U2M4.LW.Core PL/SQL

Shkrabatouskaya Vera

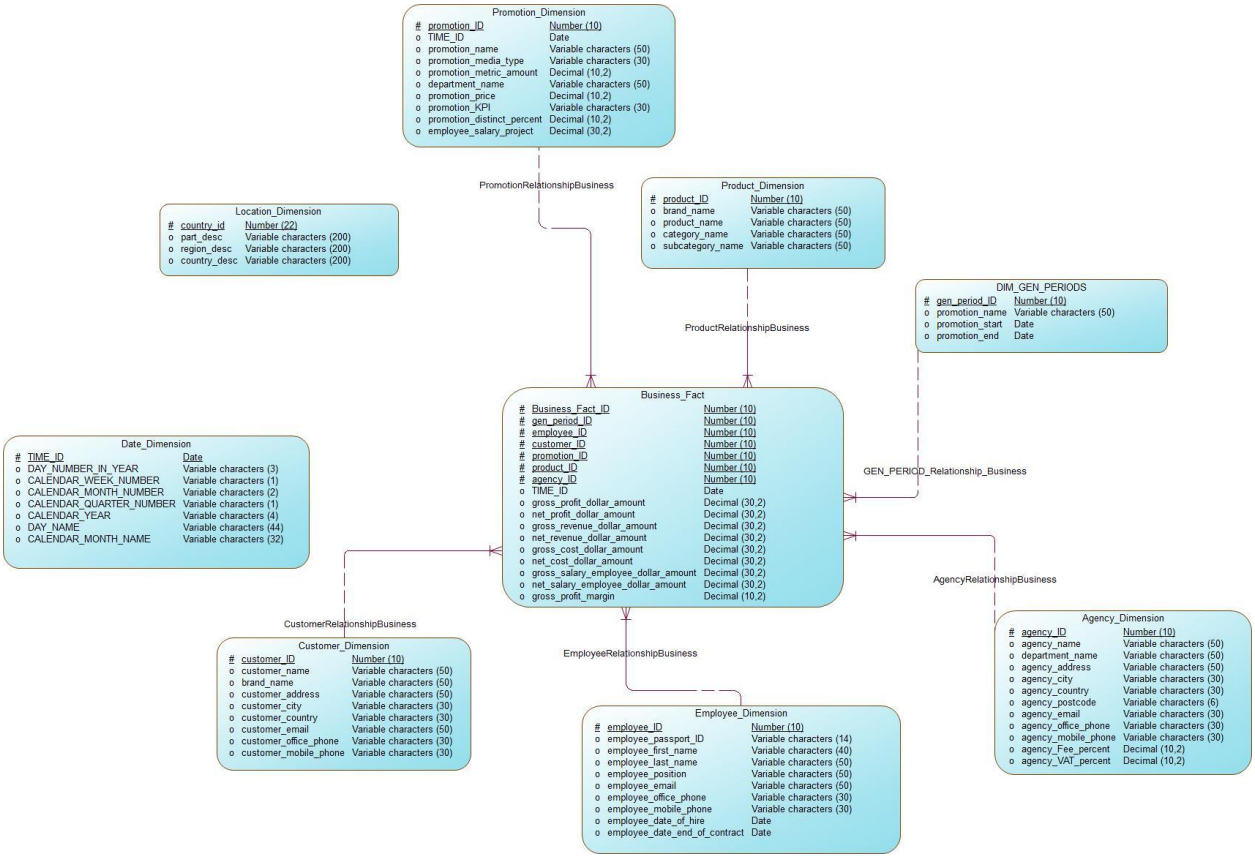
https://github.com/VeraShkrabatouskaya/DataMola_Data-Camping-2022

2. Business analyses tasks – Reports

At the customer's request, the following changes were made to the data warehouse in order to optimize costs.

History of changes

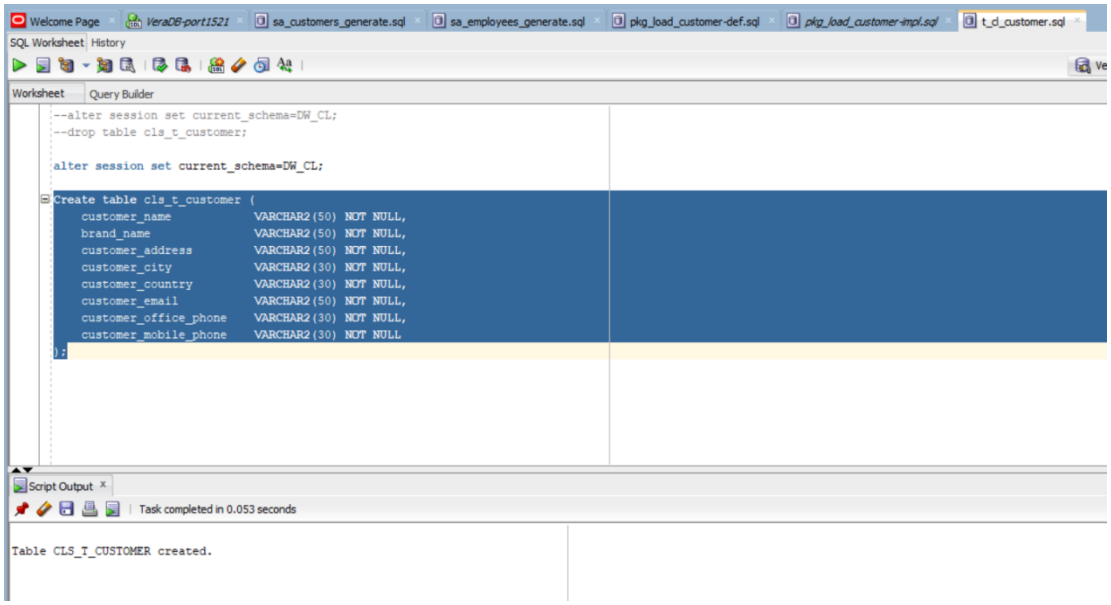
Name	Date	Reason for change	Version
Vera Shkrabatouskaya	7/25/2022	Creation of a STAR data warehouse model.	1.0 draft 1
Vera Shkrabatouskaya	7/26/2022	Changes after approval. Added details to the Business Model.	1.0 approved
Vera Shkrabatouskaya	8/3/2022	Changes after approval. The agency_country field was added to the Agency_Dimension table of the business model. The employee_salary field was renamed to employee_salary_project in Employee_Dimension.	1.1 approved
Vera Shkrabatouskaya	8/11/2022	Changes after approval. The Date_Dimension and Location_Dimension tables can be used as references, but will not be used in the data warehouse. The employee_salary_project field was moved from Employee_Dimension to Promotion_Dimension of the business model. The employee_date_of_dismissal field was renamed to employee_date_end_of_contract in Employee_Dimension. The department_name field and the agency_name field were removed from Employee_Dimension. The promotion_ID field has been renamed to promotion_name in DIM_GEN_PERIODS. Also, the promotion_description field in DIM_GEN_PERIODS has been removed. The promotion_name and promotion_media_type field in Product_Dimension have been removed. The TIME_ID field has been added to the Promotion_Dimension and Business_Fact. The customer_quantity, employee_quantity, and promotion_quantity parameters will not be calculated in this Business_Fact table. The revenue_cost_percent parameter has been replaced with gross_profit_margin due to the change of the repository's goal to calculate the margin indicator.	1.2 approved



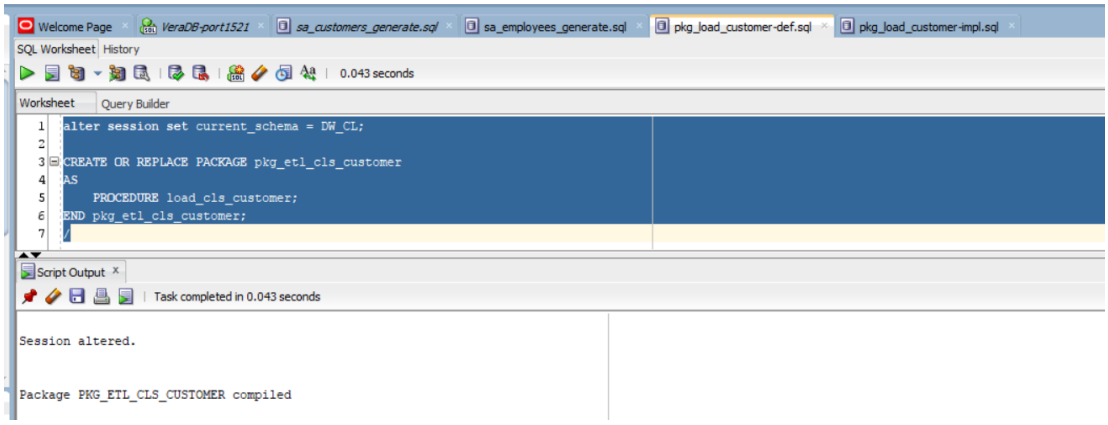
2.1. Task 01: Create Packages for Reload Dimension from SA_*

- cls_t_customer

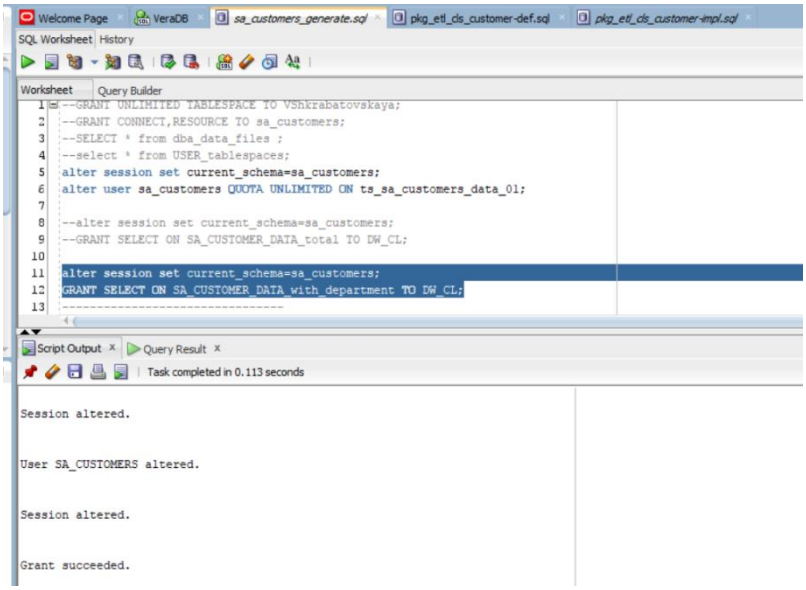
Create a cls_t_customer table at the DW - Cleansing Level.

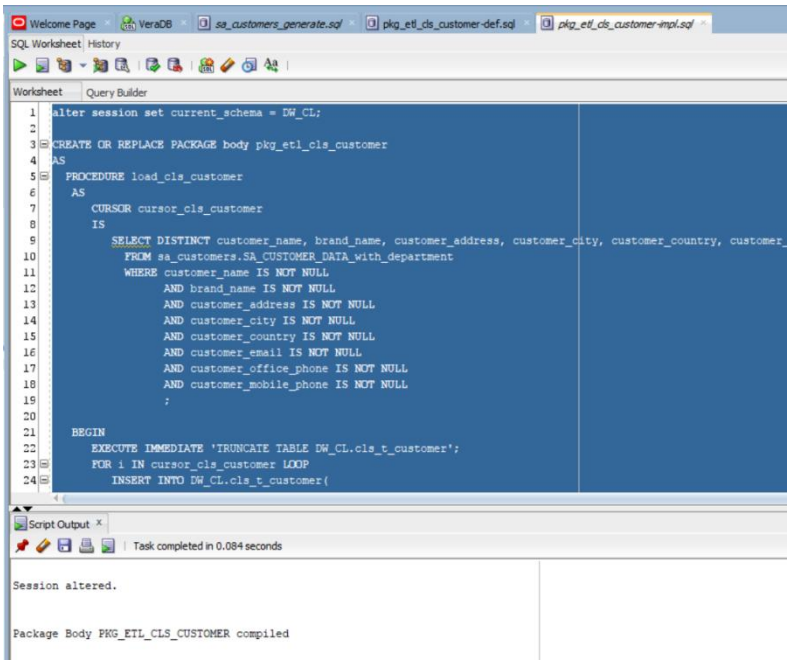


Let's create packages to get data from Storage level SA_* in DW - Cleanup level for the table cls_t_customer.



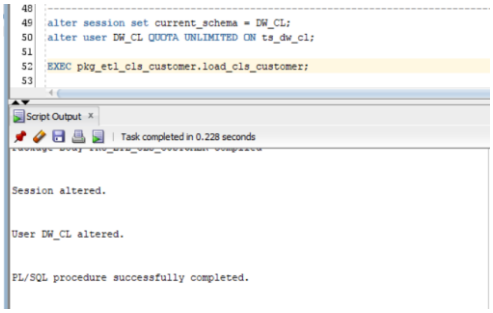
Grant permissions to user DW_CL in tablespace ts_dw_cl to use data from table SA_CUSTOMER_DATA_with_department in tablespace ts_sa_customers_data_01.

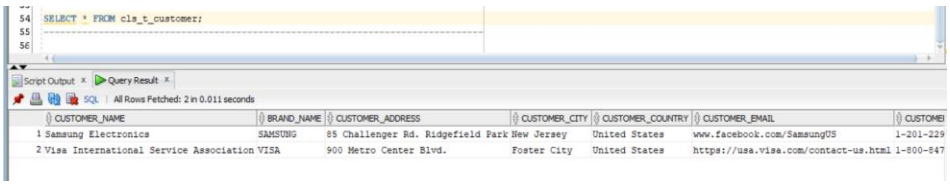




Code:

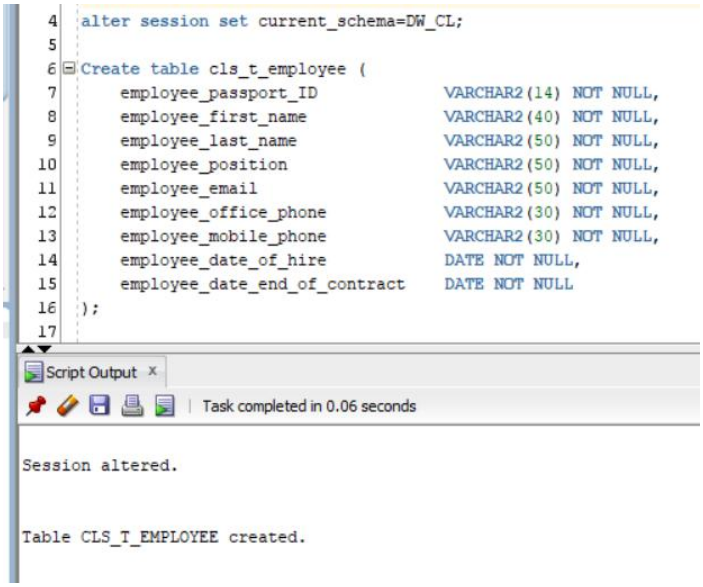
```
CREATE OR REPLACE PACKAGE body pkg_etl_cls_customer
AS
 PROCEDURE load_cls_customer
  AS
    CURSOR cursor_cls_customer
    IS
      SELECT DISTINCT customer_name, brand_name, customer_address, customer_city, customer_country, customer_email,
customer_office_phone, customer_mobile_phone
       FROM sa_customers.SA_CUSTOMER_DATA_with_department
       WHERE customer_name IS NOT NULL
           AND brand_name IS NOT NULL
           AND customer_address IS NOT NULL
           AND customer_city IS NOT NULL
           AND customer_country IS NOT NULL
           AND customer_email IS NOT NULL
           AND customer_office_phone IS NOT NULL
           AND customer_mobile_phone IS NOT NULL
           ;
  BEGIN
    EXECUTE IMMEDIATE 'TRUNCATE TABLE DW_CL.cls_t_customer';
    FOR i IN cursor_cls_customer LOOP
      INSERT INTO DW_CL.cls_t_customer(
           customer_name,
           brand_name,
           customer_address,
           customer_city,
           customer_country,
           customer_email,
           customer_office_phone,
           customer_mobile_phone)
         VALUES (
           i.customer_name,
           i.brand_name,
           i.customer_address,
           i.customer_city,
           i.customer_country,
           i.customer_email,
           i.customer_office_phone,
           i.customer_mobile_phone);
      EXIT WHEN cursor_cls_customer%NOTFOUND;
    END LOOP;
    COMMIT;
  END load_cls_customer;
END pkg_etl_cls_customer;
```





- cls_t_employee

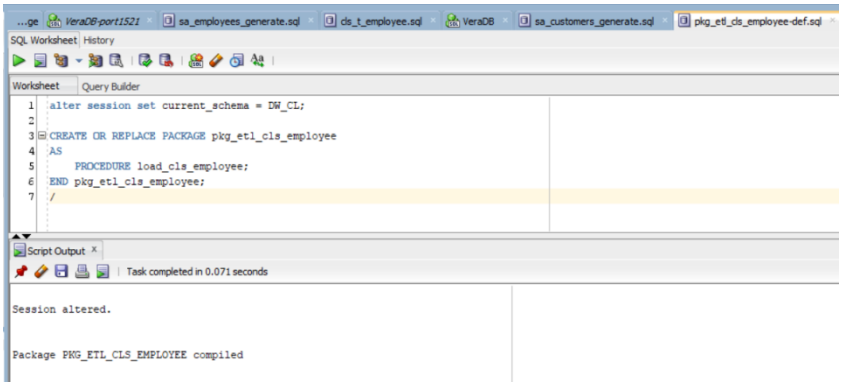
Create a cls_t_employee table at the DW - Cleansing Level.



```
alter session set current_schema=DW_CL;

Create table cls_t_employee (
    employee_passport_ID            VARCHAR2(14) NOT NULL,
    employee_first_name             VARCHAR2(40) NOT NULL,
    employee_last_name              VARCHAR2(50) NOT NULL,
    employee_position               VARCHAR2(50) NOT NULL,
    employee_email                  VARCHAR2(50) NOT NULL,
    employee_office_phone           VARCHAR2(30) NOT NULL,
    employee_mobile_phone           VARCHAR2(30) NOT NULL,
    employee_date_of_hire           DATE NOT NULL,
    employee_date_end_of_contract   DATE NOT NULL
);
```

```
Script Output
Task completed in 0.06 seconds

Session altered.

Table CLS_T_EMPLOYEE created.
```

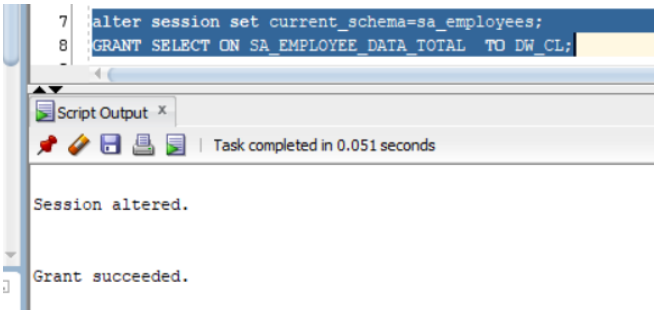
Let's create packages to get data from Storage level SA_* in DW - Cleanup level for the table cls_t_employee.



```
alter session set current_schema = DW_CL;

CREATE OR REPLACE PACKAGE pkg_etl_cls_employee
AS
    PROCEDURE load_cls_employee;
END pkg_etl_cls_employee;
/
```

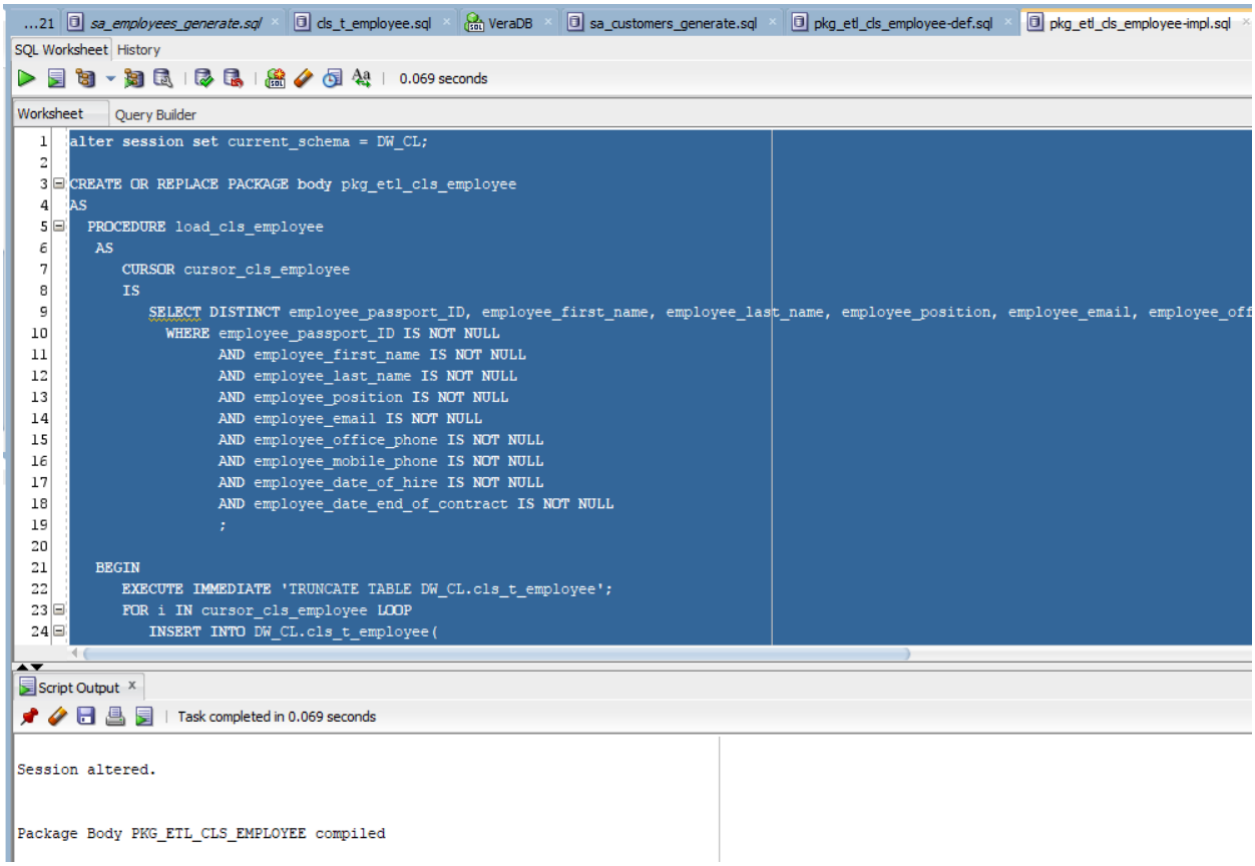
```
Script Output
Task completed in 0.071 seconds

Session altered.

Package PKG_ETL_CLS_EMPLOYEE compiled
```

Grant permissions to user DW_CL in tablespace ts_dw_cl to use data from table SA_EMPLOYEE_DATA_TOTAL in tablespace ts_sa_employees_data_01.



```
alter session set current_schema=sa_employees;
GRANT SELECT ON SA_EMPLOYEE_DATA_TOTAL  TO DW_CL;
```

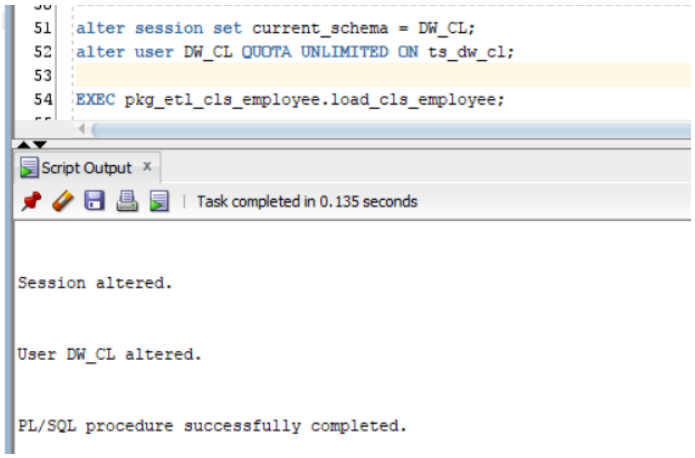
```
Script Output
Task completed in 0.051 seconds

Session altered.

Grant succeeded.
```

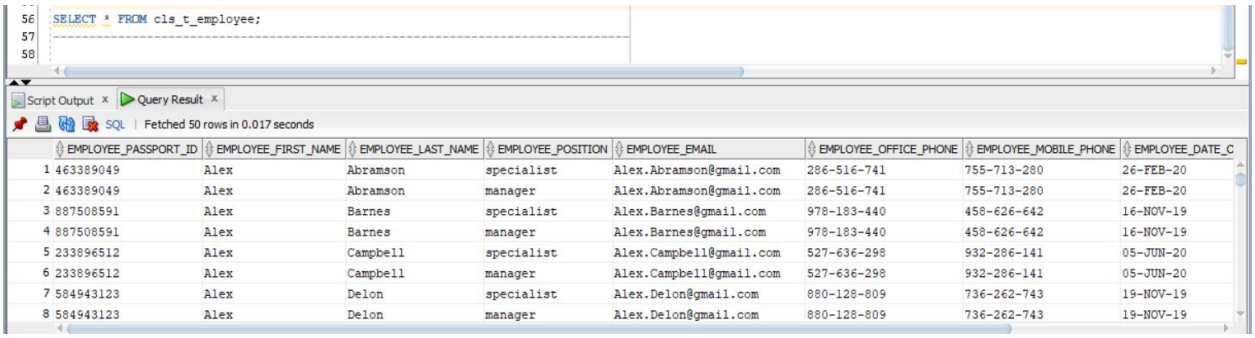


Code:

```
CREATE OR REPLACE PACKAGE body pkg_etl_cls_employee
AS
 PROCEDURE load_cls_employee
  AS
    CURSOR cursor_cls_employee
    IS
      SELECT DISTINCT employee_passport_ID, employee_first_name, employee_last_name, employee_position, employee_email,
employee_office_phone, employee_mobile_phone, employee_date_of_hire, employee_date_end_of_contract          FROM
sa_employees.SA_EMPLOYEE_DATA_TOTAL
       WHERE employee_passport_ID IS NOT NULL
          AND employee_first_name IS NOT NULL
          AND employee_last_name IS NOT NULL
          AND employee_position IS NOT NULL
          AND employee_email IS NOT NULL
          AND employee_office_phone IS NOT NULL
          AND employee_mobile_phone IS NOT NULL
          AND employee_date_of_hire IS NOT NULL
          AND employee_date_end_of_contract IS NOT NULL
          ;

  BEGIN
    EXECUTE IMMEDIATE 'TRUNCATE TABLE DW_CL.cls_t_employee';
    FOR i IN cursor_cls_employee LOOP
      INSERT INTO DW_CL.cls_t_employee(
          employee_passport_ID,
          employee_first_name,
          employee_last_name,
          employee_position,
          employee_email,
          employee_office_phone,
          employee_mobile_phone,
          employee_date_of_hire,
          employee_date_end_of_contract)
```

```
        VALUES (
          i.employee_passport_ID,
          i.employee_first_name,
          i.employee_last_name,
          i.employee_position,
          i.employee_email,
          i.employee_office_phone,
          i.employee_mobile_phone,
          i.employee_date_of_hire,
          i.employee_date_end_of_contract);
      EXIT WHEN cursor_cls_employee%NOTFOUND;
    END LOOP;

    COMMIT;
  END load_cls_employee;
END pkg_etl_cls_employee;
```

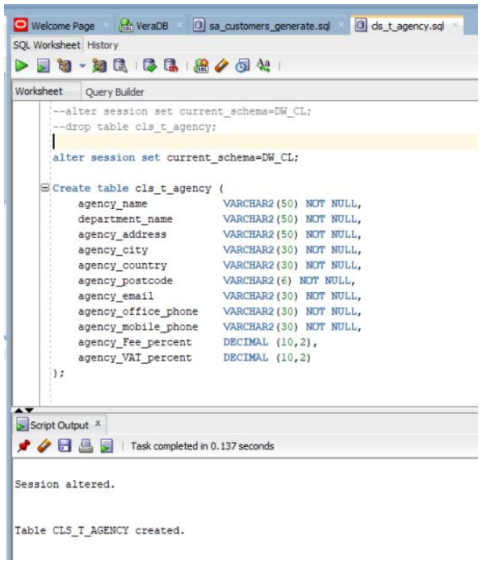




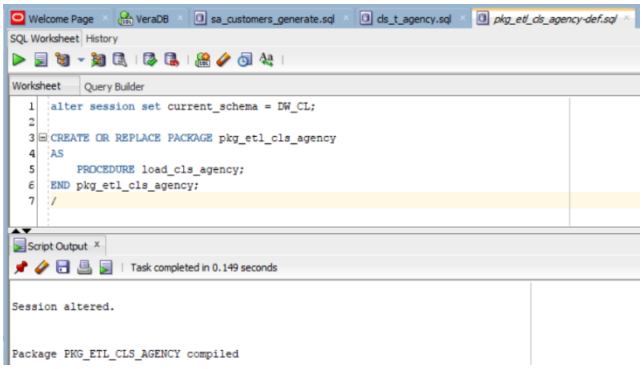
	EMPLOYEE_PASSPORT_ID	EMPLOYEE_FIRST_NAME	EMPLOYEE_LAST_NAME	EMPLOYEE_POSITION	EMPLOYEE_EMAIL	EMPLOYEE_OFFICE_PHONE	EMPLOYEE_MOBILE_PHONE	EMPLOYEE_DATE_C
1	463389049	Alex	Abramson	specialist	Alex.Abramson@gmail.com	286-516-741	755-713-280	26-FEB-20
2	463389049	Alex	Abramson	manager	Alex.Abramson@gmail.com	286-516-741	755-713-280	26-FEB-20
3	887508591	Alex	Barnes	specialist	Alex.Barnes@gmail.com	978-183-440	458-626-642	16-NOV-19
4	887508591	Alex	Barnes	manager	Alex.Barnes@gmail.com	978-183-440	458-626-642	16-NOV-19
5	233896512	Alex	Campbell	specialist	Alex.Campbell@gmail.com	527-636-298	932-286-141	05-JUN-20
6	233896512	Alex	Campbell	manager	Alex.Campbell@gmail.com	527-636-298	932-286-141	05-JUN-20
7	584943123	Alex	Delon	specialist	Alex.Delon@gmail.com	880-128-809	736-262-743	19-NOV-19
8	584943123	Alex	Delon	manager	Alex.Delon@gmail.com	880-128-809	736-262-743	19-NOV-19

- cls_t_agency

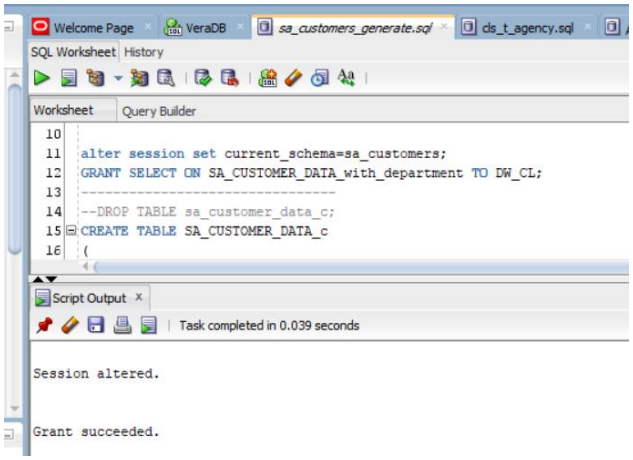
Create a cls_t_agency table at the DW - Cleansing Level.

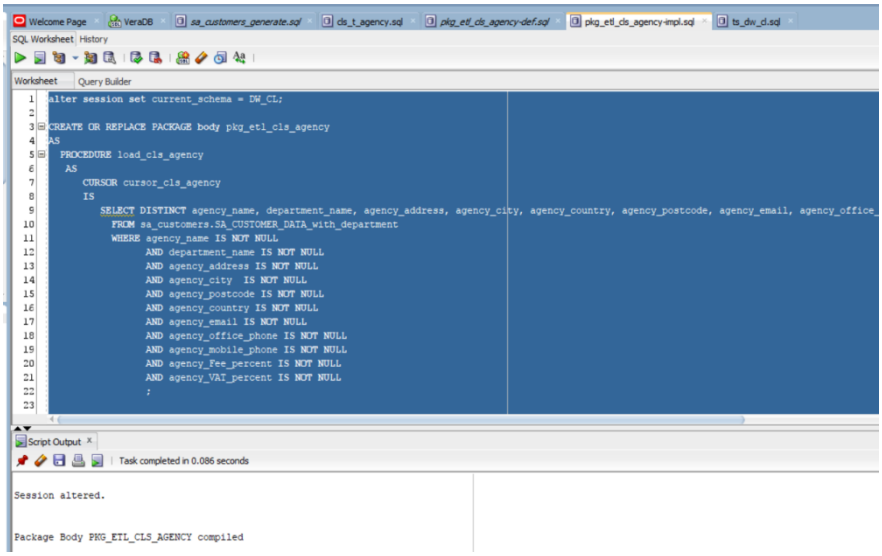


Let's create packages to get data from Storage level SA_* in DW - Cleanup level for the table cls_t_agency.



Grant permissions to user DW_CL in tablespace ts_dw_cl to use data from table SA_CUSTOMER_DATA_with_department in tablespace ts_sa_customers_data_01.





Code:

```
CREATE OR REPLACE PACKAGE body pkg_etl_cls_agency
AS
 PROCEDURE load_cls_agency
  AS
    CURSOR cursor_cls_agency
    IS
      SELECT DISTINCT agency_name, department_name, agency_address, agency_city, agency_country, agency_postcode, agency_email,
agency_office_phone, agency_mobile_phone, agency_Fee_percent, agency_VAT_percent
       FROM sa_customers.SA_CUSTOMER_DATA_with_department
       WHERE agency_name IS NOT NULL
          AND department_name IS NOT NULL
          AND agency_address IS NOT NULL
          AND agency_city  IS NOT NULL
          AND agency_postcode IS NOT NULL
          AND agency_country IS NOT NULL
          AND agency_email IS NOT NULL
          AND agency_office_phone IS NOT NULL
          AND agency_mobile_phone IS NOT NULL
          AND agency_Fee_percent IS NOT NULL
          AND agency_VAT_percent IS NOT NULL
          ;

  BEGIN
    EXECUTE IMMEDIATE 'TRUNCATE TABLE DW_CL.cls_t_agency';
    FOR i IN cursor_cls_agency LOOP
      INSERT INTO DW_CL.cls_t_agency(
          agency_name,
          department_name,
          agency_address,
          agency_city,
          agency_postcode,
          agency_country,
          agency_email,
          agency_office_phone,
          agency_mobile_phone,
          agency_Fee_percent,
          agency_VAT_percent)
        VALUES (
          i.agency_name,
          i.department_name,
          i.agency_address,
          i.agency_city,
          i.agency_postcode,
```

```
            i.agency_country,
            i.agency_email,
            i.agency_office_phone,
            i.agency_mobile_phone,
            i.agency_Fee_percent,
            i.agency_VAT_percent);
      EXIT WHEN cursor_cls_agency%NOTFOUND;
    END LOOP;

    COMMIT;
  END load_cls_agency;
END pkg_etl_cls_agency;
```

```
58  alter session set current_schema = DW_CL;
59  alter user DW_CL QUOTA UNLIMITED ON ts_dw_cl;
60
61  EXEC pkg_etl_cls_agency.load_cls_agency;
```

Script Output | Task completed in 0.361 seconds

```
Package Body PKG_ETL_CLS_AGENCY compiled


Session altered.


User DW_CL altered.


PL/SQL procedure successfully completed.
```

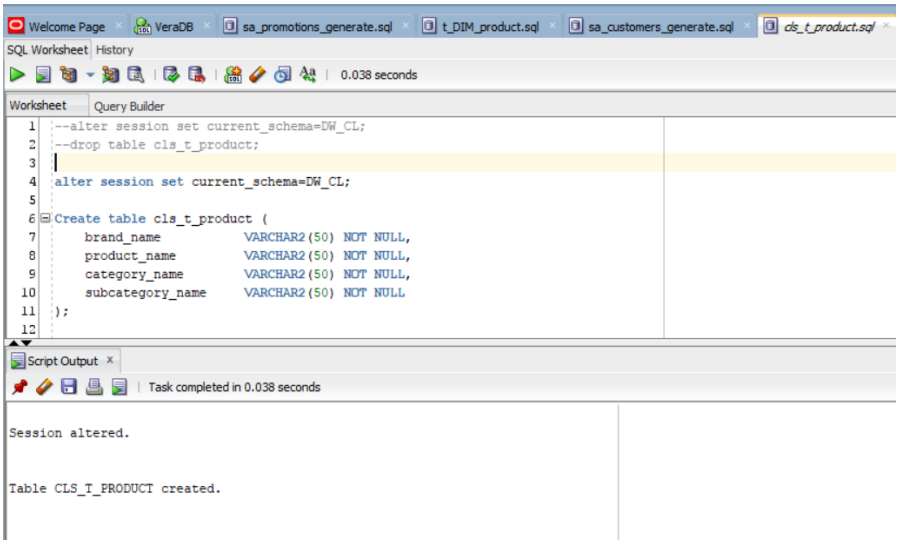
```
63  SELECT * FROM cls_t_agency;
64  ------------------------------------------------------------------------------
65
```

Script Output | Query Result | SQL | Fetched 50 rows in 0.026 seconds

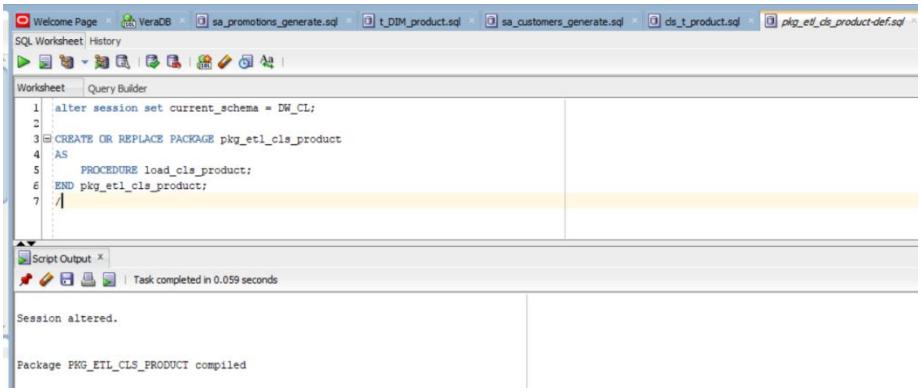
	AGENCY_NAME	DEPARTMENT_NAME	AGENCY_ADDRESS	AGENCY_CITY	AGENCY_COUNTRY	AGENCY_POSTCODE	AGENCY_EMAIL	AGENC
1	Starcom	Media Planning and Buying	375 Hudson St. Floor 12	New York	United States	10014	www.starcomww.com/contact	1-212-
2	Starcom	Media Planning and Buying	21 Harris Street, Pyrmont	Sydney	Australia	2009	www.starcomww.com/contact	612-86
3	Starcom	Media Planning and Buying	30-34 chemin Vert Street	Paris	France	75011	www.starcomww.com/contact	33-1-5
4	Starcom	Media Planning and Buying	Stadelhoferstrasse 258	Zurich	Switzerland	8001	www.starcomww.com/contact	41-43-
5	Starcom	Media Planning and Buying	21 Academias Ave. KEMA Building	Nocosia	Cyprus	2107	www.starcomww.com/contact	357-22
6	Starcom	Media Planning and Buying	29/F, Paul Y. Centre 51 Hung To Road Kwun Tong	Hong Kong	China	12345	www.starcomww.com/contact	39-051
7	Starcom	Media Planning and Buying	Paul Lincke Ufer 39-40	Berlin	Germany	10999	www.starcomww.com/contact	49-211
8	Starcom	Media Planning and Buying	21/F Henderson 688	Shanghai	China	12345	www.starcomww.com/contact	39-051
9	Starcom	Media Planning and Buying	City Stars 4th Floor	Cairo	Egypt	10521	www.starcomww.com/contact	202-0-

- cls_t_product

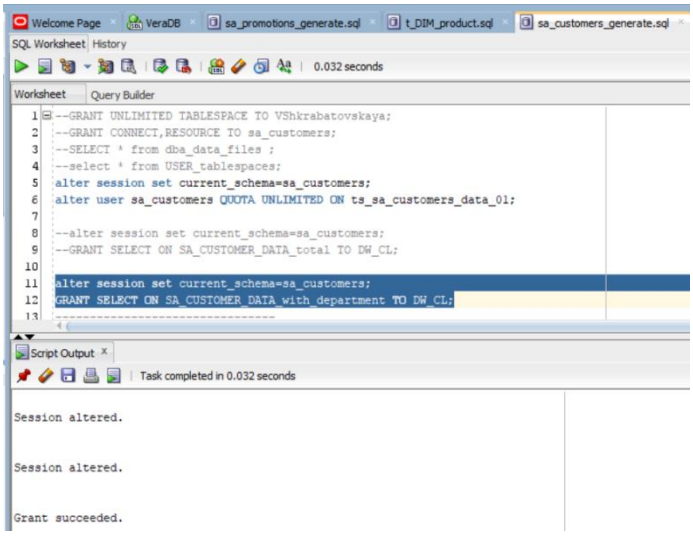
Create a cls_t_product table at the DW - Cleansing Level.

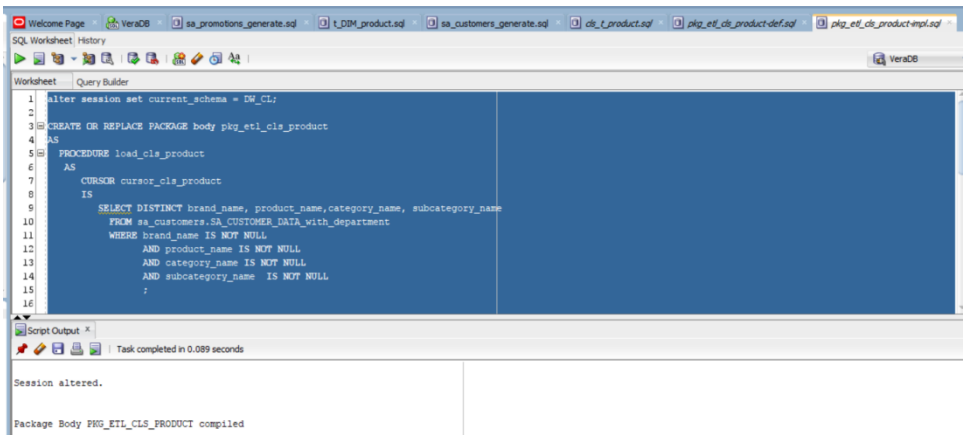


Let's create packages to get data from Storage level SA_* in DW - Cleanup level for the table cls_t_product.



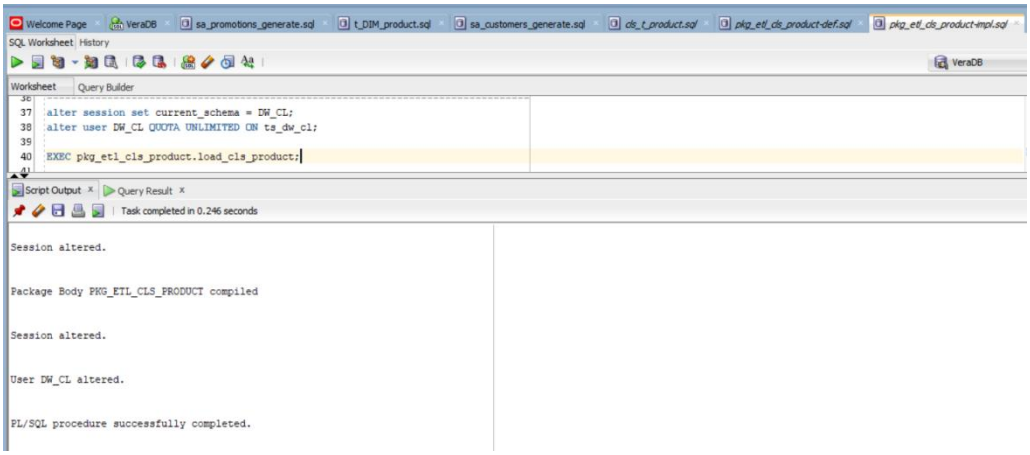
Grant permissions to user DW_CL in tablespace ts_dw_cl to use data from table SA_CUSTOMER_DATA_with_department in tablespace ts_sa_customers_data_01.





Code:

```
CREATE OR REPLACE PACKAGE body pkg_etl_cls_product
AS
 PROCEDURE load_cls_product
  AS
    CURSOR cursor_cls_product
    IS
      SELECT DISTINCT brand_name, product_name,category_name, subcategory_name
       FROM sa_customers.SA_CUSTOMER_DATA_with_department
       WHERE brand_name IS NOT NULL
            AND product_name IS NOT NULL
            AND category_name IS NOT NULL
            AND subcategory_name  IS NOT NULL
            ;

  BEGIN
    EXECUTE IMMEDIATE 'TRUNCATE TABLE DW_CL.cls_t_product';
    FOR i IN cursor_cls_product LOOP
      INSERT INTO DW_CL.cls_t_product(
           brand_name,
           product_name,
           category_name,
           subcategory_name)
         VALUES (
           i.brand_name,
           i.product_name,
           i.category_name,
           i.subcategory_name);
      EXIT WHEN cursor_cls_product%NOTFOUND;
    END LOOP;

    COMMIT;
  END load_cls_product;
END pkg_etl_cls_product;
```




```
42  SELECT * FROM cls_t_product;
43  ----------------------------------------------------------------------------
44
```

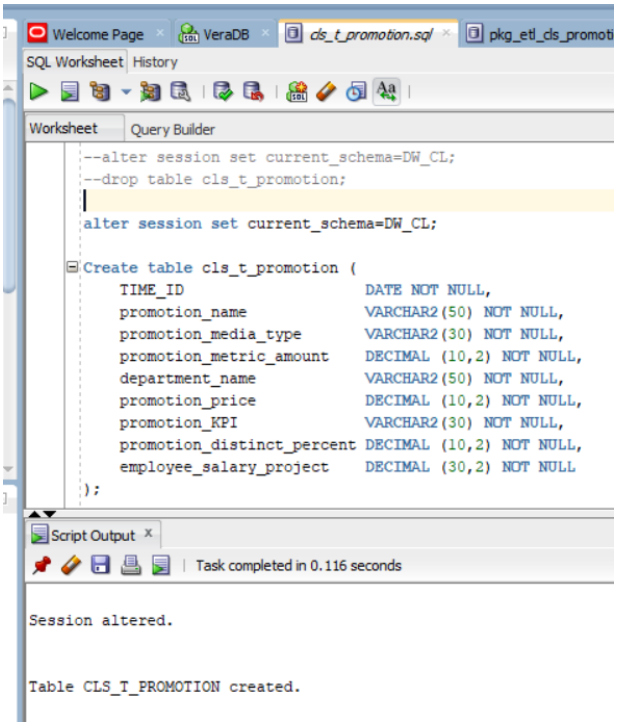
Script Output | Query Result

SQL | All Rows Fetched: 16 in 0.011 seconds

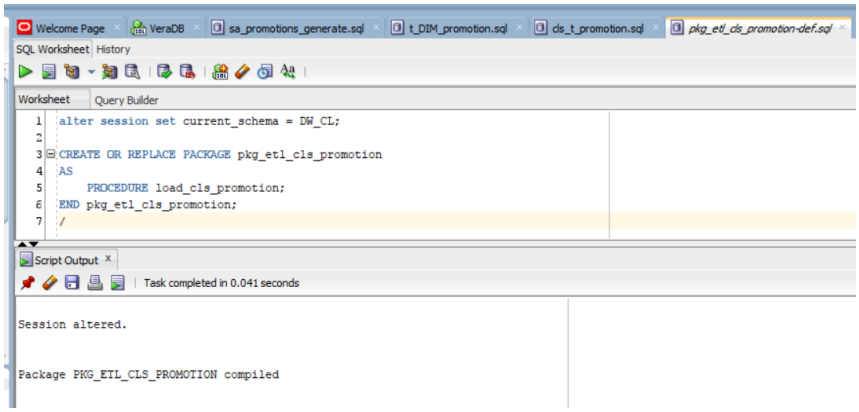
	BRAND_NAME	PRODUCT_NAME	CATEGORY_NAME	SUBCATEGORY_NAME
1	SAMSUNG	Refrigerators	Appliances	Fridge
2	SAMSUNG	Ovens	Appliances	Conventional ovens
3	SAMSUNG	Cooking ovens	Appliances	Electric cooktop
4	SAMSUNG	Washing machines	Appliances	Fully Automatic washing machine
5	SAMSUNG	Dishwashers	Appliances	Built-in Dishwasher
6	SAMSUNG	Microwave ovens	Appliances	Convection Microwave
7	SAMSUNG	Vacuum Cleaners	Appliances	Robotic vacuum cleaner
8	SAMSUNG	Smart watches	Computerized wristwatch	Smart watch for kids?
9	SAMSUNG	TVs	Appliances	QLED
10	SAMSUNG	Smartphones	Phones	Android Phones
11	SAMSUNG	Headphones	Accessories	Earbuds
12	SAMSUNG	Tablets	Tablet computer	Galaxy Tab series?
13	VISA	Sponsorship	Services	Networking sponsor
14	VISA	Card	Services	Contactless smart card

- cls_t_promotion

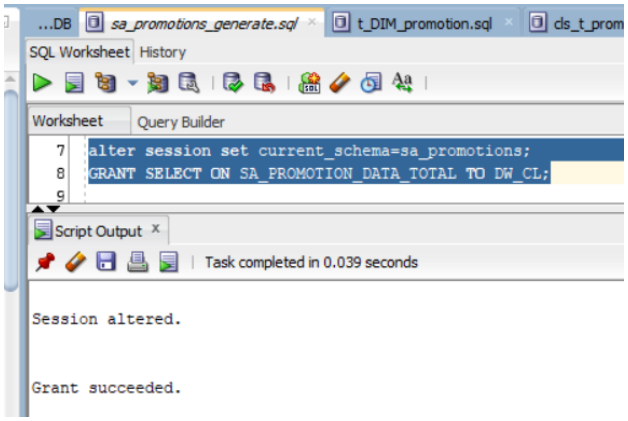
Create a cls_t_promotion table at the DW - Cleansing Level.

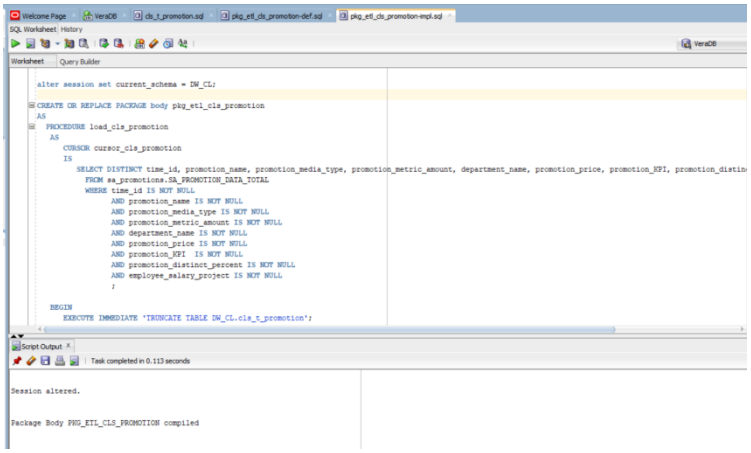


Let's create packages to get data from Storage level SA_* in DW - Cleanup level for the table cls_t_promotion.



Grant permissions to user DW_CL in tablespace ts_dw_cl to use data from table SA_PROMOTION_DATA_TOTAL in tablespace ts_sa_promotions_data_01.





Code:

```
CREATE OR REPLACE PACKAGE body pkg_etl_cls_promotion
AS
 PROCEDURE load_cls_promotion
  AS
    CURSOR cursor_cls_promotion
    IS
      SELECT DISTINCT time_id, promotion_name, promotion_media_type, promotion_metric_amount, department_name, promotion_price,
promotion_KPI, promotion_distinct_percent, employee_salary_project
       FROM sa_promotions.SA_PROMOTION_DATA_TOTAL
       WHERE time_id IS NOT NULL
           AND promotion_name IS NOT NULL
           AND promotion_media_type IS NOT NULL
           AND promotion_metric_amount IS NOT NULL
           AND department_name IS NOT NULL
           AND promotion_price IS NOT NULL
           AND promotion_KPI  IS NOT NULL
           AND promotion_distinct_percent IS NOT NULL
           AND employee_salary_project IS NOT NULL
           ;

  BEGIN
    EXECUTE IMMEDIATE 'TRUNCATE TABLE DW_CL.cls_t_promotion';
    FOR i IN cursor_cls_promotion LOOP
      INSERT INTO DW_CL.cls_t_promotion(
           time_id,
           promotion_name,
           promotion_media_type,
           promotion_metric_amount,
           department_name,
           promotion_price,
           promotion_KPI,
           promotion_distinct_percent,
           employee_salary_project)
         VALUES (
           i.time_id,
           i.promotion_name,
           i.promotion_media_type,
           i.promotion_metric_amount,
           i.department_name,
           i.promotion_price,
           i.promotion_KPI,
           i.promotion_distinct_percent,
           i.employee_salary_project);
      EXIT WHEN cursor_cls_promotion%NOTFOUND;
    END LOOP;
    COMMIT;
  END load_cls_promotion;
END pkg_etl_cls_promotion;
```

```
49  alter session set current_schema = DW_CL;
50  alter user DW_CL QUOTA UNLIMITED ON ts_dw_cl;
51
52  EXEC pkg_etl_cls_promotion.load_cls_promotion;
```

Script Output | Task completed in 15.127 seconds

```
Session altered.

Package Body PKG_ETL_CLS_PROMOTION compiled

Session altered.

User DW_CL altered.

PL/SQL procedure successfully completed.
```

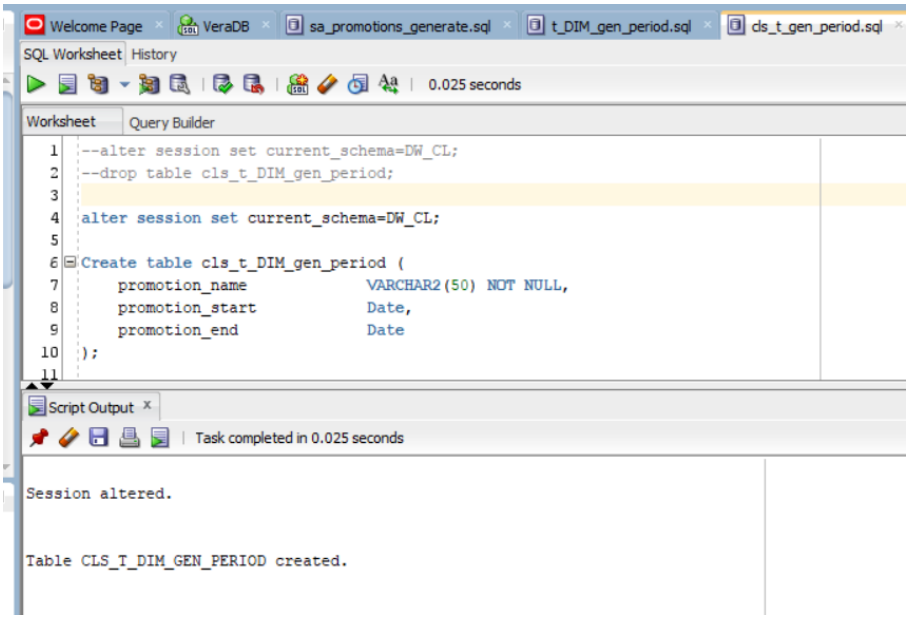
```
SELECT * FROM cls_t_promotion;
```

Script Output | Query Result | SQL | Fetched 50 rows in 0.016 seconds

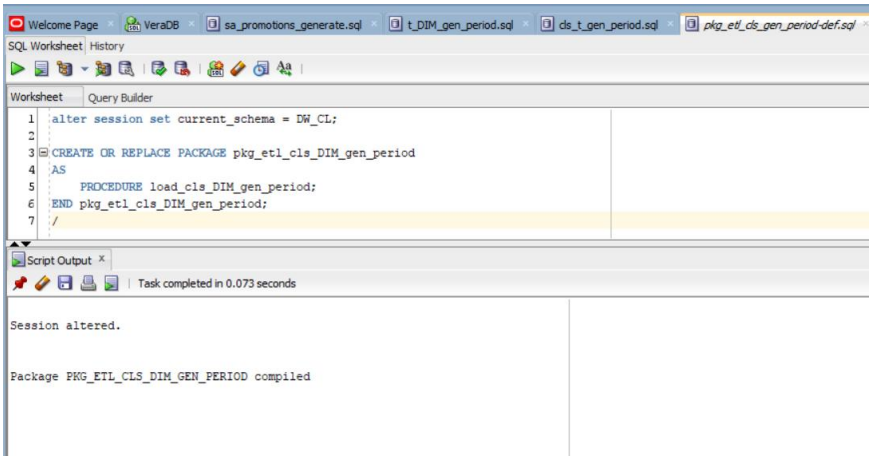
	TIME_ID	PROMOTION_NAME	PROMOTION_MEDIA_TYPE	PROMOTION_METRIC_AMOUNT	DEPARTMENT_NAME	PROMOTION_PRICE	PROMOTION_KPI	PROMOTION_DIS
1	01-JAN-22	5 SAMSUNG_Refrigerators_TV	TV	82	Media Planning and Buying	683	CPL	
2	01-JAN-22	26 SAMSUNG_Ovens_TV	TV	99	Media Planning and Buying	198	CPL	
3	01-JAN-22	49 SAMSUNG_Washing machines_TV	TV	40	Media Planning and Buying	164	CPA	
4	01-JAN-22	62 SAMSUNG_Dishwashers_TV	TV	37	Media Planning and Buying	631	CPA	
5	01-JAN-22	83 SAMSUNG_Microwave ovens_TV	TV	46	Media Planning and Buying	643	CPA	
6	01-JAN-22	105 SAMSUNG_Vacuum Cleaners_TV	TV	52	Media Planning and Buying	150	CPA	
7	01-JAN-22	128 SAMSUNG_TVs_TV	TV	44	Media Planning and Buying	624	CPA	
8	01-JAN-22	148 SAMSUNG_Smartphones_TV	TV	96	Media Planning and Buying	870	CPA	
9	01-JAN-22	166 SAMSUNG_Tablets_TV	TV	66	Media Planning and Buying	346	CPL	
10	01-JAN-22	174 SAMSUNG_Tablets_TV	TV	71	Media Planning and Buying	178	CPL	
11	01-JAN-22	175 SAMSUNG_Tablets_TV	TV	8	Media Planning and Buying	859	CPA	
12	01-JAN-22	188 SAMSUNG_Refrigerators_press	press	36	Media Planning and Buying	626	CPL	

- cls_t_DIM_gen_period

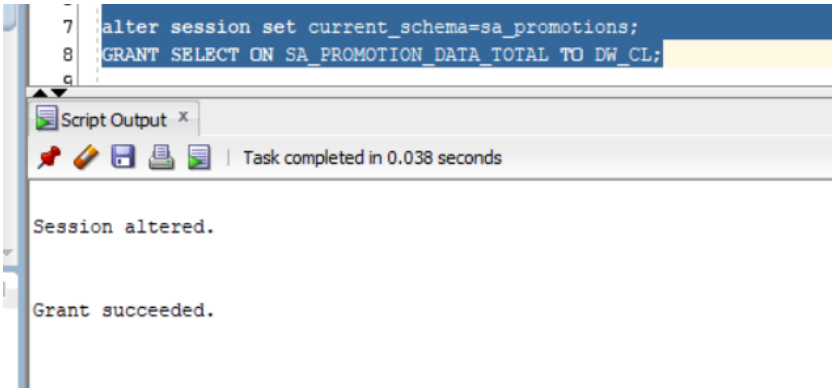
Create a cls_t_DIM_gen_period table at the DW - Cleansing Level.

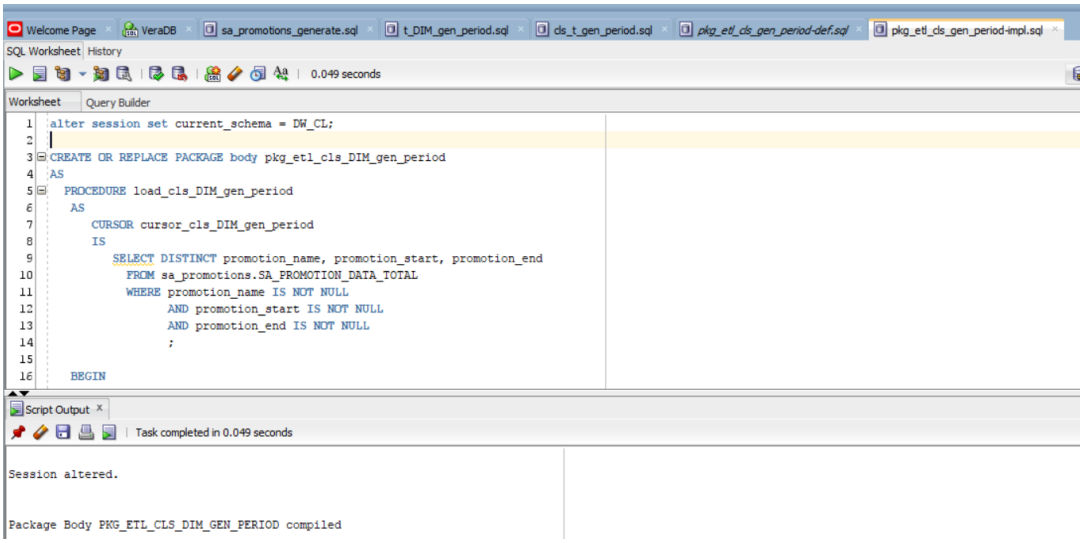


Let's create packages to get data from Storage level SA_* in DW - Cleanup level for the table cls_DIM_gen_period.



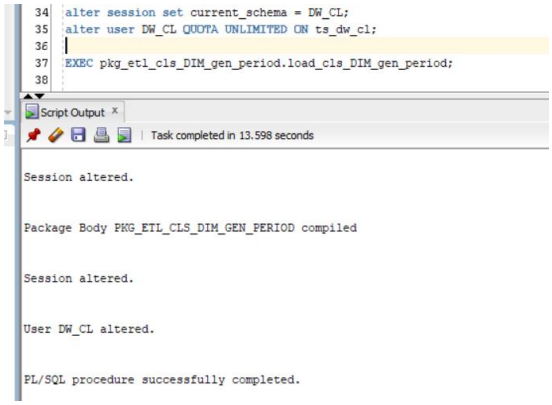
Grant permissions to user DW_CL in tablespace ts_dw_cl to use data from table SA_PROMOTION_DATA_TOTAL in tablespace ts_sa_promotions_data_01.

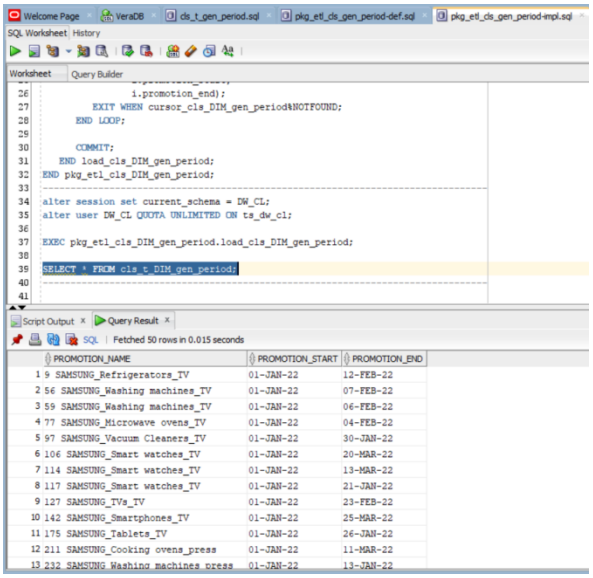




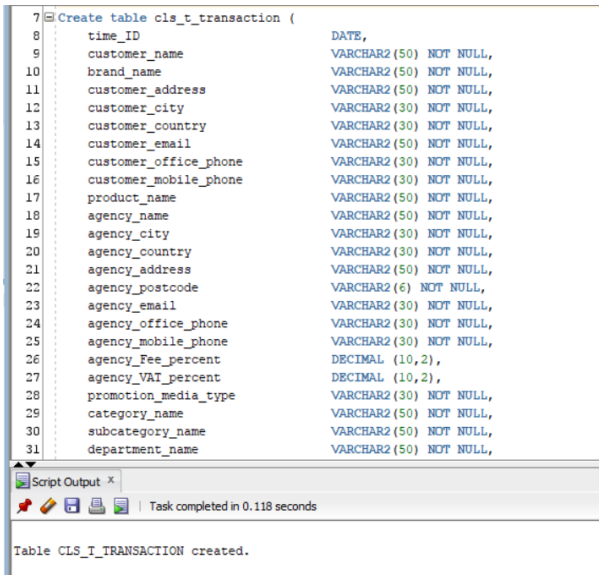
Code:

```
CREATE OR REPLACE PACKAGE body pkg_etl_cls_DIM_gen_period
AS
 PROCEDURE load_cls_DIM_gen_period
  AS
    CURSOR cursor_cls_DIM_gen_period
    IS
      SELECT DISTINCT promotion_name, promotion_start, promotion_end
       FROM sa_promotions.SA_PROMOTION_DATA_TOTAL
       WHERE promotion_name IS NOT NULL
           AND promotion_start IS NOT NULL
           AND promotion_end IS NOT NULL
           ;

  BEGIN
    EXECUTE IMMEDIATE 'TRUNCATE TABLE DW_CL.cls_t_DIM_gen_period';
    FOR i IN cursor_cls_DIM_gen_period LOOP
      INSERT INTO DW_CL.cls_t_DIM_gen_period(
           promotion_name,
           promotion_start,
           promotion_end)
         VALUES (
           i.promotion_name,
           i.promotion_start,
           i.promotion_end);
      EXIT WHEN cursor_cls_DIM_gen_period%NOTFOUND;
    END LOOP;

    COMMIT;
  END load_cls_DIM_gen_period;
END pkg_etl_cls_DIM_gen_period;
```

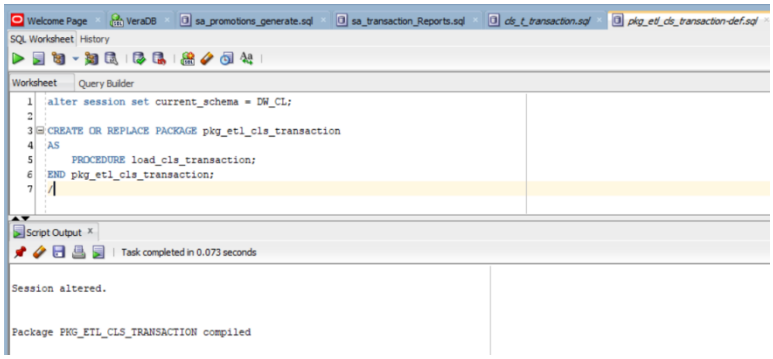




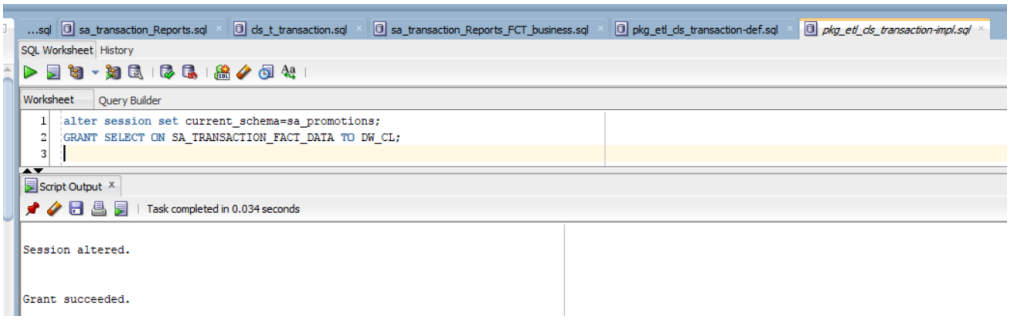
- cls_t_transaction

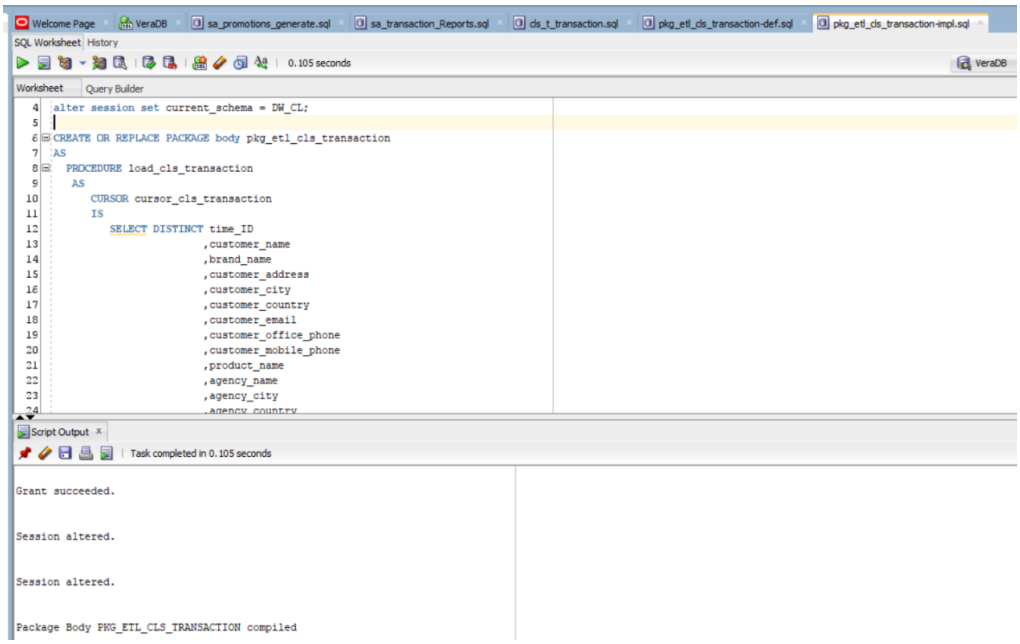


Let's create packages to get data from Storage level SA_* in DW - Cleanup level for the table cls_t_transaction.



Grant permissions to user DW_CL in tablespace ts_dw_cl to use data from table SA_ SA_TRANSACTION in tablespace ts_sa_promotions_data_01.





Code:

```
CREATE OR REPLACE PACKAGE body pkg_etl_cls_transaction
AS
 PROCEDURE load_cls_transaction
  AS
    CURSOR cursor_cls_transaction
    IS
      SELECT DISTINCT time_ID
                ,customer_name
                ,brand_name
                ,customer_address
                ,customer_city
                ,customer_country
                ,customer_email
                ,customer_office_phone
                ,customer_mobile_phone
                ,product_name
                ,agency_name
                ,agency_city
                ,agency_country
                ,agency_address
                ,agency_postcode
                ,agency_email
                ,agency_office_phone
                ,agency_mobile_phone
                ,agency_Fee_percent
                ,agency_VAT_percent
                ,promotion_media_type
                ,category_name
                ,subcategory_name
```

```
        ,department_name
        ,employee_first_name
        ,employee_last_name
        ,employee_position
        ,employee_salary_project
        ,employee_passport_ID
        ,employee_email
        ,employee_office_phone
        ,employee_mobile_phone
        ,employee_date_of_hire
        ,employee_date_end_of_contract
        ,promotion_metric_amount
        ,promotion_price
        ,promotion_KPI
        ,promotion_distinct_percent
        ,promotion_name
        ,promotion_start
        ,promotion_end
        ,gross_revenue_dollar_amount
        ,net_revenue_dollar_amount
        ,gross_cost_dollar_amount
        ,net_cost_dollar_amount
        ,gross_profit_dollar_amount
        ,net_profit_dollar_amount
        ,net_salary_employee_dollar_amount
        ,gross_salary_employee_dollar_amount
        ,gross_profit_margin_percent
FROM sa_promotions.SA_TRANSACTION_FACT_DATA
WHERE time_ID IS NOT NULL
        AND customer_name IS NOT NULL
        AND brand_name IS NOT NULL
        AND customer_address IS NOT NULL
        AND customer_city IS NOT NULL
        AND customer_country IS NOT NULL
        AND customer_email IS NOT NULL
        AND customer_office_phone IS NOT NULL
        AND customer_mobile_phone IS NOT NULL
        AND product_name IS NOT NULL
        AND agency_name IS NOT NULL
        AND agency_city IS NOT NULL
        AND agency_country IS NOT NULL
        AND agency_address IS NOT NULL
        AND agency_postcode IS NOT NULL
        AND agency_email IS NOT NULL
        AND agency_office_phone IS NOT NULL
        AND agency_mobile_phone IS NOT NULL
        AND agency_Fee_percent IS NOT NULL
        AND agency_VAT_percent IS NOT NULL
        AND promotion_media_type IS NOT NULL
        AND category_name IS NOT NULL
        AND subcategory_name IS NOT NULL
        AND department_name IS NOT NULL
        AND employee_first_name IS NOT NULL
        AND employee_last_name IS NOT NULL
        AND employee_position IS NOT NULL
        AND employee_salary_project IS NOT NULL
        AND employee_passport_ID IS NOT NULL
        AND employee_email IS NOT NULL
        AND employee_office_phone  IS NOT NULL
        AND employee_mobile_phone  IS NOT NULL
        AND employee_date_of_hire  IS NOT NULL
        AND employee_date_end_of_contract IS NOT NULL
        AND promotion_metric_amount  IS NOT NULL
        AND promotion_price  IS NOT NULL
```

```
                AND promotion_KPI  IS NOT NULL
                AND promotion_distinct_percent IS NOT NULL
                AND promotion_name IS NOT NULL
                AND promotion_start IS NOT NULL
                AND promotion_end IS NOT NULL
                AND gross_revenue_dollar_amount IS NOT NULL
                AND net_revenue_dollar_amount  IS NOT NULL
                AND gross_cost_dollar_amount  IS NOT NULL
                AND net_cost_dollar_amount  IS NOT NULL
                AND gross_profit_dollar_amount IS NOT NULL
                AND net_profit_dollar_amount  IS NOT NULL
                AND net_salary_employee_dollar_amount IS NOT NULL
                AND gross_salary_employee_dollar_amount IS NOT NULL
                AND gross_profit_margin_percent IS NOT NULL
        ;
BEGIN
  EXECUTE IMMEDIATE 'TRUNCATE TABLE DW_CL.cls_t_transaction';
  FOR i IN cursor_cls_transaction LOOP
    INSERT INTO DW_CL.cls_t_transaction(
                time_ID
               ,customer_name
               ,brand_name
               ,customer_address
               ,customer_city
               ,customer_country
               ,customer_email
               ,customer_office_phone
               ,customer_mobile_phone
               ,product_name
               ,agency_name
               ,agency_city
               ,agency_country
               ,agency_address
               ,agency_postcode
               ,agency_email
               ,agency_office_phone
               ,agency_mobile_phone
               ,agency_Fee_percent
               ,agency_VAT_percent
               ,promotion_media_type
               ,category_name
               ,subcategory_name
               ,department_name
               ,employee_first_name
               ,employee_last_name
               ,employee_position
               ,employee_salary_project
               ,employee_passport_ID
               ,employee_email
               ,employee_office_phone
               ,employee_mobile_phone
               ,employee_date_of_hire
               ,employee_date_end_of_contract
               ,promotion_metric_amount
               ,promotion_price
               ,promotion_KPI
               ,promotion_distinct_percent
               ,promotion_name
               ,promotion_start
               ,promotion_end
               ,gross_revenue_dollar_amount
               ,net_revenue_dollar_amount
               ,gross_cost_dollar_amount
               ,net_cost_dollar_amount
```

```
                ,gross_profit_dollar_amount
                ,net_profit_dollar_amount
                ,net_salary_employee_dollar_amount
                ,gross_salary_employee_dollar_amount
                ,gross_profit_margin_percent
                )
        VALUES (
                 i.time_ID
                , i.customer_name
                , i.brand_name
                , i.customer_address
                , i.customer_city
                , i.customer_country
                , i.customer_email
                , i.customer_office_phone
                , i.customer_mobile_phone
                , i.product_name
                , i.agency_name
                , i.agency_city
                , i.agency_country
                , i.agency_address
                , i.agency_postcode
                , i.agency_email
                , i.agency_office_phone
                , i.agency_mobile_phone
                , i.agency_Fee_percent
                , i.agency_VAT_percent
                , i.promotion_media_type
                , i.category_name
                , i.subcategory_name
                , i.department_name
                , i.employee_first_name
                , i.employee_last_name
                , i.employee_position
                , i.employee_salary_project
                , i.employee_passport_ID
                , i.employee_email
                , i.employee_office_phone
                , i.employee_mobile_phone
                , i.employee_date_of_hire
                , i.employee_date_end_of_contract
                , i.promotion_metric_amount
                , i.promotion_price
                , i.promotion_KPI
                , i.promotion_distinct_percent
                , i.promotion_name
                , i.promotion_start
                , i.promotion_end
                , i.gross_revenue_dollar_amount
                , i.net_revenue_dollar_amount
                , i.gross_cost_dollar_amount
                , i.net_cost_dollar_amount
                , i.gross_profit_dollar_amount
                , i.net_profit_dollar_amount
                , i.net_salary_employee_dollar_amount
                , i.gross_salary_employee_dollar_amount
                , i.gross_profit_margin_percent
                );
      EXIT WHEN cursor_cls_transaction%NOTFOUND;
    END LOOP;
    COMMIT;
  END load_cls_transaction;
END pkg_etl_cls_transaction;
```

```
190
191 alter session set current_schema = DW_CL;
192 alter user DW_CL QUOTA UNLIMITED ON ts_dw_cl;
193
194 EXEC pkg_etl_cls_transaction.load_cls_transaction;
195
```

Script Output | Task completed in 19.673 seconds

```
Package Body PKG_ETL_CLS_TRANSACTION compiled


Session altered.


User DW_CL altered.


PL/SQL procedure successfully completed.
```

```
SELECT * FROM cls_t_transaction
order by 1;
```

Script Output | Query Result | SQL | Fetched 50 rows in 0.647 seconds

	TIME_ID	CUSTOMER_NAME	BRAND_NAME	CUSTOMER_ADDRESS	CUSTOMER_CITY	CUSTOMER_COUNTRY	CUSTOMER_EMAIL
1	01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
2	01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
3	01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
4	01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
5	01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
6	01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
7	01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
8	01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
9	01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
10	01-JAN-22	Visa International Service Association	VISA	900 Metro Center Blvd.	Foster City	United States	https://usa.visa.com/contact-us.html
11	01-JAN-22	Visa International Service Association	VISA	900 Metro Center Blvd.	Foster City	United States	https://usa.visa.com/contact-us.html
12	01-JAN-22	Visa International Service Association	VISA	900 Metro Center Blvd.	Foster City	United States	https://usa.visa.com/contact-us.html
13	01-JAN-22	Visa International Service Association	VISA	900 Metro Center Blvd.	Foster City	United States	https://usa.visa.com/contact-us.html
14	01-JAN-22	Visa International Service Association	VISA	900 Metro Center Blvd.	Foster City	United States	https://usa.visa.com/contact-us.html

We can run all the procedures at the same time.

Welcome Page | VeraDB | reload_business_cls.sql

SQL Worksheet | History

Worksheet | Query Builder

```
1  alter session set current_schema = DW_CL;
2  alter user DW_CL QUOTA UNLIMITED ON ts_dw_cl;
3
4  BEGIN
5      pkg_etl_cls_customer.load_cls_customer;
6      pkg_etl_cls_employee.load_cls_employee;
7      pkg_etl_cls_agency.load_cls_agency;
8      pkg_etl_cls_DIM_gen_period.load_cls_DIM_gen_per
9      pkg_etl_cls_product.load_cls_product;
10     pkg_etl_cls_promotion.load_cls_promotion;
11     pkg_etl_cls_transaction.load_cls_transaction;
12 END;
13
```

Script Output | Task completed in 49.815 seconds

```
Session altered.


User DW_CL altered.


PL/SQL procedure successfully completed.
```

```
14 EXEC pkg_etl_cls_customer.load_cls_customer;
15 EXEC pkg_etl_cls_employee.load_cls_employee;
16 EXEC pkg_etl_cls_agency.load_cls_agency;
17 EXEC pkg_etl_cls_DIM_gen_period.load_cls_DIM_gen_period;
18 EXEC pkg_etl_cls_product.load_cls_product;
19 EXEC pkg_etl_cls_promotion.load_cls_promotion;
20 EXEC pkg_etl_cls_transaction.load_cls_transaction;
```

Script Output | Task completed in 49.708 seconds

```
PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.
```

```
SELECT * FROM DW_CL.cls_t_customer;
SELECT * FROM DW_CL.cls_t_employee;
SELECT * FROM DW_CL.cls_t_agency;
SELECT * FROM DW_CL.cls_t_DIM_gen_period;
SELECT * FROM DW_CL.cls_t_product;
SELECT * FROM DW_CL.cls_t_promotion;
SELECT * FROM cls_t_transaction;
```

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4 | Query Result 5 | Query Result 6

SQL | Fetched 50 rows in 0.018 seconds

	TIME_ID	CUSTOMER_NAME	BRAND_NAME	CUSTOMER_ADDRESS	CUSTOMER_CITY	CUSTOMER_COUNTRY	CUSTOMER_EMAIL
1	02-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
2	02-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
3	02-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
4	02-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
5	02-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
6	02-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
7	02-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
8	02-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
9	02-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
10	02-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
11	02-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
12	02-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
13	02-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
14	02-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
15	02-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
16	02-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
17	02-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
18	02-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS

Let's create all necessary tables and sequences at the DW layer for the DW_DATA user in the ts_dw_data_01 tablespace.

- DW_DATA.DIM_customer

...sql | pkg_etl_dw_customer-impl.sql | cls_t_customer.sql | pkg_etl_cls_customer-def.sql | pkg_etl_cls_customer-impl.sql | t_DIM_customer.sql

SQL Worksheet | History

Worksheet | Query Builder

```
--drop table DW_DATA.DIM_customer;
--alter session set current_schema = DW_DATA;

CREATE TABLE DW_DATA.DIM_customer(
    customer_ID NUMBER(10),
    customer_name           VARCHAR2(50),
    brand_name              VARCHAR2(50),
    customer_address        VARCHAR2(50),
    customer_city           VARCHAR2(30),
    customer_country        VARCHAR2(30),
    customer_email          VARCHAR2(50),
    customer_office_phone   VARCHAR2(30),
    customer_mobile_phone   VARCHAR2(30),

    CONSTRAINT "PK_T.DIM_customer" PRIMARY KEY(customer_ID)
);
```

Script Output | Task completed in 0.04 seconds

```
Table DW_DATA.DIM_CUSTOMER created.
```

Welcome Page | VeraDB | sq_t_DIM_customer.sql

SQL Worksheet | History

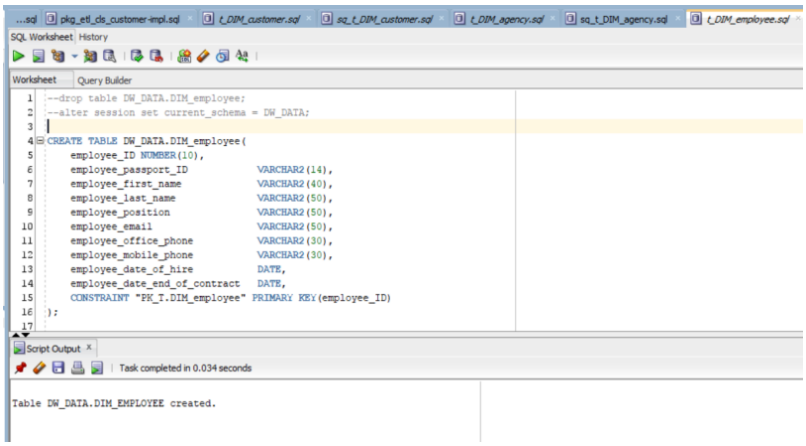
Worksheet | Query Builder

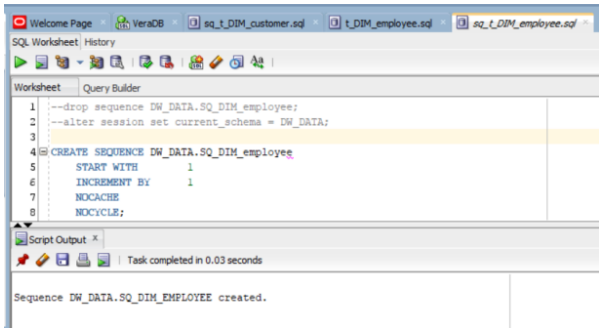
```
--drop sequence DW_DATA.SQ_DIM_customer;
--alter session set current_schema = DW_DATA;

CREATE SEQUENCE DW_DATA.SQ_DIM_customer
    START WITH      1
    INCREMENT BY    1
    NOCACHE
    NOCYCLE;
```

Script Output | Task completed in 0.032 seconds

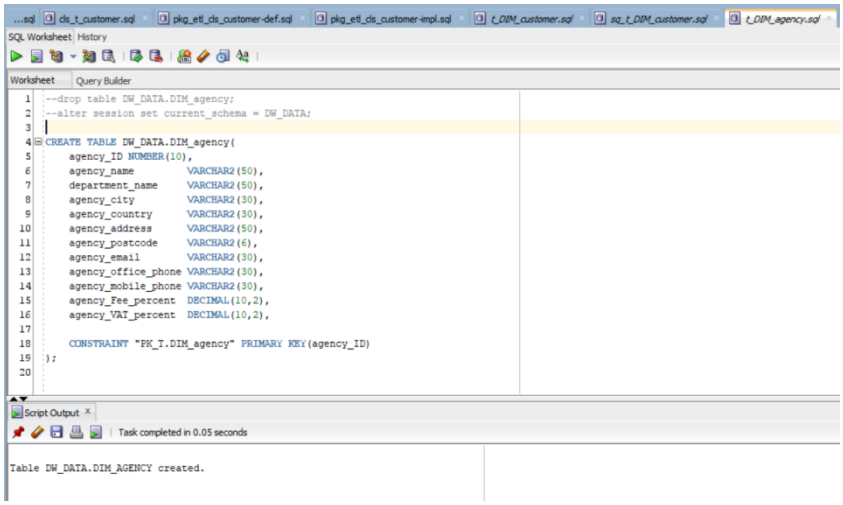
```
Sequence DW_DATA.SQ_DIM_CUSTOMER created.
```

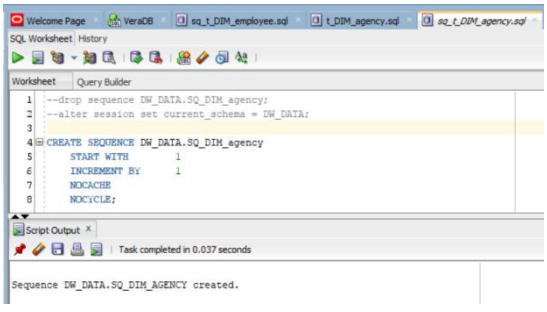
- DW_DATA.DIM_employee



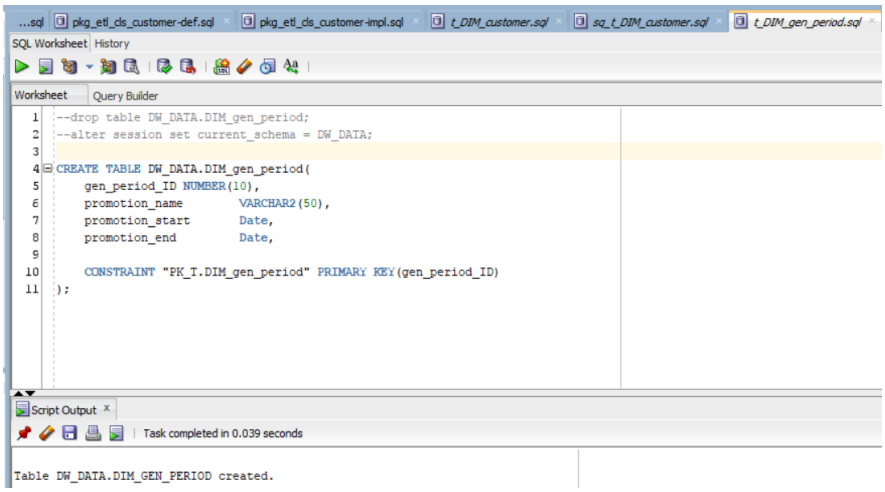


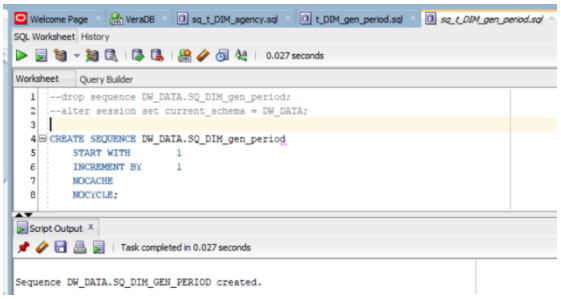
- DW_DATA.DIM_agency



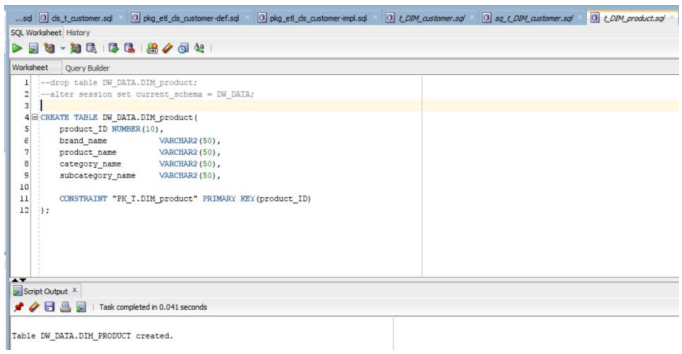


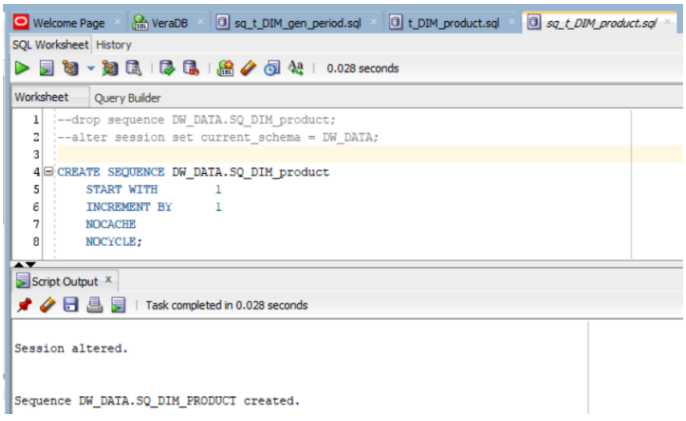
- DW_DATA.DIM_gen_period



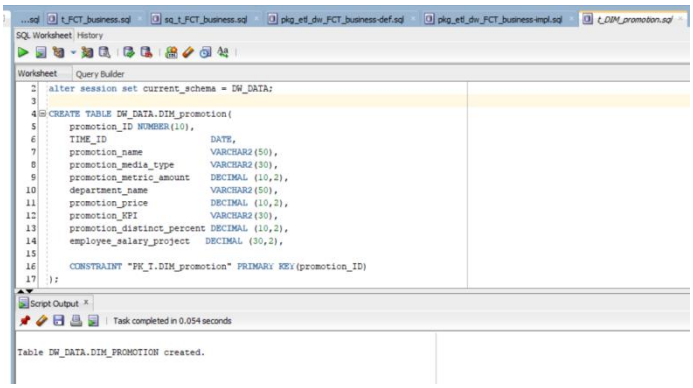


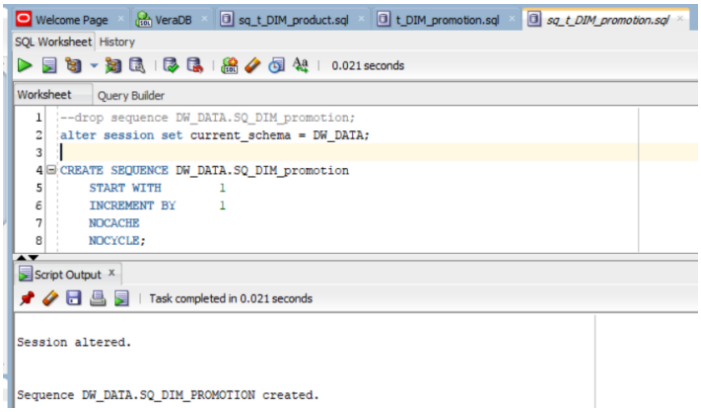
- DW_DATA.DIM_product



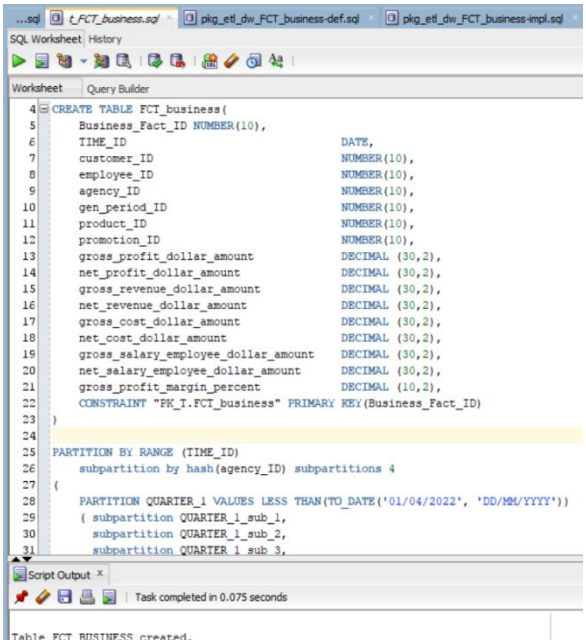


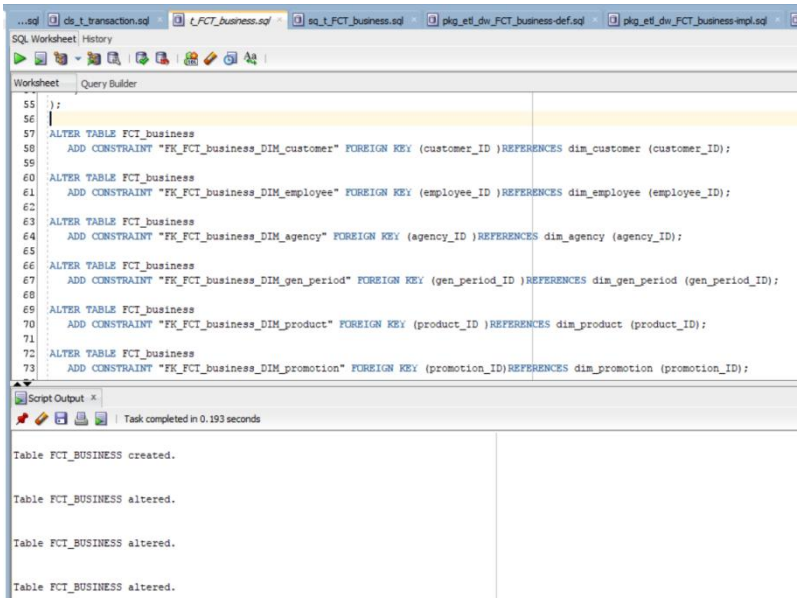
- DW_DATA.DIM_promotion

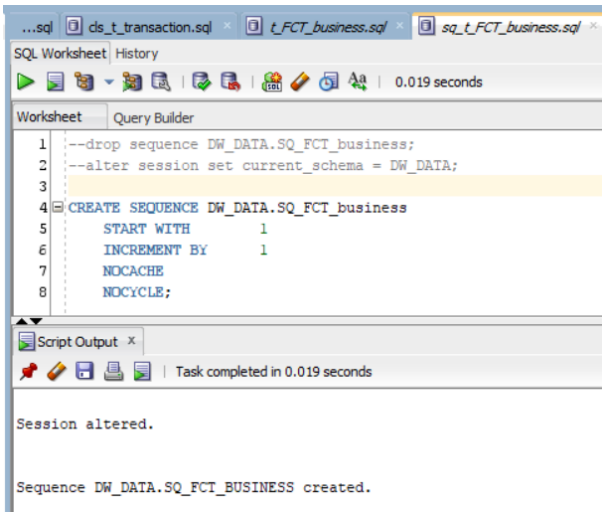




- DW_DATA.FCT_business

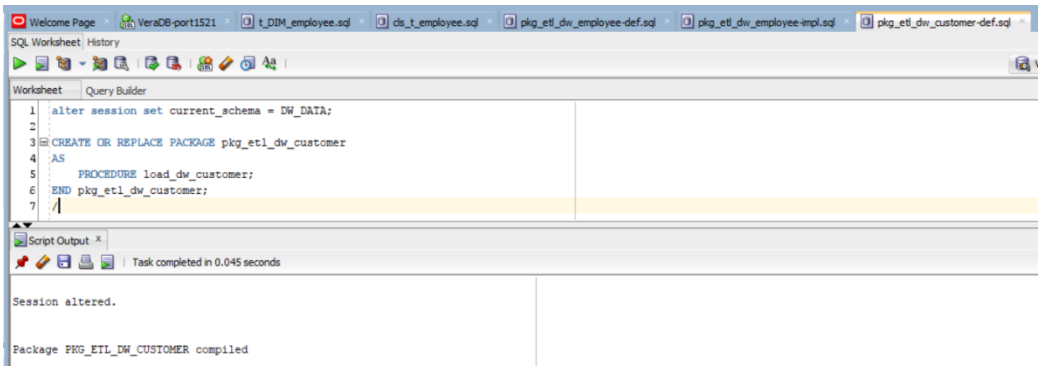






Let's create packages to move all the data from the Cleansing Level to the DW Level, with the natural keys converted to primary keys.

- pkg_etl_dw_customer



Grant permissions to table DW_CL.cls_t_customer from the Cleansing level to user DW_DATA on DW level and create pkg_etl_dw_customer-impl.

```
alter session set current_schema = DW_CL;
GRANT SELECT ON DW_CL.cls_t_customer TO DW_DATA;
```

```
Session altered.

Grant succeeded.
```

```
alter session set current_schema = DW_DATA;

CREATE OR REPLACE PACKAGE body pkg_etl_dw_customer
AS
  PROCEDURE load_dw_customer
    AS
      BEGIN
       DECLARE
            TYPE CURSOR_VARCHAR IS TABLE OF varchar2(50);
            TYPE CURSOR_NUMBER IS TABLE OF number(10);

            TYPE BIG_CURSOR IS REF CURSOR;

            ALL_INF BIG_CURSOR;

            customer_name CURSOR_VARCHAR;
            brand_name CURSOR_VARCHAR;
            customer_address CURSOR_VARCHAR;
            customer_city CURSOR_VARCHAR;
            customer_country CURSOR_VARCHAR;
            customer_email CURSOR_VARCHAR;
            customer_office_phone CURSOR_VARCHAR;
            customer_mobile_phone CURSOR_VARCHAR;
            customer_name_stage CURSOR_VARCHAR;
            customer_ID CURSOR_NUMBER;
```

```
Session altered.

Package Body PKG_ETL_DW_CUSTOMER compiled
```

Let’s select data from the DW_DATA.DIM_customer table on DW layer.

```
alter session set current_schema = DW_DATA;
alter user DW_DATA QUOTA UNLIMITED ON TS_DW_DATA_01;

EXEC pkg_etl_dw_customer.load_dw_customer;
```

```
Session altered.

User DW_DATA altered.

PL/SQL procedure successfully completed.
```

```
SELECT * FROM DW_DATA.DIM_customer;
```

All Rows Fetched: 2 in 0.004 seconds

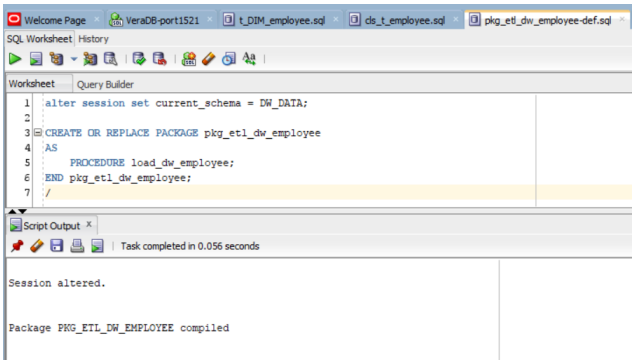
	CUSTOMER_ID	CUSTOMER_NAME	BRAND_NAME	CUSTOMER_ADDRESS	CUSTOMER_CITY	CUSTOMER_COUNTRY	CUSTOMER_EMAIL
1	1	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS
2	2	Visa International Service Association	VISA	900 Metro Center Blvd.	Foster City	United States	https://usa.visa.com/contact-us.

Code:

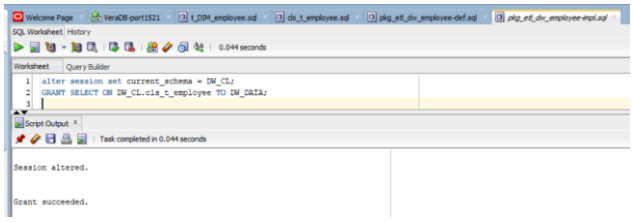
```
alter session set current_schema = DW_CL;
GRANT SELECT ON DW_CL.cls_t_customer TO DW_DATA;
alter session set current_schema = DW_DATA;

CREATE OR REPLACE PACKAGE body pkg_etl_dw_customer
AS
 PROCEDURE load_dw_customer
  AS
   BEGIN
    DECLARE
       TYPE CURSOR_VARCHAR IS TABLE OF varchar2(50);
       TYPE CURSOR_NUMBER IS TABLE OF number(10);
       TYPE BIG_CURSOR IS REF CURSOR;
       ALL_INF BIG_CURSOR;
       customer_name CURSOR_VARCHAR;
       brand_name CURSOR_VARCHAR;
       customer_address CURSOR_VARCHAR;
       customer_city CURSOR_VARCHAR;
       customer_country CURSOR_VARCHAR;
       customer_email CURSOR_VARCHAR;
       customer_office_phone CURSOR_VARCHAR;
       customer_mobile_phone CURSOR_VARCHAR;
       customer_name_stage CURSOR_VARCHAR;
       customer_ID CURSOR_NUMBER;

        BEGIN
            OPEN ALL_INF FOR
              SELECT
                  source_CL.customer_name AS  customer_name_source_CL,
                  source_CL.brand_name AS brand_name_source_CL,
                  source_CL.customer_address AS  customer_address_source_CL,
                  source_CL.customer_city AS  customer_city_source_CL,
                  source_CL.customer_country AS  customer_country_source_CL,
                  source_CL.customer_email AS customer_email_source_CL,
                  source_CL.customer_office_phone AS  customer_office_phone_source_CL,
                  source_CL.customer_mobile_phone AS  customer_mobile_phone_source_CL,
                  stage.customer_name AS customer_name_stage,
                  stage.customer_ID AS  customer_ID
                FROM (SELECT DISTINCT * FROM DW_CL.cls_t_customer) source_CL
                    LEFT JOIN DW_DATA.DIM_customer stage
                      ON (source_CL.customer_name = stage.customer_name);
            FETCH ALL_INF
            BULK COLLECT INTO
                    customer_name,
                    brand_name,
                    customer_address,
                    customer_city,
                    customer_country,
                    customer_email,
                    customer_office_phone,
                    customer_mobile_phone,
                    customer_name_stage,
                    customer_ID;
            CLOSE ALL_INF;
FOR i IN customer_ID.FIRST .. customer_ID.LAST LOOP
     IF ( customer_ID ( i ) IS NULL ) THEN
       INSERT INTO DW_DATA.DIM_customer (
                            customer_ID,
                            customer_name,
                            brand_name,
                            customer_address,
                            customer_city,
                            customer_country,
```

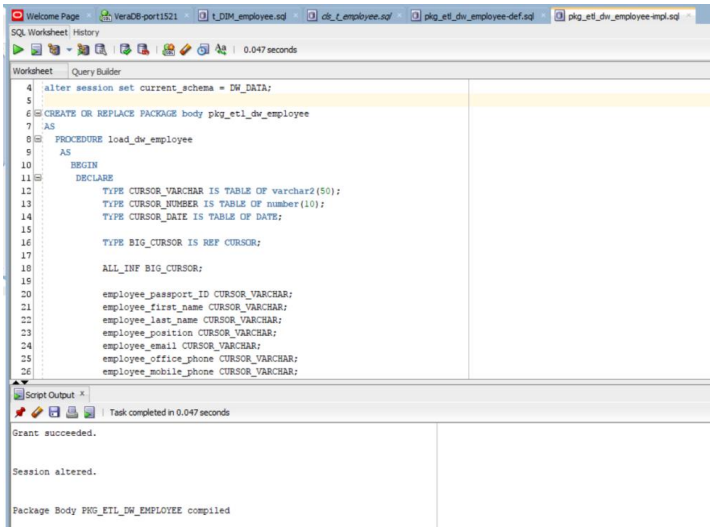
```
                    customer_email,
                    customer_office_phone,
                    customer_mobile_phone)
        VALUES ( SQ_DIM_customer.NEXTVAL,
              customer_name(i),
              brand_name(i),
              customer_address(i),
              customer_city(i),
              customer_country(i),
              customer_email(i),
              customer_office_phone(i),
              customer_mobile_phone(i)
              );
     COMMIT;

     ELSE UPDATE DW_DATA.DIM_customer
     SET
       customer_name = customer_name(i),
       brand_name = brand_name(i),
       customer_address = customer_address(i),
       customer_city = customer_city(i),
       customer_country = customer_country(i),
       customer_email = customer_email(i),
       customer_office_phone = customer_office_phone(i),
       customer_mobile_phone = customer_mobile_phone(i)
     WHERE customer_ID = customer_ID(i);
                COMMIT;
   END IF;
  END LOOP;

            END;
  END load_dw_customer;
END pkg_etl_dw_customer;
```

- pkg_etl_dw_employee



Grant permissions to table DW_CL.cls_t_employee from the Cleansing level to user DW_DATA on DW level and create pkg_etl_dw_employee-impl.



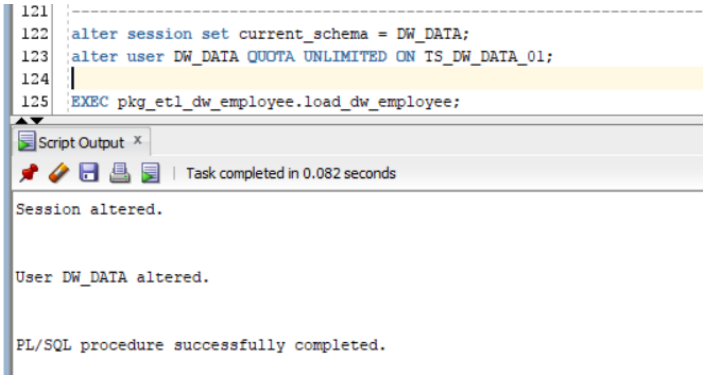


```
alter session set current_schema = DW_DATA;

CREATE OR REPLACE PACKAGE body pkg_etl_dw_employee
AS
  PROCEDURE load_dw_employee
   AS
     BEGIN
      DECLARE
           TYPE CURSOR_VARCHAR IS TABLE OF varchar2(50);
           TYPE CURSOR_NUMBER IS TABLE OF number(10);
           TYPE CURSOR_DATE IS TABLE OF DATE;

           TYPE BIG_CURSOR IS REF CURSOR;

           ALL_INF BIG_CURSOR;

           employee_passport_ID CURSOR_VARCHAR;
           employee_first_name CURSOR_VARCHAR;
           employee_last_name CURSOR_VARCHAR;
           employee_position CURSOR_VARCHAR;
           employee_email CURSOR_VARCHAR;
           employee_office_phone CURSOR_VARCHAR;
           employee_mobile_phone CURSOR_VARCHAR;
```

Script Output

Task completed in 0.047 seconds

```
Grant succeeded.

Session altered.

Package Body PKG_ETL_DW_EMPLOYEE compiled
```

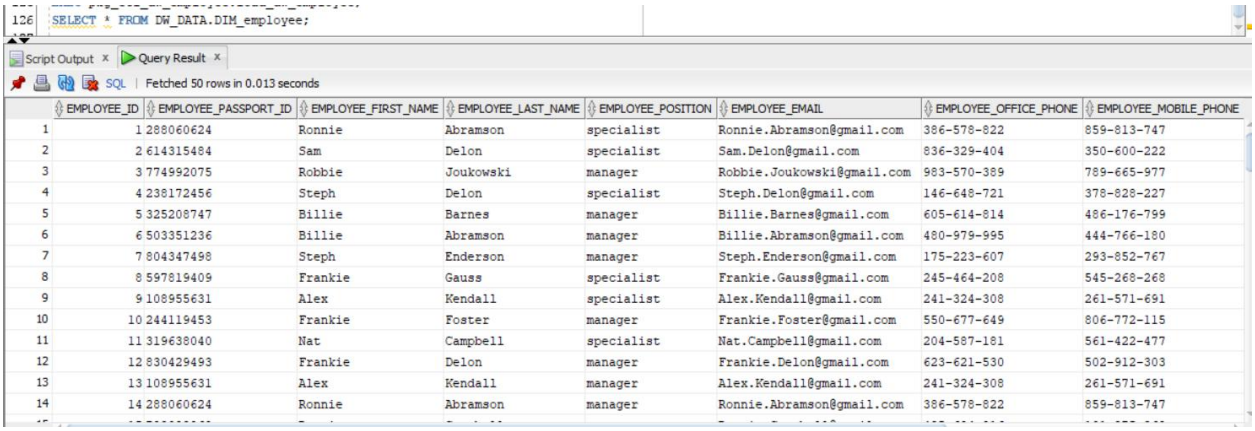


```
alter session set current_schema = DW_DATA;
alter user DW_DATA QUOTA UNLIMITED ON TS_DW_DATA_01;

EXEC pkg_etl_dw_employee.load_dw_employee;
```

Script Output

Task completed in 0.082 seconds

```
Session altered.


User DW_DATA altered.


PL/SQL procedure successfully completed.
```



```
SELECT * FROM DW_DATA.DIM_employee;
```

Script Output | Query Result

SQL | Fetched 50 rows in 0.013 seconds

	EMPLOYEE_ID	EMPLOYEE_PASSPORT_ID	EMPLOYEE_FIRST_NAME	EMPLOYEE_LAST_NAME	EMPLOYEE_POSITION	EMPLOYEE_EMAIL	EMPLOYEE_OFFICE_PHONE	EMPLOYEE_MOBILE_PHONE
1	1	288060624	Ronnie	Abramson	specialist	Ronnie.Abramson@gmail.com	386-578-822	859-813-747
2	2	614315484	Sam	Delon	specialist	Sam.Delon@gmail.com	836-329-404	350-600-222
3	3	774992075	Robbie	Joukowski	manager	Robbie.Joukowski@gmail.com	983-570-389	789-665-977
4	4	238172456	Steph	Delon	specialist	Steph.Delon@gmail.com	146-648-721	378-828-227
5	5	325208747	Billie	Barnes	manager	Billie.Barnes@gmail.com	605-614-814	486-176-799
6	6	503351236	Billie	Abramson	manager	Billie.Abramson@gmail.com	480-979-995	444-766-180
7	7	804347498	Steph	Enderson	manager	Steph.Enderson@gmail.com	175-223-607	293-852-767
8	8	597819409	Frankie	Gauss	specialist	Frankie.Gauss@gmail.com	245-464-208	545-268-268
9	9	108955631	Alex	Kendall	specialist	Alex.Kendall@gmail.com	241-324-308	261-571-691
10	10	244119453	Frankie	Foster	manager	Frankie.Foster@gmail.com	550-677-649	806-772-115
11	11	319638040	Nat	Campbell	specialist	Nat.Campbell@gmail.com	204-587-181	561-422-477
12	12	830429493	Frankie	Delon	manager	Frankie.Delon@gmail.com	623-621-530	502-912-303
13	13	108955631	Alex	Kendall	manager	Alex.Kendall@gmail.com	241-324-308	261-571-691
14	14	288060624	Ronnie	Abramson	manager	Ronnie.Abramson@gmail.com	386-578-822	859-813-747

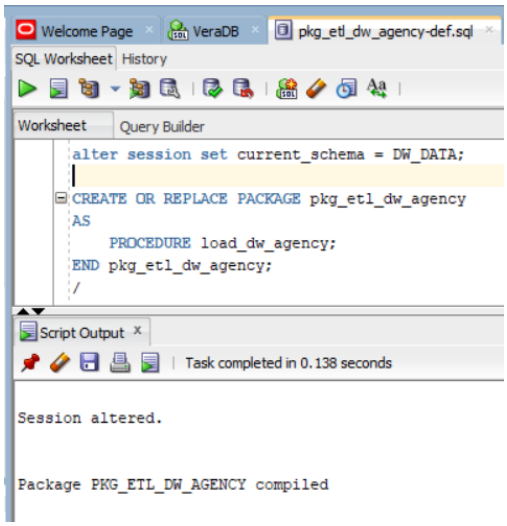
Code:

```
alter session set current_schema = DW_CL;
GRANT SELECT ON DW_CL.cls_t_employee TO DW_DATA;
alter session set current_schema = DW_DATA;

CREATE OR REPLACE PACKAGE body pkg_etl_dw_employee
AS
  PROCEDURE load_dw_employee
   AS
     BEGIN
      DECLARE
          TYPE CURSOR_VARCHAR IS TABLE OF varchar2(50);
          TYPE CURSOR_NUMBER IS TABLE OF number(10);
          TYPE CURSOR_DATE IS TABLE OF DATE;
```

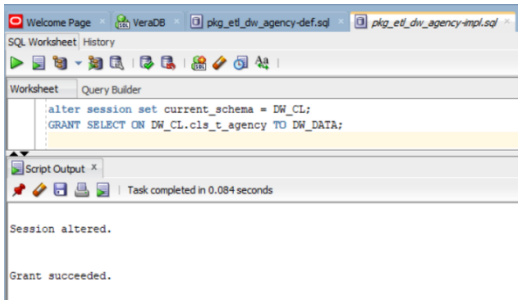
```
        TYPE BIG_CURSOR IS REF CURSOR;
        ALL_INF BIG_CURSOR;
        employee_passport_ID CURSOR_VARCHAR;
        employee_first_name CURSOR_VARCHAR;
        employee_last_name CURSOR_VARCHAR;
        employee_position CURSOR_VARCHAR;
        employee_email CURSOR_VARCHAR;
        employee_office_phone CURSOR_VARCHAR;
        employee_mobile_phone CURSOR_VARCHAR;
        employee_date_of_hire CURSOR_DATE;
        employee_date_end_of_contract CURSOR_DATE;
        employee_passport_ID_stage CURSOR_VARCHAR;
        employee_position_stage CURSOR_VARCHAR;
        employee_ID CURSOR_NUMBER;
         BEGIN
            OPEN ALL_INF FOR
              SELECT
                 source_CL.employee_passport_ID AS  employee_passport_ID_source_CL,
                 source_CL.employee_first_name AS employee_first_name_source_CL,
                 source_CL.employee_last_name AS  employee_last_name_source_CL,
                 source_CL.employee_position AS  employee_position_source_CL,
                 source_CL.employee_email AS  employee_email_source_CL,
                 source_CL.employee_office_phone AS employee_office_phone_source_CL,
                 source_CL.employee_mobile_phone AS  employee_mobile_phone_source_CL,
                 source_CL.employee_date_of_hire AS  employee_date_of_hire_source_CL,
                 source_CL.employee_date_end_of_contract AS  employee_date_end_of_contract_source_CL,
                 stage.employee_passport_ID AS employee_passport_ID_stage,
                 stage.employee_position AS employee_position_stage,
                 stage.employee_ID AS  employee_ID
                FROM (SELECT DISTINCT * FROM DW_CL.cls_t_employee) source_CL
                    LEFT JOIN DW_DATA.DIM_employee stage
                      ON (source_CL.employee_passport_ID = stage.employee_passport_ID and source_CL.employee_position =
stage.employee_position);
            FETCH ALL_INF
            BULK COLLECT INTO
                    employee_passport_ID,
                    employee_first_name,
                    employee_last_name,
                    employee_position,
                    employee_email,
                    employee_office_phone,
                    employee_mobile_phone,
                    employee_date_of_hire,
                    employee_date_end_of_contract,
                    employee_passport_ID_stage,
                    employee_position_stage,
                    employee_ID;
            CLOSE ALL_INF;
FOR i IN employee_ID.FIRST .. employee_ID.LAST LOOP
    IF ( employee_ID ( i ) IS NULL ) THEN
      INSERT INTO DW_DATA.DIM_employee (
                            employee_ID,
                            employee_passport_ID,
                            employee_first_name,
                            employee_last_name,
                            employee_position,
                            employee_email,
                            employee_office_phone,
                            employee_mobile_phone,
                            employee_date_of_hire,
                            employee_date_end_of_contract)
          VALUES ( SQ_DIM_employee.NEXTVAL,
                            employee_passport_ID(i),
                            employee_first_name(i),
```

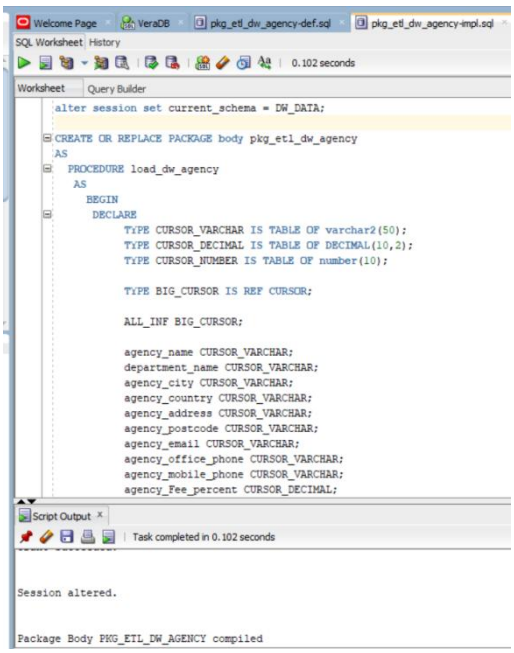
```
                        employee_last_name(i),
                        employee_position(i),
                        employee_email(i),
                        employee_office_phone(i),
                        employee_mobile_phone(i),
                        employee_date_of_hire(i),
                        employee_date_end_of_contract(i)
                );
        COMMIT;
        ELSE UPDATE DW_DATA.DIM_employee
       SET
         employee_passport_ID = employee_passport_ID(i),
         employee_first_name = employee_first_name(i),
         employee_last_name = employee_last_name(i),
         employee_position = employee_position(i),
         employee_email = employee_email(i),
         employee_office_phone = employee_office_phone(i),
         employee_mobile_phone = employee_mobile_phone(i),
         employee_date_of_hire = employee_date_of_hire(i),
         employee_date_end_of_contract = employee_date_end_of_contract(i)
       WHERE employee_ID = employee_ID(i);
                COMMIT;
     END IF;
   END LOOP;
            END;
  END load_dw_employee;
END pkg_etl_dw_employee;
```

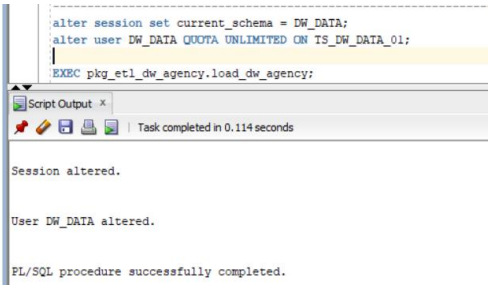
- pkg_etl_dw_agency

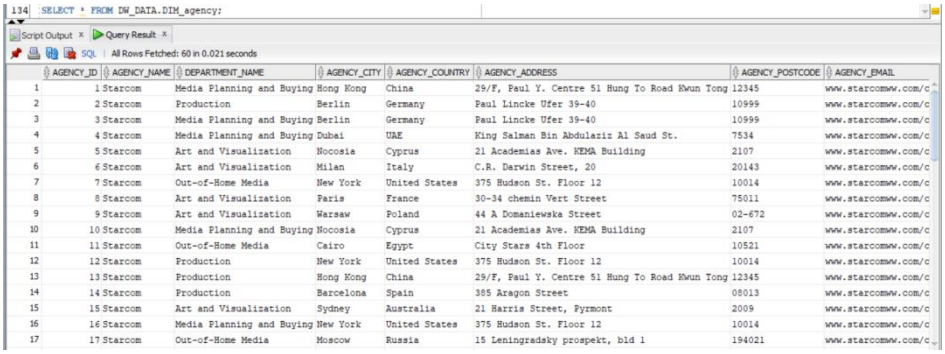


Grant permissions to table DW_CL.cls_t_agency from the Cleansing level to user DW_DATA on DW level and create pkg_etl_dw_agency-impl.









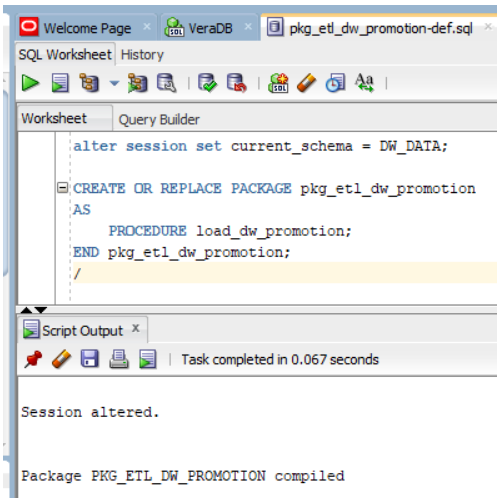
Code:

```
alter session set current_schema = DW_DATA;
CREATE OR REPLACE PACKAGE body pkg_etl_dw_agency
AS
 PROCEDURE load_dw_agency
  AS
   BEGIN
    DECLARE
       TYPE CURSOR_VARCHAR IS TABLE OF varchar2(50);
       TYPE CURSOR_DECIMAL IS TABLE OF DECIMAL(10,2);
       TYPE CURSOR_NUMBER IS TABLE OF number(10);
       TYPE BIG_CURSOR IS REF CURSOR;
       ALL_INF BIG_CURSOR;

       agency_name CURSOR_VARCHAR;
       department_name CURSOR_VARCHAR;
       agency_city CURSOR_VARCHAR;
       agency_country CURSOR_VARCHAR;
       agency_address CURSOR_VARCHAR;
       agency_postcode CURSOR_VARCHAR;
```

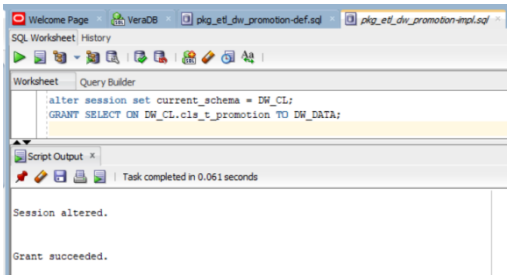
```
agency_email CURSOR_VARCHAR;
agency_office_phone CURSOR_VARCHAR;
agency_mobile_phone CURSOR_VARCHAR;
agency_Fee_percent CURSOR_DECIMAL;
agency_VAT_percent CURSOR_DECIMAL;

agency_address_stage CURSOR_VARCHAR;
agency_ID CURSOR_NUMBER;

BEGIN
    OPEN ALL_INF FOR
      SELECT
        source_CL.agency_name AS  agency_name_source_CL,
        source_CL.department_name AS department_name_source_CL,
        source_CL.agency_city AS  agency_city_source_CL,
        source_CL.agency_country AS  agency_country_source_CL,
        source_CL.agency_address AS  agency_address_source_CL,
        source_CL.agency_postcode AS agency_postcode_source_CL,
        source_CL.agency_email AS  agency_email_source_CL,
        source_CL.agency_office_phone AS  agency_office_phone_source_CL,
        source_CL.agency_mobile_phone AS agency_mobile_phone_source_CL,
        source_CL.agency_Fee_percent AS  agency_Fee_percent_source_CL,
        source_CL.agency_VAT_percent AS  agency_VAT_percent_source_CL,

        stage.agency_address AS agency_address_stage,
        stage.agency_ID AS  agency_ID
       FROM (SELECT DISTINCT * FROM DW_CL.cls_t_agency) source_CL
          LEFT JOIN DW_DATA.DIM_agency stage
            ON (source_CL.agency_address = stage.agency_address and source_CL.department_name = stage.department_name);
    FETCH ALL_INF
    BULK COLLECT INTO
          agency_name,
          department_name,
          agency_city,
          agency_country,
          agency_address,
          agency_postcode,
          agency_email,
          agency_office_phone,
          agency_mobile_phone,
          agency_Fee_percent,
          agency_VAT_percent,

          agency_address_stage,
          agency_ID;
    CLOSE ALL_INF;
FOR i IN agency_ID.FIRST .. agency_ID.LAST LOOP
  IF ( agency_ID ( i ) IS NULL ) THEN
    INSERT INTO DW_DATA.DIM_agency (
                  agency_ID,
                  agency_name,
                  department_name,
                  agency_city,
                  agency_country,
                  agency_address,
                  agency_postcode,
                  agency_email,
                  agency_office_phone,
                  agency_mobile_phone,
                  agency_Fee_percent,
                  agency_VAT_percent)
      VALUES ( DW_DATA.SQ_DIM_agency.NEXTVAL,
                  agency_name(i),
                  department_name(i),
```

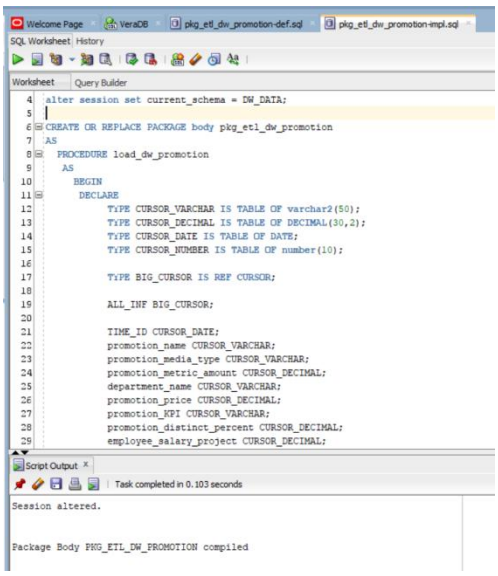
```
                    agency_city(i),
                    agency_country(i),
                    agency_address(i),
                    agency_postcode(i),
                    agency_email(i),
                    agency_office_phone(i),
                    agency_mobile_phone(i),
                    agency_Fee_percent(i),
                    agency_VAT_percent(i)
            );
    COMMIT;
    ELSE UPDATE DW_DATA.DIM_agency
    SET
      agency_name = agency_name(i),
      department_name = department_name(i),
      agency_city = agency_city(i),
      agency_country = agency_country(i),
      agency_address = agency_address(i),
      agency_postcode = agency_postcode(i),
      agency_email = agency_email(i),
      agency_office_phone = agency_office_phone(i),
      agency_mobile_phone = agency_mobile_phone(i),
      agency_Fee_percent = agency_Fee_percent(i),
      agency_VAT_percent = agency_VAT_percent(i)
    WHERE agency_ID = agency_ID(i);
              COMMIT;
  END IF;
 END LOOP;
          END;
 END load_dw_agency;
END pkg_etl_dw_agency;
```

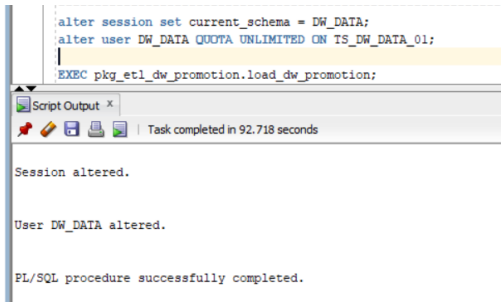
- pkg_etl_dw_promotion

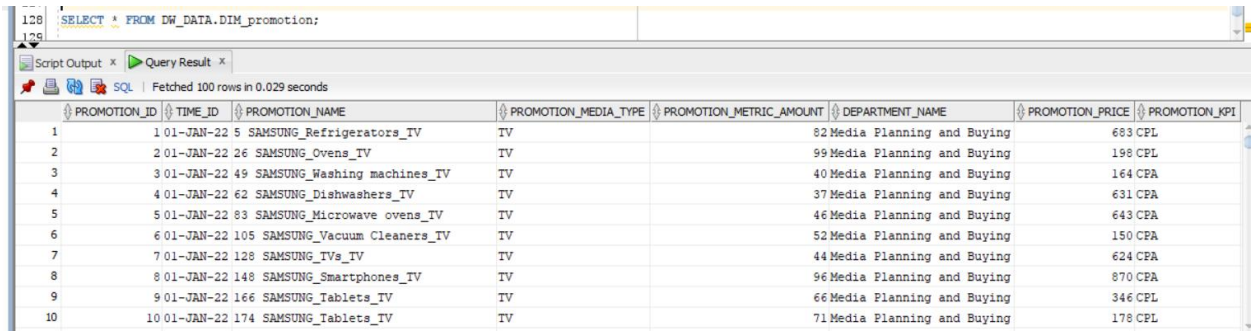


Grant permissions to table DW_CL.cls_t_promotion from the Cleansing level to user DW_DATA on DW level and create pkg_etl_dw_promotion-impl.









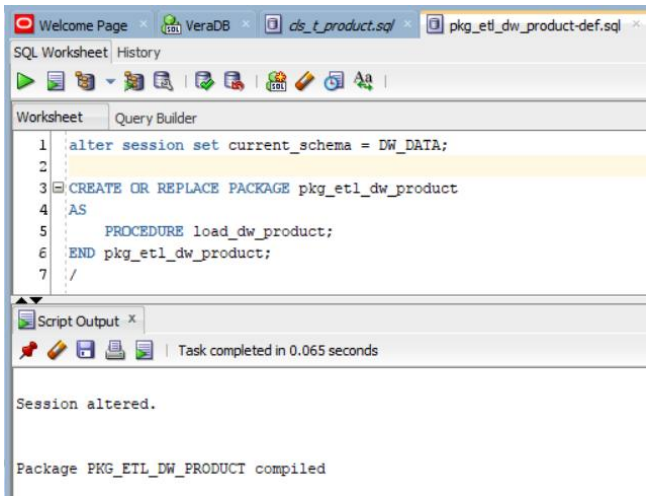
Code:

```
alter session set current_schema = DW_DATA;
CREATE OR REPLACE PACKAGE body pkg_etl_dw_promotion
AS
 PROCEDURE load_dw_promotion
  AS
    BEGIN
     DECLARE
        TYPE CURSOR_VARCHAR IS TABLE OF varchar2(50);
        TYPE CURSOR_DECIMAL IS TABLE OF DECIMAL(30,2);
        TYPE CURSOR_DATE IS TABLE OF DATE;
        TYPE CURSOR_NUMBER IS TABLE OF number(10);

        TYPE BIG_CURSOR IS REF CURSOR;

        ALL_INF BIG_CURSOR;

        TIME_ID CURSOR_DATE;
        promotion_name CURSOR_VARCHAR;
        promotion_media_type CURSOR_VARCHAR;
        promotion_metric_amount CURSOR_DECIMAL;
        department_name CURSOR_VARCHAR;
```

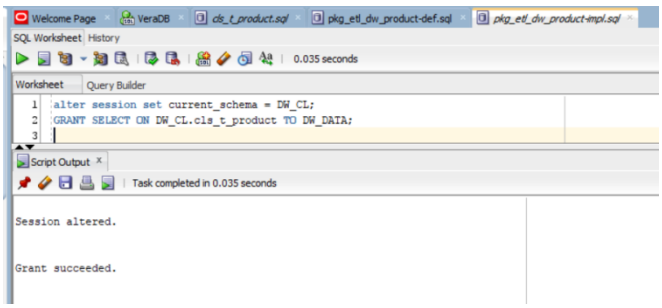
```
promotion_price CURSOR_DECIMAL;
promotion_KPI CURSOR_VARCHAR;
promotion_distinct_percent CURSOR_DECIMAL;
employee_salary_project CURSOR_DECIMAL;

promotion_name_stage CURSOR_VARCHAR;
TIME_ID_stage CURSOR_DATE;
promotion_ID CURSOR_NUMBER;

BEGIN
    OPEN ALL_INF FOR
      SELECT
        source_CL.TIME_ID AS TIME_ID_source_CL,
        source_CL.promotion_name AS  promotion_name_source_CL,
        source_CL.promotion_media_type AS  promotion_media_type_source_CL,
        source_CL.promotion_metric_amount AS  promotion_metric_amount_source_CL,
        source_CL.department_name AS department_name_source_CL,
        source_CL.promotion_price AS  promotion_price_source_CL,
        source_CL.promotion_KPI AS  promotion_KPI_source_CL,
        source_CL.promotion_distinct_percent AS promotion_distinct_percent_source_CL,
        source_CL.employee_salary_project AS  aemployee_salary_project_source_CL,

        stage.promotion_name AS promotion_name_stage,
        stage.TIME_ID AS TIME_ID_stage,
        stage.promotion_ID AS  promotion_ID
       FROM (SELECT DISTINCT * FROM DW_CL.cls_t_promotion) source_CL
          LEFT JOIN DW_DATA.DIM_promotion stage
            ON (source_CL.promotion_name = stage.promotion_name AND source_CL.TIME_ID = stage.TIME_ID)
      ;
    FETCH ALL_INF
    BULK COLLECT INTO
          TIME_ID,
          promotion_name,
          promotion_media_type,
          promotion_metric_amount,
          department_name,
          promotion_price,
          promotion_KPI,
          promotion_distinct_percent,
          employee_salary_project,

          promotion_name_stage,
          TIME_ID_stage,
          promotion_ID;
    CLOSE ALL_INF;

FOR i IN promotion_ID.FIRST .. promotion_ID.LAST LOOP
   IF ( promotion_ID ( i ) IS NULL ) THEN
     INSERT INTO DW_DATA.DIM_promotion (
                    promotion_ID,
                    TIME_ID,
                    promotion_name,
                    promotion_media_type,
                    promotion_metric_amount,
                    department_name,
                    promotion_price,
                    promotion_KPI,
                    promotion_distinct_percent,
                    employee_salary_project)
       VALUES ( DW_DATA.SQ_DIM_promotion.NEXTVAL,
                    TIME_ID(i),
                    promotion_name(i),
                    promotion_media_type(i),
                    promotion_metric_amount(i),
```

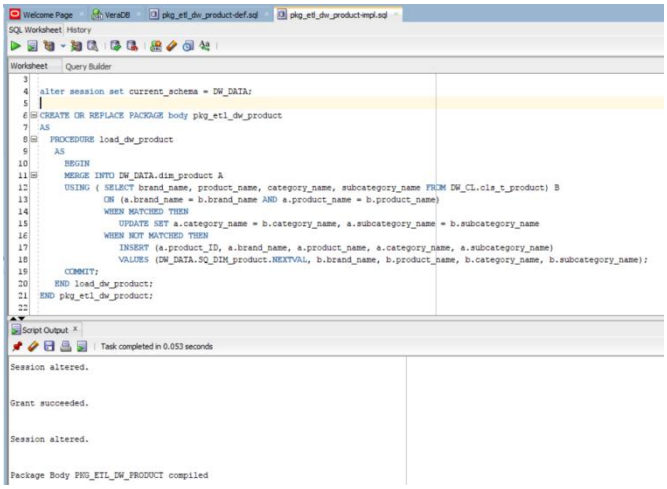
```
                    department_name(i),
                    promotion_price(i),
                    promotion_KPI(i),
                    promotion_distinct_percent(i),
                    employee_salary_project(i)
            );
     COMMIT;

     ELSE UPDATE DW_DATA.DIM_promotion
     SET
       TIME_ID = TIME_ID(i),
       promotion_name = promotion_name(i),
       promotion_media_type = promotion_media_type(i),
       promotion_metric_amount = promotion_metric_amount(i),
       department_name = department_name(i),
       promotion_price = promotion_price(i),
       promotion_KPI = promotion_KPI(i),
       promotion_distinct_percent = promotion_distinct_percent(i),
       employee_salary_project = employee_salary_project(i)
     WHERE promotion_ID = promotion_ID(i);

              COMMIT;


   END IF;

  END LOOP;

          END;
  END load_dw_promotion;
END pkg_etl_dw_promotion;
```

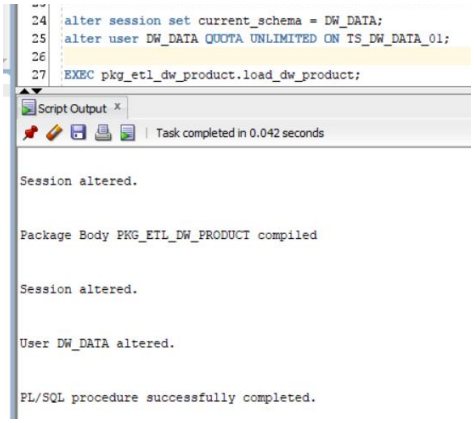
- pkg_etl_dw_product

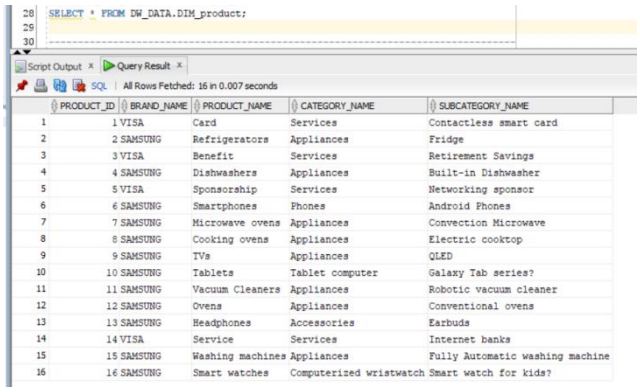


Grant permissions to table DW_CL.cls_t_product from the Cleansing level to user DW_DATA on DW level and create pkg_etl_dw_product-impl.







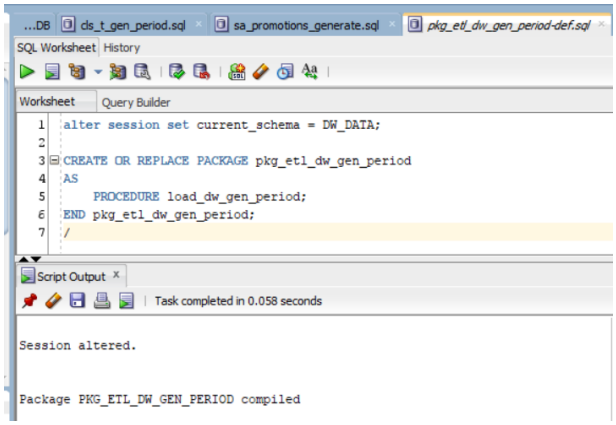


	PRODUCT_ID	BRAND_NAME	PRODUCT_NAME	CATEGORY_NAME	SUBCATEGORY_NAME
1	1	VISA	Card	Services	Contactless smart card
2	2	SAMSUNG	Refrigerators	Appliances	Fridge
3	3	VISA	Benefit	Services	Retirement Savings
4	4	SAMSUNG	Dishwashers	Appliances	Built-in Dishwasher
5	5	VISA	Sponsorship	Services	Networking sponsor
6	6	SAMSUNG	Smartphones	Phones	Android Phones
7	7	SAMSUNG	Microwave ovens	Appliances	Convection Microwave
8	8	SAMSUNG	Cooking ovens	Appliances	Electric cooktop
9	9	SAMSUNG	TVs	Appliances	QLED
10	10	SAMSUNG	Tablets	Tablet computer	Galaxy Tab series?
11	11	SAMSUNG	Vacuum Cleaners	Appliances	Robotic vacuum cleaner
12	12	SAMSUNG	Ovens	Appliances	Conventional ovens
13	13	SAMSUNG	Headphones	Accessories	Earbuds
14	14	VISA	Service	Services	Internet banks
15	15	SAMSUNG	Washing machines	Appliances	Fully Automatic washing machine
16	16	SAMSUNG	Smart watches	Computerized wristwatch	Smart watch for kids?

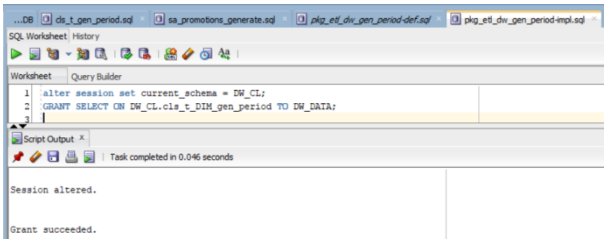
Code:

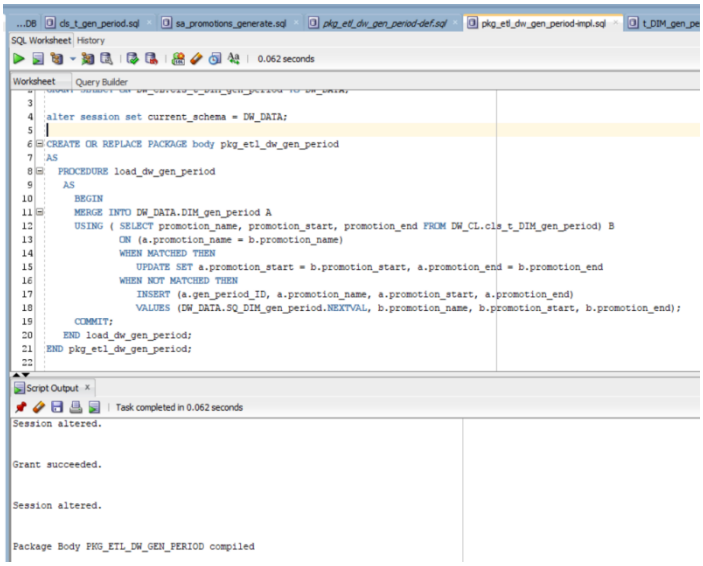
```
alter session set current_schema = DW_DATA;
CREATE OR REPLACE PACKAGE body pkg_etl_dw_product
AS
 PROCEDURE load_dw_product
  AS
   BEGIN
   MERGE INTO DW_DATA.dim_product A
   USING ( SELECT brand_name, product_name, category_name, subcategory_name FROM DW_CL.cls_t_product) B
        ON (a.brand_name = b.brand_name AND a.product_name = b.product_name)
        WHEN MATCHED THEN
          UPDATE SET a.category_name = b.category_name, a.subcategory_name = b.subcategory_name
        WHEN NOT MATCHED THEN
          INSERT (a.product_ID, a.brand_name, a.product_name, a.category_name, a.subcategory_name)
          VALUES (DW_DATA.SQ_DIM_product.NEXTVAL, b.brand_name, b.product_name, b.category_name, b.subcategory_name);
   COMMIT;
  END load_dw_product;
END pkg_etl_dw_product;
```

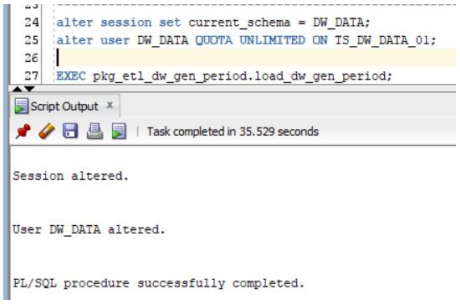
- pkg_etl_dw_gen_period



Grant permissions to table DW_CL.cls_t_DIM_gen_period from the Cleansing level to user DW_DATA on DW level and create pkg_etl_dw_product-impl.







```
28  SELECT * FROM DW_DATA.DIM_gen_period;
```

Script Output | Query Result

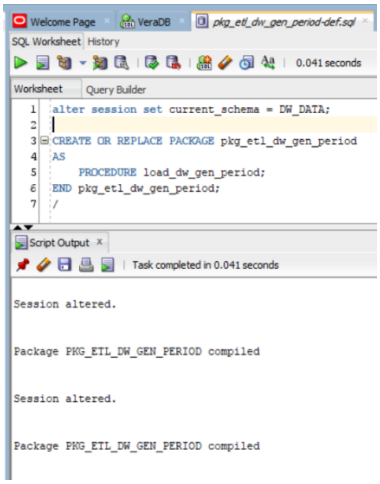
SQL | Fetched 50 rows in 0.01 seconds

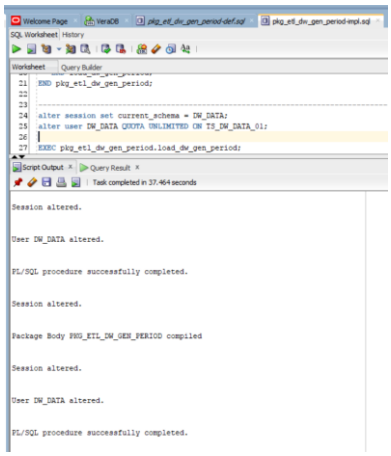
	GEN_PERIOD_ID	PROMOTION_NAME	PROMOTION_START	PROMOTION_END
1	1	9 SAMSUNG_Refrigerators_TV	01-JAN-22	12-FEB-22
2	2	56 SAMSUNG_Washing machines_TV	01-JAN-22	07-FEB-22
3	3	59 SAMSUNG_Washing machines_TV	01-JAN-22	06-FEB-22
4	4	77 SAMSUNG_Microwave ovens_TV	01-JAN-22	04-FEB-22
5	5	97 SAMSUNG_Vacuum Cleaners_TV	01-JAN-22	30-JAN-22
6	6	106 SAMSUNG_Smart watches_TV	01-JAN-22	20-MAR-22
7	7	114 SAMSUNG_Smart watches_TV	01-JAN-22	13-MAR-22
8	8	117 SAMSUNG_Smart watches_TV	01-JAN-22	21-JAN-22
9	9	127 SAMSUNG_TVs_TV	01-JAN-22	23-FEB-22
10	10	142 SAMSUNG_Smartphones_TV	01-JAN-22	25-MAR-22
11	11	175 SAMSUNG_Tablets_TV	01-JAN-22	26-JAN-22
12	12	211 SAMSUNG_Cooking ovens_press	01-JAN-22	11-MAR-22
13	13	232 SAMSUNG_Washing machines_press	01-JAN-22	13-JAN-22
14	14	235 SAMSUNG_Washing machines_press	01-JAN-22	17-MAR-22

Code:

```
alter session set current_schema = DW_DATA;
CREATE OR REPLACE PACKAGE body pkg_etl_dw_gen_period
AS
 PROCEDURE load_dw_gen_period
  AS
   BEGIN
   MERGE INTO DW_DATA.DIM_gen_period A
   USING ( SELECT promotion_name, promotion_start, promotion_end FROM DW_CL.cls_t_DIM_gen_period) B
        ON (a.promotion_name = b.promotion_name)
        WHEN MATCHED THEN
          UPDATE SET a.promotion_start = b.promotion_start, a.promotion_end = b.promotion_end
        WHEN NOT MATCHED THEN
          INSERT (a.gen_period_ID, a.promotion_name, a.promotion_start, a.promotion_end)
          VALUES (DW_DATA.SQ_DIM_gen_period.NEXTVAL, b.promotion_name, b.promotion_start, b.promotion_end);
   COMMIT;
  END load_dw_gen_period;
END pkg_etl_dw_gen_period;
```

Let's check what happens to the data after restarting the packages.





```
SELECT * FROM DW_DATA.DIM_gen_period;
```

Script Output | Query Result | SQL | Fetched 50 rows in 0.007 seconds

	GEN_PERIOD_ID	PROMOTION_NAME	PROMOTION_START	PROMOTION_END
1	1	9 SAMSUNG_Refrigerators_TV	01-JAN-22	12-FEB-22
2	2	56 SAMSUNG_Washing machines_TV	01-JAN-22	07-FEB-22
3	3	59 SAMSUNG_Washing machines_TV	01-JAN-22	06-FEB-22
4	4	77 SAMSUNG_Microwave ovens_TV	01-JAN-22	04-FEB-22
5	5	97 SAMSUNG_Vacuum Cleaners_TV	01-JAN-22	30-JAN-22
6	6	106 SAMSUNG_Smart watches_TV	01-JAN-22	20-MAR-22
7	7	114 SAMSUNG_Smart watches_TV	01-JAN-22	13-MAR-22
8	8	117 SAMSUNG_Smart watches_TV	01-JAN-22	21-JAN-22
9	9	127 SAMSUNG_TVs_TV	01-JAN-22	23-FEB-22
10	10	142 SAMSUNG_Smartphones_TV	01-JAN-22	25-MAR-22
11	11	175 SAMSUNG_Tablets_TV	01-JAN-22	26-JAN-22
12	12	211 SAMSUNG_Cooking ovens_press	01-JAN-22	11-MAR-22
13	13	232 SAMSUNG_Washing machines_press	01-JAN-22	13-JAN-22
14	14	235 SAMSUNG_Washing machines_press	01-JAN-22	17-MAR-22
15	15	244 SAMSUNG_Dishwashers_press	01-JAN-22	28-MAR-22
16	16	245 SAMSUNG_Dishwashers_press	01-JAN-22	06-MAR-22
17	17	253 SAMSUNG_Dishwashers_press	01-JAN-22	16-JAN-22
18	18	266 SAMSUNG_Microwave ovens_press	01-JAN-22	22-FEB-22
19	19	269 SAMSUNG_Microwave ovens_press	01-JAN-22	01-FEB-22
20	20	283 SAMSUNG_Vacuum Cleaners_press	01-JAN-22	27-JAN-22
21	21	289 SAMSUNG_Smart watches_press	01-JAN-22	21-FEB-22
22	22	300 SAMSUNG_Smart watches_press	01-JAN-22	30-MAR-22

As we can notice, the data does not change, all the data corresponds to the primary key.

- pkg_etl_dw_transaction

...sql | t_FCT_business.sql | pkg_etl_dw_FCT_business-def.sql

SQL Worksheet | History

Worksheet | Query Builder

```
alter session set current_schema = DW_DATA;

CREATE OR REPLACE PACKAGE pkg_etl_dw_transaction
AS
    PROCEDURE load_dw_transaction;
END pkg_etl_dw_transaction;
/
```

Script Output | Task completed in 0.062 seconds

```
Session altered.


Package PKG_ETL_DW_TRANSACTION compiled
```

Grant permissions to table DW_CL.cls_t_transaction from the Cleansing level to user DW_DATA on DW level and create pkg_etl_dw_transaction-impl.

...ge | VeraDB | reload_business_cls.sql | cls_t_transaction.sql | t_FCT_business.sql | pkg_etl_dw_FCT_business-def.sql | pkg_etl_dw_FCT_business-impl.sql

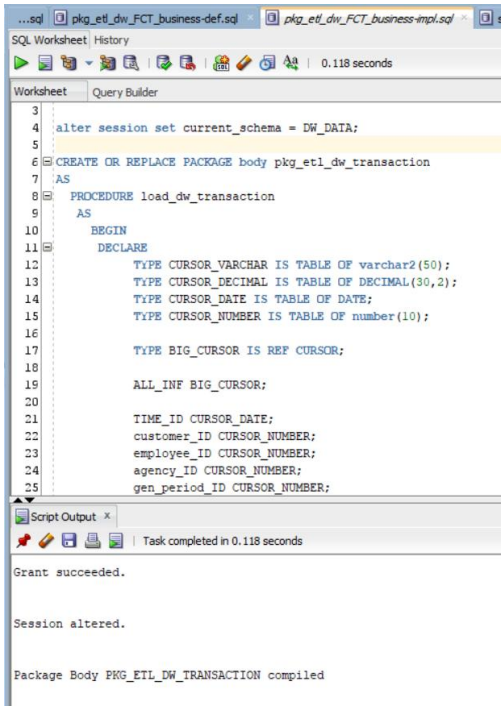
SQL Worksheet | History

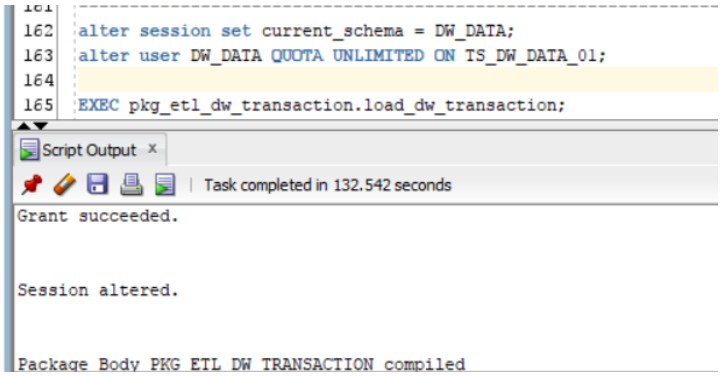
Worksheet | Query Builder

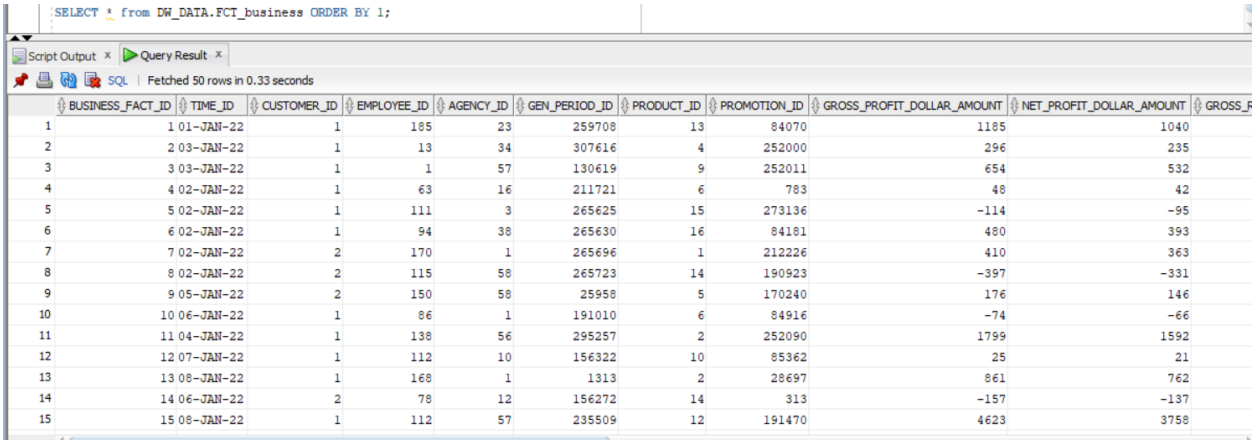
```
alter session set current_schema = DW_CL;
GRANT SELECT ON DW_CL.cls_t_transaction TO DW_DATA;
```

Script Output | Task completed in 0.035 seconds

```
Session altered.


Grant succeeded.
```







```
SELECT * from DW_DATA.FCT_business ORDER BY 1;
```

Script Output | Query Result | SQL | Fetched 50 rows in 0.33 seconds

	BUSINESS_FACT_ID	TIME_ID	CUSTOMER_ID	EMPLOYEE_ID	AGENCY_ID	GEN_PERIOD_ID	PRODUCT_ID	PROMOTION_ID	GROSS_PROFIT_DOLLAR_AMOUNT	NET_PROFIT_DOLLAR_AMOUNT	GROSS_F
1	1	01-JAN-22	1	185	23	259708	13	84070	1185	1040	
2	2	03-JAN-22	1	13	34	307616	4	252000	296	235	
3	3	03-JAN-22	1	1	57	130619	9	252011	654	532	
4	4	02-JAN-22	1	63	16	211721	6	783	48	42	
5	5	02-JAN-22	1	111	3	265625	15	273136	-114	-95	
6	6	02-JAN-22	1	94	38	265630	16	84181	480	393	
7	7	02-JAN-22	2	170	1	265696	1	212226	410	363	
8	8	02-JAN-22	2	115	58	265723	14	190923	-397	-331	
9	9	05-JAN-22	2	150	58	25958	5	170240	176	146	
10	10	06-JAN-22	1	86	1	191010	6	84916	-74	-66	
11	11	04-JAN-22	1	138	56	295257	2	252090	1799	1592	
12	12	07-JAN-22	1	112	10	156322	10	85362	25	21	
13	13	08-JAN-22	1	168	1	1313	2	28697	861	762	
14	14	06-JAN-22	2	78	12	156272	14	313	-157	-137	
15	15	08-JAN-22	1	112	57	235509	12	191470	4623	3758	

Code:

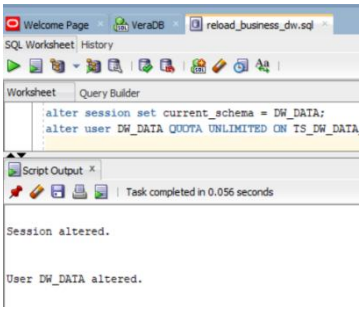
```
alter session set current_schema = DW_DATA;
CREATE OR REPLACE PACKAGE body pkg_etl_dw_transaction
AS
 PROCEDURE load_dw_transaction
  AS
   BEGIN
    DECLARE
       TYPE CURSOR_VARCHAR IS TABLE OF varchar2(50);
       TYPE CURSOR_DECIMAL IS TABLE OF DECIMAL(30,2);
       TYPE CURSOR_DATE IS TABLE OF DATE;
```

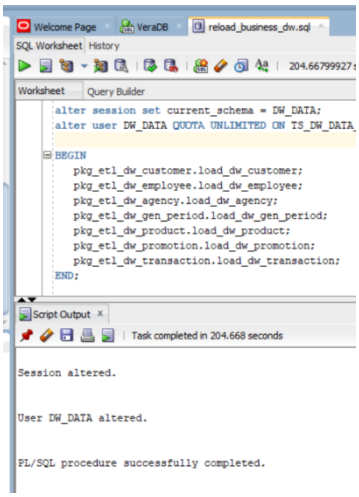
```
TYPE CURSOR_NUMBER IS TABLE OF number(10);
TYPE BIG_CURSOR IS REF CURSOR;
ALL_INF BIG_CURSOR;
TIME_ID CURSOR_DATE;
customer_ID CURSOR_NUMBER;
employee_ID CURSOR_NUMBER;
agency_ID CURSOR_NUMBER;
gen_period_ID CURSOR_NUMBER;
product_ID CURSOR_NUMBER;
promotion_ID CURSOR_NUMBER;
Business_Fact_ID CURSOR_NUMBER;
gross_profit_dollar_amount CURSOR_NUMBER;
net_profit_dollar_amount CURSOR_NUMBER;
gross_revenue_dollar_amount CURSOR_NUMBER;
net_revenue_dollar_amount CURSOR_NUMBER;
gross_cost_dollar_amount CURSOR_NUMBER;
net_cost_dollar_amount CURSOR_NUMBER;
gross_salary_employee_dollar_amount CURSOR_NUMBER;
net_salary_employee_dollar_amount CURSOR_NUMBER;
gross_profit_margin_percent CURSOR_NUMBER;
 BEGIN
     OPEN ALL_INF FOR
       SELECT
          source_CL.TIME_ID,
          cust.customer_ID,
          emp.employee_ID,
          ag.agency_ID,
          gp.gen_period_ID,
          prod.product_ID,
          prom.promotion_ID,
          fct.Business_Fact_ID,
          source_CL.gross_profit_dollar_amount,
          source_CL.net_profit_dollar_amount,
          source_CL.gross_revenue_dollar_amount,
          source_CL.net_revenue_dollar_amount,
          source_CL.gross_cost_dollar_amount,
          source_CL.net_cost_dollar_amount,
          source_CL.gross_salary_employee_dollar_amount,
          source_CL.net_salary_employee_dollar_amount,
          source_CL.gross_profit_margin_percent
        FROM (SELECT DISTINCT * FROM DW_CL.cls_t_transaction) source_CL
            LEFT JOIN DW_DATA.DIM_customer cust
              ON (source_CL.customer_name = cust.customer_name)
            LEFT JOIN DW_DATA.DIM_employee emp
              ON (source_CL.employee_passport_ID = emp.employee_passport_ID AND source_CL.employee_position =
emp.employee_position)
            LEFT JOIN DW_DATA.DIM_agency ag
              ON (source_CL.agency_address = ag.agency_address AND source_CL.department_name = ag.department_name)
            LEFT JOIN DW_DATA.DIM_gen_period gp
              ON (source_CL.promotion_name = gp.promotion_name AND source_CL.promotion_start = gp.promotion_start)
            LEFT JOIN DW_DATA.DIM_product prod
              ON (source_CL.brand_name = prod.brand_name AND source_CL.product_name = prod.product_name)
            LEFT JOIN DW_DATA.DIM_promotion prom
              ON (source_CL.time_id = prom.time_id AND source_CL.promotion_name = prom.promotion_name)
            LEFT JOIN DW_DATA.FCT_business fct
              ON (cust.customer_ID=fct.customer_ID
          AND emp.employee_id=fct.employee_id
          AND ag.agency_id=fct.agency_id
          AND gp.gen_period_id=fct.gen_period_id
          AND prod.product_id=fct.product_id
          AND prom.promotion_id=fct.promotion_id
          AND prom.time_id=fct.time_id
          AND source_CL.gross_profit_dollar_amount=fct.gross_profit_dollar_amount
          AND source_CL.net_profit_dollar_amount=fct.net_profit_dollar_amount
```

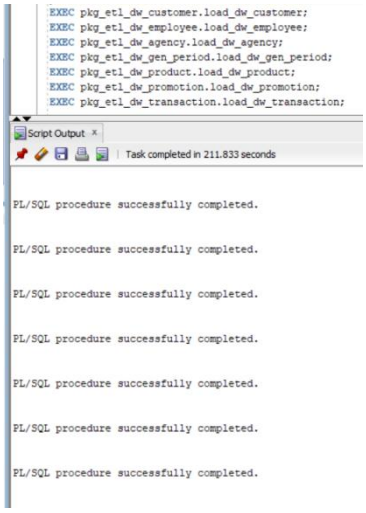
```
                AND source_CL.gross_revenue_dollar_amount=fct.gross_revenue_dollar_amount
                AND source_CL.net_revenue_dollar_amount=fct.net_revenue_dollar_amount
                AND source_CL.gross_cost_dollar_amount=fct.gross_cost_dollar_amount
                AND source_CL.net_cost_dollar_amount=fct.net_cost_dollar_amount
                AND source_CL.gross_salary_employee_dollar_amount=fct.gross_salary_employee_dollar_amount
                AND source_CL.net_salary_employee_dollar_amount=fct.net_salary_employee_dollar_amount
                AND source_CL.gross_profit_margin_percent=fct.gross_profit_margin_percent
              );
        FETCH ALL_INF
        BULK COLLECT INTO
                TIME_ID,
                customer_ID,
                employee_ID,
                agency_ID,
                gen_period_ID,
                product_ID,
                promotion_ID,
                Business_Fact_ID,
                gross_profit_dollar_amount,
                net_profit_dollar_amount,
                gross_revenue_dollar_amount,
                net_revenue_dollar_amount,
                gross_cost_dollar_amount,
                net_cost_dollar_amount,
                gross_salary_employee_dollar_amount,
                net_salary_employee_dollar_amount,
                gross_profit_margin_percent;
        CLOSE ALL_INF;
FOR i IN Business_Fact_ID.FIRST .. Business_Fact_ID.LAST LOOP
    IF ( Business_Fact_ID ( i ) IS NULL ) THEN
      INSERT INTO DW_DATA.FCT_business (
                        Business_Fact_ID,
                        TIME_ID,
                        customer_ID,
                        employee_ID,
                        agency_ID,
                        gen_period_ID,
                        product_ID,
                        promotion_ID,
                        gross_profit_dollar_amount,
                        net_profit_dollar_amount,
                        gross_revenue_dollar_amount,
                        net_revenue_dollar_amount,
                        gross_cost_dollar_amount,
                        net_cost_dollar_amount,
                        gross_salary_employee_dollar_amount,
                        net_salary_employee_dollar_amount,
                        gross_profit_margin_percent)
        VALUES ( DW_DATA.SQ_FCT_business.NEXTVAL,
                        TIME_ID(i),
                        customer_ID(i),
                        employee_ID(i),
                        agency_ID(i),
                        gen_period_ID(i),
                        product_ID(i),
                        promotion_ID(i),
                        ROUND(gross_profit_dollar_amount(i),2),
                        ROUND(net_profit_dollar_amount(i),2),
                        ROUND(gross_revenue_dollar_amount(i),2),
                        ROUND(net_revenue_dollar_amount(i),2),
                        ROUND(gross_cost_dollar_amount(i),2),
                        ROUND(net_cost_dollar_amount(i),2),
                        ROUND(gross_salary_employee_dollar_amount(i),2),
                        ROUND(net_salary_employee_dollar_amount(i),2),
```

```
                    ROUND(gross_profit_margin_percent(i),2)
                );
        COMMIT;
      END IF;
    END LOOP;
            END;
  END load_dw_transaction;
END pkg_etl_dw_transaction;
```

Run all procedures at the DW level.







```
SELECT * FROM DW_DATA.DIM_customer;
SELECT * FROM DW_DATA.DIM_employee;
SELECT * FROM DW_DATA.DIM_agency;
SELECT * FROM DW_DATA.DIM_gen_period;
SELECT * FROM DW_DATA.DIM_product;
SELECT * FROM DW_DATA.DIM_promotion;
SELECT * from DW_DATA.FCT_business;
```

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4 | Query Result 5 | Query Result 6

SQL | Fetched 50 rows in 0.014 seconds

	BUSINESS_FACT_ID	TIME_ID	CUSTOMER_ID	EMPLOYEE_ID	AGENCY_ID	GEN_PERIOD_ID	PRODUCT_ID	PROMOTION_ID	GROSS_PROFIT_DOLLAR_AMOUNT	NET_PROFIT_DOLLAR_AMOUNT	GROSS_R
1	958391	22-FEB-22	1	122	54	6100	11	233689	-187	-173	
2	958394	20-MAR-22	1	147	34	331239	6	217108	-324	-258	
3	958415	21-MAR-22	1	103	40	146654	16	97873	-210	-167	
4	958422	22-MAR-22	1	122	13	179143	12	140660	46	41	
5	958434	15-MAR-22	1	129	27	226634	12	96566	203	170	
6	958631	21-JAN-22	2	62	11	246757	14	22780	415	364	
7	958632	23-JAN-22	1	14	40	29527	2	167381	129	102	
8	958634	24-JAN-22	1	67	53	29611	2	88463	1433	1365	
9	958648	10-JAN-22	1	28	40	213187	4	191763	0	0	
10	958657	03-JAN-22	1	139	34	209	11	27848	90	71	
11	958668	16-FEB-22	1	95	40	278173	16	105288	105	84	
12	958676	20-FEB-22	1	59	50	59842	12	199642	657	543	
13	958681	28-FEB-22	1	199	30	5575	6	136128	628	583	
14	958683	26-FEB-22	1	69	34	196930	8	128478	-389	-309	
15	958692	29-JAN-22	2	181	34	288891	5	211996	-46	-37	
16	958701	01-FEB-22	1	77	27	31619	10	261558	99	83	
17	958707	11-FEB-22	1	26	34	88972	10	219897	245	195	
18	958709	11-FEB-22	2	68	54	89047	14	296561	-104	-97	