Data Warehouse
Project Report

Submitted By:

Ali Kamal

19I-1865

BSDS(N)

Table of Contents

Introduction

The main purpose of this document is to report on the Data Warehouse project undertaken by me for the Date Warehouse and Business Intelligence semester project.

Project Overview

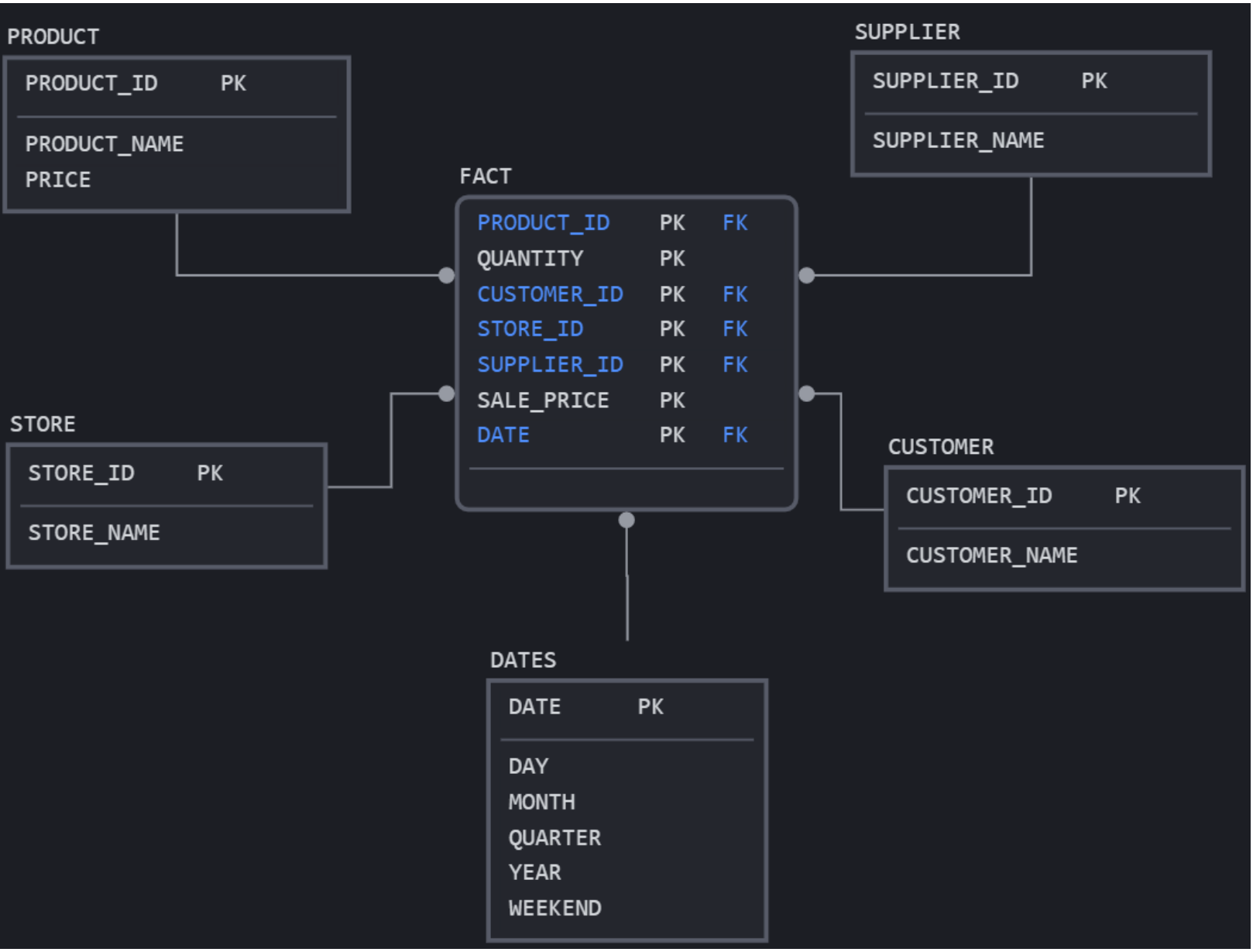
The project can be divided into three stages, which are basically the steps of ETL:

1. **Extracting** the data from the databases into Java
2. Performing relevant **transformations** on the loaded data from the databases in Java
3. **Loading** the transformed data into the Data Warehouse Star Schema

These ETL steps were performed with the help of the MESHJOIN algorithm.

Schema for Data Warehouse

After analyzing the databases provided, I opted to go for a simple Star Schema for my Data Warehouse design, as it seemed like the best fit for the current data. I did not see any advantage in going for other schemas (Snowflake etc.). I created separate dimension tables for “Product”, “Supplier”, “Store”, “Customer”, and “Dates”, with the Fact table having the “Sale Price” as its own attribute.


PRODUCT
PRODUCT_ID PK
PRODUCT_NAME
PRICE
SUPPLIER
SUPPLIER_ID PK
SUPPLIER_NAME
FACT
PRODUCT_ID PK FK
QUANTITY PK
CUSTOMER_ID PK FK
STORE_ID PK FK
SUPPLIER_ID PK FK
SALE_PRICE PK
DATE PK FK
STORE
STORE_ID PK
STORE_NAME
CUSTOMER
CUSTOMER_ID PK
CUSTOMER_NAME
DATES
DATE PK
DAY
MONTH
QUARTER
YEAR
WEEKEND

MESHJOIN algorithm

The basic crux of my MESHJOIN algorithm is as follows. I read and store the master data according to the partition size (10 partitions of size 10). I then read the transactional data (in blocks of 50) into a multi-hash map. For joining the two tables, I basically iterate over the current partition of master data, and get its "PRODUCT_ID". I then get that same "PRODUCT_ID" from the multi-hash map, where I have stored the transactional data. If there exists any data with same "PRODUCT_ID" (i-e. joinable data), it will be inserted into the Data Warehouse into their respective dimension tables (E.g. "Product ID" and "Product Name" into the "Product" dimension table etc.), and the fact table. In the fact table, the "Sale Price" attribute is calculated using "Quantity" from transactional data, and "Price" from master data. A queue is maintained in order keep track of the partition of transactional data that has 'seen' all the blocks of master data. Once the size of the queue becomes 'full', the head of the queue is de-queued (i-e the oldest of the partitions containing 50 transactional data records gets de-queued). Since our queue size and number of partitions of master data is the same (10), once the queue becomes full, we are basically ensured that the oldest partition in the queue (present at the head of the queue), has 'seen' all the partitions of the master data, and hence all records from that partition have been joined.

Shortcomings of Mesh Join

1. Does not make full efficient use of memory, as it still keeps all the joined records in memory.
2. As it still keeps all the joined records in memory, searching speed of algorithm is not the fastest it can be.
3. Requires external dependency (I had to externally import Apache's MultiHashMap), so the algorithm is not very portable.

What I learned from the project

I learned quite a lot from this project, honestly speaking. Firstly, I learnt how to use Java, which I had never used, and hence did not know. Learning Java was a pleasant experience, and not as difficult as I initially thought it to be. I also learnt how to implement mesh join, and how to overcome the various challenged that came with it (coding in an unfamiliar language, learning new concepts etc.). Similarly, I grew familiar with the practical aspects of Data Warehousing, and how to move Database(s) to a Data Warehouse. I learnt the basics of ETL, which will hopefully help

me in my professional career. After implementing the OLAP queries, I understood why the need for a Data Warehouse, and how it can easily handle queries related to particular subjects (e.g. quarterly analysis etc.).

OLAP queries results

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

SUPPLIER_ID	Quarter 1	Quarter 2	Quarter 3	Quarter 4	January	February	March	April	May	June	July	August	September	October	November	December
SP-1	630	557	564	668	226	170	234	172	186	199	202	181	181	210	257	201
SP-10	531	495	569	545	177	181	173	164	140	191	174	231	164	180	206	159
SP-11	396	383	457	342	94	152	150	152	95	136	162	164	131	115	133	94
SP-12	800	630	715	730	195	298	307	200	185	245	247	198	270	263	275	192
SP-13	634	675	711	581	226	179	229	227	231	217	197	271	243	235	167	179
SP-14	459	580	502	590	144	155	160	220	165	195	148	131	223	144	230	216
SP-15	471	682	553	628	122	173	176	218	265	199	160	216	177	162	232	234
SP-16	740	699	707	791	210	271	259	247	242	210	229	268	210	193	297	301
SP-17	639	593	581	501	207	203	229	220	178	195	233	177	171	205	129	167
SP-18	6 1548	509	589	491	157	228	232	101	177	231	182	201	206	146	132	213
SP-19	939	896	733	787	363	321	255	235	303	358	265	222	246	223	258	306
SP-2	632	634	670	612	187	220	225	229	204	201	227	265	178	237	192	183
SP-20	781	807	699	824	237	286	258	281	267	259	251	217	231	288	275	261
SP-3	1516	1548	1565	1719	543	452	521	545	504	499	459	529	577	598	616	505
SP-4	622	507	502	538	259	195	168	154	196	157	196	121	185	182	154	202
SP-5	513	614	547	542	223	124	166	224	217	173	210	150	187	191	187	164
SP-6	533	565	586	598	191	174	168	158	224	183	225	154	207	195	173	230
SP-7	541	555	535	505	227	127	187	113	234	208	190	158	187	171	156	178
SP-8	465	649	608	516	175	143	147	216	265	168	229	184	195	165	129	222
SP-9	1469	1125	1332	1167	511	503	455	353	464	308	431	403	498	431	391	345

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 Name	Product Name	Total Sales
Albany	Apples	79
Albany	Applesauce	102
Albany	Asparagus	37
Albany	Avocados	81
Albany	Bagels	99
Albany	Baked beans	62
Albany	Bananas	53
Albany	Basil	45
Albany	BBQ sauce	66
Albany	Berries	79
Albany	Black pepper	65
Albany	Bouillon cubes	73
Albany	Breakfasts	79
Albany	Broccoli	95
Albany	Burritos	58
Albany	Carrots	84
Albany	Cauliflower	39
Albany	Celery	65
Albany	Cereal	98
Albany	Cherries	40

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

PRODUCT_NAME	Amount 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	Quarter 2	Quarter 3	Quarter 4
Carrots	202	60	107	115
Cucumbers	181	153	176	145
Lettuce / Greens	126	166	107	170
Squash	156	89	100	121
Bananas	138	207	101	164
Cherries	70	153	152	140
Kiwis	176	115	120	192
Melon	116	180	188	140
Burritos	205	137	158	115
Popsicles	183	111	153	129
Fries / Tater tots	108	129	100	163
Hot sauce	122	130	127	148
Salsa	127	133	147	141
Syrup	129	150	182	167
Pasta	178	89	164	131
Tea	165	135	147	164
Vegetable oil	140	126	114	142
Veggies	190	107	154	114
Black pepper	147	82	123	186
Ginger	124	125	146	111

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

PRODUCT_NAME	First Half of Year	Second Half of Year	Total Yearly Sale
Carrots	262	222	484
Cucumbers	334	321	655
Lettuce / Greens	292	277	569
Squash	245	221	466
Bananas	345	265	610
Cherries	223	292	515
Kiwis	291	312	603
Melon	296	328	624
Burritos	342	273	615
Popsicles	294	282	576

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

Tomatoes has two suppliers, with starkly different prices (1.79 and 19.40), which are not feasibly possible, so this is an anomaly.

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-9	P-1000	527.25
S-8	P-1000	228.00
S-7	P-1000	441.75
S-6	P-1000	470.25
S-5	P-1000	256.50
S-4	P-1000	57.00
S-3	P-1000	698.25
S-2	P-1000	527.25
S-10	P-1000	114.00
S-9	P-1001	1568.61