\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CREATE TABLE DATE\_DIMENSION

(

DATE\_KEY DATE NOT NULL,

FULL\_DATE\_DESCRIPTION VARCHAR2(64) NOT NULL,

DAY\_OF\_WEEK NUMBER(1,0) NOT NULL,

DAY\_OF\_MONTH NUMBER(2,0) NOT NULL,

DAY\_OF\_YEAR NUMBER(3,0) NOT NULL,

LAST\_DAY\_OF\_WEEK\_INDICATOR CHAR(1) NOT NULL,

LAST\_DAY\_OF\_MONTH\_INDICATOR CHAR(1) NOT NULL,

WEEK\_ENDING\_DATE DATE NOT NULL,

MONTH\_NUMBER NUMBER(2,0) NOT NULL,

MONTH\_NAME VARCHAR2(32) NOT NULL,

YEAR\_MONTH CHAR(32) NOT NULL,

QUARTER\_NUMBER NUMBER(1,0) NOT NULL,

YEAR\_QUARTER CHAR(32) NOT NULL,

YEAR\_NUMBER NUMBER(4,0) NOT NULL,

CONSTRAINT DATE\_DIMENSION\_PK PRIMARY KEY (DATE\_KEY)

)

—--------------------------------------------------------------------------------------------------------

Procedure to insert dates in the date dime

++++++++++++++++++++++++++++++++++++++++++++

create or replace PROCEDURE sp\_DATE\_DIMENSION(v\_START\_YEAR IN INT, v\_END\_YEAR IN INT) AS

--Declare two variables as DATE datatypes

v\_CURRENT\_DATE DATE;

v\_END\_DATE DATE;

BEGIN

--Assign the start year and end year to it's respective variables

v\_CURRENT\_DATE := TO\_DATE('0101' || v\_START\_YEAR, 'MMDDYYYY');

v\_END\_DATE := TO\_DATE('1231' || v\_END\_YEAR, 'MMDDYYYY');

--Clear/Dump what is currently stored in the table

DELETE FROM DATE\_DIMENSION;

--Check the condition to see if the start year is less than the end year (Input Parameters)

WHILE v\_CURRENT\_DATE <= v\_END\_DATE

LOOP

--DATE\_DIMENSION Table

INSERT INTO DATE\_DIMENSION

(

DATE\_KEY,

FULL\_DATE\_DESCRIPTION,

DAY\_OF\_WEEK,

DAY\_OF\_MONTH,

DAY\_OF\_YEAR,

LAST\_DAY\_OF\_WEEK\_INDICATOR,

LAST\_DAY\_OF\_MONTH\_INDICATOR,

WEEK\_ENDING\_DATE,

MONTH\_NUMBER,

MONTH\_NAME,

YEAR\_MONTH,

QUARTER\_NUMBER,

YEAR\_QUARTER,

YEAR\_NUMBER

)

VALUES

(

v\_CURRENT\_DATE, --DATE\_KEY

TO\_CHAR(v\_CURRENT\_DATE, 'Day, Month DD, YYYY'), --FULL\_DATE\_DESCRIPTION

TO\_NUMBER(TO\_CHAR(v\_CURRENT\_DATE, 'D')) -1, --DAY\_OF\_WEEK

TO\_CHAR(v\_CURRENT\_DATE,'DD'), --DAY\_OF\_MONTH

TO\_CHAR(v\_CURRENT\_DATE,'DDD'), --DAY\_OF\_YEAR

CASE --LAST\_DAY\_OF\_WEEK\_INDICATOR

WHEN TO\_CHAR(v\_CURRENT\_DATE,'FMDay') = 'Saturday' THEN 'Y'

ELSE 'N'

END,

CASE --LAST\_DAY\_OF\_MONTH\_INDICATOR

WHEN LAST\_DAY(v\_CURRENT\_DATE) = v\_CURRENT\_DATE THEN 'Y'

ELSE 'N'

END,

CASE --WEEK\_ENDING\_DATE OF CURRENT WEEK ENDING ON SATURDAY

WHEN TO\_CHAR(v\_CURRENT\_DATE,'FMDay') = 'Saturday' THEN v\_CURRENT\_DATE

ELSE NEXT\_DAY(v\_CURRENT\_DATE,'SATURDAY')

END,

TO\_CHAR(v\_CURRENT\_DATE,'MM'), --MONTH\_NUMBER

TO\_CHAR(v\_CURRENT\_DATE,'MONTH'), --MONTH\_NAME

TO\_CHAR(v\_CURRENT\_DATE,'MONTH YYYY'), --YEAR\_MONTH

TO\_CHAR(v\_CURRENT\_DATE,'Q'), --QUARTER\_NUMBER

TO\_CHAR(v\_CURRENT\_DATE,'YYYY Q'), --YEAR\_QUARTER

TO\_CHAR(v\_CURRENT\_DATE,'YYYY') --YEAR\_NUMBER

);

--Increment and assign the current date value to be re-evaluated

v\_CURRENT\_DATE := v\_CURRENT\_DATE + 1;

END LOOP;

END;

====================================================

–Calling

declare

begin

sp\_DATE\_DIMENSION (2013, 2022);

End;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CREATE TABLE DATE\_DIM

as

(

select

TO\_NUMBER(TO\_CHAR(DATE\_D,'j')) as DATE\_J,

DATE\_D,

TO\_CHAR(DATE\_D,'YYYY-MM-DD') as DATE\_V,

TO\_NUMBER(TO\_CHAR(DATE\_D,'YYYY')) as YEAR\_NUM,

TO\_NUMBER(TO\_CHAR(DATE\_D,'Q')) as QUARTER\_NUM,

TO\_NUMBER(TO\_CHAR(DATE\_D,'MM')) as MONTH\_NUM,

TRIM(TO\_CHAR(DATE\_D,'Month','nls\_date\_language=english')) as MONTH\_DESC,

TO\_NUMBER(TO\_CHAR(DATE\_D,'IW')) as ISO\_WEEK\_NUM,

TO\_NUMBER(TO\_CHAR(DATE\_D,'IYYY')) as ISO\_YEAR\_NUM,

TO\_NUMBER(TO\_CHAR(DATE\_D,'DD')) as DAY\_OF\_MONTH\_NUM,

TO\_NUMBER(TO\_CHAR(DATE\_D,'D')) as DAY\_OF\_WEEK\_NUM,

TRIM(TO\_CHAR(DATE\_D,'Day','nls\_date\_language=english')) as DAY\_OF\_WEEK\_DESC,

(CASE WHEN TRIM(TO\_CHAR(DATE\_D,'Day','nls\_date\_language=english')) IN ('Saturday','Sunday') THEN 'Weekend' ELSE 'Weekday' END) as DAY\_TYPE\_DESC

from

(

select

to\_date('1999-12-31','YYYY-MM-DD')+ROWNUM as DATE\_D

from

dual

connect by level <= to\_date('2029-12-31','YYYY-MM-DD')-to\_date('1999-12-31','YYYY-MM-DD')

)

);

================================================================

DECLARE

CURSOR Add\_Sequence\_Cursor IS

SELECT u.table\_name , c.column\_name , u.constraint\_type , b.object\_type , t.data\_type

FROM user\_constraints u , user\_cons\_columns c , USER\_OBJECTS b , user\_tab\_columns t

WHERE u.CONSTRAINT\_NAME = c.constraint\_name

and b.object\_name = c.table\_name

and c.column\_name = t.column\_name

and c.table\_name = t.table\_name

and b.object\_type = 'TABLE'

and t.data\_type = 'NUMBER'

and u.constraint\_type = 'P' ;

v\_count number(1);

BEGIN

FOR seq\_record IN Add\_Sequence\_Cursor LOOP

select count(\*) into v\_count from user\_sequences where SEQUENCE\_NAME=seq\_record.table\_name||'\_SEQ';

IF v\_count = 0 THEN

EXECUTE IMMEDIATE 'CREATE SEQUENCE '||seq\_record.table\_name||'\_SEQ

START WITH 600

increment by 1

MAXVALUE 999999999999999999999999999';

END IF;

EXECUTE IMMEDIATE 'CREATE OR REPLACE TRIGGER '||seq\_record.table\_name||'\_TRG

BEFORE INSERT

ON '||seq\_record.table\_name||'

REFERENCING NEW AS New OLD AS Old

FOR EACH ROW

BEGIN

:new.'||seq\_record.column\_name||' := '||seq\_record.table\_name||'\_SEQ.nextval;

END; ';

END LOOP;

END;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

create table company\_Dim (

Company\_key number (10)constraint comp\_pk primary key ,

Sympol varchar2(50),

Company\_name varchar2(200),

Sec\_filling varchar2(200),

GICS\_Sector varchar2(200) ,

GICS\_Industry\_group varchar2(200),

GICS\_Sector\_hirarchy varchar2(200),

GICS\_sub\_industry varchar2(200),

Headquarters\_location varchar2(200),

Date\_first\_added date,

CIK number(20),

Founded varchar2(50));

……………………………………………………….

Table (Yearly\_stock\_vs\_gold\_vs\_crude\_oil\_vs\_inflation\_vs\_interest\_analysis\_fact)

—---------------------------------------------------------------------------------------------------------------

CREATE TABLE YRLY\_INF\_INT\_FACT

(DATE\_KEY DATE constraint year\_date\_fk references DATE\_DIMENSION(DATE\_KEY),

INFLATION\_CONSUMER\_PRICES NUMBER(8,3),

INFLATION\_GDP\_DEFLATOR NUMBER(8,3),

REAL\_INTEREST\_RATE NUMBER(8,3),

LENDING\_INTEREST\_RATE NUMBER(8,3));

—------------------------------------------------------------------------------------------------------------------

CREATE TABLE STOCK\_PRICE\_daily\_FACT

(COMPANY\_KEY NUMBER(10)constraint compny\_fk references company\_Dim (Company\_key),

DATE\_KEY DATE constraint daily\_date\_fk references DATE\_DIMENSION(DATE\_KEY),

OPEN NUMBER(10,2),

HIGH NUMBER(10,2),

LOW NUMBER(10,2),

CLOSE NUMBER(10,2),

VOLUME NUMBER(10,2));

—--------------------------------------------------------------------------------------------------------------------

CREATE TABLE MNTHLY\_GD\_OIL\_FACT

(DATE\_KEY DATE constraint mnth\_date\_fk references DATE\_DIMENSION(DATE\_KEY),

GOLD\_PRICE NUMBER(10,2),

CRUDE\_OIL\_PRICE NUMBER(10,2));

—-----------------------------------------------------------------------------------------------------------------

CREATE TABLE STOCK\_PRICE\_FACT

(COMPANY\_KEY NUMBER(10)constraint compan\_fk references company\_Dim (Company\_key),

DATE\_KEY DATE constraint stck\_date\_fk references DATE\_DIMENSION(DATE\_KEY),

Price NUMBER(10,3),

"Price/Earnings" NUMBER(10,3),

Dividend\_Yield NUMBER(10,3),

"Earnings/Share" NUMBER(10,3),

"52\_Week\_Low" NUMBER(10,3),

"52\_Week\_High" NUMBER(10,3),

Market\_Cap NUMBER(20),

EBITDA NUMBER(20),

"Price/Sales" NUMBER(10,3),

"Price/Book" NUMBER(10,3),

SEC\_Filings varchar2(250));