-------------------------------------------------CREATE SELECT---------------------------------------------------------------------

Create table STOCK as

(SELECT Stock\_Name,Ticker\_Symbol,Cat\_Id,BRAND\_VALUE,(LAST\_VALUE + CHANGE\_VALUE) as current\_value

FROM Industrials

union

SELECT Stock\_Name,Ticker\_Symbol,Cat\_Id,BRAND\_VALUE,(LAST\_VALUE + CHANGE\_VALUE) as current\_value

FROM ConsumerDiscretionary

union

SELECT Stock\_Name,Ticker\_Symbol,Cat\_Id,BRAND\_VALUE,(LAST\_VALUE + CHANGE\_VALUE) as current\_value

FROM Healthcare

union

SELECT Stock\_Name,Ticker\_Symbol,Cat\_Id,BRAND\_VALUE,(LAST\_VALUE + CHANGE\_VALUE) as current\_value

FROM Energysector

union

SELECT Stock\_Name,Ticker\_Symbol,Cat\_Id,BRAND\_VALUE,(LAST\_VALUE + CHANGE\_VALUE) as current\_value

FROM Technology);

ALTER TABLE STOCK

ADD (CONSTRAINT PK\_TICKER\_SYMBOL PRIMARY KEY(TICKER\_SYMBOL),

FOREIGN KEY(CAT\_ID) REFERENCES CATEGORY(CAT\_ID) );

ALTER TABLE STOCK

ADD (RISK VARCHAR(10),

INVESTMENT VARCHAR(10));

-- Import Data into table STOCK\_A from file

------------------------------------------------------UPDATE STOCK------------------------------------------------------------

update stock s

set

Risk = (select a.risk from stock\_a a where a.TICKER\_SYMBOL = s.TICKER\_SYMBOL ),

Investment = (select a.investment from stock\_a a where a.TICKER\_SYMBOL = s.TICKER\_SYMBOL );

------------------------------------------------updating Values to low/moderate/high-----------------------------------

UPDATE STOCK

SET RISK =

CASE WHEN risk < 10 THEN 'LOW'

WHEN risk < 20 THEN 'MODERATE'

ELSE 'HIGH'

END,

INVESTMENT=

CASE WHEN investment < 10 THEN 'LOW'

WHEN investment < 20 THEN 'MODERATE'

ELSE 'HIGH'

END;

UPDATE STOCK

SET BRAND\_VALUE=

CASE WHEN brand\_value < 357130

THEN 'LOW'

WHEN brand\_value < 800000 THEN 'MODERATE'

ELSE 'HIGH'

END;

--#2--Evaluation Table-----------------------------------

create table evaluation as

(select Ticker\_Symbol, (News+Dividends\_Announced+Product\_Release+New\_Contract+Employee\_Layoff+Takeover+Mgmt\_Changes+Scandals+EPS+PE\_Ratio+Exchange\_Rate)as Parameter\_Value

from Stock\_Parameter);

ALTER TABLE EVALUATION

ADD CONSTRAINT FK\_TICKER\_SYMBOL1 FOREIGN KEY(TICKER\_SYMBOL) REFERENCES STOCK(TICKER\_SYMBOL);

---------------------------------------------------#1--Stock\_Decision Table------------------------------------------------------

create table STOCK\_DECISION as (

SELECT Rownum as Dec\_ID, e.parameter\_value,S.\*

FROM STOCK S

inner join evaluation E

on E.ticker\_symbol = S.ticker\_symbol) ;

ALTER TABLE STOCK\_DECISION

ADD (CONSTRAINT PK\_DECISION\_ID PRIMARY KEY(Dec\_ID) ,

CONSTRAINT FK\_TICKER\_SYMBOL FOREIGN KEY(TICKER\_SYMBOL) REFERENCES STOCK(TICKER\_SYMBOL));

--#3--Client\_Stock\_Match Table---------------------------

create table client\_stock\_match AS

select concat('M0',rownum) as Match\_ID,S.\*,C.Client\_ID, St.Strategy\_ID from STOCK\_DECISION S

inner join Client\_Category C

on S.Cat\_ID = C.Cat\_ID

inner join Strategy\_Category St

on C.Client\_ID = St.Client\_ID

and C.Cat\_ID = St.Cat\_ID

and S.Brand\_Value = St.Brand\_Value

and S.Investment = St.Investment

and S.Parameter\_Value >= St.Parameter\_Value

order by Match\_ID,C.Cat\_ID;

ALTER TABLE CLIENT\_STOCK\_MATCH

ADD (CONSTRAINT PK\_MATCH\_ID PRIMARY KEY(Match\_ID),

CONSTRAINT FK\_CLIENT\_ID FOREIGN KEY(CLIENT\_ID) REFERENCES CLIENT(CLIENT\_ID),

CONSTRAINT FK\_STRATEGY\_ID FOREIGN KEY(STRATEGY\_ID) REFERENCES Client\_strat(STRATEGY\_ID),

FOREIGN KEY(CAT\_ID) REFERENCES CATEGORY(CAT\_ID));

----------------------------------------------------#4--Client\_Stock\_Proposal Table----------------------------

Create table Client\_Stock\_Proposal as

select CC.Brokerage\_Rate, CS.MATCH\_ID, CS.Stock\_Name, CS.Ticker\_symbol, CS.Cat\_Id, CS.Current\_value, CS.Client\_Id

from client\_stock\_match CS

inner join Client\_Category CC

on CC.Client\_Id = CS.Client\_Id

and CS.Cat\_Id = CC.Cat\_Id;

ALTER TABLE Client\_Stock\_Proposal

ADD (CONSTRAINT FK1\_CLIENT\_ID FOREIGN KEY(CLIENT\_ID) REFERENCES CLIENT(CLIENT\_ID),

CONSTRAINT FK\_MATCH\_ID FOREIGN KEY(MATCH\_ID) REFERENCES CLIENT\_STOCK\_MATCH(MATCH\_ID),

FOREIGN KEY(CAT\_ID) REFERENCES CATEGORY(CAT\_ID),

FOREIGN KEY(Ticker\_symbol) REFERENCES STOCK(Ticker\_symbol));

-----------------------------PROPOSAL STATUS-----------------------------------------------------------

--#5--Proposal\_Status Table------------------------------------

create table PROPOSAL\_STATUS AS(

select concat('T0',rownum) as TRANS\_ID,C.MATCH\_ID,C.Client\_Id,C.Ticker\_symbol,SYSDATE AS TRANS\_DATE

FROM client\_stock\_match C);

ALTER TABLE PROPOSAL\_STATUS

ADD (

PRIMARY KEY(TRANS\_ID),

FOREIGN KEY(MATCH\_ID) REFERENCES CLIENT\_STOCK\_MATCH(MATCH\_ID)) ;

ALTER TABLE PROPOSAL\_STATUS

ADD (NOOFSTOCKS NUMBER(4) default 0 ,

STATUS VARCHAR(6) DEFAULT 'NIL');

ALTER TABLE PROPOSAL\_STATUS

ADD CONSTRAINT CHECK\_STATUS CHECK (STATUS IN ('ACCEPT','REJECT','NIL') AND

((STATUS = 'ACCEPT' and NOOFSTOCKS > 0) OR

(STATUS = 'REJECT' and NOOFSTOCKS = 0) OR

(STATUS = 'NIL')));

--------------------IMPORTANT TO CREATE TREND USING TRANS DATE--------

update proposal\_status SET TRANS\_DATE =

CASE WHEN mod(TO\_NUMBER(SUBSTR(TRANS\_ID,2)),3)=0 Then TO\_DATE('01-apr-19','dd-mon-yy')

WHEN mod(TO\_NUMBER(SUBSTR(TRANS\_ID,2)),3)=1 Then TO\_DATE('01-mar-19','dd-mon-yy')

ELSE TO\_DATE('01-feb-19','dd-mon-yy')

END;

-----------------------------STATUS UPDATE FOR PROPOSAL STATUS----

update proposal\_status p SET p.STATUS =

Case

when (p.TRANS\_DATE = TO\_DATE('01-feb-19','dd-mon-yy') and mod(TO\_NUMBER(SUBSTR(p.TRANS\_ID,2)),2)=0) Then to\_char('REJECT')

when (p.TRANS\_DATE = TO\_DATE('01-mar-19','dd-mon-yy') and mod(TO\_NUMBER(SUBSTR(p.TRANS\_ID,2)),4)=0) Then to\_char('REJECT')

when (p.TRANS\_DATE = TO\_DATE('01-apr-19','dd-mon-yy') and mod(TO\_NUMBER(SUBSTR(p.TRANS\_ID,2)),9)=0 and mod(TO\_NUMBER(SUBSTR(p.TRANS\_ID,2)),2)=1 ) Then to\_char('REJECT')

ELSE to\_char('NIL')

END;

------------------------CREATING TRIGGER------------------------------

set serveroutput on

CREATE OR REPLACE TRIGGER T\_Mail

AFTER UPDATE OF STATUS ON PROPOSAL\_STATUS

FOR EACH ROW

WHEN(NEW.STATUS ='ACCEPT')

BEGIN

INSERT INTO MAIL(V\_TO,CLIENT\_ID)

(SELECT C.EMAIL,C.CLIENT\_ID

FROM CLIENT C

WHERE C.CLIENT\_ID=:NEW.CLIENT\_ID);

if updating then

DBMS\_OUTPUT.PUT\_LINE ('MAIL HAS BEEN SENT');

end if;

END;

/

----------------------------------Updating STATUS TO ACCEPT and NOOFSSTOCKS------------------------------------

update proposal\_status SET STATUS = 'ACCEPT', noofstocks = '2' where status = 'NIL';

------------------------------------------------------------------------------------------------------------------------------------------

--#6--Trans\_Header Table---------------------------------------

create table TRANS\_HEADER AS(

select p.TRANS\_ID,c.client\_id,c.Ticker\_Symbol,c.Stock\_Name,c.Cat\_ID

from Client\_Stock\_Proposal c

inner join PROPOSAL\_STATUS p

on p.MATCH\_ID=c.MATCH\_ID

and p.STATUS='ACCEPT');

ALTER TABLE TRANS\_HEADER

ADD CONSTRAINT FK\_HTRANS\_ID FOREIGN KEY(TRANS\_ID) REFERENCES PROPOSAL\_STATUS(TRANS\_ID) ;

--#7--Trans\_Detail Table------------------------------------------

create table TRANS\_DETAIL AS

select p.TRANS\_ID,c.Ticker\_Symbol,c.Stock\_Name,c.Cat\_ID,p.NOOFSTOCKS,c.current\_value,c.brokerage\_rate,p.TRANS\_DATE

from Client\_Stock\_Proposal c

inner join PROPOSAL\_STATUS p

on p.MATCH\_ID=c.MATCH\_ID

inner join TRANS\_HEADER t

on t.client\_id = c.client\_id

and p.STATUS='ACCEPT';

ALTER TABLE TRANS\_DETAIL

ADD CONSTRAINT FK\_DTRANS\_ID FOREIGN KEY(TRANS\_ID) REFERENCES PROPOSAL\_STATUS(TRANS\_ID) ;

--#8--Trans\_Deletion Table-------------------------------------------

create table TRANS\_DELETION AS

select p.TRANS\_ID,p.MATCH\_ID,c.client\_id,c.Ticker\_Symbol,c.Stock\_Name,p.TRANS\_DATE

from Client\_Stock\_Proposal c

inner join PROPOSAL\_STATUS p

on p.MATCH\_ID = c.MATCH\_ID

and p.STATUS='REJECT';

ALTER TABLE TRANS\_DElETION

ADD CONSTRAINT FK\_TRANS\_ID FOREIGN KEY(TRANS\_ID) REFERENCES PROPOSAL\_STATUS(TRANS\_ID) ;

--#9--Profit Table----------------------------------------------------

create table PROFIT as

select concat('P0',rownum) as Profit\_ID,c.client\_id,c.Ticker\_Symbol,c.Stock\_Name,p.TRANS\_ID,c.Brokerage\_rate,p.noofstocks,c.Brokerage\_rate\*p.noofstocks as Profit\_Amount

from Client\_Stock\_Proposal c

inner join PROPOSAL\_STATUS p

on p.MATCH\_ID = c.MATCH\_ID ;

ALTER TABLE PROFIT

ADD CONSTRAINT PK\_PROFIT\_ID PRIMARY KEY(PROFIT\_ID);

**/\***

**----------Update of Proposal\_Status table by the Client:---------**

UPDATE PROPOSAL\_STATUS

SET NOOFSTOCKS='&NOOFSTOCKS',

STATUS='&STATUS',

TRANS\_DATE=SYSDATE

WHERE CLIENT\_ID = '&CLIENT\_ID'

\*/