

U1M7.LW.Dimension and Facts Basics

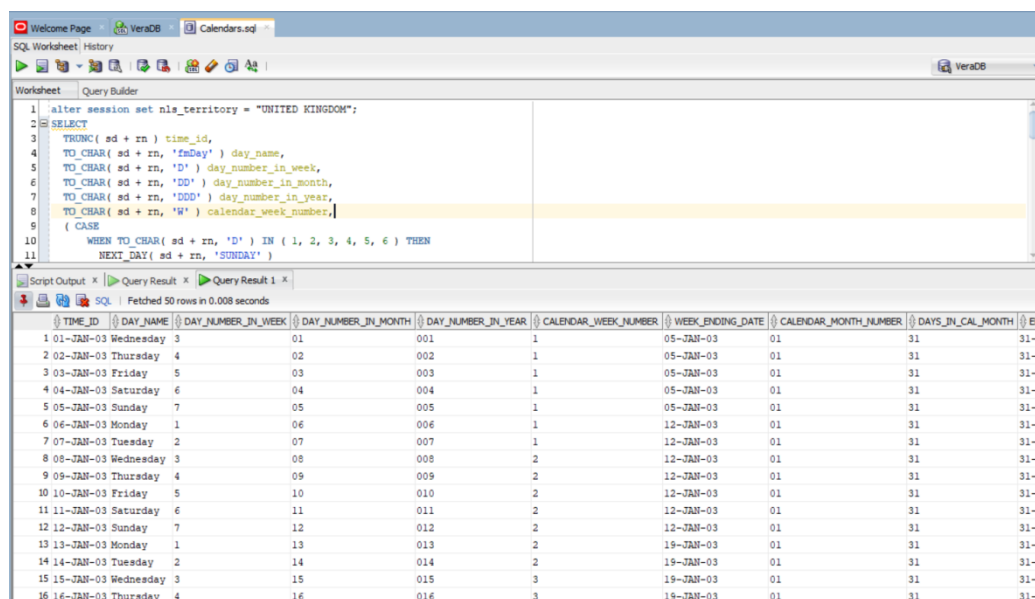
Part 1

Shkrabatouskaya Vera

https://github.com/VeraShkrabatouskaya/DataMola_Data-Camping-2022

2. Create and populate Dimension of TIME DW – Layer

Using External Resources to Populate Time dims:



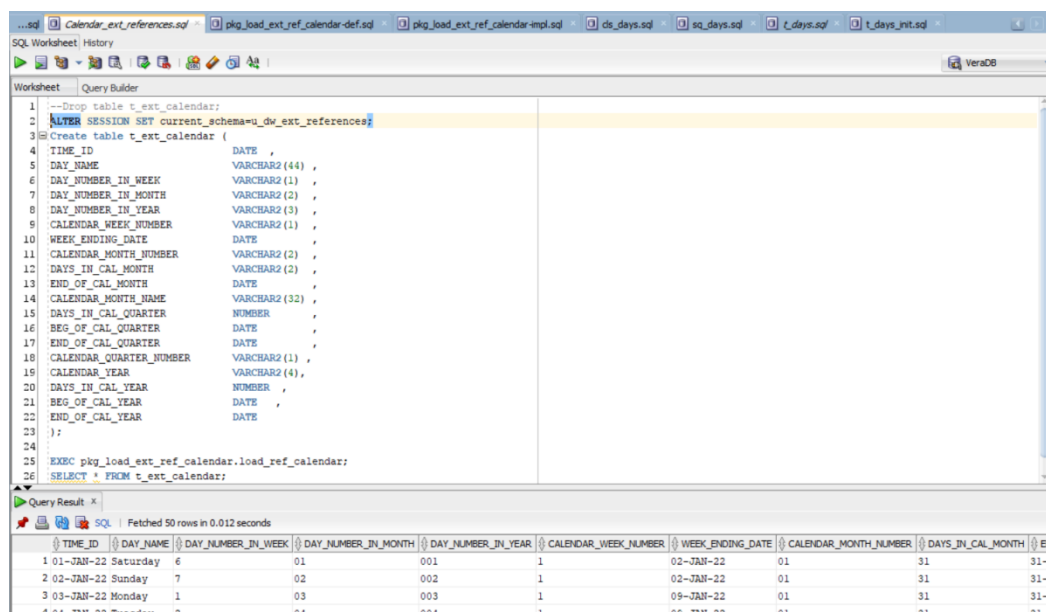
The screenshot shows a SQL Worksheet with a query to create a time dimension table. The query is as follows:

```
1 alter session set nls_territory = 'UNITED KINGDOM';
2
3 SELECT
4   TRUNC( sd + rn ) time_id,
5   TO_CHAR( sd + rn, 'fmDay' ) day_name,
6   TO_CHAR( sd + rn, 'D' ) day_number_in_week,
7   TO_CHAR( sd + rn, 'DD' ) day_number_in_month,
8   TO_CHAR( sd + rn, 'DDD' ) day_number_in_year,
9   TO_CHAR( sd + rn, 'W' ) calendar_week_number,
10  ( CASE
11    WHEN TO_CHAR( sd + rn, 'D' ) IN ( 1, 2, 3, 4, 5, 6 ) THEN
12      NEXT_DAY( sd + rn, 'SUNDAY' )
13  )
```

The results show 16 rows of data for the first week of January 2003:

TIME_ID	DAY_NAME	DAY_NUMBER_IN_WEEK	DAY_NUMBER_IN_MONTH	DAY_NUMBER_IN_YEAR	CALENDAR_WEEK_NUMBER	WEEK_ENDING_DATE	CALENDAR_MONTH_NUMBER	DAYS_IN_CAL_MONTH	EP
01-JAN-03	Wednesday	3	01	001	1	05-JAN-03	01	31	31-
02-JAN-03	Thursday	4	02	002	1	05-JAN-03	01	31	31-
03-JAN-03	Friday	5	03	003	1	05-JAN-03	01	31	31-
04-JAN-03	Saturday	6	04	004	1	05-JAN-03	01	31	31-
05-JAN-03	Sunday	7	05	005	1	05-JAN-03	01	31	31-
06-JAN-03	Monday	1	06	006	1	12-JAN-03	01	31	31-
07-JAN-03	Tuesday	2	07	007	1	12-JAN-03	01	31	31-
08-JAN-03	Wednesday	3	08	008	2	12-JAN-03	01	31	31-
09-JAN-03	Thursday	4	09	009	2	12-JAN-03	01	31	31-
10-JAN-03	Friday	5	10	010	2	12-JAN-03	01	31	31-
11-JAN-03	Saturday	6	11	011	2	12-JAN-03	01	31	31-
12-JAN-03	Sunday	7	12	012	2	12-JAN-03	01	31	31-
13-JAN-03	Monday	1	13	013	2	19-JAN-03	01	31	31-
14-JAN-03	Tuesday	2	14	014	2	19-JAN-03	01	31	31-
15-JAN-03	Wednesday	3	15	015	3	19-JAN-03	01	31	31-
16-JAN-03	Thursday	4	16	016	3	19-JAN-03	01	31	31-

For creating Objects on DW layer, we used External Table t_ext_calendar.



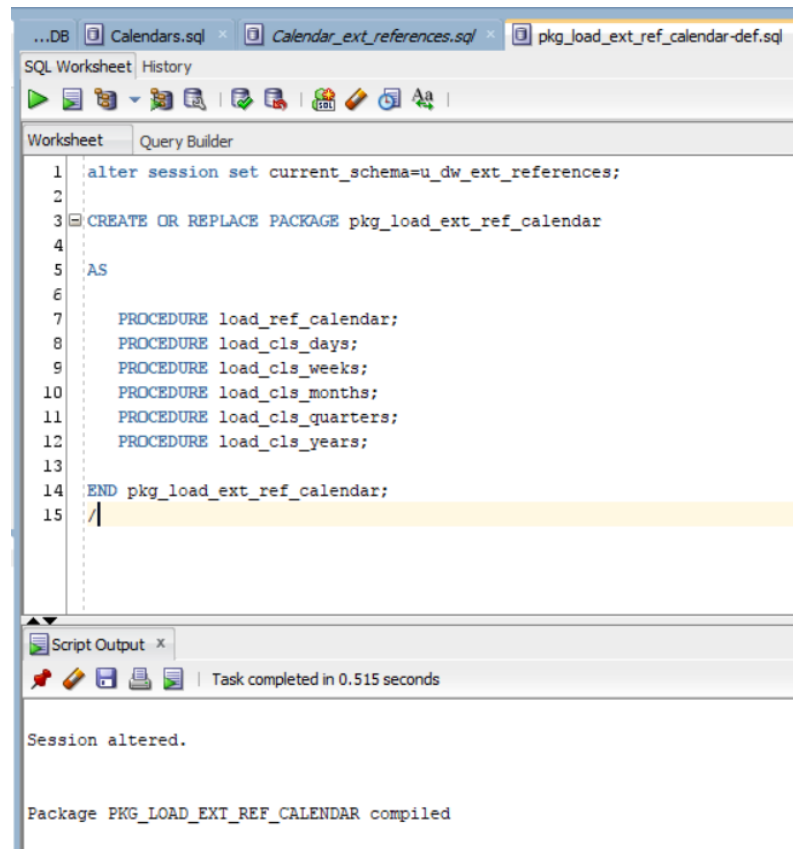
The screenshot shows a SQL Worksheet with a query to create an external table. The query is as follows:

```
1 --Drop table t_ext_calendar;
2 ALTER SESSION SET current_schema=u_dw_ext_references;
3 Create table t_ext_calendar (
4   TIME_ID DATE ,
5   DAY_NAME VARCHAR2(44) ,
6   DAY_NUMBER_IN_WEEK VARCHAR2(1) ,
7   DAY_NUMBER_IN_MONTH VARCHAR2(2) ,
8   DAY_NUMBER_IN_YEAR VARCHAR2(3) ,
9   CALENDAR_WEEK_NUMBER VARCHAR2(1) ,
10  WEEK_ENDING_DATE DATE ,
11  CALENDAR_MONTH_NUMBER VARCHAR2(2) ,
12  DAYS_IN_CAL_MONTH VARCHAR2(2) ,
13  END_OF_CAL_MONTH DATE ,
14  CALENDAR_MONTH_NAME VARCHAR2(32) ,
15  DAYS_IN_CAL_QUARTER NUMBER ,
16  BEG_OF_CAL_QUARTER DATE ,
17  END_OF_CAL_QUARTER DATE ,
18  CALENDAR_QUARTER_NUMBER VARCHAR2(1) ,
19  CALENDAR_YEAR VARCHAR2(4) ,
20  DAYS_IN_CAL_YEAR NUMBER ,
21  BEG_OF_CAL_YEAR DATE ,
22  END_OF_CAL_YEAR DATE ,
23 );
24
25 EXEC pkg_load_ext_ref_calendar.load_ref_calendar;
26 SELECT * FROM t_ext_calendar;
```

The results show 4 rows of data for the first week of January 2022:

TIME_ID	DAY_NAME	DAY_NUMBER_IN_WEEK	DAY_NUMBER_IN_MONTH	DAY_NUMBER_IN_YEAR	CALENDAR_WEEK_NUMBER	WEEK_ENDING_DATE	CALENDAR_MONTH_NUMBER	DAYS_IN_CAL_MONTH	EP
01-JAN-22	Saturday	6	01	001	1	02-JAN-22	01	31	31-
02-JAN-22	Sunday	7	02	002	1	02-JAN-22	01	31	31-
03-JAN-22	Monday	1	03	003	1	09-JAN-22	01	31	31-
04-JAN-22	Tuesday	2	04	004	1	09-JAN-22	01	31	31-

We used packages to divide the calendar data into different layers and load them into cleansing tables.



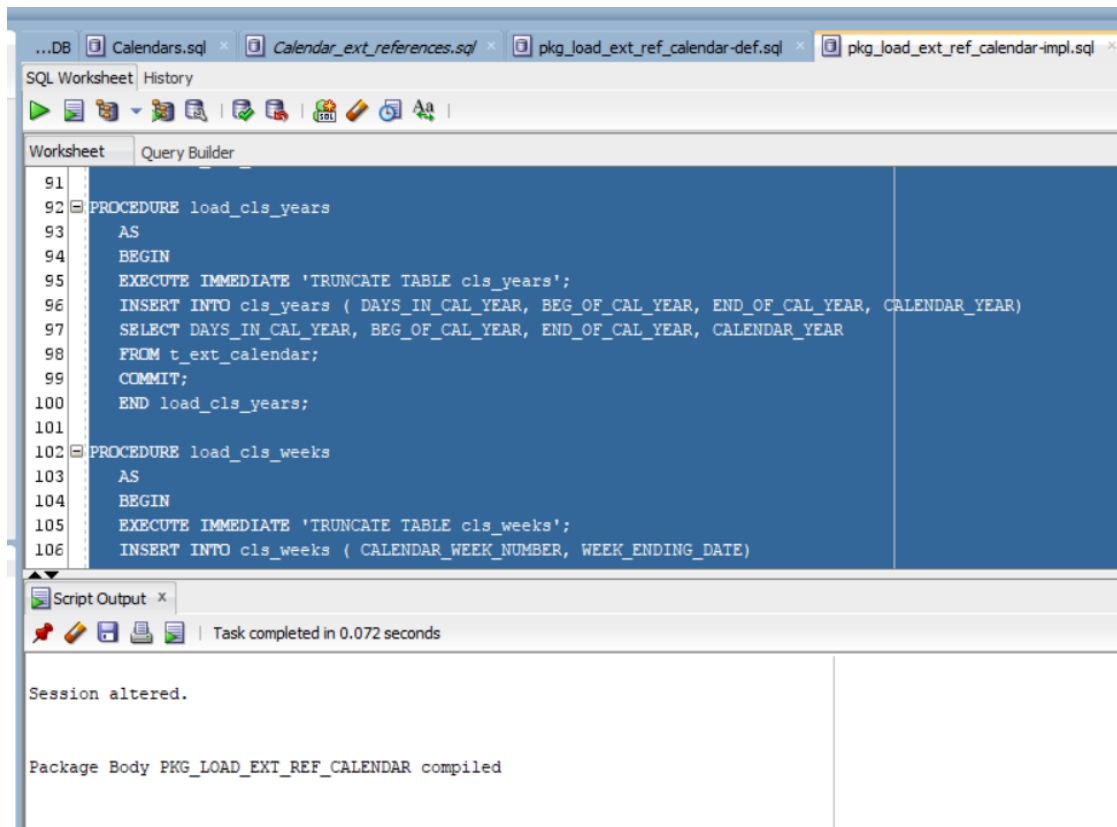
```
1 alter session set current_schema=u_dw_ext_references;
2
3 CREATE OR REPLACE PACKAGE pkg_load_ext_ref_calendar
4
5 AS
6
7     PROCEDURE load_ref_calendar;
8     PROCEDURE load_cls_days;
9     PROCEDURE load_cls_weeks;
10    PROCEDURE load_cls_months;
11    PROCEDURE load_cls_quarters;
12    PROCEDURE load_cls_years;
13
14 END pkg_load_ext_ref_calendar;
15 /
```

Script Output

Task completed in 0.515 seconds

Session altered.

Package PKG_LOAD_EXT_REF_CALENDAR compiled



```
91
92 PROCEDURE load_cls_years
93 AS
94 BEGIN
95     EXECUTE IMMEDIATE 'TRUNCATE TABLE cls_years';
96     INSERT INTO cls_years ( DAYS_IN_CAL_YEAR, BEG_OF_CAL_YEAR, END_OF_CAL_YEAR, CALENDAR_YEAR)
97     SELECT DAYS_IN_CAL_YEAR, BEG_OF_CAL_YEAR, END_OF_CAL_YEAR, CALENDAR_YEAR
98     FROM t_ext_calendar;
99     COMMIT;
100 END load_cls_years;
101
102 PROCEDURE load_cls_weeks
103 AS
104 BEGIN
105     EXECUTE IMMEDIATE 'TRUNCATE TABLE cls_weeks';
106     INSERT INTO cls_weeks ( CALENDAR_WEEK_NUMBER, WEEK_ENDING_DATE)
```

Script Output

Task completed in 0.072 seconds

Session altered.

Package Body PKG_LOAD_EXT_REF_CALENDAR compiled

2.1. Task 01: CREATE DW.T_DAYS

Create Object T_DAYS on DW layer:

- cls

The screenshot shows the SQL Developer interface with the following SQL script in the Worksheet:

```
1 ALTER SESSION SET current_schema=u_dw_ext_references;
2
3 CREATE TABLE cls_days(
4
5   DAY_NAME                VARCHAR2(44) ,
6   DAY_NUMBER_IN_WEEK      VARCHAR2(1)  ,
7   DAY_NUMBER_IN_MONTH     VARCHAR2(2)  ,
8   DAY_NUMBER_IN_YEAR      VARCHAR2(3)  ,
9 )
10 TABLESPACE TS_REFERENCES_EXT_DATA_01;
11
12 EXEC pkg_load_ext_ref_calendar.load_cls_days;
13 select * from cls_days;
```

The Query Result shows 15 rows of data:

	DAY_NAME	DAY_NUMBER_IN_WEEK	DAY_NUMBER_IN_MONTH	DAY_NUMBER_IN_YEAR
1	Saturday	6	01	001
2	Sunday	7	02	002
3	Monday	1	03	003
4	Tuesday	2	04	004
5	Wednesday	3	05	005
6	Thursday	4	06	006
7	Friday	5	07	007
8	Saturday	6	08	008
9	Sunday	7	09	009
10	Monday	1	10	010
11	Tuesday	2	11	011
12	Wednesday	3	12	012
13	Thursday	4	13	013
14	Friday	5	14	014
15	Saturday	6	15	015

- sq

The screenshot shows the SQL Developer interface with the following SQL script in the Worksheet:

```
1 alter session set current_schema=u_dw_references;
2
3 create sequence u_dw_references.sq_day_id start with 1;
4
5 grant SELECT on u_dw_references.sq_day_id to u_dw_ext_references;
```

- t

The screenshot shows the SQL Developer interface with the following SQL script in the Worksheet:

```
1 --creating table with id PK
2 alter session set current_schema = u_dw_references;
3 --drop table t_days;
4 --alter user u_dw_references quota unlimited on TS_REFERENCES_DATA_01;
5
6 CREATE TABLE t_days(
7   DAY_ID                NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
8   DAY_NAME              VARCHAR2(44) ,
9   DAY_NUMBER_IN_WEEK    VARCHAR2(1)  ,
10  DAY_NUMBER_IN_MONTH    VARCHAR2(2)  ,
11  DAY_NUMBER_IN_YEAR     VARCHAR2(3)  ,
12  CONSTRAINT "PK_DW.T_DAYS" PRIMARY KEY(day_id) USING INDEX TABLESPACE ts_references_idx_01
13 );
```

- t_init

SQL Worksheet History

Worksheet Query Builder

```

1 alter session set current_schema = u_dw_references;
2 INSERT INTO t_days(
3   DAY_NAME
4   ,
5   DAY_NUMBER_IN_WEEK
6   ,
7   DAY_NUMBER_IN_MONTH
8   ,
9   DAY_NUMBER_IN_YEAR
10  )
11  SELECT DAY_NAME
12         ,
13         DAY_NUMBER_IN_WEEK
14         ,
15         DAY_NUMBER_IN_MONTH
16         ,
17         DAY_NUMBER_IN_YEAR
18  FROM u_dw_ext_references.cls_days;
19 commit;
20
21 select * from t_days;

```

Query Result x

SQL | Fetched 50 rows in 0.007 seconds

DAY_ID	DAY_NAME	DAY_NUMBER_IN_WEEK	DAY_NUMBER_IN_MONTH	DAY_NUMBER_IN_YEAR
1	Saturday	6	01	001
2	Sunday	7	02	002
3	Monday	1	03	003
4	Tuesday	2	04	004
5	Wednesday	3	05	005
6	Thursday	4	06	006
7	Friday	5	07	007
8	Saturday	6	08	008
9	Sunday	7	09	009
10	Monday	1	10	010
11	Tuesday	2	11	011
12	Wednesday	3	12	012
13	Thursday	4	13	013

- W

SQL Worksheet History

Worksheet Query Builder

```

1 create or replace view u_dw_references.w_days as
2 SELECT
3   DAY_ID
4   ,
5   DAY_NAME
6   ,
7   DAY_NUMBER_IN_WEEK
8   ,
9   DAY_NUMBER_IN_MONTH
10  ,
11  DAY_NUMBER_IN_YEAR
12  FROM t_days;
13
14 grant DELETE,INSERT,UPDATE,SELECT on u_dw_references.w_days to u_dw_ext_references;

```

Script Output x

Task completed in 0.1 seconds

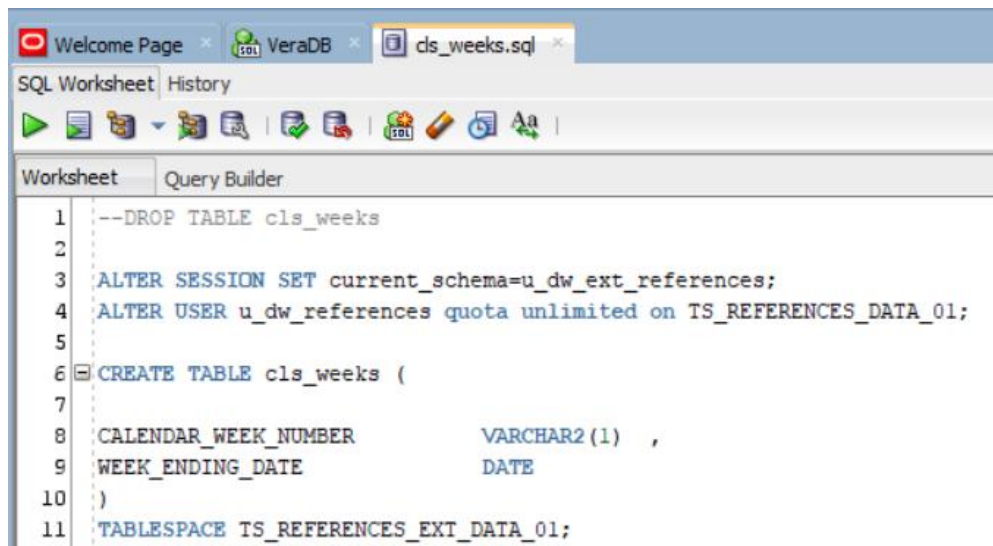
View U_DW_REFERENCES.W_DAYS created.

Grant succeeded.

2.2. Task 02: CREATE DW.T_WEEKS

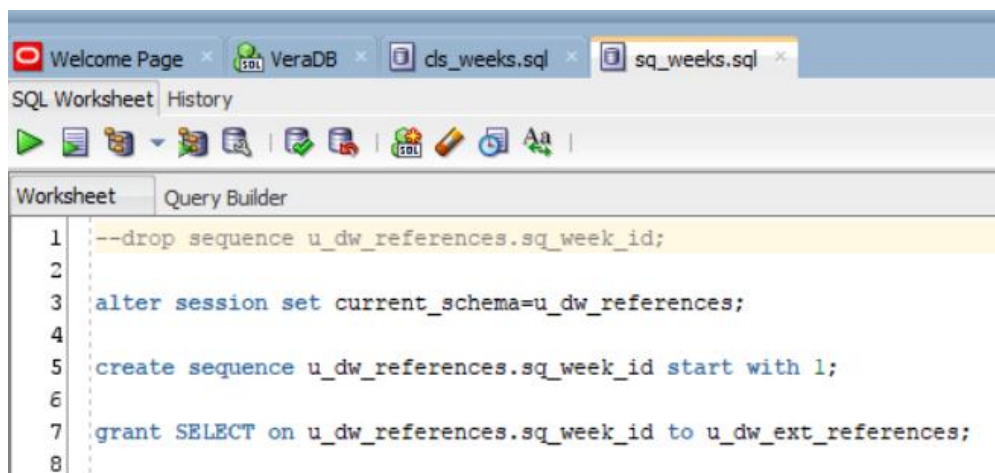
Create Object T_WEEKS on DW layer:

- cls



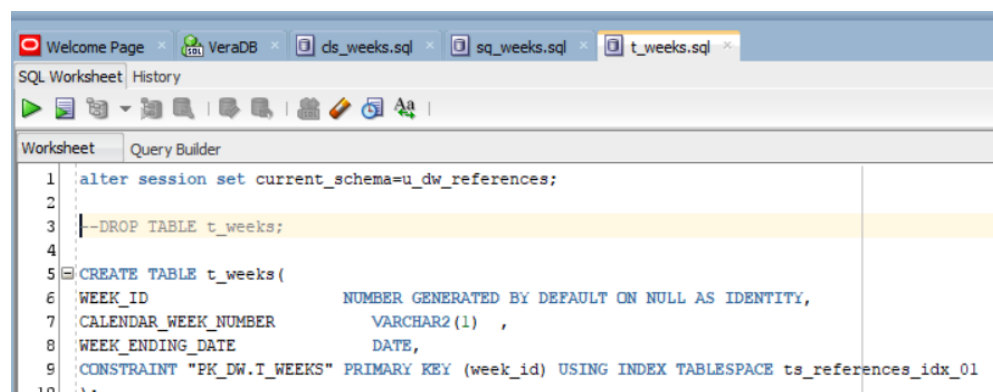
```
1  --DROP TABLE cls_weeks
2
3  ALTER SESSION SET current_schema=u_dw_ext_references;
4  ALTER USER u_dw_references quota unlimited on TS_REFERENCES_DATA_01;
5
6  CREATE TABLE cls_weeks (
7
8  CALENDAR_WEEK_NUMBER          VARCHAR2(1) ,
9  WEEK_ENDING_DATE              DATE
10 )
11 TABLESPACE TS_REFERENCES_EXT_DATA_01;
```

- sq



```
1  --drop sequence u_dw_references.sq_week_id;
2
3  alter session set current_schema=u_dw_references;
4
5  create sequence u_dw_references.sq_week_id start with 1;
6
7  grant SELECT on u_dw_references.sq_week_id to u_dw_ext_references;
8
```

- t



```
1  alter session set current_schema=u_dw_references;
2
3  --DROP TABLE t_weeks;
4
5  CREATE TABLE t_weeks(
6  WEEK_ID                      NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
7  CALENDAR_WEEK_NUMBER        VARCHAR2(1) ,
8  WEEK_ENDING_DATE            DATE,
9  CONSTRAINT "PK_DW.T_WEEKS" PRIMARY KEY (week_id) USING INDEX TABLESPACE ts_references_idx_01
10 )
```

- t_init

The screenshot shows the SQL Developer interface with a script named 't_weeks_init.sql'. The script contains the following SQL statements:

```

1 alter session set current_schema=u_dw_references;
2
3 insert into t_weeks (
4     CALENDAR_WEEK_NUMBER,
5     WEEK_ENDING_DATE
6 )
7     select CALENDAR_WEEK_NUMBER,
8            WEEK_ENDING_DATE FROM u_dw_ext_references.cis_weeks;
9
10 commit;
11
12 select * from t_weeks;

```

The 'Query Result' pane shows the output of the final SELECT statement, displaying 50 rows. The columns are WEEK_ID, CALENDAR_WEEK_NUMBER, and WEEK_ENDING_DATE. The data shows weeks 4 through 7, all ending on 09-JAN-22.

WEEK_ID	CALENDAR_WEEK_NUMBER	WEEK_ENDING_DATE
4	4 1	09-JAN-22
5	5 1	09-JAN-22
6	6 1	09-JAN-22
7	7 1	09-JAN-22

- w

The screenshot shows the SQL Developer interface with a script named 'w_weeks.sql'. The script contains the following SQL statements:

```

1 --drop view u_dw_references.v_weeks;
2
3 -----
4 -- View: v_weeks
5 -----
6
7 create or replace view u_dw_references.v_weeks as
8     select WEEK_ID,
9            CALENDAR_WEEK_NUMBER,
10           WEEK_ENDING_DATE
11     from t_weeks;
12
13 grant DELETE,INSERT,UPDATE,SELECT on u_dw_references.v_weeks to u_dw_ext_references;

```

The 'Script Output' pane shows the message: 'View U_DW_REFERENCES.V_WEEKS created.' and 'Grant succeeded.'

2.3. Task 03: CREATE DW.T_MONTHS

Create Object T_MONTHS on DW layer:

- cls

The screenshot shows the SQL Developer interface with a script named 'cls_months.sql'. The script contains the following SQL statements:

```

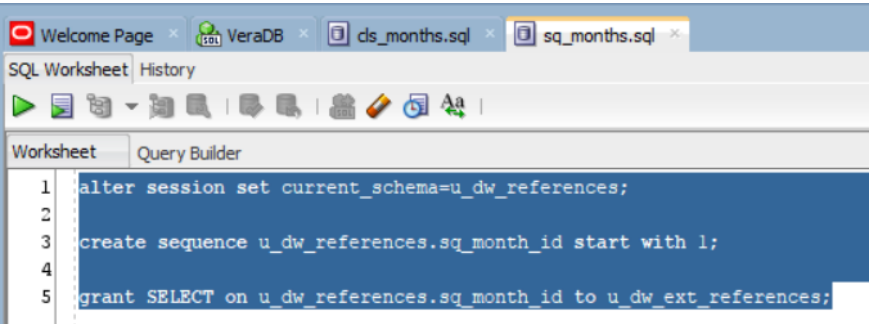
1 ALTER SESSION SET current_schema=u_dw_ext_references;
2
3 CREATE TABLE cls_months (
4
5     CALENDAR_MONTH_NUMBER      VARCHAR2(2) ,
6     DAYS_IN_CAL_MONTH          VARCHAR2(2) ,
7     END_OF_CAL_MONTH           DATE ,
8     CALENDAR_MONTH_NAME        VARCHAR2(32)
9 )
10 TABLESPACE TS_REFERENCES_EXT_DATA_01;
11
12 EXEC pkg_load_ext_ref_calendar.load_cls_months;
13 select * from cls_months;

```

The 'Query Result' pane shows the output of the final SELECT statement, displaying 50 rows. The columns are CALENDAR_MONTH_NUMBER, DAYS_IN_CAL_MONTH, END_OF_CAL_MONTH, and CALENDAR_MONTH_NAME. The data shows months 1 through 4, all ending on 31-JAN-22.

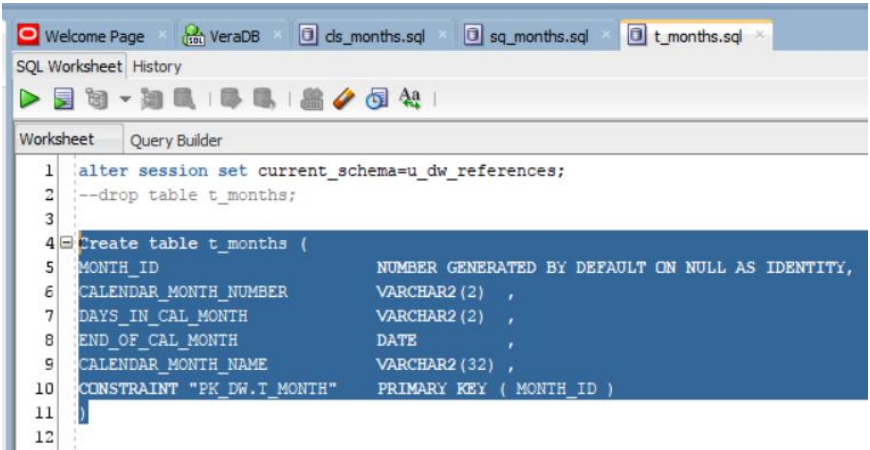
CALENDAR_MONTH_NUMBER	DAYS_IN_CAL_MONTH	END_OF_CAL_MONTH	CALENDAR_MONTH_NAME
1 01	31	31-JAN-22	January
2 01	31	31-JAN-22	January
3 01	31	31-JAN-22	January
4 01	31	31-JAN-22	January

- sq



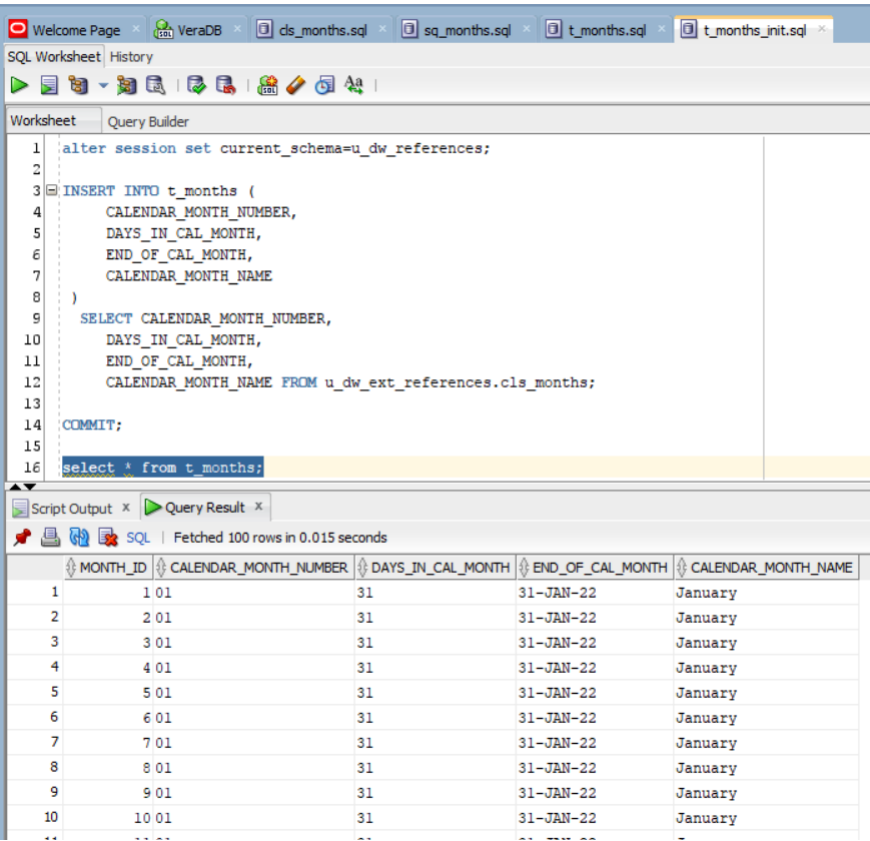
```
1 alter session set current_schema=u_dw_references;
2
3 create sequence u_dw_references.sq_month_id start with 1;
4
5 grant SELECT on u_dw_references.sq_month_id to u_dw_ext_references;
```

- t



```
1 alter session set current_schema=u_dw_references;
2 --drop table t_months;
3
4 create table t_months (
5     MONTH_ID                NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
6     CALENDAR_MONTH_NUMBER    VARCHAR2(2) ,
7     DAYS_IN_CAL_MONTH        VARCHAR2(2) ,
8     END_OF_CAL_MONTH         DATE ,
9     CALENDAR_MONTH_NAME      VARCHAR2(32) ,
10    CONSTRAINT "PK_DW.T_MONTH" PRIMARY KEY ( MONTH_ID )
11 )
12
```

- t_init



```
1 alter session set current_schema=u_dw_references;
2
3 INSERT INTO t_months (
4     CALENDAR_MONTH_NUMBER,
5     DAYS_IN_CAL_MONTH,
6     END_OF_CAL_MONTH,
7     CALENDAR_MONTH_NAME
8 )
9 SELECT CALENDAR_MONTH_NUMBER,
10        DAYS_IN_CAL_MONTH,
11        END_OF_CAL_MONTH,
12        CALENDAR_MONTH_NAME FROM u_dw_ext_references.cls_months;
13
14 COMMIT;
15
16 select * from t_months;
```

Script Output x Query Result x

SQL | Fetched 100 rows in 0.015 seconds

	MONTH_ID	CALENDAR_MONTH_NUMBER	DAYS_IN_CAL_MONTH	END_OF_CAL_MONTH	CALENDAR_MONTH_NAME
1	1	01	31	31-JAN-22	January
2	2	01	31	31-JAN-22	January
3	3	01	31	31-JAN-22	January
4	4	01	31	31-JAN-22	January
5	5	01	31	31-JAN-22	January
6	6	01	31	31-JAN-22	January
7	7	01	31	31-JAN-22	January
8	8	01	31	31-JAN-22	January
9	9	01	31	31-JAN-22	January
10	10	01	31	31-JAN-22	January
..

- W

The screenshot shows a SQL Worksheet with the following SQL code:

```

6
7 create or replace view u_dw_references.w_months as
8   SELECT month_id
9         , calendar_month_number
10        , days_in_cal_month
11        , end_of_cal_month
12        , calendar_month_name
13   FROM t_months;
14
15 comment on column u_dw_references.w_months.month_id is
16   'Identifier of the Month';
17
18 comment on column u_dw_references.w_months.days_in_cal_month is
19   'Number of days in month';
20
21 comment on column u_dw_references.w_months.end_of_cal_month is
22   'Last day of month';
23
24 comment on column u_dw_references.w_months.calendar_month_name is
25   'Month name';
26
27 grant DELETE, INSERT, UPDATE, SELECT on u_dw_references.w_months to u_dw_ext_references;
28

```

The Script Output pane shows the following messages:

```

View U_DW_REFERENCES.W_MONTHS created.

Grant succeeded.

```

2.4. Task 04: CREATE DW.T_QUARTERS

Create Object T_QUARTERS on DW layer:

- cls

The screenshot shows a SQL Worksheet with the following SQL code:

```

1 ALTER SESSION SET current_schema=u_dw_ext_references;
2
3 CREATE TABLE cls_quarters(
4
5   DAYS_IN_CAL_QUARTER      NUMBER,
6   BEG_OF_CAL_QUARTER      DATE,
7   END_OF_CAL_QUARTER      DATE,
8   CALENDAR_QUARTER_NUMBER VARCHAR2(1)
9 )
10 TABLESPACE TS_REFERENCES_EXT_DATA_01;
11
12
13 EXEC pkg_load_ext_ref_calendar.load_cls_quarters;
14 select * from cls_quarters;

```

The Script Output pane shows the following messages:

```

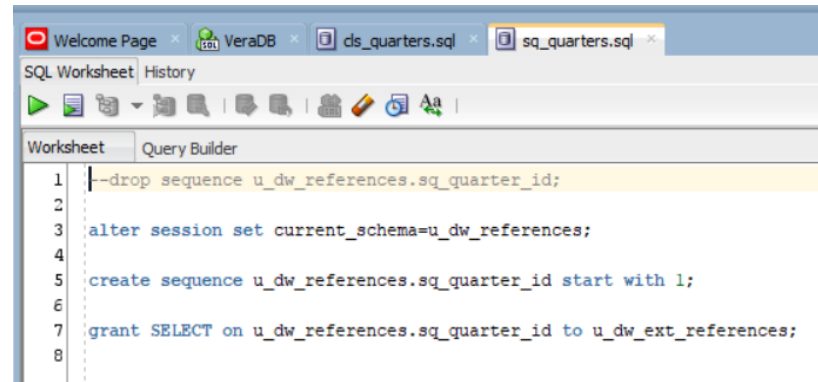
SQL | Fetched 50 rows in 0.014 seconds

```

The Query Result pane shows the following data:

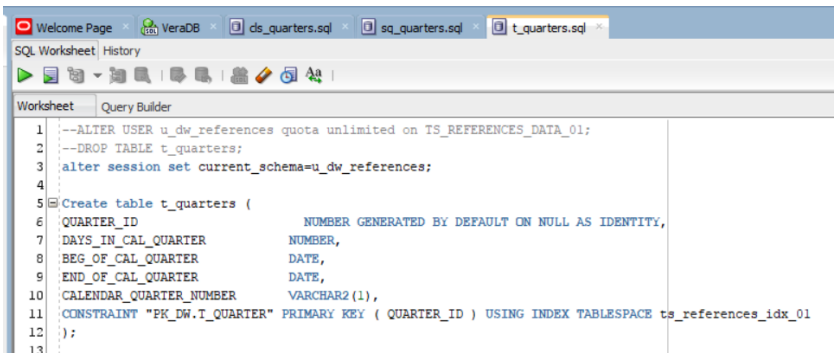
	DAYS_IN_CAL_QUARTER	BEG_OF_CAL_QUARTER	END_OF_CAL_QUARTER	CALENDAR_QUARTER_NUMBER
1	90	01-JAN-22	31-MAR-22	1
2	90	01-JAN-22	31-MAR-22	1
3	90	01-JAN-22	31-MAR-22	1
4	90	01-JAN-22	31-MAR-22	1
5	90	01-JAN-22	31-MAR-22	1
6	90	01-JAN-22	31-MAR-22	1
7	90	01-JAN-22	31-MAR-22	1
8	90	01-JAN-22	31-MAR-22	1
9	90	01-JAN-22	31-MAR-22	1
10	90	01-JAN-22	31-MAR-22	1

- sq



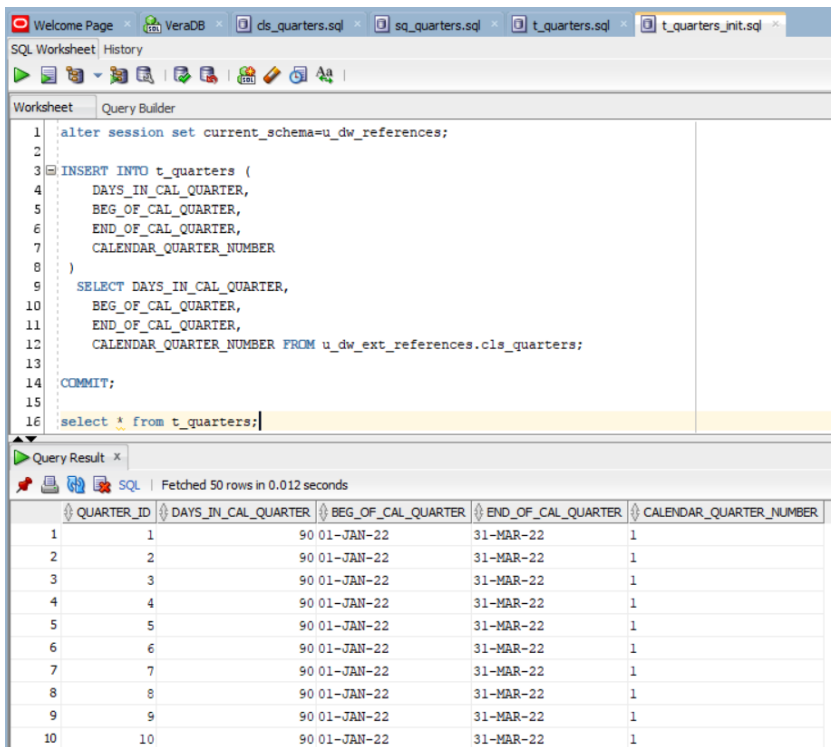
```
1  --drop sequence u_dw_references.sq_quarter_id;
2
3  alter session set current_schema=u_dw_references;
4
5  create sequence u_dw_references.sq_quarter_id start with 1;
6
7  grant SELECT on u_dw_references.sq_quarter_id to u_dw_ext_references;
8
```

- t



```
1  --ALTER USER u_dw_references quota unlimited on TS_REFERENCES_DATA_01;
2  --DROP TABLE t_quarters;
3  alter session set current_schema=u_dw_references;
4
5  Create table t_quarters (
6  QUARTER_ID              NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
7  DAYS_IN_CAL_QUARTER     NUMBER,
8  BEG_OF_CAL_QUARTER      DATE,
9  END_OF_CAL_QUARTER      DATE,
10 CALENDAR_QUARTER_NUMBER VARCHAR2(1),
11 CONSTRAINT "PK_DW.T_QUARTER" PRIMARY KEY ( QUARTER_ID ) USING INDEX TABLESPACE ts_references_idx_01
12 );
13
```

- t_init



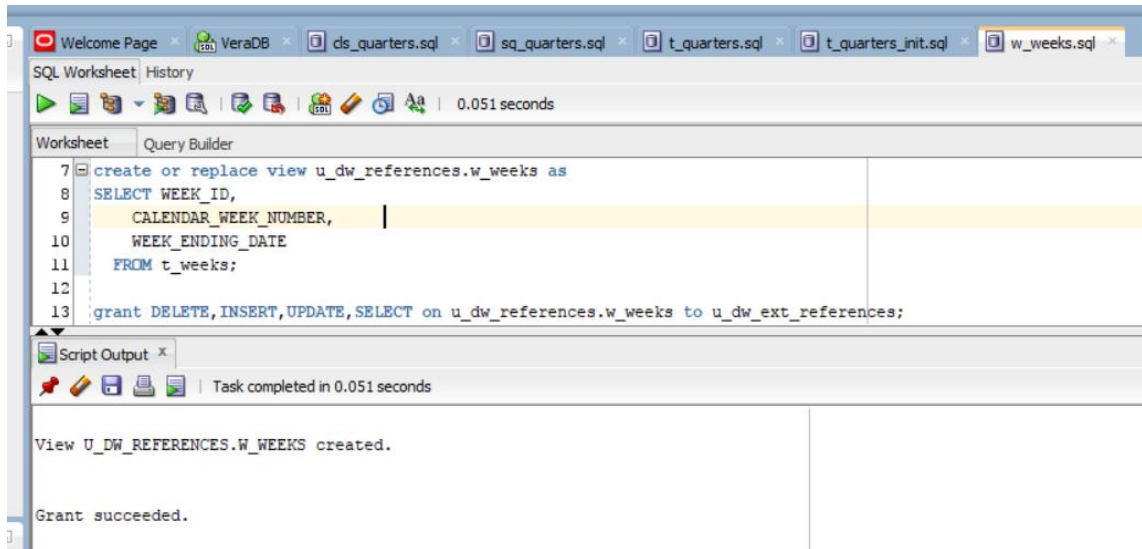
```
1  alter session set current_schema=u_dw_references;
2
3  INSERT INTO t_quarters (
4  DAYS_IN_CAL_QUARTER,
5  BEG_OF_CAL_QUARTER,
6  END_OF_CAL_QUARTER,
7  CALENDAR_QUARTER_NUMBER
8  )
9  SELECT DAYS_IN_CAL_QUARTER,
10 BEG_OF_CAL_QUARTER,
11 END_OF_CAL_QUARTER,
12 CALENDAR_QUARTER_NUMBER FROM u_dw_ext_references.cls_quarters;
13
14 COMMIT;
15
16 select * from t_quarters;
```

Query Result x

SQL | Fetched 50 rows in 0.012 seconds

QUARTER_ID	DAYS_IN_CAL_QUARTER	BEG_OF_CAL_QUARTER	END_OF_CAL_QUARTER	CALENDAR_QUARTER_NUMBER
1	1	90 01-JAN-22	31-MAR-22	1
2	2	90 01-JAN-22	31-MAR-22	1
3	3	90 01-JAN-22	31-MAR-22	1
4	4	90 01-JAN-22	31-MAR-22	1
5	5	90 01-JAN-22	31-MAR-22	1
6	6	90 01-JAN-22	31-MAR-22	1
7	7	90 01-JAN-22	31-MAR-22	1
8	8	90 01-JAN-22	31-MAR-22	1
9	9	90 01-JAN-22	31-MAR-22	1
10	10	90 01-JAN-22	31-MAR-22	1

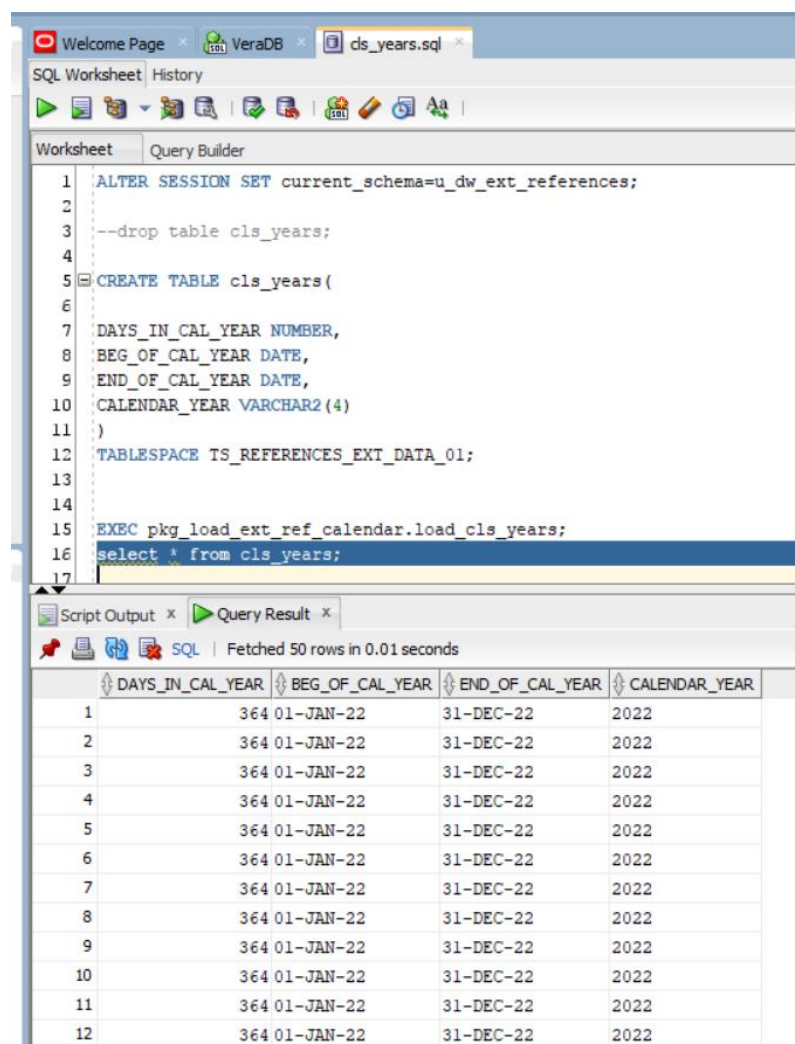
- W



2.5. Task 05: CREATE DW.T_YEARS

Create Object T_YEARS on DW layer:

- cls



- sq

```

1  --drop sequence u_dw_references.sq_quarter_id;
2
3  alter session set current_schema=u_dw_references;
4
5  create sequence u_dw_references.sq_year_id start with 1;
6
7  grant SELECT on u_dw_references.sq_year_id to u_dw_ext_references;
8
9

```

- t

```

1  alter session set current_schema=u_dw_references;
2  --drop table t_months;
3
4  Create table t_years (
5      YEAR_ID NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
6      DAYS_IN_CAL_YEAR NUMBER,
7      BEG_OF_CAL_YEAR DATE,
8      END_OF_CAL_YEAR DATE,
9      CALENDAR_YEAR VARCHAR2(4),
10     CONSTRAINT "FK_DW.T_YEAR" PRIMARY KEY ( YEAR_ID ) USING INDEX TABLESPACE ts_references_idx_01
11 );

```

- t_init

```

1  alter session set current_schema=u_dw_references;
2
3  INSERT INTO t_years (
4      DAYS_IN_CAL_YEAR,
5      BEG_OF_CAL_YEAR,
6      END_OF_CAL_YEAR,
7      CALENDAR_YEAR
8  )
9      SELECT DAYS_IN_CAL_YEAR,
10         BEG_OF_CAL_YEAR,
11         END_OF_CAL_YEAR,
12         CALENDAR_YEAR FROM u_dw_ext_references.cls_years;
13
14 COMMIT;
15
16 SELECT * FROM t_years;

```

Script Output x Query Result x

SQL | Fetched 50 rows in 0.01 seconds

	YEAR_ID	DAYS_IN_CAL_YEAR	BEG_OF_CAL_YEAR	END_OF_CAL_YEAR	CALENDAR_YEAR
1	1	364	01-JAN-22	31-DEC-22	2022
2	2	364	01-JAN-22	31-DEC-22	2022
3	3	364	01-JAN-22	31-DEC-22	2022
4	4	364	01-JAN-22	31-DEC-22	2022
5	5	364	01-JAN-22	31-DEC-22	2022
6	6	364	01-JAN-22	31-DEC-22	2022
7	7	364	01-JAN-22	31-DEC-22	2022
8	8	364	01-JAN-22	31-DEC-22	2022
9	9	364	01-JAN-22	31-DEC-22	2022
10	10	364	01-JAN-22	31-DEC-22	2022
11	11	364	01-JAN-22	31-DEC-22	2022

- W

The screenshot shows a SQL IDE with multiple tabs. The active tab is 'w_years.sql'. The SQL script in the editor is as follows:

```

1 create or replace view u_dw_references.w_years as
2 SELECT YEAR_ID,
3       DAYS_IN_CAL_YEAR,
4       BEG_OF_CAL_YEAR,
5       END_OF_CAL_YEAR,
6       CALENDAR_YEAR
7 FROM t_years;
8
9 grant DELETE,INSERT,UPDATE,SELECT on u_dw_references.w_years to u_dw_ext_references;
10

```

Below the editor, the 'Script Output' pane shows the execution results:

```

View U_DW_REFERENCES.W_YEARS created.

Grant succeeded.

```

The task completed in 0.063 seconds.

Physical Diagram:

T_DAYS		
<u>DAY_ID</u>	NUMBER	<pk>
DAY_NAME	VARCHAR2(44)	
DAY_NUMBER_IN_WEEK	VARCHAR2(1)	
DAY_NUMBER_IN_MONTH	VARCHAR2(2)	
DAY_NUMBER_IN_YEAR	VARCHAR2(3)	

T_WEEKS		
<u>WEEK_ID</u>	NUMBER	<pk>
CALENDAR_WEEK_NUMBER	VARCHAR2(1)	
WEEK_ENDING_DATE	DATE	

T_MONTHS		
<u>MONTH_ID</u>	NUMBER	<pk>
CALENDAR_MONTH_NUMBER	VARCHAR2(2)	
DAYS_IN_CAL_MONTH	VARCHAR2(2)	
END_OF_CAL_MONTH	DATE	
CALENDAR_MONTH_NAME	VARCHAR2(32)	

T_QUARTERS		
<u>QUARTER_ID</u>	NUMBER	<pk>
DAYS_IN_CAL_QUARTER	NUMBER	
BEG_OF_CAL_QUARTER	DATE	
END_OF_CAL_QUARTER	DATE	
CALENDAR_QUARTER_NUMBER	VARCHAR2(1)	

T_YEARS		
<u>YEAR_ID</u>	NUMBER	<pk>
DAYS_IN_CAL_YEAR	NUMBER	
BEG_OF_CAL_YEAR	DATE	
END_OF_CAL_YEAR	DATE	
CALENDAR_YEAR	VARCHAR2(4)	