**DATA WAREHOUSING PROJECT REPORT**

**NAME: HAMZA KHAN**

**SECTION: A**

**ROLL NO: 16K-3638**

**INLJ ALGORITHM:**

create procedure INLJLOOP()

is

cursor ms is select t.transaction\_id as TID, t.product\_id as PID, t.customer\_id as CID, t.customer\_name as Cname, t.store\_id as STID, t.store\_name as STname,

t.t\_date as TD,t.Quantity as Q, m.product\_name as Pname, m.supplier\_id as SID, m.supplier\_name as Sname,m.price

from transaction t join masterdata m where t.PID = m.product\_id

MM int; --------------------------------- Variable to extract month from Date ---------------------------------------

YY int; --------------------------------- Variable to extract year from Date ---------------------------------------

QQ int; --------------------------------- Variable to extract quarter from Date ---------------------------------------

temp ms%rowtype ; ---------------------------- To fetch records row wise and insert them into their respective tables -----------------------------------

fetch ms into temp;

begin

open ms;

loop

----------------------------------- Data insertion into DimCustomer -------------------------------------------

BEGIN

merge into DimCustomer

using ( select temp.CID, temp.Cname from temp )

on ( DimCustomer.CID = temp.CID )

when matched then

update set

DimCustomer.Cname = temp.Cname;

when not matched then

INSERT VALUES ( temp.CID,temp.Cname)

END

----------------------------------- Data insertion into DimProduct -------------------------------------------

BEGIN

merge into DimProduct

using ( select temp.price, temp.PID, temp.Pname from temp )

on ( DimProduct.PID = temp.PID )

when matched then

update set

DimProduct.Pname = temp.Pname;

when not matched then

INSERT VALUES ( temp.price, temp.PID, temp.Pname)

END

----------------------------------- Data insertion into DimSupplier -------------------------------------------

BEGIN

merge into DimSupplier

using ( select temp.SID, temp.Sname from temp )

on ( DimCustomer.SID = temp.SID )

when matched then

update set

DimSupplier.Sname = temp.Sname;

when not matched then

INSERT VALUES ( temp.SID,temp.Sname)

END

----------------------------------- Data insertion into DimStore -------------------------------------------

BEGIN

merge into DimStore

using ( select temp.STID, temp.STname from temp )

on ( DimDate.DID = temp. )

when matched then

update set

DimStore.STname = temp.STname;

when not matched then

INSERT VALUES ( temp.STID,temp.STname)

END

----------------------------------- Data insertion into DimDate -------------------------------------------

DD= select extract ( day from temp.t\_date);

MM= select extract ( month from temp.t\_date)

YY= select extract ( year from temp.t\_date);

QQ= select extract ( quarter from temp.t\_date);

BEGIN

merge into DimDate

using ( select DD, MM, YY, QQ )

on ( DimDate.STID = temp.TD )

when matched then

update set

DimDate.T\_Date = DD;

when not matched then

INSERT VALUES ( TD, DD, MM, QQ, YY)

END

----------------------------------- Data insertion into FactSales -------------------------------------------

BEGIN

merge into FactSales

using ( select temp.Q, temp.TID, temp.CID, temp.SID, temp.STID, temp.PID temp.STID, temp.STname from temp )

on ( FactSales.TID = temp.TID )

when matched then

update set

DimStore.STname = temp.STname;

when not matched then

INSERT ( Quantity, Sales, C\_ID, S\_ID, ST\_ID, P\_ID, D\_ID, T\_ID ) VALUES ( temp.Q, temp.Q \* temp.price, temp.CID, temp.SID, temp.STID, temp.PID, temp.DID, temp.TID)

END

exit when ms%notfound;

end loop;

close ms;

end;

**OLAP QUERIES:**

1) select \* from (

Select PID, Pname, t\_year, sum(Sales)

from ( (DimProduct P join FactSales S where P.PID=S.P\_ID) and (DimDate D join FactSales S where D.DID=S.D\_ID) )

GROUP BY PID, Pname, t\_year having t\_year=2016 order by Sales DESC ) limit 1;

2) select \* from (

Select SID, Sname, t\_month, t\_year, sum(Sales)

from ( (DimSupplier S join FactSales F where S.SID= F.S\_ID) and (DimDate D join FactSales F where D.DID=F.D\_ID) )

GROUP BY SID, Sname, t\_month, t\_year having t\_month=8 and t\_year=2016 order by Sales DESC ) limit 3;

3) select \* from (

Select STID, STname, t\_month, t\_year, sum(Sales)

from ( (DimStore S join FactSales F where S.STID=F.ST\_ID) and (DimDate D join FactSales F where D.DID=F.D\_ID) )

GROUP BY STID, STname, t\_month, t\_year having t\_month=8 and t\_year=2016 order by Sales DESC ) limit 3;

4) select \* from (

Select PID, Pname, Quantity, t\_year, sum(Sales)

from ( ( DimProduct P join FactSales F where P.PID=F.P\_ID) and (DimDate D join FactSales F where D.DID=F.D\_ID) )

GROUP BY order by Sales DESC ) limit 1;

5) Select PID, t\_quarter, sum(Sales)

from ( ( DimProduct P join FactSales F where P.PID=F.P\_ID) and ( DimDate D join FactSales F where D.DID=F.D\_DID) )

GROUP BY t\_quarter

6) CREATE MATERIALIZED VIEW STOREANALYSIS\_MV

BUILD IMMEDIATE

REFRESH FAST

ON DEMAND AS

Select STID, PID, sum(Sales) from ( ( DimStore S join FactSales F where S.STID=F.ST\_ID) and ( DimProduct P join FactSales F where P.PID=F.P\_ID) )

GROUP BY STID, PID, Sales

**SUMMARY:**

Through this project, I revised by Database concepts on how to extract data and further manipulating it through joins. I further learned new commands such as MERGE which inserts data into a table from another table based on a specific condition, how to create Materialized Views and how to limit the no. of rows which are being selected through LIMIT operator. I also learned about the INLJ Loop which is the core part of this project and how it joins data based on a common attribute. Overall the project taught me how to insert data into a warehouse according to its need and how it can be manipulated which we learned through the queries given in the project.