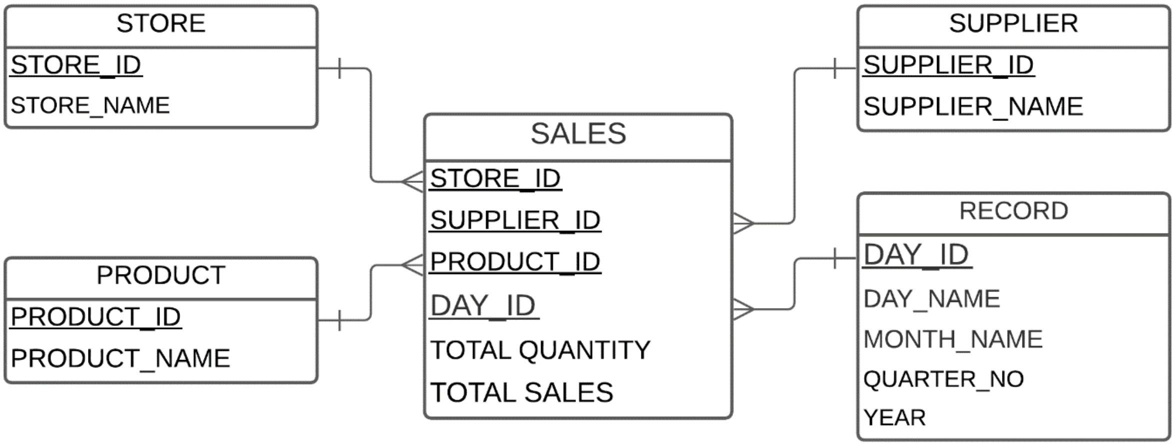
The First Step to design a Data Warehouse is to decide our Subject Areas etc. Like Fact Table, Dimensions & Level of granularity. In this Project, Our Dimensions are

- STORE
- SUPPLIER
- RECORD
- PRODUCT

& Fact Table is SALES.

In Our Data Warehouse Prototype, we have no level of granularity in each Dimension. Data in Data Warehouse is Summarized on the basics of Days. Like Total Quantity Sold & Total Sales in each day, against each product, each supplier & each store.

Schema for DWH



As we can see from the Schema of DWH, we have 4 dimensions. The Total Quantity & Total Sales will be store against each product from different

Suppliers at different Store across Pakistan/Islamabad for each day. Our data is no doubt more summarized than Simple Normalize Database.

MESHJOIN Algorithm

The MESHJOIN (Mesh Join) algorithm has been introduced by Polyzotis in 2008 with objective of implementing the join operation in the transformation phase of ETL.

As in MESHJOIN we require a Queue in which we load data in chunks from database and store only Product id in Queues & remaining data in Multi Hash Map Against each unique Product id. In my Algorithm, I am not just saving Product id in Queue but also transactional id. Because when we have join to all Master Data against each Queue Element than we have to send that Queue Top Data to Data Warehouse after deleting that data from Multi Hash Map. The transactional id which I store earlier come handy to delete data from Multi Hash Map.

Algorithm

- ➢ Loop till <201 Time: (200*50 Batch Size = 10,000)
    - o DB Result to store 50 Data points from transactions Data Table
    - o Store DB Result in Multi Hash Map
    - o Store (Product id, Transaction id) for each data point in Queue
    - o Master Data to store 10 Data points from Master Data Table
    - o Join Master Data & Queue with each other & Store Data in Hash Map
    - o If Queue.Siz() ==10:
        - ▪ Pop 1 partition contain 50 Data points
        - ▪ Send Pop Data to Data Ware House
    - o Loop until 200 iterations

Three Shortcomings in Mesh Join

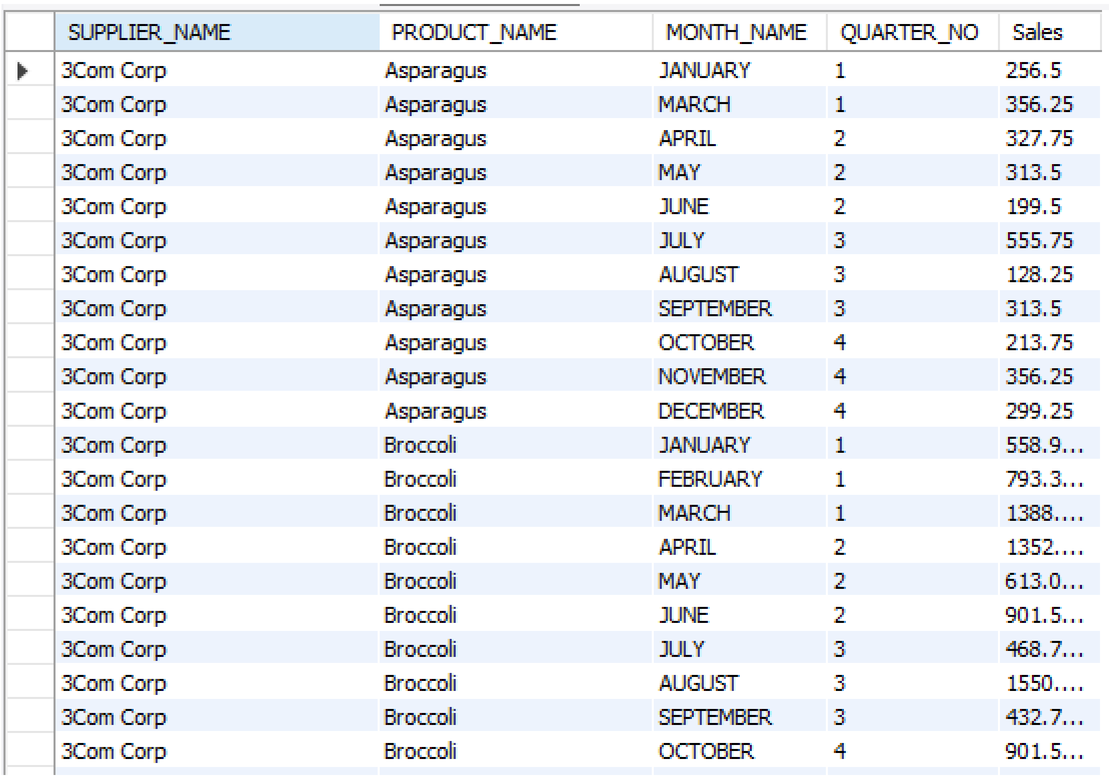
- In Mesh join, there is a dependency that partitions size for both Stream Data & Master Data present in Disk, must be same. This hinder the optimal distribution of memory among join components.
- In real world, the sales can found a skewed distribution. Like 20% of the products generate 80% of revenue & remaining 80% products generate 20% revenue in total. So, that mean 20% of Master Data are use more often used than 80%. But in Mesh Join Load Master Data in Infinite Loop, that is not a good approach as 20% must be reload after every partitions.
- If Master Size is more than Transactional Data, then Mesh Join will be affected. Because Each tuple of Transactional Data has to compare with all Master Data to find the Suitable tuple from Master Table. This will affect the Performance of Mesh Join.

Reference

- R-MESHJOIN for Near-Real-Time Data Warehousing
- X-HYBRIDJOIN for Near-Real-Time Data Warehousing

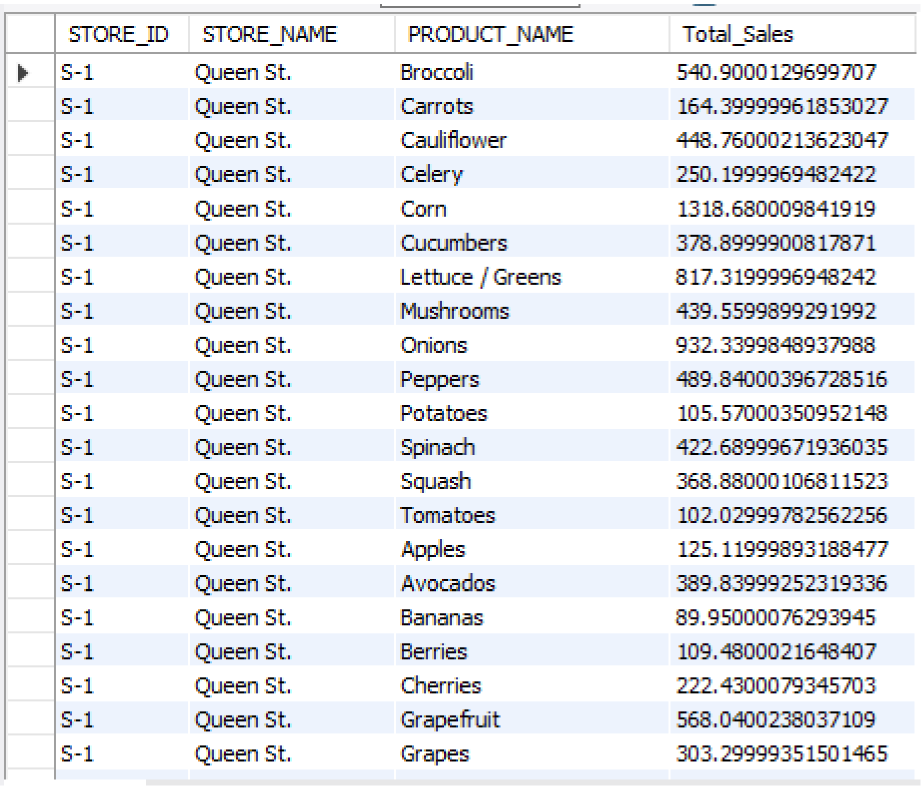
Query Results:

1. Present total sales of all products supplied by each supplier with respect to quarter and month.



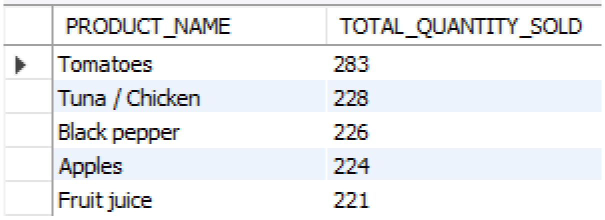
SUPPLIER_NAME	PRODUCT_NAME	MONTH_NAME	QUARTER_NO	Sales
3Com Corp	Asparagus	JANUARY	1	256.5
3Com Corp	Asparagus	MARCH	1	356.25
3Com Corp	Asparagus	APRIL	2	327.75
3Com Corp	Asparagus	MAY	2	313.5
3Com Corp	Asparagus	JUNE	2	199.5
3Com Corp	Asparagus	JULY	3	555.75
3Com Corp	Asparagus	AUGUST	3	128.25
3Com Corp	Asparagus	SEPTEMBER	3	313.5
3Com Corp	Asparagus	OCTOBER	4	213.75
3Com Corp	Asparagus	NOVEMBER	4	356.25
3Com Corp	Asparagus	DECEMBER	4	299.25
3Com Corp	Broccoli	JANUARY	1	558.9...
3Com Corp	Broccoli	FEBRUARY	1	793.3...
3Com Corp	Broccoli	MARCH	1	1388....
3Com Corp	Broccoli	APRIL	2	1352....
3Com Corp	Broccoli	MAY	2	613.0...
3Com Corp	Broccoli	JUNE	2	901.5...
3Com Corp	Broccoli	JULY	3	468.7...
3Com Corp	Broccoli	AUGUST	3	1550....
3Com Corp	Broccoli	SEPTEMBER	3	432.7...
3Com Corp	Broccoli	OCTOBER	4	901.5...

2. Present total sales of each product sold by each store. The output should be organized store wise and then product wise under each store.



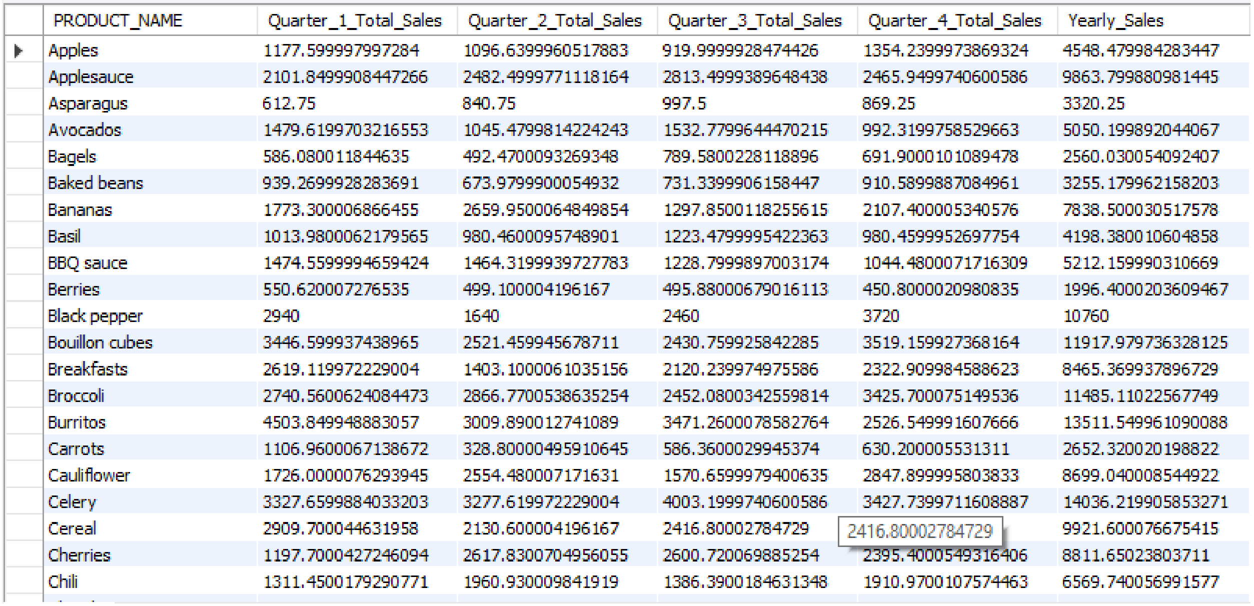
STORE_ID	STORE_NAME	PRODUCT_NAME	Total_Sales
S-1	Queen St.	Broccoli	540.9000129699707
S-1	Queen St.	Carrots	164.39999961853027
S-1	Queen St.	Cauliflower	448.76000213623047
S-1	Queen St.	Celery	250.1999969482422
S-1	Queen St.	Corn	1318.680009841919
S-1	Queen St.	Cucumbers	378.8999900817871
S-1	Queen St.	Lettuce / Greens	817.3199996948242
S-1	Queen St.	Mushrooms	439.5599899291992
S-1	Queen St.	Onions	932.3399848937988
S-1	Queen St.	Peppers	489.84000396728516
S-1	Queen St.	Potatoes	105.57000350952148
S-1	Queen St.	Spinach	422.68999671936035
S-1	Queen St.	Squash	368.88000106811523
S-1	Queen St.	Tomatoes	102.02999782562256
S-1	Queen St.	Apples	125.11999893188477
S-1	Queen St.	Avocados	389.83999252319336
S-1	Queen St.	Bananas	89.95000076293945
S-1	Queen St.	Berries	109.4800021648407
S-1	Queen St.	Cherries	222.4300079345703
S-1	Queen St.	Grapefruit	568.0400238037109
S-1	Queen St.	Grapes	303.29999351501465

3. Find the 5 most popular products sold over the weekends.



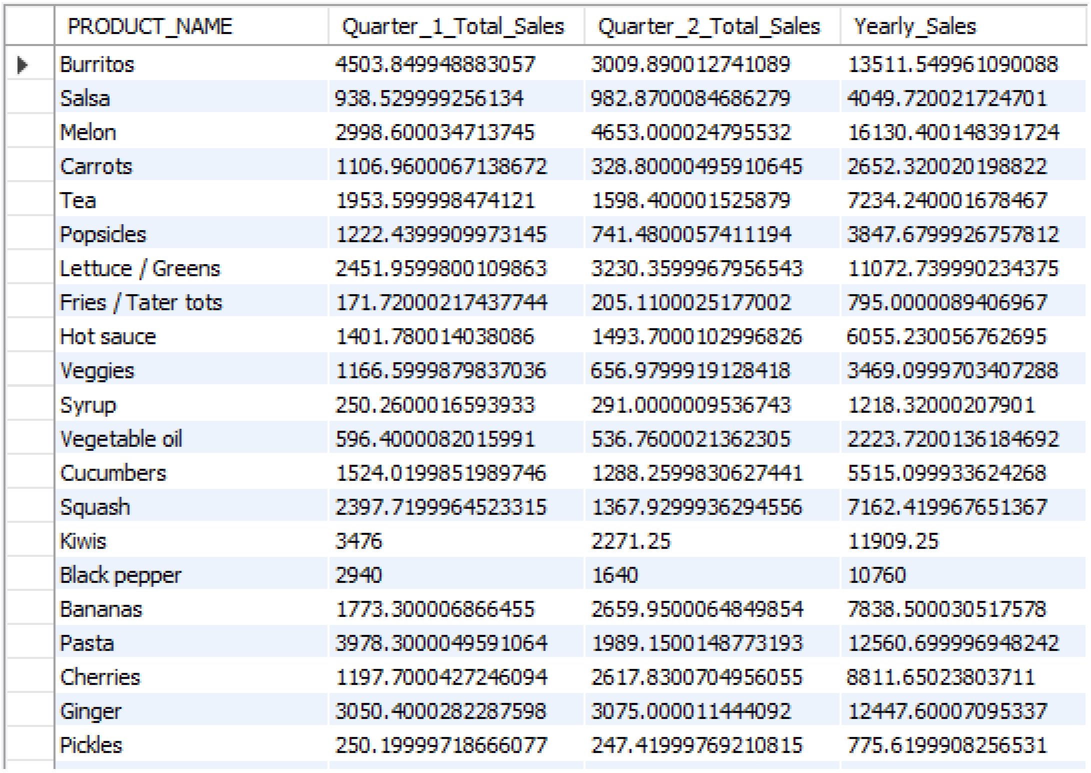
PRODUCT_NAME	TOTAL_QUANTITY_SOLD
Tomatoes	283
Tuna / Chicken	228
Black pepper	226
Apples	224
Fruit juice	221

4. Present the quarterly sales of each product for year 2016 using drill down query concept. Note: each quarter sale must be a column.



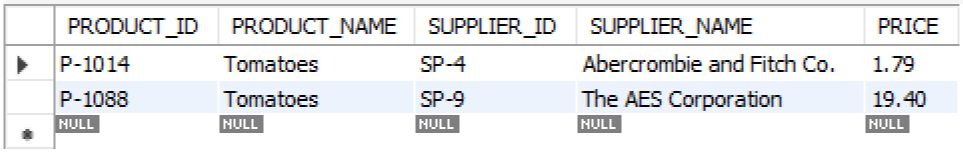
	PRODUCT_NAME	Quarter_1_Total_Sales	Quarter_2_Total_Sales	Quarter_3_Total_Sales	Quarter_4_Total_Sales	Yearly_Sales
▶	Apples	1177.599997997284	1096.6399960517883	919.9999928474426	1354.2399973869324	4548.479984283447
	Applesauce	2101.8499908447266	2482.4999771118164	2813.4999389648438	2465.9499740600586	9863.799880981445
	Asparagus	612.75	840.75	997.5	869.25	3320.25
	Avocados	1479.6199703216553	1045.4799814224243	1532.7799644470215	992.3199758529663	5050.199892044067
	Bagels	586.080011844635	492.4700093269348	789.5800228118896	691.9000101089478	2560.030054092407
	Baked beans	939.2699928283691	673.9799900054932	731.3399906158447	910.5899887084961	3255.179962158203
	Bananas	1773.300006866455	2659.9500064849854	1297.8500118255615	2107.400005340576	7838.500030517578
	Basil	1013.9800062179565	980.4600095748901	1223.4799995422363	980.4599952697754	4198.380010604858
	BBQ sauce	1474.5599994659424	1464.3199939727783	1228.7999897003174	1044.4800071716309	5212.159990310669
	Berries	550.620007276535	499.100004196167	495.88000679016113	450.8000020980835	1996.4000203609467
	Black pepper	2940	1640	2460	3720	10760
	Bouillon cubes	3446.599937438965	2521.459945678711	2430.759925842285	3519.159927368164	11917.979736328125
	Breakfasts	2619.119972229004	1403.1000061035156	2120.239974975586	2322.909984588623	8465.369937896729
	Broccoli	2740.5600624084473	2866.7700538635254	2452.0800342559814	3425.700075149536	11485.1102256749
	Burritos	4503.849948883057	3009.890012741089	3471.2600078582764	2526.549991607666	13511.549961090088
	Carrots	1106.9600067138672	328.80000495910645	586.3600029945374	630.200005531311	2652.320020198822
	Cauliflower	1726.0000076293945	2554.480007171631	1570.6599979400635	2847.899995803833	8699.040008544922
	Celery	3327.6599884033203	3277.619972229004	4003.1999740600586	3427.7399711608887	14036.219905853271
	Cereal	2909.700044631958	2130.600004196167	2416.80002784729	2416.80002784729	9921.600076675415
	Cherries	1197.7000427246094	2617.8300704956055	2600.720069885254	2395.4000549316406	8811.65023803711
	Chili	1311.4500179290771	1960.930009841919	1386.3900184631348	1910.9700107574463	6569.740056991577

5. Extract total sales of each product for the first and second half of year 2016 along with its total yearly sales.



	PRODUCT_NAME	Quarter_1_Total_Sales	Quarter_2_Total_Sales	Yearly_Sales
▶	Burritos	4503.849948883057	3009.890012741089	13511.549961090088
	Salsa	938.529999256134	982.8700084686279	4049.720021724701
	Melon	2998.600034713745	4653.000024795532	16130.400148391724
	Carrots	1106.9600067138672	328.80000495910645	2652.320020198822
	Tea	1953.599998474121	1598.400001525879	7234.240001678467
	Popsicles	1222.4399909973145	741.4800057411194	3847.6799926757812
	Lettuce / Greens	2451.9599800109863	3230.3599967956543	11072.739990234375
	Fries / Tater tots	171.72000217437744	205.1100025177002	795.0000089406967
	Hot sauce	1401.780014038086	1493.7000102996826	6055.230056762695
	Veggies	1166.5999879837036	656.9799919128418	3469.0999703407288
	Syrup	250.2600016593933	291.0000009536743	1218.32000207901
	Vegetable oil	596.4000082015991	536.7600021362305	2223.7200136184692
	Cucumbers	1524.0199851989746	1288.2599830627441	5515.099933624268
	Squash	2397.7199964523315	1367.9299936294556	7162.419967651367
	Kiwis	3476	2271.25	11909.25
	Black pepper	2940	1640	10760
	Bananas	1773.300006866455	2659.9500064849854	7838.500030517578
	Pasta	3978.3000049591064	1989.1500148773193	12560.699996948242
	Cherries	1197.7000427246094	2617.8300704956055	8811.65023803711
	Ginger	3050.4000282287598	3075.000011444092	12447.60007095337
	Pickles	250.19999718666077	247.41999769210815	775.6199908256531

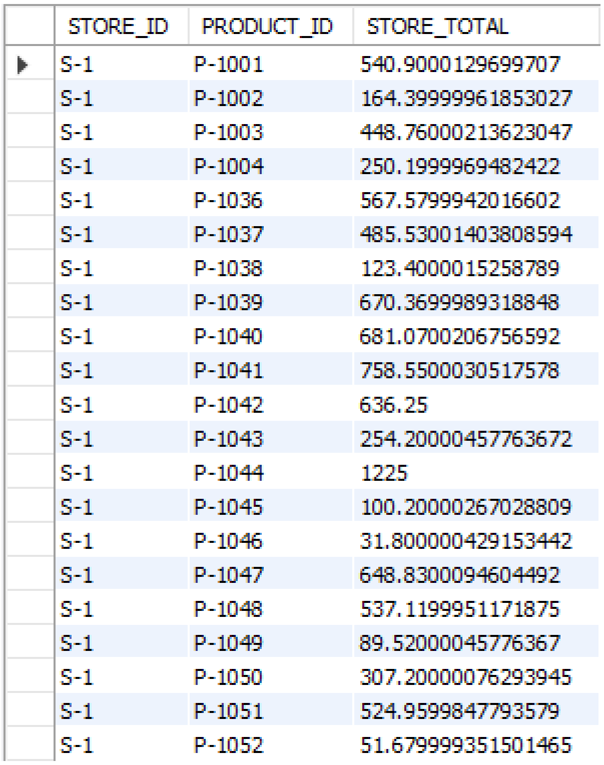
6. Find an anomaly in the data warehouse dataset. write a query to show the anomaly and explain the anomaly in your project report.



PRODUCT_ID	PRODUCT_NAME	SUPPLIER_ID	SUPPLIER_NAME	PRICE
P-1014	Tomatoes	SP-4	Abercrombie and Fitch Co.	1.79
P-1088	Tomatoes	SP-9	The AES Corporation	19.40
NULL	NULL	NULL	NULL	NULL

The only Anomaly that I find is in master data table, because same product distributed by two different Supplier with a very high difference in price.

7. Create a materialized view with name “STOREANALYSIS_MV” that presents the product-wise sales analysis for each store.



STORE_ID	PRODUCT_ID	STORE_TOTAL
S-1	P-1001	540.9000129699707
S-1	P-1002	164.39999961853027
S-1	P-1003	448.76000213623047
S-1	P-1004	250.1999969482422
S-1	P-1036	567.5799942016602
S-1	P-1037	485.53001403808594
S-1	P-1038	123.4000015258789
S-1	P-1039	670.3699989318848
S-1	P-1040	681.0700206756592
S-1	P-1041	758.5500030517578
S-1	P-1042	636.25
S-1	P-1043	254.20000457763672
S-1	P-1044	1225
S-1	P-1045	100.20000267028809
S-1	P-1046	31.800000429153442
S-1	P-1047	648.8300094604492
S-1	P-1048	537.1199951171875
S-1	P-1049	89.52000045776367
S-1	P-1050	307.20000076293945
S-1	P-1051	524.9599847793579
S-1	P-1052	51.679999351501465

What did you learn from the project?

I learned from this is that

- That how can we de-normalize data for Data Warehouse to make it summarize to make decisions & to apply Data Mining Algorithm's to analyze trends.
- We use our Customer etc. data in order to make more feasible decision, which is only possible with we knowledge of Data Warehouse.
- As I am from BS (AI), Now I understand to potential of data. By using both AI/ML & DWH knowledge I can make data driven application, which can help us to make a sound decision in any field of AI, which is the core component of AI in coming years.