```
--6.11 Dimension Loads in Snowflake

/****************************************
Course: IMT 577
Assignment: Module 6
Notes: Create Dim Date and load with
two years of dates. Loads 20 years of
dates.

****************************************/

--======================================================
-------------DIM_DATE
--=====================================================
-- Create table script for Dimension DIM_DATE
create or replace table DIM_DATE (
        DATE_PKEY                       number(9) PRIMARY KEY,
        DATE                            date not null,
        FULL_DATE_DESC                  varchar(64) not null,
        DAY_NUM_IN_WEEK                 number(1) not null,
        DAY_NUM_IN_MONTH                number(2) not null,
        DAY_NUM_IN_YEAR                 number(3) not null,
        DAY_NAME                        varchar(10) not null,
        DAY_ABBREV                      varchar(3) not null,
        WEEKDAY_IND                     varchar(64) not null,
        US_HOLIDAY_IND                  varchar(64) not null,
    /*<COMPANYNAME>*/_HOLIDAY_IND varchar(64) not null,
    MONTH_END_IND             varchar(64) not null,
    WEEK_BEGIN_DATE_NKEY      number(9) not null,
    WEEK_BEGIN_DATE           date not null,
    WEEK_END_DATE_NKEY        number(9) not null,
    WEEK_END_DATE             date not null,
    WEEK_NUM_IN_YEAR          number(9) not null,
    MONTH_NAME                     varchar(10) not null,
    MONTH_ABBREV              varchar(3) not null,
    MONTH_NUM_IN_YEAR         number(2) not null,
    YEARMONTH                 varchar(10) not null,
    QUARTER                        number(1) not null,
    YEARQUARTER               varchar(10) not null,
    YEAR                      number(5) not null,
    FISCAL_WEEK_NUM           number(2) not null,
    FISCAL_MONTH_NUM          number(2) not null,
    FISCAL_YEARMONTH     varchar(10) not null,
    FISCAL_QUARTER            number(1) not null,
    FISCAL_YEARQUARTER        varchar(10) not null,
    FISCAL_HALFYEAR           number(1) not null,
    FISCAL_YEAR               number(5) not null,
    SQL_TIMESTAMP             timestamp_ntz,
    CURRENT_ROW_IND           char(1) default 'Y',
    EFFECTIVE_DATE            date default to_date(current_timestamp),
    EXPIRATION_DATE           date default To_date('9999-12-31')
)
```

```
comment = 'Type 0 Dimension Table Housing Calendar and Fiscal Year Date Attributes';


-- Populate data into DIM_DATE

insert into DIM_DATE

select DATE_PKEY,

               DATE_COLUMN,

     FULL_DATE_DESC,

               DAY_NUM_IN_WEEK,

               DAY_NUM_IN_MONTH,

               DAY_NUM_IN_YEAR,

               DAY_NAME,

               DAY_ABBREV,

               WEEKDAY_IND,

               US_HOLIDAY_IND,

     COMPANY_HOLIDAY_IND,

               MONTH_END_IND,

               WEEK_BEGIN_DATE_NKEY,

               WEEK_BEGIN_DATE,

               WEEK_END_DATE_NKEY,

               WEEK_END_DATE,

               WEEK_NUM_IN_YEAR,

               MONTH_NAME,

               MONTH_ABBREV,

               MONTH_NUM_IN_YEAR,

               YEARMONTH,

               CURRENT_QUARTER,
               YEARQUARTER,
               CURRENT_YEAR,
               FISCAL_WEEK_NUM,
               FISCAL_MONTH_NUM,
               FISCAL_YEARMONTH,
               FISCAL_QUARTER,
               FISCAL_YEARQUARTER,
               FISCAL_HALFYEAR,
               FISCAL_YEAR,
               SQL_TIMESTAMP,
               CURRENT_ROW_IND,
               EFFECTIVE_DATE,
               EXPIRA_DATE
           from


        --( select to_date('01-25-2019 23:25:11.120','MM-DD-YYYY HH24:MI:SS.FF') as DD, /*<<Modify date for preferred table start date*/
        --( select to_date('2013-01-01 00:00:01','YYYY-MM-DD HH24:MI:SS') as DD, /*<<Modify date for preferred table start date*/
            ( select to_date('2012-12-31 23:59:59','YYYY-MM-DD HH24:MI:SS') as DD, /*<<Modify date for preferred table start date*/
                   seq1() as Sl,row_number() over (order by Sl) as row_numbers,
                   dateadd(day,row_numbers,DD) as V_DATE,
                   case when date_part(dd, V_DATE) < 10 and date_part(mm, V_DATE) > 9 then
```

```
                        date_part(year, V_DATE)||date_part(mm, 
V_DATE)||'0'||date_part(dd, V_DATE)
                        when date_part(dd, V_DATE) < 10 and  date_part(mm, V_DATE) < 
10 then
                        date_part(year, V_DATE)||'0'||date_part(mm, 
V_DATE)||'0'||date_part(dd, V_DATE)
                        when date_part(dd, V_DATE) > 9 and  date_part(mm, V_DATE) < 
10 then
                        date_part(year, V_DATE)||'0'||date_part(mm, 
V_DATE)||date_part(dd, V_DATE)
                        when date_part(dd, V_DATE) > 9 and  date_part(mm, V_DATE) > 9 
then
                        date_part(year, V_DATE)||date_part(mm, 
V_DATE)||date_part(dd, V_DATE) end as DATE_PKEY,
                V_DATE as DATE_COLUMN,
                dayname(dateadd(day,row_numbers,DD)) as DAY_NAME_1,
                case
                        when dayname(dateadd(day,row_numbers,DD)) = 'Mon' then 
'Monday'
                        when dayname(dateadd(day,row_numbers,DD)) = 'Tue' then 
'Tuesday'
                        when dayname(dateadd(day,row_numbers,DD)) = 'Wed' then 
'Wednesday'
                        when dayname(dateadd(day,row_numbers,DD)) = 'Thu' then 
'Thursday'
                        when dayname(dateadd(day,row_numbers,DD)) = 'Fri' then 
'Friday'
                        when dayname(dateadd(day,row_numbers,DD)) = 'Sat' then 
'Saturday'
```

```
                                 when dayname(dateadd(day,row_numbers,DD)) = 'Sun' then
'Sunday' end ||', '||

                         case when monthname(dateadd(day,row_numbers,DD)) ='Jan' then
'January'

                                   when monthname(dateadd(day,row_numbers,DD)) ='Feb' then
'February'

                                   when monthname(dateadd(day,row_numbers,DD)) ='Mar' then
'March'

                                   when monthname(dateadd(day,row_numbers,DD)) ='Apr' then
'April'

                                   when monthname(dateadd(day,row_numbers,DD)) ='May' then
'May'

                                   when monthname(dateadd(day,row_numbers,DD)) ='Jun' then
'June'

                                   when monthname(dateadd(day,row_numbers,DD)) ='Jul' then
'July'

                                   when monthname(dateadd(day,row_numbers,DD)) ='Aug' then
'August'

                                   when monthname(dateadd(day,row_numbers,DD)) ='Sep' then
'September'

                                   when monthname(dateadd(day,row_numbers,DD)) ='Oct' then
'October'

                                   when monthname(dateadd(day,row_numbers,DD)) ='Nov' then
'November'

                                   when monthname(dateadd(day,row_numbers,DD)) ='Dec' then
'December' end

                                   ||' '|| to_varchar(dateadd(day,row_numbers,DD), ' dd, yyyy') as
FULL_DATE_DESC,

                         dateadd(day,row_numbers,DD) as V_DATE_1,

                         dayofweek(V_DATE_1)+1 as DAY_NUM_IN_WEEK,
                         Date_part(dd,V_DATE_1) as DAY_NUM_IN_MONTH,

                         dayofyear(V_DATE_1) as DAY_NUM_IN_YEAR,

                         case

                                 when dayname(V_DATE_1) = 'Mon' then 'Monday'

                                 when dayname(V_DATE_1) = 'Tue' then 'Tuesday'

                                 when dayname(V_DATE_1) = 'Wed' then 'Wednesday'

                                 when dayname(V_DATE_1) = 'Thu' then 'Thursday'

                                 when dayname(V_DATE_1) = 'Fri' then 'Friday'

                                 when dayname(V_DATE_1) = 'Sat' then 'Saturday'

                                 when dayname(V_DATE_1) = 'Sun' then 'Sunday' end as
                     DAY_NAME,

                         dayname(dateadd(day,row_numbers,DD)) as DAY_ABBREV,

                         case

                                 when dayname(V_DATE_1) = 'Sun' and dayname(V_DATE_1) =
'Sat' then

     'Not-Weekday'

                                 else 'Weekday' end as WEEKDAY_IND,

                              case

                                 when (DATE_PKEY = date_part(year, V_DATE)||'0101' or
DATE_PKEY = date_part(year, V_DATE)||'0704' or

                                 DATE_PKEY = date_part(year, V_DATE)||'1225' or DATE_PKEY =
date_part(year, V_DATE)||'1226') then

                                 'Holiday'

                                 when monthname(V_DATE_1) ='May' and
dayname(last_day(V_DATE_1)) = 'Wed'

                                 and dateadd(day,-2,last_day(V_DATE_1)) = V_DATE_1  then

                                 'Holiday'
```

```
                                    when monthname(V_DATE_1) ='May' and
dayname(last_day(V_DATE_1)) = 'Thu'
                                    and dateadd(day,-3,last_day(V_DATE_1)) = V_DATE_1  then
                                    'Holiday'
                                    when monthname(V_DATE_1) ='May' and
dayname(last_day(V_DATE_1)) = 'Fri'
                                    and dateadd(day,-4,last_day(V_DATE_1)) = V_DATE_1 then
                                    'Holiday'
                                    when monthname(V_DATE_1) ='May' and
dayname(last_day(V_DATE_1)) = 'Sat'
                                    and dateadd(day,-5,last_day(V_DATE_1)) = V_DATE_1  then
                                    'Holiday'
                                    when monthname(V_DATE_1) ='May' and
dayname(last_day(V_DATE_1)) = 'Sun'
                                    and dateadd(day,-6,last_day(V_DATE_1)) = V_DATE_1 then
                                    'Holiday'
                                    when monthname(V_DATE_1) ='May' and
dayname(last_day(V_DATE_1)) = 'Mon'
                                    and last_day(V_DATE_1) = V_DATE_1 then
                                    'Holiday'
                                    when monthname(V_DATE_1) ='May' and
dayname(last_day(V_DATE_1)) = 'Tue'
                                    and dateadd(day,-1 ,last_day(V_DATE_1)) = V_DATE_1  then
                                    'Holiday'
                                    when monthname(V_DATE_1) ='Sep' and
dayname(date_part(year, V_DATE_1)||'-09-01') = 'Wed'
                                    and dateadd(day,5,(date_part(year, V_DATE_1)||'-09-01')) =
V_DATE_1  then
```

```
                                'Holiday'
                                when monthname(V_DATE_1) ='Sep' and
dayname(date_part(year, V_DATE_1)||'-09-01') = 'Thu'
                                and dateadd(day,4,(date_part(year, V_DATE_1)||'-09-01')) =
V_DATE_1  then
                                'Holiday'
                                when monthname(V_DATE_1) ='Sep' and
dayname(date_part(year, V_DATE_1)||'-09-01') = 'Fri'
                                and dateadd(day,3,(date_part(year, V_DATE_1)||'-09-01')) =
V_DATE_1 then
                                'Holiday'
                                when monthname(V_DATE_1) ='Sep' and
dayname(date_part(year, V_DATE_1)||'-09-01') = 'Sat'
                                and dateadd(day,2,(date_part(year, V_DATE_1)||'-09-01')) =
V_DATE_1  then
                                'Holiday'
                                when monthname(V_DATE_1) ='Sep' and
dayname(date_part(year, V_DATE_1)||'-09-01') = 'Sun'
                                and dateadd(day,1,(date_part(year, V_DATE_1)||'-09-01')) =
V_DATE_1 then
                                'Holiday'
                                when monthname(V_DATE_1) ='Sep' and
dayname(date_part(year, V_DATE_1)||'-09-01') = 'Mon'
                                and date_part(year, V_DATE_1)||'-09-01' = V_DATE_1 then
                                'Holiday'
                                when monthname(V_DATE_1) ='Sep' and
dayname(date_part(year, V_DATE_1)||'-09-01') = 'Tue'
                                and dateadd(day,6 ,(date_part(year, V_DATE_1)||'-09-01')) =
V_DATE_1  then
```

```
                        'Holiday'
                        when monthname(V_DATE_1) ='Nov' and
dayname(date_part(year, V_DATE_1)||'-11-01') = 'Wed'
                        and (dateadd(day,23,(date_part(year, V_DATE_1)||'-11-01')) =
V_DATE_1  or
                                dateadd(day,22,(date_part(year, V_DATE_1)||'-11-01')) =
V_DATE_1 ) then
                        'Holiday'
                        when monthname(V_DATE_1) ='Nov' and
dayname(date_part(year, V_DATE_1)||'-11-01') = 'Thu'
                        and ( dateadd(day,22,(date_part(year, V_DATE_1)||'-11-01')) =
V_DATE_1 or
                                dateadd(day,21,(date_part(year, V_DATE_1)||'-11-01')) =
V_DATE_1 ) then
                        'Holiday'
                        when monthname(V_DATE_1) ='Nov' and
dayname(date_part(year, V_DATE_1)||'-11-01') = 'Fri'
                        and ( dateadd(day,21,(date_part(year, V_DATE_1)||'-11-01')) =
V_DATE_1 or
                                dateadd(day,20,(date_part(year, V_DATE_1)||'-11-01')) =
V_DATE_1 ) then
                         'Holiday'
                        when monthname(V_DATE_1) ='Nov' and
dayname(date_part(year, V_DATE_1)||'-11-01') = 'Sat'
                        and ( dateadd(day,27,(date_part(year, V_DATE_1)||'-11-01')) =
V_DATE_1 or
                                dateadd(day,26,(date_part(year, V_DATE_1)||'-11-01')) =
V_DATE_1 ) then
                        'Holiday'
```

```
                when monthname(V_DATE_1) ='Nov' and
dayname(date_part(year, V_DATE_1)||'-11-01') = 'Sun'
                and ( dateadd(day,26,(date_part(year, V_DATE_1)||'-11-01')) =
V_DATE_1 or
                    dateadd(day,25,(date_part(year, V_DATE_1)||'-11-01')) =
V_DATE_1 ) then
                'Holiday'
                when monthname(V_DATE_1) ='Nov' and
dayname(date_part(year, V_DATE_1)||'-11-01') = 'Mon'
                and (dateadd(day,25,(date_part(year, V_DATE_1)||'-11-01')) =
V_DATE_1 or
                    dateadd(day,24,(date_part(year, V_DATE_1)||'-11-01')) =
V_DATE_1 ) then
                'Holiday'
                when monthname(V_DATE_1) ='Nov' and
dayname(date_part(year, V_DATE_1)||'-11-01') = 'Tue'
                and (dateadd(day,24,(date_part(year, V_DATE_1)||'-11-01')) =
V_DATE_1 or
                    dateadd(day,23,(date_part(year, V_DATE_1)||'-11-01')) =
V_DATE_1 ) then
                 'Holiday'
                else
                'Not-Holiday' end as US_HOLIDAY_IND,
            /*Modify the following for Company Specific Holidays*/
            case
                when (DATE_PKEY = date_part(year, V_DATE)||'0101' or
DATE_PKEY = date_part(year, V_DATE)||'0219'
                or DATE_PKEY = date_part(year, V_DATE)||'0528' or DATE_PKEY =
date_part(year, V_DATE)||'0704'
```

```
                        or DATE_PKEY = date_part(year, V_DATE)||'1225' )then
                        'Holiday'
        when monthname(V_DATE_1) ='Mar' and dayname(last_day(V_DATE_1)) = 'Fri'
                        and last_day(V_DATE_1) = V_DATE_1 then
                        'Holiday'
                        when monthname(V_DATE_1) ='Mar' and
dayname(last_day(V_DATE_1)) = 'Sat'
                        and dateadd(day,-1,last_day(V_DATE_1)) = V_DATE_1  then
                        'Holiday'
                        when monthname(V_DATE_1) ='Mar' and
dayname(last_day(V_DATE_1)) = 'Sun'
                        and dateadd(day,-2,last_day(V_DATE_1)) = V_DATE_1 then
                        'Holiday'
                        when monthname(V_DATE_1) ='Apr' and
dayname(date_part(year, V_DATE_1)||'-04-01') = 'Tue'
        and dateadd(day,3,(date_part(year, V_DATE_1)||'-04-01')) = V_DATE_1 then
                        'Holiday'
                        when monthname(V_DATE_1) ='Apr' and
dayname(date_part(year, V_DATE_1)||'-04-01') = 'Wed'
                        and dateadd(day,2,(date_part(year, V_DATE_1)||'-04-01')) =
V_DATE_1 then
                        'Holiday'
                        when monthname(V_DATE_1) ='Apr' and
dayname(date_part(year, V_DATE_1)||'-04-01') = 'Thu'
        and dateadd(day,1,(date_part(year, V_DATE_1)||'-04-01')) = V_DATE_1 then
                        'Holiday'
                        when monthname(V_DATE_1) ='Apr' and
dayname(date_part(year, V_DATE_1)||'-04-01') = 'Fri'
```

```
            and date_part(year, V_DATE_1)||'-04-01' = V_DATE_1 then
            'Holiday'
    when monthname(V_DATE_1) ='Apr' and dayname(date_part(year, V_DATE_1)||'-04-
01') = 'Wed'
            and dateadd(day,5,(date_part(year, V_DATE_1)||'-04-01')) =
V_DATE_1  then
            'Holiday'
            when monthname(V_DATE_1) ='Apr' and
dayname(date_part(year, V_DATE_1)||'-04-01') = 'Thu'
            and dateadd(day,4,(date_part(year, V_DATE_1)||'-04-01')) =
V_DATE_1  then
            'Holiday'
            when monthname(V_DATE_1) ='Apr' and
dayname(date_part(year, V_DATE_1)||'-04-01') = 'Fri'
            and dateadd(day,3,(date_part(year, V_DATE_1)||'-04-01')) =
V_DATE_1 then
            'Holiday'
            when monthname(V_DATE_1) ='Apr' and
dayname(date_part(year, V_DATE_1)||'-04-01') = 'Sat'
            and dateadd(day,2,(date_part(year, V_DATE_1)||'-04-01')) =
V_DATE_1  then
            'Holiday'
            when monthname(V_DATE_1) ='Apr' and
dayname(date_part(year, V_DATE_1)||'-04-01') = 'Sun'
            and dateadd(day,1,(date_part(year, V_DATE_1)||'-04-01')) =
V_DATE_1 then
            'Holiday'
            when monthname(V_DATE_1) ='Apr' and
dayname(date_part(year, V_DATE_1)||'-04-01') = 'Mon'
```

```
          and date_part(year, V_DATE_1)||'-04-01'= V_DATE_1 then
                                'Holiday'
                                when monthname(V_DATE_1) ='Apr' and
dayname(date_part(year, V_DATE_1)||'-04-01') = 'Tue'
                                and dateadd(day,6 ,(date_part(year, V_DATE_1)||'-04-01')) =
V_DATE_1  then
                                'Holiday'
                                when monthname(V_DATE_1) ='Sep' and
dayname(date_part(year, V_DATE_1)||'-09-01') = 'Wed'
                                and dateadd(day,5,(date_part(year, V_DATE_1)||'-09-01')) =
V_DATE_1  then
                                'Holiday'
                                when monthname(V_DATE_1) ='Sep' and
dayname(date_part(year, V_DATE_1)||'-09-01') = 'Thu'
                                and dateadd(day,4,(date_part(year, V_DATE_1)||'-09-01')) =
V_DATE_1  then
                                'Holiday'
                                when monthname(V_DATE_1) ='Sep' and
dayname(date_part(year, V_DATE_1)||'-09-01') = 'Fri'
                                and dateadd(day,3,(date_part(year, V_DATE_1)||'-09-01')) =
V_DATE_1 then
                                'Holiday'
                                when monthname(V_DATE_1) ='Sep' and
dayname(date_part(year, V_DATE_1)||'-09-01') = 'Sat'
                                and dateadd(day,2,(date_part(year, V_DATE_1)||'-09-01')) =
V_DATE_1  then
                                'Holiday'
                                when monthname(V_DATE_1) ='Sep' and
dayname(date_part(year, V_DATE_1)||'-09-01') = 'Sun'
```

```
                        and dateadd(day,1,(date_part(year, V_DATE_1)||'-09-01')) =
V_DATE_1 then
                        'Holiday'
                        when monthname(V_DATE_1) ='Sep' and
dayname(date_part(year, V_DATE_1)||'-09-01') = 'Mon'
        and date_part(year, V_DATE_1)||'-09-01' = V_DATE_1 then
                        'Holiday'
                        when monthname(V_DATE_1) ='Sep' and
dayname(date_part(year, V_DATE_1)||'-09-01') = 'Tue'
                        and dateadd(day,6 ,(date_part(year, V_DATE_1)||'-09-01')) =
V_DATE_1  then
                        'Holiday'
                        when monthname(V_DATE_1) ='Nov' and
dayname(date_part(year, V_DATE_1)||'-11-01') = 'Wed'
                        and dateadd(day,23,(date_part(year, V_DATE_1)||'-11-01')) =
V_DATE_1 then
                        'Holiday'
                        when monthname(V_DATE_1) ='Nov' and
dayname(date_part(year, V_DATE_1)||'-11-01') = 'Thu'
                        and dateadd(day,22,(date_part(year, V_DATE_1)||'-11-01')) =
V_DATE_1 then
                        'Holiday'
                        when monthname(V_DATE_1) ='Nov' and
dayname(date_part(year, V_DATE_1)||'-11-01') = 'Fri'
                        and dateadd(day,21,(date_part(year, V_DATE_1)||'-11-01')) =
V_DATE_1  then
                         'Holiday'
                        when monthname(V_DATE_1) ='Nov' and
dayname(date_part(year, V_DATE_1)||'-11-01') = 'Sat'
```

```
                and dateadd(day,27,(date_part(year, V_DATE_1)||'-11-01')) =
V_DATE_1 then
                'Holiday'
                when monthname(V_DATE_1) ='Nov' and
dayname(date_part(year, V_DATE_1)||'-11-01') = 'Sun'
                and dateadd(day,26,(date_part(year, V_DATE_1)||'-11-01')) =
V_DATE_1 then
                'Holiday'
                when monthname(V_DATE_1) ='Nov' and
dayname(date_part(year, V_DATE_1)||'-11-01') = 'Mon'
                and dateadd(day,25,(date_part(year, V_DATE_1)||'-11-01')) =
V_DATE_1 then
                'Holiday'
                when monthname(V_DATE_1) ='Nov' and
dayname(date_part(year, V_DATE_1)||'-11-01') = 'Tue'
                and dateadd(day,24,(date_part(year, V_DATE_1)||'-11-01')) =
V_DATE_1  then
                 'Holiday'
                else
                'Not-Holiday' end as COMPANY_HOLIDAY_IND,
            case
                when last_day(V_DATE_1) = V_DATE_1 then
                'Month-end'
                else 'Not-Month-end' end as MONTH_END_IND,


            case when date_part(mm,date_trunc('week',V_DATE_1)) < 10 and
date_part(dd,date_trunc('week',V_DATE_1)) < 10 then
                    date_part(yyyy,date_trunc('week',V_DATE_1))||'0'||
                                          date_part(mm,date_trunc('week',V_DATE_1))||'0'||
                                          date_part(dd,date_trunc('week',V_DATE_1))
                                  when date_part(mm,date_trunc('week',V_DATE_1)) < 10 and
date_part(dd,date_trunc('week',V_DATE_1)) > 9 then

        date_part(yyyy,date_trunc('week',V_DATE_1))||'0'||

        date_part(mm,date_trunc('week',V_DATE_1))||date_part(dd,date_trunc('week',V_DATE
_1))
                                  when date_part(mm,date_trunc('week',V_DATE_1)) > 9 and
date_part(dd,date_trunc('week',V_DATE_1)) < 10 then

        date_part(yyyy,date_trunc('week',V_DATE_1))||date_part(mm,date_trunc('week',V_DAT
E_1))||
                                                  '0'||date_part(dd,date_trunc('week',V_DATE_1))
                                 when date_part(mm,date_trunc('week',V_DATE_1)) > 9 and
date_part(dd,date_trunc('week',V_DATE_1)) > 9 then
                                                  date_part(yyyy,date_trunc('week',V_DATE_1))||
                                                  date_part(mm,date_trunc('week',V_DATE_1))||
                                                  date_part(dd,date_trunc('week',V_DATE_1)) end as
WEEK_BEGIN_DATE_NKEY,
                         date_trunc('week',V_DATE_1) as WEEK_BEGIN_DATE,


                         case when  date_part(mm,last_day(V_DATE_1,'week')) < 10 and
date_part(dd,last_day(V_DATE_1,'week')) < 10 then
                                          date_part(yyyy,last_day(V_DATE_1,'week'))||'0'||
                                          date_part(mm,last_day(V_DATE_1,'week'))||'0'||
                                          date_part(dd,last_day(V_DATE_1,'week'))
```

```
                               when  date_part(mm,last_day(V_DATE_1,'week')) < 10 and
date_part(dd,last_day(V_DATE_1,'week')) > 9 then
                                       date_part(yyyy,last_day(V_DATE_1,'week'))||'0'||

date_part(mm,last_day(V_DATE_1,'week'))||date_part(dd,last_day(V_DATE_1,'week'))
                               when  date_part(mm,last_day(V_DATE_1,'week')) > 9 and
date_part(dd,last_day(V_DATE_1,'week')) < 10  then

date_part(yyyy,last_day(V_DATE_1,'week'))||date_part(mm,last_day(V_DATE_1,'week'))||'0'||
                                       date_part(dd,last_day(V_DATE_1,'week'))
                               when  date_part(mm,last_day(V_DATE_1,'week')) > 9 and
date_part(dd,last_day(V_DATE_1,'week')) > 9 then
                                       date_part(yyyy,last_day(V_DATE_1,'week'))||
                                       date_part(mm,last_day(V_DATE_1,'week'))||
                                       date_part(dd,last_day(V_DATE_1,'week')) end as
WEEK_END_DATE_NKEY,
                       last_day(V_DATE_1,'week') as WEEK_END_DATE,
                       week(V_DATE_1) as WEEK_NUM_IN_YEAR,
                       case when monthname(V_DATE_1) ='Jan' then 'January'
                                when monthname(V_DATE_1) ='Feb' then 'February'
                                when monthname(V_DATE_1) ='Mar' then 'March'
                                when monthname(V_DATE_1) ='Apr' then 'April'
                                when monthname(V_DATE_1) ='May' then 'May'
                                when monthname(V_DATE_1) ='Jun' then 'June'
                                when monthname(V_DATE_1) ='Jul' then 'July'
                                when monthname(V_DATE_1) ='Aug' then 'August'
                                when monthname(V_DATE_1) ='Sep' then 'September'
                                when monthname(V_DATE_1) ='Oct' then 'October'
```

```
                              when monthname(V_DATE_1) ='Nov' then 'November'
                              when monthname(V_DATE_1) ='Dec' then 'December' end as
MONTH_NAME,
                     monthname(V_DATE_1) as MONTH_ABBREV,
                     month(V_DATE_1) as MONTH_NUM_IN_YEAR,
                     case when month(V_DATE_1) < 10 then
                     year(V_DATE_1)||'-0'||month(V_DATE_1)
                     else year(V_DATE_1)||'-'||month(V_DATE_1) end as YEARMONTH,
                     quarter(V_DATE_1) as CURRENT_QUARTER,
                     year(V_DATE_1)||'-0'||quarter(V_DATE_1) as YEARQUARTER,
                     year(V_DATE_1) as CURRENT_YEAR,
                     /*Modify the following based on company fiscal year - assumes Jan 01*/
      to_date(year(V_DATE_1)||'-01-01','YYYY-MM-DD') as FISCAL_CUR_YEAR,
      to_date(year(V_DATE_1) -1||'-01-01','YYYY-MM-DD') as FISCAL_PREV_YEAR,
                     case when   V_DATE_1 < FISCAL_CUR_YEAR then
                     datediff('week', FISCAL_PREV_YEAR,V_DATE_1)
                     else
                     datediff('week', FISCAL_CUR_YEAR,V_DATE_1)  end as
FISCAL_WEEK_NUM  ,
                     decode(datediff('MONTH',FISCAL_CUR_YEAR, V_DATE_1)+1 ,-2,10,-
1,11,0,12,
          datediff('MONTH',FISCAL_CUR_YEAR, V_DATE_1)+1 ) as FISCAL_MONTH_NUM,
                     concat( year(FISCAL_CUR_YEAR)
                              ,case when to_number(FISCAL_MONTH_NUM) = 10 or
                                                 to_number(FISCAL_MONTH_NUM) = 11 or
               to_number(FISCAL_MONTH_NUM) = 12  then
                                                 '-'||FISCAL_MONTH_NUM
```

```
                                        else  concat('-0',FISCAL_MONTH_NUM) end ) as
FISCAL_YEARMONTH,
                        case when quarter(V_DATE_1) = 4 then 4
                                 when quarter(V_DATE_1) = 3 then 3
                                 when quarter(V_DATE_1) = 2 then 2
                                 when quarter(V_DATE_1) = 1 then 1 end as FISCAL_QUARTER,


                        case when   V_DATE_1 < FISCAL_CUR_YEAR then
                                        year(FISCAL_CUR_YEAR)
                                        else year(FISCAL_CUR_YEAR)+1 end
                                        ||'-0'||case when quarter(V_DATE_1) = 4 then 4
                                         when quarter(V_DATE_1) = 3 then 3
                                         when quarter(V_DATE_1) = 2 then 2
                                         when quarter(V_DATE_1) = 1 then 1 end as
FISCAL_YEARQUARTER,
                        case when quarter(V_DATE_1) = 4  then 2 when quarter(V_DATE_1) = 3
then 2
                                when quarter(V_DATE_1) = 1  then 1 when quarter(V_DATE_1) = 2
then 1
                        end as FISCAL_HALFYEAR,
                        year(FISCAL_CUR_YEAR) as FISCAL_YEAR,
                        to_timestamp_ntz(V_DATE) as SQL_TIMESTAMP,
                        'Y' as CURRENT_ROW_IND,
                        to_date(current_timestamp) as EFFECTIVE_DATE,
                        to_date('9999-12-31') as EXPIRA_DATE
                        --from table(generator(rowcount => 8401)) /*<< Set to generate 20 years.
Modify rowcount to increase or decrease size*/
```

```
                from table(generator(rowcount => 730)) /*<< Set to generate 20 years. Modify rowcount to increase or decrease size*/
   )v;


--Miscellaneous queries
--select * from  DIM_DATE
--ORDER BY DATE;



--delete from DIM_DATE;
--select* from dim_product;


-- =============================================
-- DIM_PRODUCT
-- =============================================
-- Create DIM_PRODUCT table
CREATE OR REPLACE TABLE DIM_PRODUCT (
   DimProductID INT IDENTITY(1,1) CONSTRAINT PK_DimProductID PRIMARY KEY NOT NULL, --Surrogate Key(Identity key autoincremets it by one step size)
   ProductID INT,
   ProductTypeID	INT,
   ProductCategoryID INT,
   ProductName VARCHAR(255),
   ProductType VARCHAR(255),
   ProductCategory VARCHAR(255),
   Product_Cost FLOAT(10),
   Product_RetailPrice FLOAT(10),
    Product_WholesalePrice FLOAT,
    Product_RetailProfit FLOAT,
    Product_WholesaleProfit FLOAT,
    Product_ProfitMarginPercent FLOAT
    );

-- Insert matched product data
INSERT INTO DIM_PRODUCT (
    ProductID,
    ProductTypeID,
    ProductCategoryID,
    ProductName,
    ProductType,
    ProductCategory,
    Product_Cost,
    Product_RetailPrice,
    Product_WholesalePrice,
    Product_RetailProfit,
    Product_WholesaleProfit,
    Product_ProfitMarginPercent
)
SELECT
    CAST(sp.ProductID AS INT) AS ProductID,-- the inital data had these as character dataype hence typecasting to integer
    CAST(spt.PRODUCTTYPEID AS INT) AS ProductTypeID,
    CAST(spc.PRODUCTCATEGORYID AS INT) AS ProductCategoryID,
    sp.Product,

    spt.PRODUCTTYPE,

    spc.PRODUCTCATEGORY,

    Price,

    WholesalePrice,

    Cost,

    Price - Cost AS Product_RetailProfit,

    WholesalePrice - Cost AS Product_WholesaleProfit,

    ROUND(COALESCE((((COALESCE(sp.Price - sp.Cost, 0) / COALESCE(sp.Price, 1)) * 100) +
((COALESCE(sp.WholesalePrice - sp.Cost, 0) / COALESCE(sp.WholesalePrice, 1)) * 100)) / 2, -1), 2)
AS Product_ProfitMarginPercentage

FROM staging_product sp

LEFT JOIN STAGING_PRODUCTTYPE spt ON sp.ProductTypeID = spt.PRODUCTTYPEID

LEFT JOIN STAGING_PRODUCTCATEGORY spc ON spt.PRODUCTCATEGORYID =
spc.PRODUCTCATEGORYID;


-- inserting unknowns

INSERT INTO DIM_PRODUCT

(

    DimProductID,

    ProductID,

    ProductTypeID,

    ProductCategoryID,

    ProductName,

    ProductType,

    ProductCategory,

    Product_Cost,
    Product_RetailPrice,
    Product_WholesalePrice,
    Product_RetailProfit,
    Product_WholesaleProfit,
    Product_ProfitMarginPercent
)
VALUES
(
    -1, --int
    -1,
    -1,
    -1,
    'Unknown',--varchar
    'Unknown',
    'Unknown',
    -1.0,--float
    -1.0,
    -1.0,
    -1.0,
    -1.0,
    -1.0

);
--we are creating the unknows in all these tables because we will be referring to these tables in our fact table and fact tables cannot have unknowns, hence if there is any value that is unknown in our initial excel files, we will replace it with these unknown values
```

```
SELECT * FROM DIM_PRODUCT;


-- ===============================================
-- DIM_CUSTOMER
-- ===============================================
-- Create DIM_CUSTOMER table
CREATE OR REPLACE TABLE DIM_CUSTOMER(
   DimCustomerID INT IDENTITY(1,1) CONSTRAINT PK_DimCustomerID PRIMARY KEY NOT NULL,
   DimLocationID INT,
   CustomerID VARCHAR(255),
   CustomerFullName VARCHAR(255),
   CustomerFirstName VARCHAR(255),
   CustomerLastName VARCHAR(255),
   CustomerGender VARCHAR(255),
   CONSTRAINT FK_Customer_Location FOREIGN KEY (DimLocationID) REFERENCES
DIM_LOCATION(DimLocationID)
 );

-- Insert matched customer data
INSERT INTO DIM_CUSTOMER (
   DimLocationID,
   CustomerID,
   CustomerFullName,
   CustomerFirstName,
   CustomerLastName,
   CustomerGender
)
SELECT
   CAST(L.DimLocationID AS INT) AS DimLocationID,
   C.CUSTOMERID,
   CONCAT(C.FIRSTNAME, ' ', C.LASTNAME) AS CustomerFullName,
   C.FIRSTNAME AS CustomerFirstName,
   C.LASTNAME AS CustomerLastName,
   C.GENDER AS CustomerGender,

FROM
   STAGING_CUSTOMER C
LEFT JOIN
   DIM_LOCATION L ON C.ADDRESS = L.Address
               AND C.CITY = L.City
               AND C.POSTALCODE = L.PostalCode
               AND C.STATEPROVINCE = L.State_Province
               AND C.COUNTRY = L.Country;
SELECT * FROM DIM_CUSTOMER;

-- inserting unknowns
INSERT INTO DIM_CUSTOMER
(
   DimCustomerID,
   DimLocationID,
   CustomerID,
   CustomerFullName,

   CustomerFirstName,

   CustomerLastName,

   CustomerGender


)

VALUES

(

   -1,

   -1,

   'Unknown',

   'Unknown',

   'Unknown',

   'Unknown',

   'Unknown'


);


select * from dim_customer;



-- ====================================================

-- DIM_LOCATION

-- ====================================================

-- Create DIM_LOCATION table

CREATE OR REPLACE TABLE DIM_LOCATION (
   DimLocationID  INT AUTOINCREMENT PRIMARY KEY,
   Address VARCHAR(255),
   City VARCHAR(255),
   State_Province VARCHAR(255),
   PostalCode VARCHAR(255),
   Country VARCHAR(255)
   );

-- Insert matched location data
INSERT INTO DIM_LOCATION (
   Address,
   City,
   PostalCode,
   State_Province,
   Country
)
SELECT
       r.Address AS Address,
       r.City AS City,
       r.StateProvince AS State_Province,
       r.PostalCode AS PostalCode,
       r.Country AS Country
   FROM
       STAGING_RESELLER r

   UNION
   SELECT
       c.Address AS Address,
       c.City AS City,
       c.StateProvince AS State_Province,
       c.PostalCode AS PostalCode,
       c.Country AS Country
   FROM
       STAGING_CUSTOMER c

   UNION

   SELECT
       s.Address AS Address,
       s.City AS City,
       s.StateProvince AS State_Province,
       s.PostalCode AS PostalCode,
       s.Country AS Country
   FROM
       STAGING_STORE s;

SELECT * FROM DIM_LOCATION;



-- ============================================
-- DIM_STORE

-- =============================================

-- Create DIM_STORE table

CREATE OR REPLACE TABLE DIM_STORE(

    DimStoreID INT IDENTITY(1,1) CONSTRAINT PK_DimStoreID PRIMARY KEY NOT NULL,

    DimLocationID INT,

    StoreID INT,

    StoreNumber INT,

    StoreManager VARCHAR(255),

    CONSTRAINT FK_Store_Location FOREIGN KEY (DimLocationID) REFERENCES DIM_LOCATION(DimLocationID)


);


-- Insert matched store data

INSERT INTO DIM_STORE (

    DimLocationID,

    StoreID,

    StoreNumber,

    StoreManager

)

SELECT

    l.DimLocationID,

    CAST(s.STOREID AS INT),

    CAST(s.STORENUMBER AS INT),

    s.STOREMANAGER
```

```
FROM STAGING_STORE s
JOIN DIM_LOCATION l
  ON TRIM(UPPER(s.ADDRESS)) = TRIM(UPPER(l.Address))
 AND TRIM(UPPER(s.CITY)) = TRIM(UPPER(l.City))
 AND TRIM(UPPER(s.POSTALCODE)) = TRIM(UPPER(l.PostalCode))
 AND TRIM(UPPER(s.STATEPROVINCE)) = TRIM(UPPER(l.State_Province))
 AND TRIM(UPPER(s.COUNTRY)) = TRIM(UPPER(l.Country));

-- inserting unknowns

INSERT INTO DIM_STORE

(
    DimLocationID,
    StoreID,
    StoreNumber,
    StoreManager
)
VALUES
(
    -1,
    -1,
    -1,
    'Unknown'

);
```

```
SELECT * FROM DIM_STORE;



-- =============================================
-- DIM_RESELLER
-- =============================================

-- Create DIM_RESELLER table
CREATE OR REPLACE TABLE DIM_RESELLER (
    DimResellerID INT IDENTITY(1,1) CONSTRAINT PK_DimResellerID PRIMARY KEY NOT NULL,
    DimLocationID INT,
    ResellerID VARCHAR(255),
    ResellerName VARCHAR(255),
    Contact VARCHAR(255),
    PhoneNumber VARCHAR(255),
    EmailAddress VARCHAR(255),
    CONSTRAINT FK_Reseller_Location FOREIGN KEY (DimLocationID) REFERENCES
DIM_LOCATION(DimLocationID)
);

-- Insert matched reseller data
INSERT INTO DIM_RESELLER (
    DimLocationID,
    ResellerID,
    ResellerName,
    Contact,
    PhoneNumber,
    EmailAddress
)
SELECT
    l.DimLocationID,
    r.RESELLERID,
    r.RESELLERNAME,
    r.CONTACT,
    r.PHONENUMBER,
    r.EMAILADDRESS
FROM IMT577_DW_APEKSHA_TEJWANI_STAGING.PUBLIC.STAGING_RESELLER r
LEFT JOIN DIM_LOCATION l
  ON REPLACE(TRIM(UPPER(r.ADDRESS)), '.', '') = REPLACE(TRIM(UPPER(l.Address)), '.', '')
 AND REPLACE(TRIM(UPPER(r.CITY)), '.', '') = REPLACE(TRIM(UPPER(l.City)), '.', '')
 AND REPLACE(TRIM(UPPER(r.POSTALCODE)), ' ', '') = REPLACE(TRIM(UPPER(l.PostalCode)), ' ', '')
 AND REPLACE(TRIM(UPPER(r.STATEPROVINCE)), '.', '') =
REPLACE(TRIM(UPPER(l.State_Province)), '.', '')
 AND REPLACE(TRIM(UPPER(r.COUNTRY)), '.', '') = REPLACE(TRIM(UPPER(l.Country)), '.', '');

-- inserting unknowns
INSERT INTO DIM_RESELLER

(
    DimResellerID,
    DimLocationID,
    ResellerID,
    ResellerName,
    Contact,
    PhoneNumber,
    EmailAddress
)
VALUES
(
    -1,
    -1,
    'Unknown',
    'Unknown',
    'Unknown',
    'Unknown',
    'Unknown'

);

select * from dim_reseller;



-- =============================================
-- DIM_CHANNEL
-- =============================================
-- Create DIM_CHANNEL table
CREATE OR REPLACE TABLE DIM_CHANNEL(
DimChannelID INT IDENTITY(1,1) CONSTRAINT PK_DimPChannelID PRIMARY KEY NOT NULL, --
Surrogate Key(Identity key autoincremets it by one step size)
   ChannelID INT,
   ChannelCategoryID  INT,
   ChannelName VARCHAR(255),
   ChannelCategory VARCHAR(255)


);


-- Insert matched channel data
INSERT INTO DIM_CHANNEL (
   ChannelID,
   ChannelCategoryID,
   ChannelName,
   ChannelCategory
)
SELECT
   CAST(c.ChannelID AS INT) AS ChannelID,-- the inital data had these as character dataype
hence typecasting to integer
   CAST(c.ChannelCategoryID AS INT) AS ChannelCategoryID,
   c.Channel,
   cc.ChannelCategory,
FROM STAGING_CHANNEL c
```

```
LEFT JOIN STAGING_CHANNELCATEGORY cc ON c.CHANNELCATEGORYID = cc.CHANNELCATEGORYID;


-- inserting unknowns
INSERT INTO DIM_CHANNEL (
   DimChannelID,ChannelID,
   ChannelCategoryID,
   ChannelName,
   ChannelCategory
)
VALUES (
   -1,-1,          -- Default ChannelID and DimChannelID
   -1,          -- Default ChannelCategoryID
   'Unknown',      -- ChannelName
   'Unknown'      -- ChannelCategory
);
SELECT * FROM DIM_CHANNEL;
```