**Project Report**

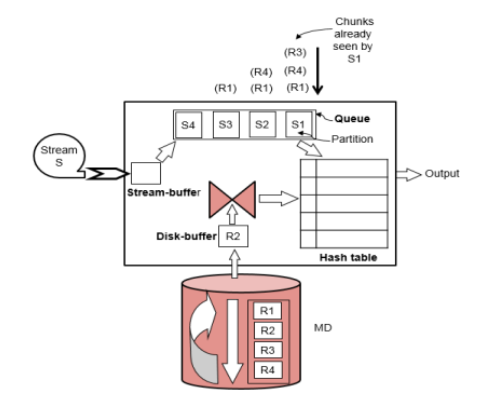
**PROJECT OVERVIEW:**

In this project, I have implemented Data Warehouse of Metro superstore so I can perform any type of anaylsis on products, customers ,suppliers and sepecially from sales point of view of metro superstore. e.g I can analyze results with respect to different dimensions sales w.r.t (products,time,supplier,stores and many more) like show total sales of all products supplied by each supplier with respect to quarter and month or Find the 5 most popular products sold over the weekends or Extract total sales of each product for the first and second half of year 2017 along with its total yearly sales or Determine the top 3 store names who generated highest sales in September, 2017 and many many more anaylsis I can perform onmy sales from w.r.t different dimensions mentioned above.

I was given data which compromises of 2 parts Transactions & Masterdata. In transaction Part the data coming from every transaction contains transaction\_id, product\_id, customer\_id, store\_id, store\_name, time\_id, T\_date, quantity. While from master data have 2 tables one is coutomer table which have customer\_id, customer\_name and sceond tale is Product table which have product\_id, product\_name, supplier\_id, supplier\_name, price. After that I implemented ETL phase In, ETL phase to transform data I used algorithm called meshjoin algorithm. Mesh join is basically design to tackel with near real time data coming from transaction at real time. Then when I transform data I sent it to warehouse which have star schema of metro store for analysis purpose and then from there I used to perform different type of anaylsis as mentiond in above paragraph also that’s my project overview now below I describe the complete implementation of mesh join algorithm I used in ETL phase.

**MESHJOIN OVERVIEW:**

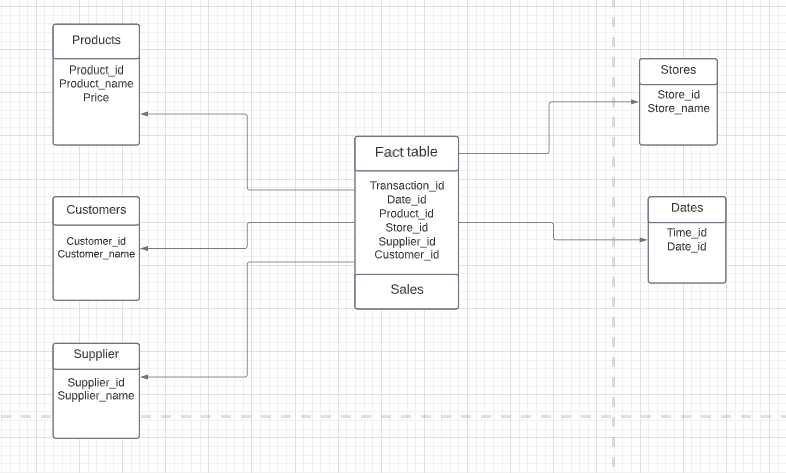
Mesh join is basically design to tackel with near real time data coming from transaction at real time. I used this algorithm in my ETL phase of project to transform data.I have implemented meshjoin in Java using Queue and MultiHashTable. Queue is divided into 10 partitions (equivalent to Masterdata Partitions). In queue, Transaction\_id and Product\_id is stored. In MultiHashTable product\_id is the key while whole tuple are its values. How it actually looks like I insert a picture below to show the view of its working.



Now I think It is easy to understand from looking upto the working view of mesh join that how it work. Now how I implement it like as transaction occur it come to stream buffer and I set condition when stream buffer have 50 transactions I took those 50 transaction data from stream buffer and send product\_id and transaction\_id to queue and in hashtable I send product\_id as key and the whole rest of tuple which came from tranctional data as values of that key. Then I read 50 tuples from master data table into disk buffer and match Product\_id which came from master data into disk buffer with the hashtable key if the key and Product\_id match then I insert remaining master data table into hashtable and caluclate sales by using both master data and transaction data(multiplying both price came from master data and quantity from trasnacation data for each tarncation)if not match then next 10 tuples from master\_data table load into disk buffer and again match this working done in loop until all the tuples in chunks of 10 from master\_data load into disk\_buffer one by one and match with hashtable data . During this whole process on the other hand transaction data in tuples of 50 also keep loading into stream buffer and when size limt of stream buffer equals to 50 then it loads into queue which maintains the count that either all the data which came from stream buffer into queue and also load in hashtable matched with all the data loads into dsik buffer from master data table. Once queue size equals to limit we set for our queue then it means that all the data have been checked for the transaction tuple which loads first w.r.t other 9 tuples load into queue then that one tuple from queue as well as from hashtable removed and send that tuple from hashtable into warehouse which in my case I send that one data from hashtable into my dbms warehouse star-schema which I named warehouse where I have created table by using one of the known warehouse schema called star-schema

And then by using those tables data which came from my meshjoin I apply queries and anaylze sales of metro store where I have calculated sales with respect to every dimension.

**Star Schema:**



**Queries:**

Query 1 --

(select f.store\_id,s.store\_name, sum(f.sale) as Sale\_Sept\_2017 from facts f,stores s where

f.store\_id=s.store\_id and

monthname(f.date\_id)="September" and year(f.date\_id)="2017"

group by f.store\_id

order by sum(f.sale)desc

limit 3 );

-- Query 2 --

select s.SUPPLIER\_ID, SUM(sa.SALE) as Total\_sale

from suppliers s, facts sa,dates d

where weekday(d.date\_id)=5 OR weekday(d.date\_id)=6 and sa.SUPPLIER\_ID = s.SUPPLIER\_ID

group by SUPPLIER\_ID

order by sale limit 10;

For next weekend just set weekday(d.date\_id)=12 and 13 in where condition.

-- Query3(a)

select store\_id,supplier\_id,quarter(date\_id),year(date\_id),SUM(SALE)AS "SALE"

from facts

GROUP BY SUPPLIER\_ID,STORE\_ID,quarter(date\_id),year(date\_id)

order by store\_id,supplier\_id,quarter(date\_id),year(date\_id);

#Query3(b)

select store\_id,supplier\_id,monthname(date\_id),year(date\_id),SUM(SALE)AS "SALE"

from facts

GROUP BY SUPPLIER\_ID,STORE\_ID,month(date\_id),year(date\_id)

order by store\_id,supplier\_id,month(date\_id),year(date\_id);

Query 4 --

SELECT STORE\_ID,PRODUCT\_ID,SUM(SALE)

FROM FACTS

GROUP BY STORE\_ID,PRODUCT\_ID

ORDER BY STORE\_ID,PRODUCT\_ID;

Query--5

SELECT SUM(CASE WHEN QUARTER(d.DATE\_ID)=1 THEN SALE END) AS Quarter1,

SUM(CASE WHEN QUARTER(d.DATE\_ID)=2 THEN SALE END) AS Quarter2,

SUM(CASE WHEN QUARTER(d.DATE\_ID)=3 THEN SALE END) AS Quarter3,

SUM(CASE WHEN QUARTER(d.DATE\_ID)=4 THEN SALE END) AS Quarter4

FROM facts f , dates d

WHERE d.date\_ID= f.date\_ID;

-- Query 6

SELECT PRODUCT\_ID,SUM(SALE) AS SALE FROM FACTS

WHERE WEEKDAY(DATE\_ID) = 5 OR WEEKDAY(DATE\_ID)=6

GROUP BY (PRODUCT\_ID)

ORDER BY SALE DESC

LIMIT 0,5;

-- Query 7 --

SELECT store\_ID,supplier\_ID,product\_ID, sum(s.SALE)

FROM facts s

GROUP BY store\_ID,supplier\_ID,product\_ID WITH ROLLUP;

This query sum sale with respect to supplier id like for same suppliers it gives one total sale in the table .

Query 8 --

SELECT PRODUCT\_ID,

(CASE

WHEN QUARTER(DATE\_ID) < 3 THEN "First Half"

WHEN QUARTER(DATE\_ID) > 2 THEN "Second Half"

END) AS YEAR\_PART ,SUM(SALE)

FROM FACTS

WHERE YEAR(DATE\_ID)=2017

GROUP BY year\_part,PRODUCT\_ID

ORDER BY product\_id,YEAR\_PART;

-- Query 9 --

SELECT product\_name,product\_id,Price from productss

where product\_name in

(SELECT product\_name

FROM productss

GROUP BY product\_name

HAVING COUNT(product\_name) > 1);

The data set have same product\_name with more then 2 product\_ id’s and prices.

-- QUERY 10 --

#---------------------------------------------------------

DROP TABLE IF EXISTS DW\_analysis;

CREATE TABLE DW\_analysis(

`STORE\_NAME` VARCHAR(30) NOT NULL,

`PRODUCT\_NAME` VARCHAR(30) NOT NULL,

`SALE` DECIMAL(10,2) DEFAULT (0.0) NOT NULL

);

INSERT INTO DW\_analysis

SELECT STORE\_Name , PRODUCT\_Name,SUM(SALE) AS SALE

FROM FACTS f,productss p,stores s

where f.store\_id=s.store\_id and f.product\_id=p.product\_id

GROUP BY s.STORE\_NAME,p.PRODUCT\_NAME

ORDER BY s.STORE\_NAME,p.PRODUCT\_NAME;

SELECT \* FROM DW\_analysis;

**What I learnt?**

I learnt Java language and got the practical idea of whole Data Warehouse phases (ETL, aggregation). I learnt that how mesh join works and how on large scale data handled how large scale data processed and then used. I also learnt that how on warehouse queries handle.