```
--Create Clients table in landing
create or replace TABLE "MYDB"."CLOCKIFY_LANDING".CLIENTS (
address VARCHAR,
archived BOOLEAN,
currencyCode VARCHAR,
currencyId VARCHAR,
email VARCHAR,
id VARCHAR,
name VARCHAR,
note VARCHAR,
workspaceId VARCHAR
);

--Insert into Clients table
insert into "MYDB"."CLOCKIFY_LANDING".CLIENTS (
address,
archived,
currencyCode,
currencyId,
email,
id,
name,
note,
workspaceId
)
select
value:address::varchar as address,
value:archived::boolean as archived,
value:currencyCode::varchar as currencyCode,
value:currencyId::varchar as currencyId,
value:email::varchar as email,
value:id::varchar as id,
value:name::varchar as name,
value:note::varchar as note,
value:workspaceId::varchar as workspaceId
from dwh_ucla2024_clockify.clients as cl;

--select count(*) from dwh_ucla2024_clockify.clients;
--168 rows

----------

--Create Projects table in landing
create or replace TABLE "MYDB"."CLOCKIFY_LANDING".PROJECTS (
```

```
archived BOOLEAN,
billable BOOLEAN,
budgetEstimateActive BOOLEAN,
budgetEstimate NUMBER(9,0),
budgetEstimateIncludeExpenses BOOLEAN,
budgetEstimateResetOption VARCHAR,
budgetEstimateType VARCHAR,
clientId VARCHAR,
clientName VARCHAR,
color VARCHAR,
costRate NUMBER(7,3),
duration VARCHAR,
estimateDuration VARCHAR,
estimateType VARCHAR,
hourlyRateAmount NUMBER(7,3),
hourlyRateCurrency VARCHAR,
id VARCHAR,
memberships ARRAY,
name VARCHAR,
note VARCHAR,
public BOOLEAN,
template BOOLEAN,
timeEstimateActive BOOLEAN,
timeEstimate VARCHAR,
timeEstimateincludeNonBillable BOOLEAN,
timeEstimateResetOption VARCHAR,
timeEstimateType VARCHAR,
workspaceId VARCHAR
);

--Insert into Projects table
insert into "MYDB"."CLOCKIFY_LANDING".PROJECTS(
archived,
billable,
budgetEstimateActive,
budgetEstimate,
budgetEstimateIncludeExpenses,
budgetEstimateResetOption,
budgetEstimateType,
clientId,
clientName,
color,
costRate,
duration,
```

```
estimateDuration,
estimateType,
hourlyRateAmount,
hourlyRateCurrency,
id,
memberships,
name,
note,
public,
template,
timeEstimateActive,
timeEstimate,
timeEstimateincludeNonBillable,
timeEstimateResetOption,
timeEstimateType,
workspaceId
)
select
value:archived::boolean as archived,
value:billable::boolean as billable,
value:budgetEstimate:active::boolean as budgetEstimateActive,
value:budgetEstimate:estimate::number(9,0) as budgetEstimate,
value:budgetEstimate:includeExpenses::boolean as budgetEstimateIncludeExpenses,
value:budgetEstimate:resetOption::varchar as budgetEstimateResetOption,
value:budgetEstimate:type::varchar as budgetEstimateType,
value:clientId::varchar as clientId,
value:clientName::varchar as clientName,
value:color::varchar as color,
value:costRate::number(7,3) as costRate,
value:duration::varchar as duration,
value:estimate:estimate::varchar as estimateDuration,
value:estimate:type::varchar as estimateType,
value:hourlyRate:amount::number(7,3) as hourlyRateAmount,
value:hourlyRate:currency::varchar as hourlyRateCurrency,
value:id::varchar as id,
value:memberships::array as memberships,
value:name::varchar as name,
value:note::varchar as note,
value:public::boolean as public,
value:template::boolean as template,
value:timeEstimate:active::boolean as timeEstimateActive,
value:timeEstimate:estimate::varchar as timeEstimate,
value:timeEstimate:includeNonBillable::boolean as timeEstimateincludeNonBillable,
value:timeEstimate:resetOption::varchar as timeEstimateResetOption,
```

```
value:timeEstimate:type::varchar as timeEstimateType,
value:workspaceId::varchar as workspaceId
from dwh_ucla2024_clockify.projects as p;

--select count(*) from dwh_ucla2024_clockify.projects;
--434 rows

----------

--Create Tasks table in landing
create or replace TABLE "MYDB"."CLOCKIFY_LANDING".TASKS (
assigneeId VARCHAR,
assigneeIds ARRAY,
billable BOOLEAN,
budgetEstimate NUMBER(7,3),
costRate NUMBER(7,3),
duration VARCHAR,
estimate VARCHAR,
hourlyRate NUMBER(7,3),
id VARCHAR,
name VARCHAR,
projectId VARCHAR,
status VARCHAR,
userGroupIds ARRAY
);


--Insert into Tasks table
insert into "MYDB"."CLOCKIFY_LANDING".TASKS (
assigneeId,
assigneeIds,
billable,
budgetEstimate,
costRate,
duration,
estimate,
hourlyRate,
id,
name,
projectId,
status,
userGroupIds
)
select
```

```
value:assigneeId::varchar as assigneeId,
value:assigneeIds::array as assigneeIds,
value:billable::boolean as billable,
value:budgetEstimate::number(7,3) as budgetEstimate,
value:costRate::number(7,3) as costRate,
value:duration::varchar as duration,
value:estimate::varchar as estimate,
value:hourlyRate::number(7,3) as hourlyRate,
value:id::varchar as id,
value:name::varchar as name,
value:projectId::varchar as projectId,
value:status::varchar as status,
value:userGroupIds::array as userGroupIds
from dwh_ucla2024_clockify.tasks as ts;

--select count(*) from dwh_ucla2024_clockify.tasks;
--2384 rows

----------

--Create Time_Entries table in landing
create or replace TABLE "MYDB"."CLOCKIFY_LANDING".TIME_ENTRIES (
billable BOOLEAN,
customFieldValues ARRAY,
description VARCHAR,
id VARCHAR,
isLocked BOOLEAN,
kioskId VARCHAR,
projectId VARCHAR,
tagIds ARRAY,
taskId VARCHAR,
startTime TIMESTAMPTZ,
endTime TIMESTAMPTZ,
duration VARCHAR,
type VARCHAR,
userId VARCHAR,
workspaceId VARCHAR,
loadTimestamp TIMESTAMPTZ
);

--Insert into Time_entries table
insert into "MYDB"."CLOCKIFY_LANDING".TIME_ENTRIES (
billable,
customFieldValues,
```

```
description,
id,
isLocked,
kioskId,
projectId,
tagIds,
taskId,
startTime,
endTime,
duration,
type,
userId,
workspaceId,
loadTimestamp
)
select
json_data:billable::boolean as billable,
json_data:customFieldValues::array as customFieldValues,
json_data:description::varchar as description,
json_data:id::varchar as id,
json_data:isLocked::boolean as isLocked,
json_data:kioskId::varchar as kioskId,
json_data:projectId::varchar as projectId,
json_data:tagIds::array as tagIds,
json_data:taskId::varchar as taskId,
json_data:timeInterval:start::TIMESTAMPTZ as startTime,
json_data:timeInterval:end::TIMESTAMPTZ as endTime,
json_data:timeInterval:duration::varchar as duration,
json_data:type::varchar as type,
json_data:userId::varchar as userId,
json_data:workspaceId::varchar as workspaceId,
load_timestamp as loadTimestamp
from dwh_ucla2024_clockify.time_entries as te;

--select count(*) from dwh_ucla2024_clockify.time_entries;
--40194 rows

----------

--Create Users Table in landing
create or replace TABLE "MYDB"."CLOCKIFY_LANDING".USERS (
activeWorkspace VARCHAR,
customFields ARRAY,
defaultWorkspace VARCHAR,
```

```
email VARCHAR,
id VARCHAR,
memberships ARRAY,
name VARCHAR,
profilePicture VARCHAR,
alerts BOOLEAN,
approval BOOLEAN,
collapseAllProjectLists BOOLEAN,
dashboardPinToTop BOOLEAN,
dashboardSelection VARCHAR,
dashboardViewType VARCHAR,
dateFormat VARCHAR,
groupSimilarEntriesDisabled BOOLEAN,
isCompactViewOn BOOLEAN,
lang VARCHAR,
longRunning BOOLEAN,
multiFactorEnabled BOOLEAN,
myStartOfDay VARCHAR,
onboarding BOOLEAN,
projectListCollapse NUMBER(3,0),
projectPickerTaskFilter BOOLEAN,
pto BOOLEAN,
reminders BOOLEAN,
scheduledReports BOOLEAN,
scheduling BOOLEAN,
sendNewsletter BOOLEAN,
showOnlyWorkingDays BOOLEAN,
summaryReportSettingsGroup VARCHAR,
summaryReportSettingsSubGroup VARCHAR,
summaryReportSettingsGroupTheme VARCHAR,
summaryReportSettingsGroupTimeFormat VARCHAR,
summaryReportSettingsGroupTimeTrackingManual BOOLEAN,
summaryReportSettingsGroupTimeZone VARCHAR,
summaryReportSettingsGroupWeekStart VARCHAR,
summaryReportSettingsGroupWeeklyUpdates BOOLEAN,
status VARCHAR
);

--Insert into Users Tables
insert into "MYDB"."CLOCKIFY_LANDING".USERS (
activeWorkspace,
customFields,
defaultWorkspace,
email,
```

```
id,
memberships,
name,
profilePicture,
alerts,
approval,
collapseAllProjectLists,
dashboardPinToTop,
dashboardSelection,
dashboardViewType,
dateFormat,
groupSimilarEntriesDisabled,
isCompactViewOn,
lang,
longRunning,
multiFactorEnabled,
myStartOfDay,
onboarding,
projectListCollapse,
projectPickerTaskFilter,
pto,
reminders,
scheduledReports,
scheduling,
sendNewsletter,
showOnlyWorkingDays,
summaryReportSettingsGroup,
summaryReportSettingsSubGroup,
summaryReportSettingsGroupTheme,
summaryReportSettingsGroupTimeFormat,
summaryReportSettingsGroupTimeTrackingManual,
summaryReportSettingsGroupTimeZone,
summaryReportSettingsGroupWeekStart,
summaryReportSettingsGroupWeeklyUpdates,
status
)
select
value:activeWorkspace::varchar as activeWorkspace,
value:customFields::array as customFields,
value:defaultWorkspace::varchar as defaultWorkspace,
value:email::varchar as email,
value:id::varchar as id,
value:memberships::array as memberships,
value:name::varchar as name,
```

```
value:profilePicture::varchar as profilePicture,
value:settings:alerts::boolean as alerts,
value:settings:approval::boolean as approval,
value:settings:collapseAllProjectLists::boolean as collapseAllProjectLists,
value:settings:dashboardPinToTop::boolean as dashboardPinToTop,
value:settings:dashboardSelection::varchar as dashboardSelection,
value:settings:dashboardViewType::varchar as dashboardViewType,
value:settings:dateFormat::varchar as dateFormat,
value:settings:groupSimilarEntriesDisabled::boolean as groupSimilarEntriesDisabled,
value:settings:isCompactViewOn::boolean as isCompactViewOn,
value:settings:lang::varchar as lang,
value:settings:longRunning::boolean as longRunning,
value:settings:multiFactorEnabled::boolean as multiFactorEnabled,
value:settings:myStartOfDay::varchar as myStartOfDay,
value:settings:onboarding::boolean as onboarding,
value:settings:projectListCollapse::number(3,0) as projectListCollapse,
value:settings:projectPickerTaskFilter::boolean as projectPickerTaskFilter,
value:settings:pto::boolean as pto,
value:settings:reminders::boolean as reminders,
value:settings:scheduledReports::boolean as scheduledReports,
value:settings:scheduling::boolean as scheduling,
value:settings:sendNewsletter::boolean as sendNewsletter,
value:settings:showOnlyWorkingDays::boolean as showOnlyWorkingDays,
value:settings:summaryReportSettings:group::varchar as summaryReportSettingsGroup,
value:settings:summaryReportSettings:subgroup::varchar as
summaryReportSettingsSubGroup,
value:settings:theme::varchar as summaryReportSettingsGroupTheme,
value:settings:timeFormat::varchar as summaryReportSettingsGroupTimeFormat,
value:settings:timeTrackingManual::boolean as
summaryReportSettingsGroupTimeTrackingManual,
value:settings:timeZone::varchar as summaryReportSettingsGroupTimeZone,
value:settings:weekStart::varchar as summaryReportSettingsGroupWeekStart,
value:settings:weeklyUpdates::boolean as summaryReportSettingsGroupWeeklyUpdates,
value:status::varchar as status
from dwh_ucla2024_clockify.users as u;

--select count(*) from dwh_ucla2024_clockify.users;
--119 rows

use warehouse my_wh;
use database mydb;
use schema CLOCKIFY_ANALYTICS;
USE SCHEMA
FWLOYYD_PZA82601_DATASLEEK_DW_UCLA2024.DWH_UCLA2024_CLOCKIFY;
```

```
--Create Clockify_Analytics Schema
create or replace SCHEMA "MYDB"."CLOCKIFY_ANALYTICS";

--create clients sequence
create or replace sequence CLIENTS_DIM;

--select * from "CLOCKIFY_LANDING".clients;

--Create Clients table in landing
create or replace TABLE "MYDB"."CLOCKIFY_ANALYTICS".CLIENTS_DIM (
client_key NUMBER(38,0) NOT NULL autoincrement,
client_id VARCHAR,
client_name VARCHAR,
client_email VARCHAR,
client_address VARCHAR,
client_workspace_id VARCHAR
);

--Insert into Clients table
insert into "MYDB"."CLOCKIFY_ANALYTICS".CLIENTS_DIM (
client_key,
client_id,
client_name,
client_email,
client_address,
client_workspace_id
)
select
CLIENTS_DIM.nextval as client_key,
id,
name,
email,
address,
workspaceId
from "MYDB"."CLOCKIFY_LANDING".clients as cl;

----------

--create projects sequence
create or replace sequence PROJECTS_DIM;

select * from "CLOCKIFY_LANDING".projects;
```

```
--Create Projects table in analytics
create or replace TABLE "MYDB"."CLOCKIFY_ANALYTICS".PROJECTS_DIM (
project_key NUMBER(38,0),
project_id VARCHAR,
project_name VARCHAR,
project_short_name VARCHAR,
project_billable BOOLEAN,
project_client_key NUMBER(38,0),
project_duration_hours NUMBER(8,3),
project_estimate_duration_hours NUMBER(7,3),
project_estimate_type VARCHAR,
project_workspace_id VARCHAR
);


--Insert into Projects table
insert into "MYDB"."CLOCKIFY_ANALYTICS".PROJECTS_DIM(
project_key,
project_id,
project_name,
project_short_name,
project_billable,
project_client_key,
project_duration_hours,
project_estimate_duration_hours,
project_workspace_id
)
select
PROJECTS_DIM.nextval as project_key,
p.id,
p.name,
trim(split(p.name,'-')[0]),
p.billable,
cl.client_key,
((case
  when charindex('H',p.duration)>0 then
    substr(p.duration,3, charindex('H',p.duration)-charindex('PT',p.duration)-2)
else '0' end) +
((case
  when (charindex('H',p.duration)>0) and (charindex('M',p.duration)>0) then
    substr(p.duration,charindex('H',p.duration)+1, charindex('M',p.duration)-
charindex('H',p.duration)-1)
  when (charindex('H',p.duration)=0) and (charindex('M',p.duration)>0) then
    substr(p.duration,3, charindex('M',p.duration)-charindex('PT',p.duration)-2)
```

```
else '0' end)/60) +
((case
  when (charindex('M',p.duration)>0) and (charindex('S',p.duration)>0) then
    substr(p.duration,charindex('M',p.duration)+1, charindex('S',p.duration)-
charindex('M',p.duration)-1)
  when (charindex('H',p.duration)>0) and (charindex('M',p.duration)=0) and
(charindex('S',p.duration)>0) then
    substr(p.duration,charindex('H',p.duration)+1, charindex('S',p.duration)-
charindex('H',p.duration)-1)
  when (charindex('H',p.duration)=0) and (charindex('M',p.duration)=0) and
(charindex('S',p.duration)>0) then
    substr(p.duration,charindex('PT',p.duration)+2, charindex('S',p.duration)-
charindex('PT',p.duration)-2)
else '0' end)/3600)) as duration_hours,
((case
  when charindex('H',p.estimateDuration)>0 then
    substr(p.estimateDuration,3, charindex('H',p.estimateDuration)-
charindex('PT',p.estimateDuration)-2)
else '0' end) +
((case
  when (charindex('H',p.estimateDuration)>0) and (charindex('M',p.estimateDuration)>0) then
    substr(p.estimateDuration,charindex('H',p.estimateDuration)+1,
charindex('M',p.estimateDuration)-charindex('H',p.estimateDuration)-1)
  when (charindex('H',p.estimateDuration)=0) and (charindex('M',p.estimateDuration)>0) then
    substr(p.estimateDuration,3, charindex('M',p.estimateDuration)-
charindex('PT',p.estimateDuration)-2)
else '0' end)/60) +
((case
  when (charindex('M',p.estimateDuration)>0) and (charindex('S',p.estimateDuration)>0) then
    substr(p.estimateDuration,charindex('M',p.estimateDuration)+1,
charindex('S',p.estimateDuration)-charindex('M',p.estimateDuration)-1)
  when (charindex('H',p.estimateDuration)>0) and (charindex('M',p.estimateDuration)=0) and
(charindex('S',p.estimateDuration)>0) then
    substr(p.estimateDuration,charindex('H',p.estimateDuration)+1,
charindex('S',p.estimateDuration)-charindex('H',p.estimateDuration)-1)
  when (charindex('H',p.estimateDuration)=0) and (charindex('M',p.estimateDuration)=0) and
(charindex('S',p.estimateDuration)>0) then
    substr(p.estimateDuration,charindex('PT',p.estimateDuration)+2,
charindex('S',p.estimateDuration)-charindex('PT',p.estimateDuration)-2)
else '0' end)/3600)) as estimate_duration_hours,
p.workspaceId
from "MYDB"."CLOCKIFY_LANDING".PROJECTS as p
left join "MYDB"."CLOCKIFY_ANALYTICS".CLIENTS_DIM cl
on p.clientid = cl.client_id;
```

```
--create tasks sequence
create or replace sequence TASKS_DIM;

select * from "CLOCKIFY_LANDING".tasks;

--Create Tasks table in analytics
create or replace TABLE "MYDB"."CLOCKIFY_ANALYTICS".TASKS_DIM (
task_key NUMBER(38,0),
task_id VARCHAR,
task_name VARCHAR,
task_assignee_id VARCHAR,
task_assignee_ids ARRAY,
task_billable BOOLEAN,
task_duration_hours NUMBER(8,3),
task_estimate_duration_hours NUMBER(7,3),
task_project_key INTEGER,
task_status VARCHAR(16777216),
task_user_group_ids ARRAY
);


--Insert into Tasks table
insert into "MYDB"."CLOCKIFY_ANALYTICS".TASKS_DIM (
task_key,
task_id,
task_name,
task_assignee_id,
task_assignee_ids,
task_billable,
task_duration_hours,
task_estimate_duration_hours,
task_project_key,
task_status,
task_user_group_ids
)
select
TASKS_DIM.nextval as task_key,
ts.id,
ts.name,
ts.assigneeId,
```

```
ts.assigneeIds,
ts.billable,
((case
  when charindex('H',ts.duration)>0 then
    substr(ts.duration,3, charindex('H',ts.duration)-charindex('PT',ts.duration)-2)
else '0' end) +
((case
  when (charindex('H',ts.duration)>0) and (charindex('M',ts.duration)>0) then
    substr(ts.duration,charindex('H',ts.duration)+1, charindex('M',ts.duration)-
charindex('H',ts.duration)-1)
  when (charindex('H',ts.duration)=0) and (charindex('M',ts.duration)>0) then
    substr(ts.duration,3, charindex('M',ts.duration)-charindex('PT',ts.duration)-2)
else '0' end)/60) +
((case
  when (charindex('M',ts.duration)>0) and (charindex('S',ts.duration)>0) then
    substr(ts.duration,charindex('M',ts.duration)+1, charindex('S',ts.duration)-
charindex('M',ts.duration)-1)
  when (charindex('H',ts.duration)>0) and (charindex('M',ts.duration)=0) and
(charindex('S',ts.duration)>0) then
    substr(ts.duration,charindex('H',ts.duration)+1, charindex('S',ts.duration)-
charindex('H',ts.duration)-1)
  when (charindex('H',ts.duration)=0) and (charindex('M',ts.duration)=0) and
(charindex('S',ts.duration)>0) then
    substr(ts.duration,charindex('PT',ts.duration)+2, charindex('S',ts.duration)-
charindex('PT',ts.duration)-2)
else '0' end)/3600)) as duration_hours,
((case
  when charindex('H',ts.estimate)>0 then
    substr(ts.estimate,3, charindex('H',ts.estimate)-charindex('PT',ts.estimate)-2)
else '0' end) +
((case
  when (charindex('H',ts.estimate)>0) and (charindex('M',ts.estimate)>0) then
    substr(ts.estimate,charindex('H',ts.estimate)+1, charindex('M',ts.estimate)-
charindex('H',ts.estimate)-1)
  when (charindex('H',ts.estimate)=0) and (charindex('M',ts.estimate)>0) then
    substr(ts.estimate,3, charindex('M',ts.estimate)-charindex('PT',ts.estimate)-2)
else '0' end)/60) +
((case
  when (charindex('M',ts.estimate)>0) and (charindex('S',ts.estimate)>0) then
    substr(ts.estimate,charindex('M',ts.estimate)+1, charindex('S',ts.estimate)-
charindex('M',ts.estimate)-1)
  when (charindex('H',ts.estimate)>0) and (charindex('M',ts.estimate)=0) and
(charindex('S',ts.estimate)>0) then
```

```
   substr(ts.estimate,charindex('H',ts.estimate)+1, charindex('S',ts.estimate)-
charindex('H',ts.estimate)-1)
  when (charindex('H',ts.estimate)=0) and (charindex('M',ts.estimate)=0) and
(charindex('S',ts.estimate)>0) then
   substr(ts.estimate,charindex('PT',ts.estimate)+2, charindex('S',ts.estimate)-
charindex('PT',ts.estimate)-2)
else '0' end)/3600)) as estimate_duration_hours,
p.project_key,
ts.status,
ts.userGroupIds
from "MYDB"."CLOCKIFY_LANDING".TASKS as ts
left join "MYDB"."CLOCKIFY_ANALYTICS".PROJECTS_DIM p
on ts.projectId = p.project_id;

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

--create tasks sequence
create or replace sequence USERS_DIM;

select * from "CLOCKIFY_LANDING".users;

--Create Users users in analytics
create or replace TABLE "MYDB"."CLOCKIFY_ANALYTICS".USERS_DIM (
user_key NUMBER(38,0),
user_id VARCHAR,
user_email VARCHAR,
user_name VARCHAR,
user_status VARCHAR
);

--Insert into Users Tables
insert into "MYDB"."CLOCKIFY_ANALYTICS".USERS_DIM (
user_key,
user_id,
user_email,
user_name,
user_status
)
select
USERS_DIM.nextval as user_key,
id,
email,
name,
status
```

```
from "MYDB"."CLOCKIFY_LANDING".USERS  as u;


------- Create Date table in analytics

CREATE OR REPLACE TABLE "MYDB"."CLOCKIFY_ANALYTICS".DATE_DIM(
        DATE_KEY                        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 DATE_DIM
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('1999-01-01 00:00:00','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 => 15000)) /*<< Set to generate 20 years.
Modify rowcount to increase or decrease size*/
        )v;



----------


--Create Time_Entries table in analytics
create or replace TABLE "MYDB"."CLOCKIFY_ANALYTICS".TIME_ENTRIES_FACT (
time_entry_id VARCHAR,
--project_key NUMBER(38,0),
task_key NUMBER(38,0),
user_key NUMBER(38,0),
billable BOOLEAN,
--custom_field_values ARRAY,
description VARCHAR,
is_locked BOOLEAN,
kiosk_id VARCHAR,
tag_ids ARRAY,
start_time TIMESTAMPTZ,
start_time_date_key NUMBER(9,0),
end_time TIMESTAMPTZ,
duration VARCHAR,
duration_hours NUMBER(8,3),
type VARCHAR,
workspace_id VARCHAR,
load_timestamp TIMESTAMPTZ
```

```
);

--Insert into Time_entries table
insert into "MYDB"."CLOCKIFY_ANALYTICS".TIME_ENTRIES_FACT (
time_entry_id,
--project_key,
task_key,
user_key,
billable,
--custom_field_values,
description,
is_locked,
kiosk_id,
tag_ids,
start_time,
start_time_date_key,
end_time,
duration,
duration_hours,
type,
workspace_id,
load_timestamp
)
select
te.id,
--p.project_key,
t.task_key,
u.user_key,
te.billable,
--te.customFieldValues,
te.description,
te.isLocked,
te.kioskId,
te.tagIds,
te.startTime,
ds.date_key,
te.endTime,
te.duration,
((case
  when charindex('H',te.duration)>0 then
    substr(te.duration,3, charindex('H',te.duration)-charindex('PT',te.duration)-2)
else '0' end) +
((case
  when (charindex('H',te.duration)>0) and (charindex('M',te.duration)>0) then
```

```
   substr(te.duration,charindex('H',te.duration)+1, charindex('M',te.duration)-
charindex('H',te.duration)-1)
  when (charindex('H',te.duration)=0) and (charindex('M',te.duration)>0) then
   substr(te.duration,3, charindex('M',te.duration)-charindex('PT',te.duration)-2)
else '0' end)/60) +
((case
  when (charindex('M',te.duration)>0) and (charindex('S',te.duration)>0) then
   substr(te.duration,charindex('M',te.duration)+1, charindex('S',te.duration)-
charindex('M',te.duration)-1)
  when (charindex('H',te.duration)>0) and (charindex('M',te.duration)=0) and
(charindex('S',te.duration)>0) then
   substr(te.duration,charindex('H',te.duration)+1, charindex('S',te.duration)-
charindex('H',te.duration)-1)
  when (charindex('H',te.duration)=0) and (charindex('M',te.duration)=0) and
(charindex('S',te.duration)>0) then
   substr(te.duration,charindex('PT',te.duration)+2, charindex('S',te.duration)-
charindex('PT',te.duration)-2)
else '0' end)/3600)) as duration_hours,
te.type,
te.workspaceId,
te.loadTimestamp
from "MYDB"."CLOCKIFY_LANDING".TIME_ENTRIES as te
left join "MYDB"."CLOCKIFY_ANALYTICS".USERS_DIM as u
on te.userid=u.user_id
left join "MYDB"."CLOCKIFY_ANALYTICS".TASKS_DIM as t
on te.taskid=t.task_id
left join "MYDB"."CLOCKIFY_ANALYTICS".PROJECTS_DIM as p
on te.projectid=p.project_id
inner join "MYDB"."CLOCKIFY_ANALYTICS".DATE_DIM ds
ON to_number(to_varchar(to_date(te.starttime),'YYYYMMDD')) = ds.DATE_KEY;
-- inner join "MYDB"."CLOCKIFY_ANALYTICS".DATE_DIM de ON
-- to_number(to_varchar(to_date(end_time),'YYYYMMDD')) = de.DATE_KEY;




—– KPI QUERIES

-- KPI 1 Total Project time/Employee vs Estimate time
SELECT USER_NAME, ROUND(SUM(TOTAL_TASK_TIME),2) AS TOTAL_PROJECT_TIME,
ROUND(SUM(ESTIMATE_TIME),2) AS TOTAL_ESTIMATE_TIME
FROM
(SELECT
```

```
te.USER_KEY,
t.TASK_KEY,
SUM(te.DURATION_HOURS) AS TOTAL_TASK_TIME,
MAX(t.TASK_ESTIMATE_DURATION_HOURS) AS ESTIMATE_TIME
FROM "MYDB"."CLOCKIFY_ANALYTICS"."TASKS_DIM" t
RIGHT OUTER JOIN "MYDB"."CLOCKIFY_ANALYTICS"."TIME_ENTRIES_FACT" te ON
t.TASK_KEY = te.TASK_KEY
GROUP BY t.TASK_KEY, te.USER_KEY) s
JOIN "MYDB"."CLOCKIFY_ANALYTICS"."USERS_DIM" u ON s.USER_KEY = u.USER_KEY
GROUP BY u.USER_NAME
ORDER BY u.USER_NAME;


-- KPI 1.5 Project Time vs. Estimate Time per Project
select
p.project_name,
p.project_short_name,
round(sum(te.duration_hours),2) as total_time_hours,
round(max(p.project_duration_hours),2) as project_duration_estimate,
from time_entries_fact te
inner join tasks_dim t
on te.task_key=t.task_key
inner join projects_dim p
on t.project_key = p.project_key
group by
p.project_name,
p.project_short_name
order by total_time_hours desc;


-- KPI 6 Top 20 employees who's running overtime
SELECT USER_NAME, (ROUND(SUM(TOTAL_TASK_TIME),2) -
ROUND(SUM(ESTIMATE_TIME),2)) AS UNDERESTIMATED_TIME,
ROUND(SUM(TOTAL_TASK_TIME),2) AS TOTAL_PROJECT_TIME,
ROUND(SUM(ESTIMATE_TIME),2) AS TOTAL_ESTIMATE_TIME
FROM
(SELECT
te.USER_KEY,
t.TASK_KEY,
SUM(te.DURATION_HOURS) AS TOTAL_TASK_TIME,
MAX(t.TASK_ESTIMATE_DURATION_HOURS) AS ESTIMATE_TIME
FROM "MYDB"."CLOCKIFY_ANALYTICS"."TASKS_DIM" t
RIGHT OUTER JOIN "MYDB"."CLOCKIFY_ANALYTICS"."TIME_ENTRIES_FACT" te ON
t.TASK_KEY = te.TASK_KEY
```

```
GROUP BY t.TASK_KEY, te.USER_KEY) s
JOIN "MYDB"."CLOCKIFY_ANALYTICS"."USERS_DIM" u ON s.USER_KEY = u.USER_KEY
GROUP BY u.USER_NAME
HAVING (ROUND(SUM(TOTAL_TASK_TIME),2) - ROUND(SUM(ESTIMATE_TIME),2)) > 0
AND ROUND(SUM(ESTIMATE_TIME),2) != 0
ORDER BY UNDERESTIMATED_TIME DESC
LIMIT 20;


-- KPI 2 Billable hours by Client
SELECT CASE WHEN c.CLIENT_NAME IS NULL THEN 'Anonymous Clients' ELSE
C.CLIENT_NAME END,
ROUND(SUM(te.DURATION_HOURS),2) AS BILLABLE_HOURS
FROM "MYDB"."CLOCKIFY_ANALYTICS"."TIME_ENTRIES_FACT" te
LEFT OUTER JOIN "MYDB"."CLOCKIFY_ANALYTICS"."TASKS_DIM" t ON te.TASK_KEY =
t.TASK_KEY
LEFT OUTER JOIN "MYDB"."CLOCKIFY_ANALYTICS"."PROJECTS_DIM" p ON
t.PROJECT_KEY = p.PROJECT_KEY
lEFT OUTER JOIN "MYDB"."CLOCKIFY_ANALYTICS"."CLIENTS_DIM" c ON p.CLIENT_KEY
= c.CLIENT_KEY
WHERE te.BILLABLE = 'TRUE'
GROUP BY c.CLIENT_NAME
ORDER BY BILLABLE_HOURS DESC;


-- KPI 3 Non-billable hours by Client
SELECT CASE WHEN c.CLIENT_NAME IS NULL THEN 'Anonymous Clients' ELSE
C.CLIENT_NAME END,
ROUND(SUM(te.DURATION_HOURS),2) AS BILLABLE_HOURS
FROM "MYDB"."CLOCKIFY_ANALYTICS"."TIME_ENTRIES_FACT" te
LEFT OUTER JOIN "MYDB"."CLOCKIFY_ANALYTICS"."TASKS_DIM" t ON te.TASK_KEY =
t.TASK_KEY
LEFT OUTER JOIN "MYDB"."CLOCKIFY_ANALYTICS"."PROJECTS_DIM" p ON
t.PROJECT_KEY = p.PROJECT_KEY
lEFT OUTER JOIN "MYDB"."CLOCKIFY_ANALYTICS"."CLIENTS_DIM" c ON p.CLIENT_KEY
= c.CLIENT_KEY
WHERE te.BILLABLE = 'FALSE'
GROUP BY c.CLIENT_NAME
ORDER BY BILLABLE_HOURS DESC;


-- KPI 4 Avg Duration by Billable / NonBillable Task
SELECT CASE WHEN TASK_BILLABLE = 'TRUE' THEN 'BILLABLE' ELSE 'NON BILLABLE'
END AS BILLABLE,
```

```
ROUND(AVG(TASK_DURATION_HOURS),2) AS AVG_TASK_DURATION,
ROUND(SUM(TASK_DURATION_HOURS),2) AS TOTAL_TASK_DURATION
FROM "MYDB"."CLOCKIFY_ANALYTICS"."TASKS_DIM"
GROUP BY TASK_BILLABLE;


-- KPI 5 Avg and Total Duration by Project Type (proj name)
SELECT PROJECT_SHORT_NAME AS PROJECT_TYPE,
ROUND(AVG(PROJECT_DURATION_HOURS),2) AS AVG_PROJECT_DURATION,
ROUND(SUM(PROJECT_DURATION_HOURS),2) AS TOTAL_PROJECT_DURATION
FROM "MYDB"."CLOCKIFY_ANALYTICS"."PROJECTS_DIM"
GROUP BY PROJECT_SHORT_NAME
ORDER BY AVG_PROJECT_DURATION DESC;
```