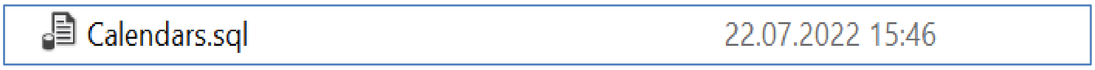
Lab Report #7
Sadovskaya Veronika

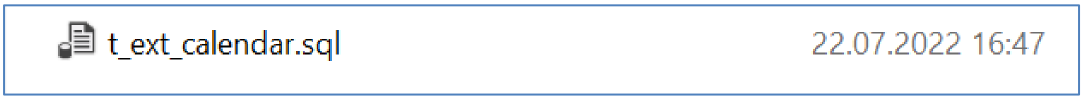
GitHub: https://github.com/sdveronika/DataMola22

Data Source - this is the script that generated the calendar.



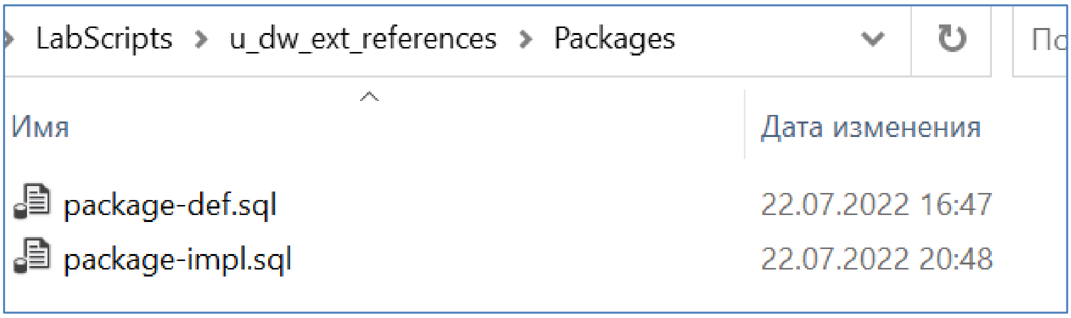
Calendars.sql	22.07.2022 15:46

Next, an external table was created, which is not included in the tablespace, for overloading data into it – it's **Stage.**

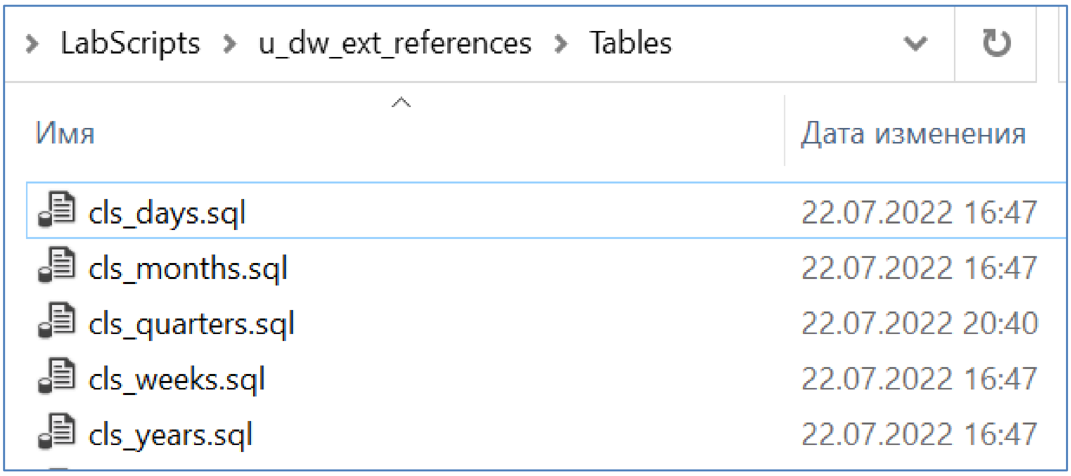


t_ext_calendar.sql	22.07.2022 16:47

From it, using the package, the data was divided and placed in cleasing tables - this is the **DW**.



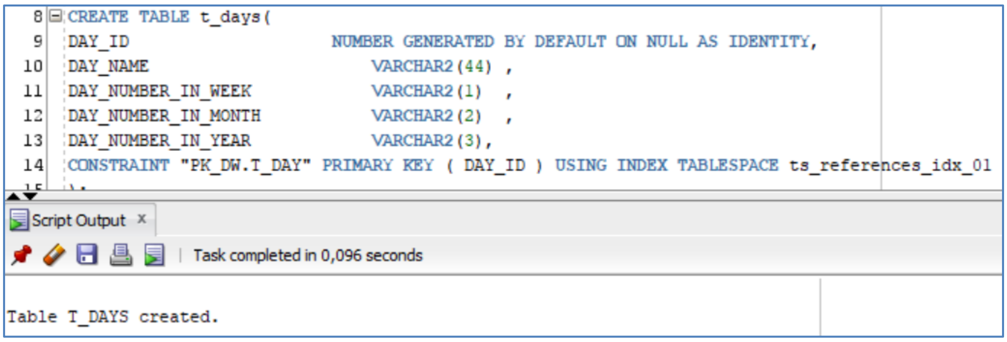
Имя	Дата изменения
package-def.sql	22.07.2022 16:47
package-impl.sql	22.07.2022 20:48



Имя	Дата изменения
cls_days.sql	22.07.2022 16:47
cls_months.sql	22.07.2022 16:47
cls_quarters.sql	22.07.2022 20:40
cls_weeks.sql	22.07.2022 16:47
cls_years.sql	22.07.2022 16:47

Further, from cleansing tables, the following tables were filled.

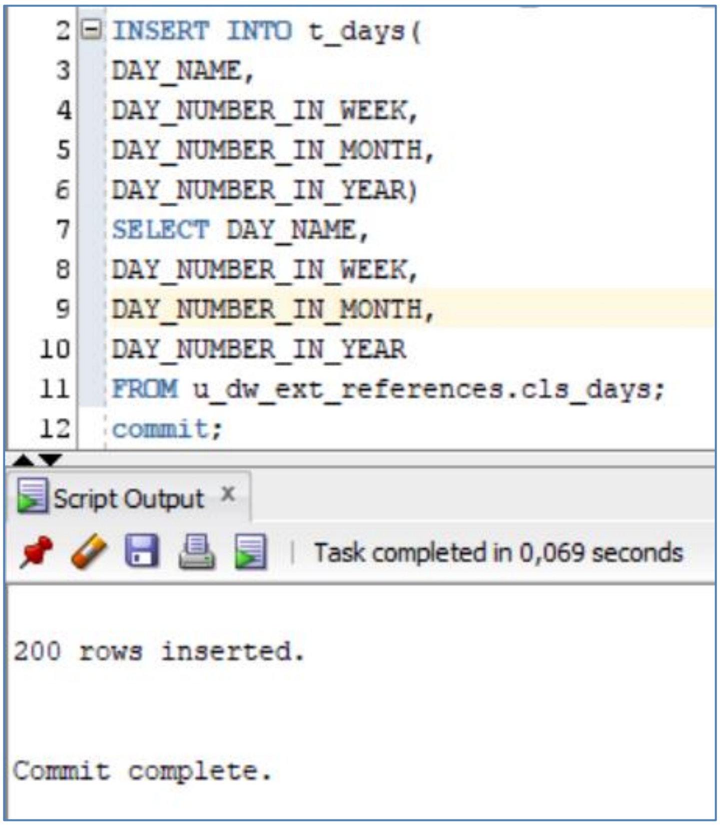
Task 01: CREATE DW.T_DAYS



```
CREATE TABLE t_days(
DAY_ID                          NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
DAY_NAME                             VARCHAR2(44) ,
DAY_NUMBER_IN_WEEK                   VARCHAR2(1)  ,
DAY_NUMBER_IN_MONTH                  VARCHAR2(2)  ,
DAY_NUMBER_IN_YEAR                   VARCHAR2(3),
CONSTRAINT "PK_DW.T_DAY" PRIMARY KEY ( DAY_ID ) USING INDEX TABLESPACE ts_references_idx_01
);
```

```
Table T_DAYS created.
```

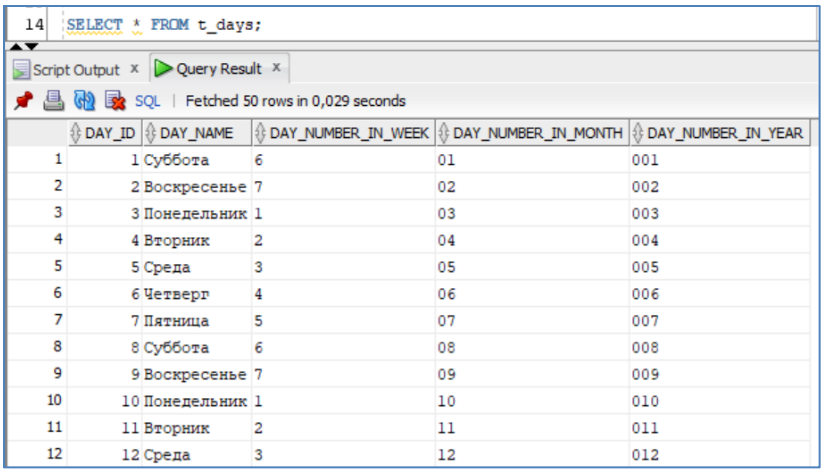
Figure 1.1 – Create table t_days



```
INSERT INTO t_days(
DAY_NAME,
DAY_NUMBER_IN_WEEK,
DAY_NUMBER_IN_MONTH,
DAY_NUMBER_IN_YEAR)
SELECT DAY_NAME,
DAY_NUMBER_IN_WEEK,
DAY_NUMBER_IN_MONTH,
DAY_NUMBER_IN_YEAR
FROM u_dw_ext_references.cls_days;
commit;
```

```
200 rows inserted.


Commit complete.
```

Figure 1.2 – Populate table t_days

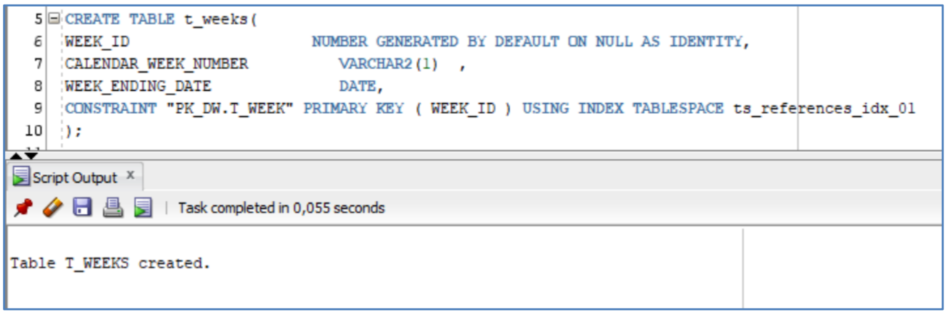


```
SELECT * FROM t_days;
```

	DAY_ID	DAY_NAME	DAY_NUMBER_IN_WEEK	DAY_NUMBER_IN_MONTH	DAY_NUMBER_IN_YEAR
1	1	Суббота	6	01	001
2	2	Воскресенье	7	02	002
3	3	Понедельник	1	03	003
4	4	Вторник	2	04	004
5	5	Среда	3	05	005
6	6	Четверг	4	06	006
7	7	Пятница	5	07	007
8	8	Суббота	6	08	008
9	9	Воскресенье	7	09	009
10	10	Понедельник	1	10	010
11	11	Вторник	2	11	011
12	12	Среда	3	12	012

Figure 1.3 – Select from table t_days

Task 02: CREATE DW.T_WEEKS

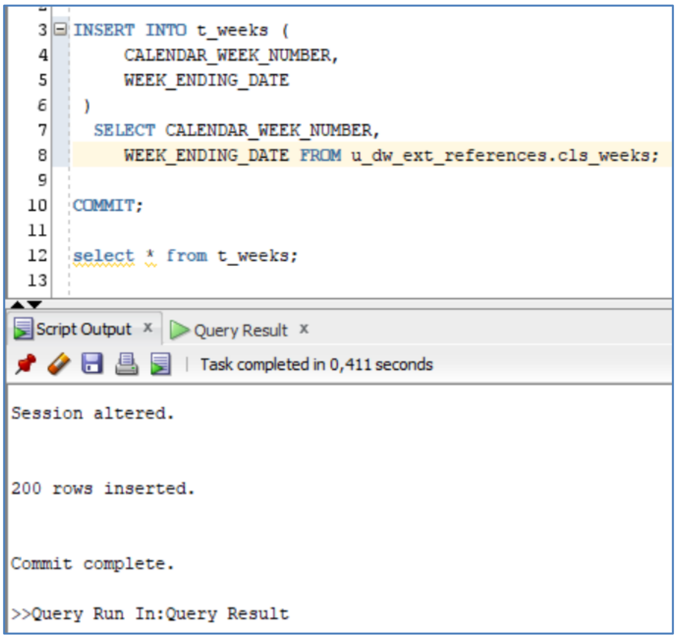


```
CREATE TABLE t_weeks(
WEEK_ID                       NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
CALENDAR_WEEK_NUMBER             VARCHAR2(1)  ,
WEEK_ENDING_DATE                 DATE,
CONSTRAINT "PK_DW.T_WEEK" PRIMARY KEY ( WEEK_ID ) USING INDEX TABLESPACE ts_references_idx_01
);
```

Task completed in 0,055 seconds

```
Table T_WEEKS created.
```

Figure 2.1 – Create table t_weeks

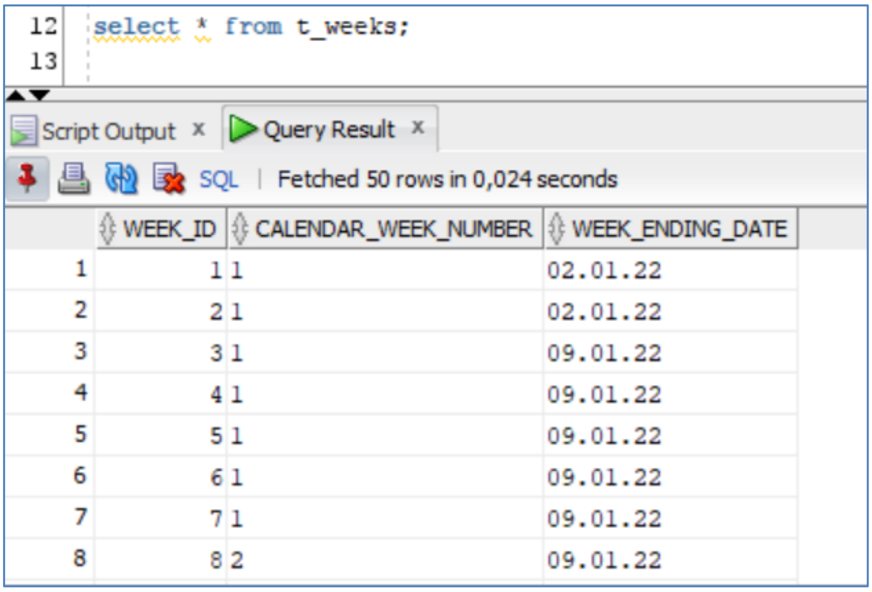


```
INSERT INTO t_weeks (
      CALENDAR_WEEK_NUMBER,
      WEEK_ENDING_DATE
 )
  SELECT CALENDAR_WEEK_NUMBER,
      WEEK_ENDING_DATE FROM u_dw_ext_references.cls_weeks;

COMMIT;

select * from t_weeks;
```

Task completed in 0,411 seconds

```
Session altered.


200 rows inserted.


Commit complete.

>>Query Run In:Query Result
```

Figure 2.2 – Populate table t_ weeks



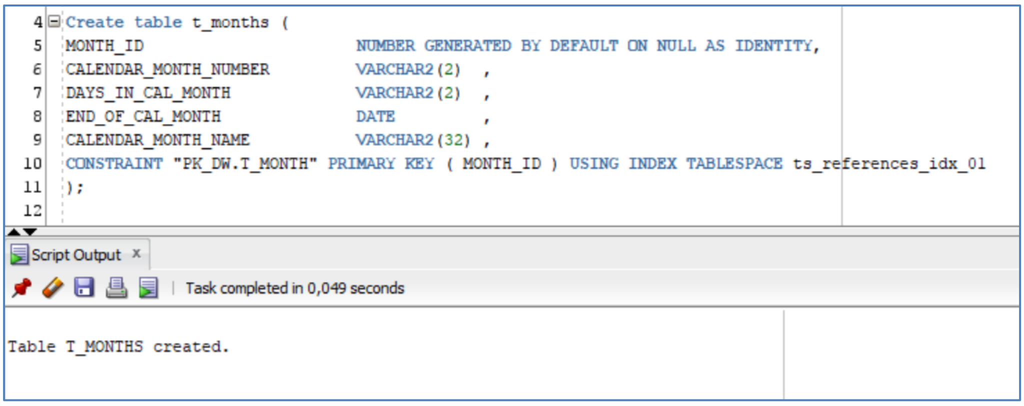
```
select * from t_weeks;
```

Fetched 50 rows in 0,024 seconds

	WEEK_ID	CALENDAR_WEEK_NUMBER	WEEK_ENDING_DATE
1	1	1	02.01.22
2	2	1	02.01.22
3	3	1	09.01.22
4	4	1	09.01.22
5	5	1	09.01.22
6	6	1	09.01.22
7	7	1	09.01.22
8	8	2	09.01.22

Figure 2.3 – Select from table t_ weeks

Task 03: CREATE DW.T_MONTHS

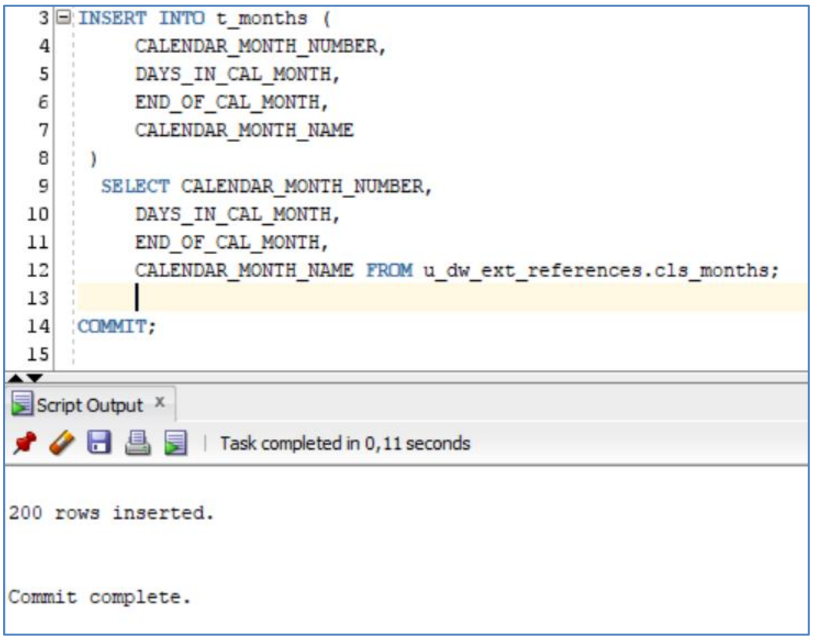


```
Create table t_months (
MONTH_ID                    NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
CALENDAR_MONTH_NUMBER       VARCHAR2(2)  ,
DAYS_IN_CAL_MONTH           VARCHAR2(2)  ,
END_OF_CAL_MONTH            DATE         ,
CALENDAR_MONTH_NAME         VARCHAR2(32) ,
CONSTRAINT "PK_DW.T_MONTH" PRIMARY KEY ( MONTH_ID ) USING INDEX TABLESPACE ts_references_idx_01
);
```

Task completed in 0,049 seconds

```
Table T_MONTHS created.
```

Figure 3.1 – Create table t_months

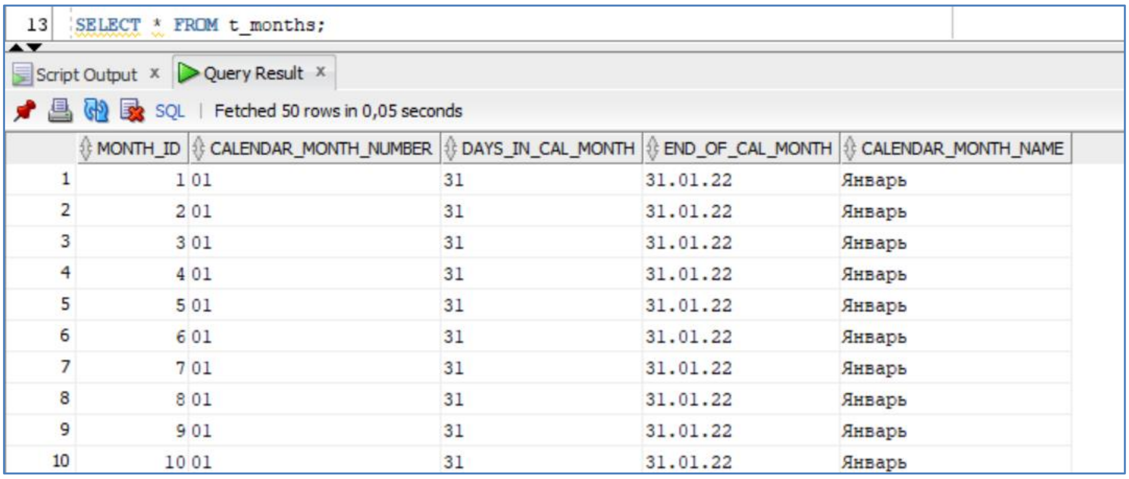


```
INSERT INTO t_months (
      CALENDAR_MONTH_NUMBER,
      DAYS_IN_CAL_MONTH,
      END_OF_CAL_MONTH,
      CALENDAR_MONTH_NAME
 )
  SELECT CALENDAR_MONTH_NUMBER,
      DAYS_IN_CAL_MONTH,
      END_OF_CAL_MONTH,
      CALENDAR_MONTH_NAME FROM u_dw_ext_references.cls_months;

COMMIT;
```

Task completed in 0,11 seconds

```
200 rows inserted.


Commit complete.
```

Figure 3.2 – Populate table t_ months



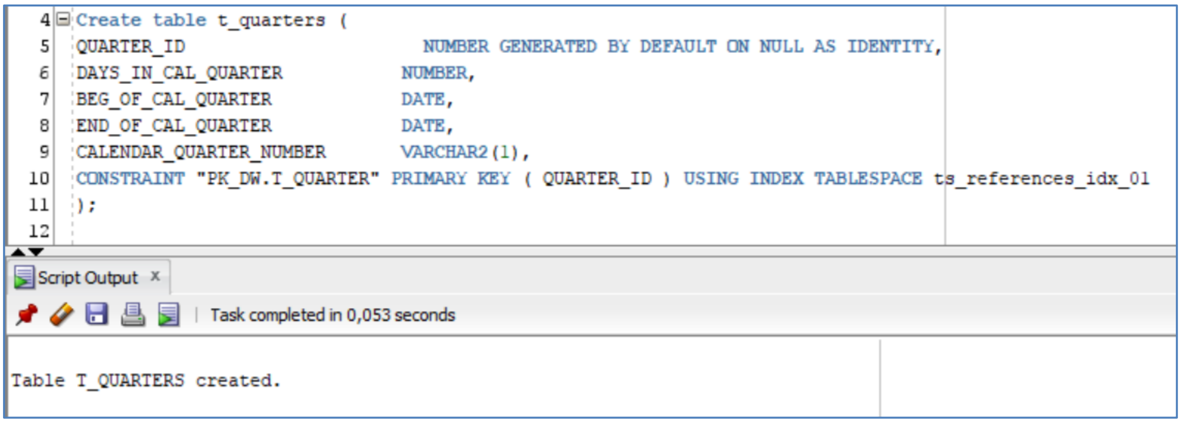
```
SELECT * FROM t_months;
```

Fetched 50 rows in 0,05 seconds

	MONTH_ID	CALENDAR_MONTH_NUMBER	DAYS_IN_CAL_MONTH	END_OF_CAL_MONTH	CALENDAR_MONTH_NAME
1	1	01	31	31.01.22	Январь
2	2	01	31	31.01.22	Январь
3	3	01	31	31.01.22	Январь
4	4	01	31	31.01.22	Январь
5	5	01	31	31.01.22	Январь
6	6	01	31	31.01.22	Январь
7	7	01	31	31.01.22	Январь
8	8	01	31	31.01.22	Январь
9	9	01	31	31.01.22	Январь
10	10	01	31	31.01.22	Январь

Figure 3.3 – Select from table t_ months

Task 04: CREATE DW.T_QUARTERS

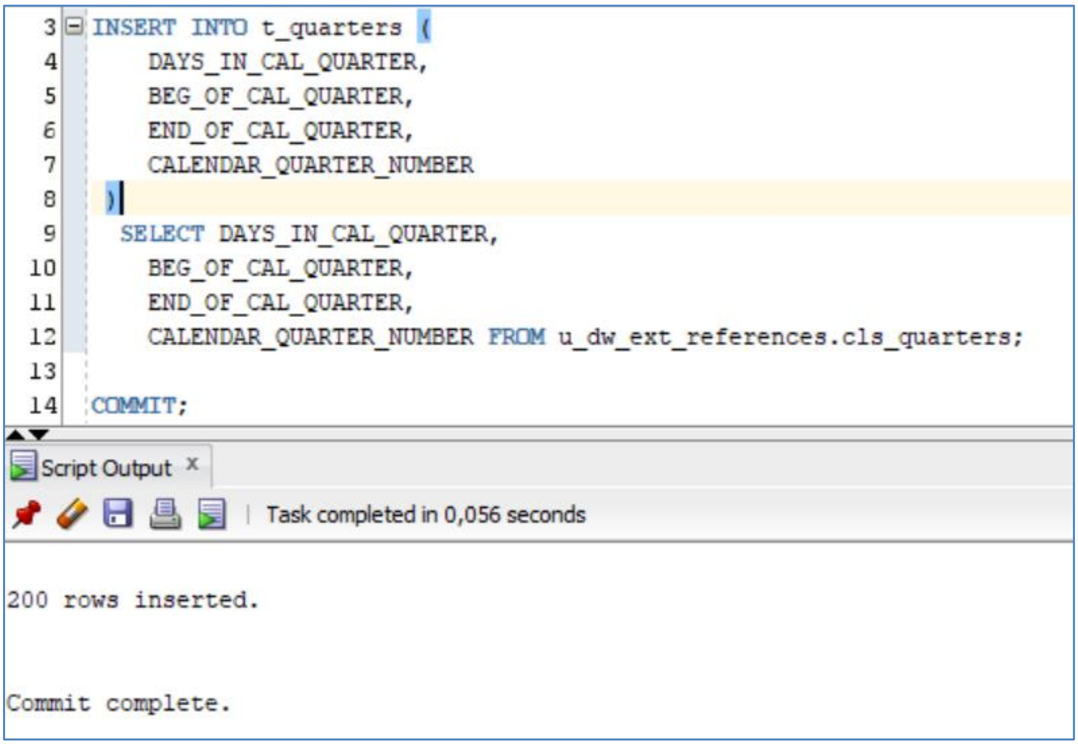


```
Create table t_quarters (
QUARTER_ID                    NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
DAYS_IN_CAL_QUARTER         NUMBER,
BEG_OF_CAL_QUARTER          DATE,
END_OF_CAL_QUARTER          DATE,
CALENDAR_QUARTER_NUMBER     VARCHAR2(1),
CONSTRAINT "PK_DW.T_QUARTER" PRIMARY KEY ( QUARTER_ID ) USING INDEX TABLESPACE ts_references_idx_01
);
```

Task completed in 0,053 seconds

```
Table T_QUARTERS created.
```

Figure 4.1 – Create table t_quarters

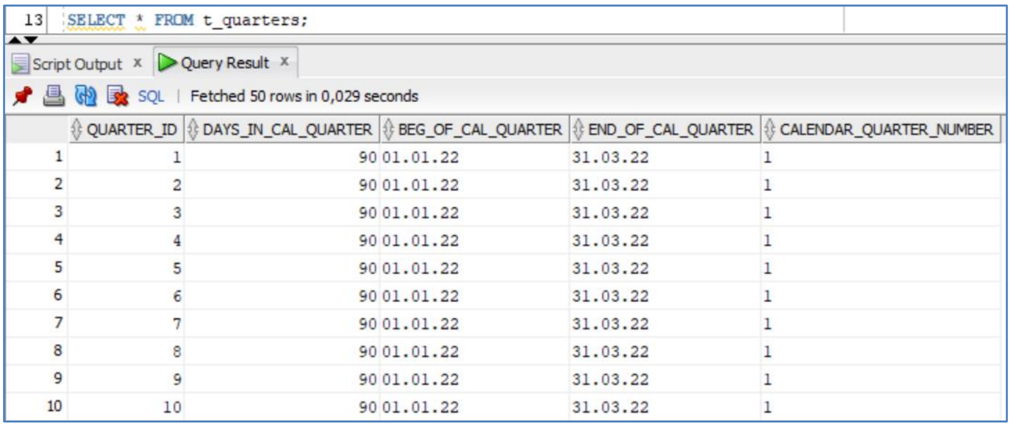


```
INSERT INTO t_quarters (
    DAYS_IN_CAL_QUARTER,
    BEG_OF_CAL_QUARTER,
    END_OF_CAL_QUARTER,
    CALENDAR_QUARTER_NUMBER
)
  SELECT DAYS_IN_CAL_QUARTER,
    BEG_OF_CAL_QUARTER,
    END_OF_CAL_QUARTER,
    CALENDAR_QUARTER_NUMBER FROM u_dw_ext_references.cls_quarters;

COMMIT;
```

Task completed in 0,056 seconds

```
200 rows inserted.


Commit complete.
```

Figure 4.2 – Populate table t_ quarters



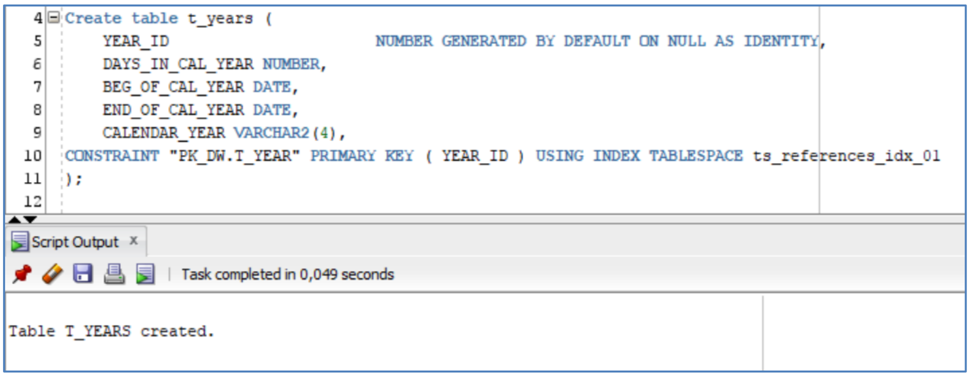
```
SELECT * FROM t_quarters;
```

Fetched 50 rows in 0,029 seconds

	QUARTER_ID	DAYS_IN_CAL_QUARTER	BEG_OF_CAL_QUARTER	END_OF_CAL_QUARTER	CALENDAR_QUARTER_NUMBER
1	1	90	01.01.22	31.03.22	1
2	2	90	01.01.22	31.03.22	1
3	3	90	01.01.22	31.03.22	1
4	4	90	01.01.22	31.03.22	1
5	5	90	01.01.22	31.03.22	1
6	6	90	01.01.22	31.03.22	1
7	7	90	01.01.22	31.03.22	1
8	8	90	01.01.22	31.03.22	1
9	9	90	01.01.22	31.03.22	1
10	10	90	01.01.22	31.03.22	1

Figure 4.3 – Select from table t_ quarters

Task 05: CREATE DW.T_YEARS

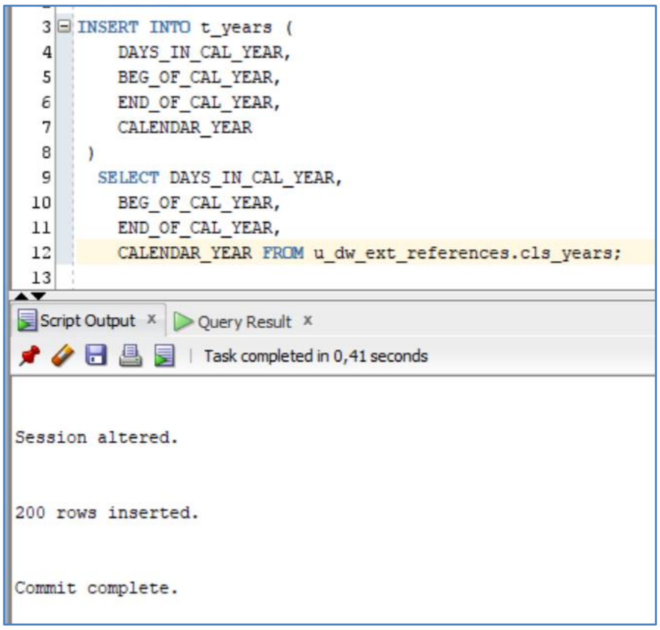


```sql
Create table t_years (
    YEAR_ID                    NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
    DAYS_IN_CAL_YEAR NUMBER,
    BEG_OF_CAL_YEAR DATE,
    END_OF_CAL_YEAR DATE,
    CALENDAR_YEAR VARCHAR2(4),
CONSTRAINT "PK_DW.T_YEAR" PRIMARY KEY ( YEAR_ID ) USING INDEX TABLESPACE ts_references_idx_01
);
```

Task completed in 0,049 seconds

```
Table T_YEARS created.
```

Figure 5.1 – Create table t_years

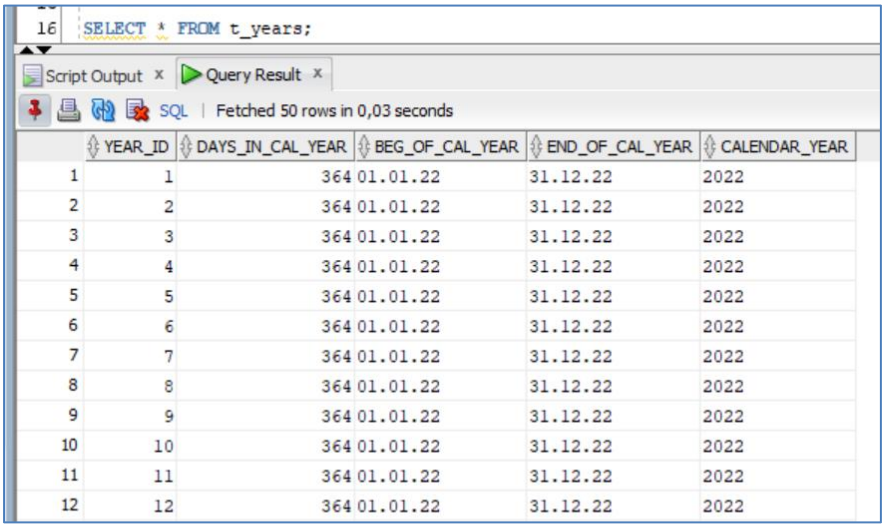


```sql
INSERT INTO t_years (
    DAYS_IN_CAL_YEAR,
    BEG_OF_CAL_YEAR,
    END_OF_CAL_YEAR,
    CALENDAR_YEAR
 )
  SELECT DAYS_IN_CAL_YEAR,
    BEG_OF_CAL_YEAR,
    END_OF_CAL_YEAR,
    CALENDAR_YEAR FROM u_dw_ext_references.cls_years;
```

Task completed in 0,41 seconds

```
Session altered.


200 rows inserted.


Commit complete.
```

Figure 5.2 – Populate table t_ years

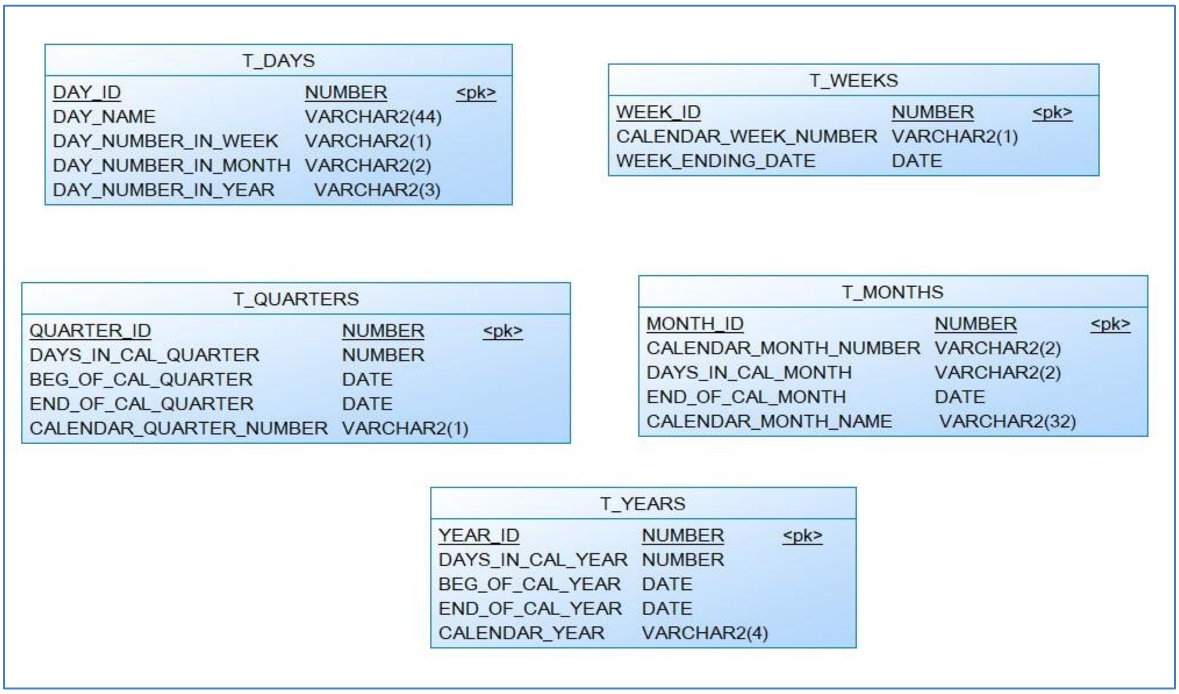


```sql
SELECT * FROM t_years;
```

Fetched 50 rows in 0,03 seconds

	YEAR_ID	DAYS_IN_CAL_YEAR	BEG_OF_CAL_YEAR	END_OF_CAL_YEAR	CALENDAR_YEAR
1	1	364	01.01.22	31.12.22	2022
2	2	364	01.01.22	31.12.22	2022
3	3	364	01.01.22	31.12.22	2022
4	4	364	01.01.22	31.12.22	2022
5	5	364	01.01.22	31.12.22	2022
6	6	364	01.01.22	31.12.22	2022
7	7	364	01.01.22	31.12.22	2022
8	8	364	01.01.22	31.12.22	2022
9	9	364	01.01.22	31.12.22	2022
10	10	364	01.01.22	31.12.22	2022
11	11	364	01.01.22	31.12.22	2022
12	12	364	01.01.22	31.12.22	2022

Figure 5.3 – Select from table t_ years



T_DAYS		
DAY_ID	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		
WEEK_ID	NUMBER	<pk>
CALENDAR_WEEK_NUMBER	VARCHAR2(1)	
WEEK_ENDING_DATE	DATE	

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

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

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

Figure – General Physical Diagram