

# U2M4.LW.Core PL/SQL

## Shkrabatouskaya Vera

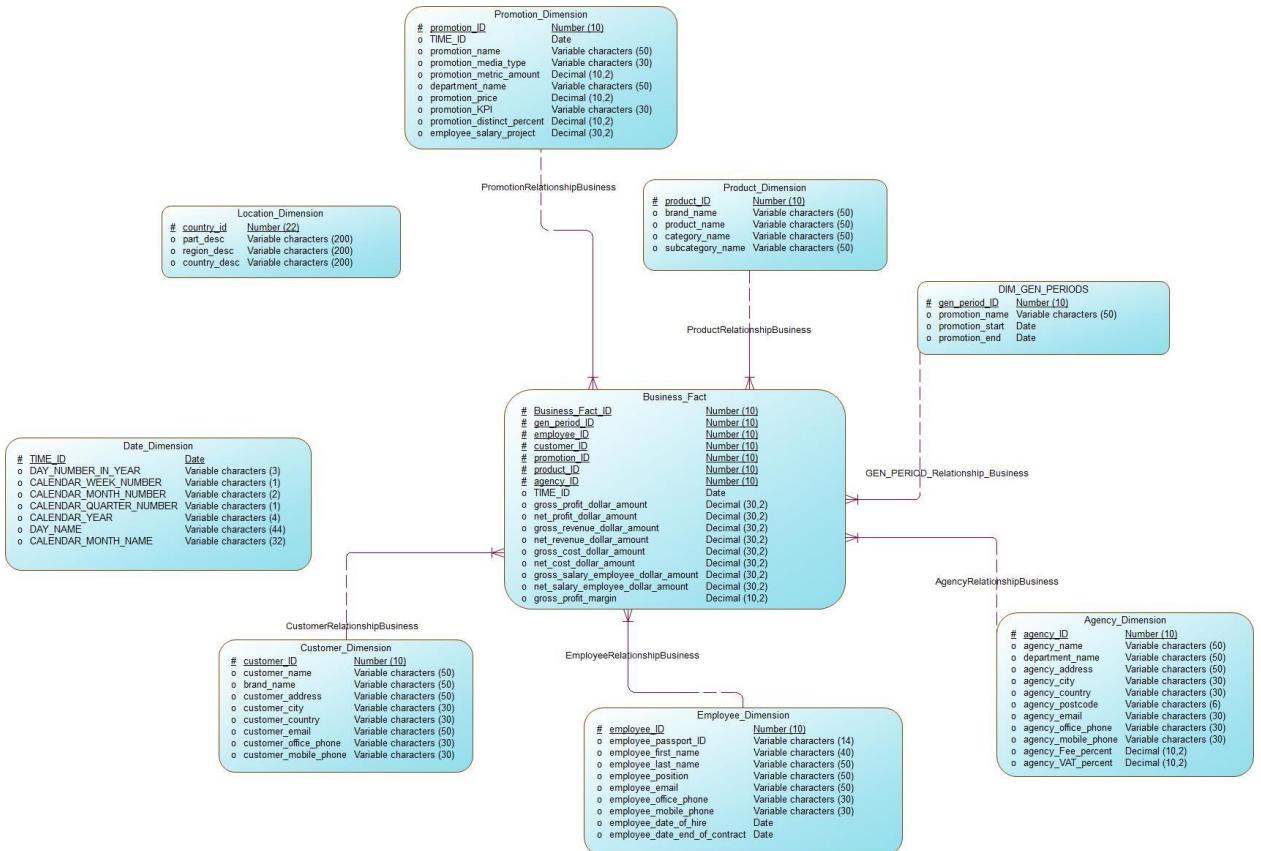
[https://github.com/VeraShkrabatouskaya/DataMola\\_Data-Camping-2022](https://github.com/VeraShkrabatouskaya/DataMola_Data-Camping-2022)

### 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.

```

--alter session set current_schema=DW_CL;
--drop table cls_t_customer;

alter session set current_schema=DW_CL;

Create table cls_t_customer (
    customer_name VARCHAR2(50) NOT NULL,
    brand_name VARCHAR2(50) NOT NULL,
    customer_address VARCHAR2(50) NOT NULL,
    customer_city VARCHAR2(30) NOT NULL,
    customer_country VARCHAR2(30) NOT NULL,
    customer_email VARCHAR2(50) NOT NULL,
    customer_office_phone VARCHAR2(30) NOT NULL,
    customer_mobile_phone VARCHAR2(30) NOT NULL
);

```

Script Output: Table CLS\_T\_CUSTOMER created. Task completed in 0.053 seconds

Let's create packages to get data from Storage level SA\_\* in DW - Cleanup level for the table cls\_t\_customer.

```

1 alter session set current_schema = DW_CL;
2
3 CREATE OR REPLACE PACKAGE pkg_etl_cls_customer
4 AS
5   PROCEDURE load_cls_customer;
6 END pkg_etl_cls_customer;
7 /

```

Session altered.

Package PKG\_ETL\_CLS\_CUSTOMER compiled

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.

```

1 --GRANT UNLIMITED TABLESPACE TO VShkrabatovskaya;
2 --GRANT CONNECT,RESOURCE TO sa_customers;
3 --SELECT * from dba_data_files ;
4 --select * from USER tablespaces;
5 alter session set current_schema=sa_customers;
6 alter user sa_customers QUOTA UNLIMITED ON ts_sa_customers_data_01;
7
8 --alter session set current_schema=sa_customers;
9 --GRANT SELECT ON SA_CUSTOMER_DATA_total TO DW_CL;
10
11 alter session set current_schema=sa_customers;
12 GRANT SELECT ON SA_CUSTOMER DATA with_department TO DW_CL;
13

```

Session altered.

User SA\_CUSTOMERS altered.

Session altered.

Grant succeeded.

```

1 alter session set current_schema = DW_CL;
2
3 CREATE OR REPLACE PACKAGE BODY pkg_etl_cls_customer
4 AS
5   PROCEDURE load_cls_customer
6   AS
7     CURSOR cursor_cls_customer
8     IS
9       SELECT DISTINCT customer_name, brand_name, customer_address, customer_city, customer_country, customer_
10         FROM sa_customers.SA_CUSTOMER DATA with_department
11         WHERE customer_name IS NOT NULL
12           AND brand_name IS NOT NULL
13           AND customer_address IS NOT NULL
14           AND customer_city IS NOT NULL
15           AND customer_country IS NOT NULL
16           AND customer_email IS NOT NULL
17           AND customer_office_phone IS NOT NULL
18           AND customer_mobile_phone IS NOT NULL
19           ;
20
21   BEGIN
22     EXECUTE IMMEDIATE 'TRUNCATE TABLE DW_CL.cls_t_customer';
23   FOR i IN cursor_cls_customer LOOP
24     INSERT INTO DW_CL.cls_t_customer(

```

Session altered.

Package Body PKG\_ETL\_CLS\_CUSTOMER compiled

Code:

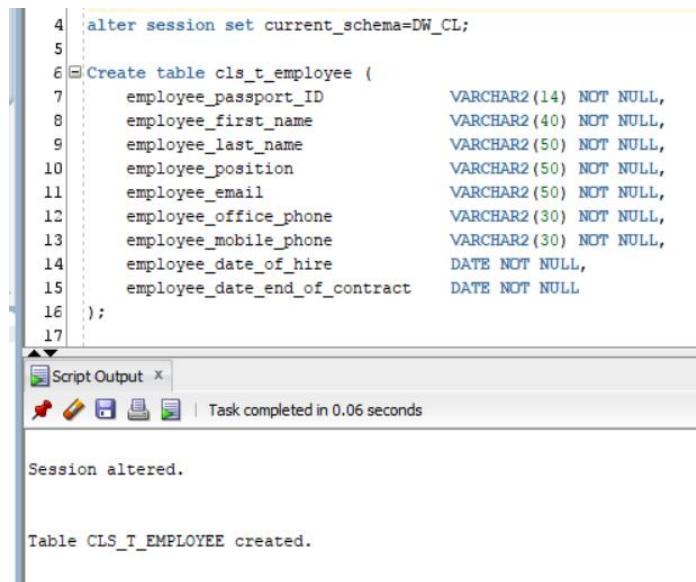
```
CREATE OR REPLACE PACKAGE body pkg_etl_cls_customer
AS
  PROCEDURE load_cls_customer
  AS
    CURSOR cursor_cls_customer
    IS
      SELECT DISTINCT customer_name, brand_name, customer_address, customer_city, customer_country, customer_email,
customer_office_phone, customer_mobile_phone
      FROM 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;
```

The screenshot shows the Oracle SQL Developer interface with two panes. The top pane displays the execution of a PL/SQL block. The code includes setting the current schema to DW\_CL, altering the user quota, executing the package procedure, and committing the transaction. The bottom pane shows the results of a query on the DW\_CL.cls\_t\_customer table, displaying two rows of customer data.

CUSTOMER_NAME	BRAND_NAME	CUSTOMER_ADDRESS	CUSTOMER_CITY	CUSTOMER_COUNTRY	CUSTOMER_EMAIL	CUSTOMER_PHONE
Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS	1-201-229
Visa International Service Association	VISA	900 Metro Center Blvd.	Foster City	United States	https://usa.visa.com/contact-us.html	1-800-847

- `cls_t_employee`

Create a `cls_t_employee` table at the DW - Cleansing Level.



```

4  alter session set current_schema=DW_CL;
5
6  Create table cls_t_employee (
7      employee_passport_ID          VARCHAR2(14) NOT NULL,
8      employee_first_name           VARCHAR2(40) NOT NULL,
9      employee_last_name            VARCHAR2(50) NOT NULL,
10     employee_position             VARCHAR2(50) NOT NULL,
11     employee_email                VARCHAR2(50) NOT NULL,
12     employee_office_phone         VARCHAR2(30) NOT NULL,
13     employee_mobile_phone         VARCHAR2(30) NOT NULL,
14     employee_date_of_hire          DATE NOT NULL,
15     employee_date_end_of_contract DATE NOT NULL
16 );
17

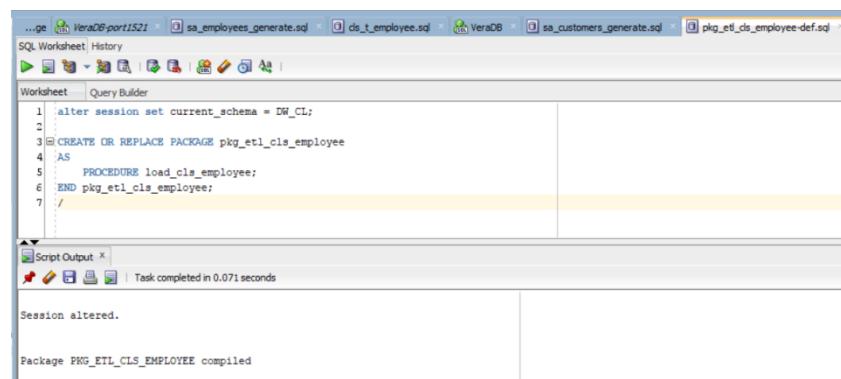
```

Script Output X | 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`.



```

1  alter session set current_schema = DW_CL;
2
3  CREATE OR REPLACE PACKAGE pkg_etl_cls_employee
4  AS
5      PROCEDURE load_cls_employee;
6  END pkg_etl_cls_employee;
7 /

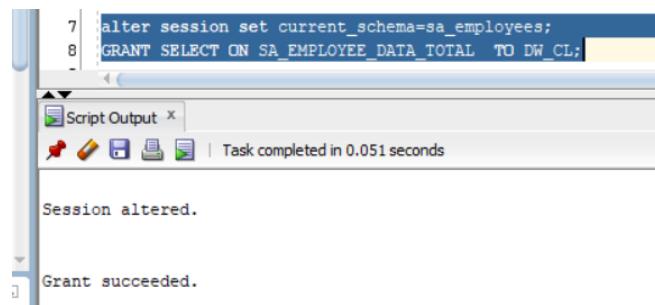
```

Script Output X | 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`.



```

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

```

Script Output X | Task completed in 0.051 seconds

Session altered.

Grant succeeded.

The screenshot shows the Oracle SQL Developer interface. The top bar has tabs for multiple SQL scripts, with 'sa\_employees\_generate.sql' currently selected. Below the tabs is a toolbar with various icons. The main area is divided into two panes: 'Worksheet' on the left and 'Script Output' on the right.

**Worksheet:**

```

1 alter session set current_schema = DW_CL;
2
3 CREATE OR REPLACE PACKAGE body pkg_etl_cls_employee
4 AS
5   PROCEDURE load_cls_employee
6     AS
7       CURSOR cursor_cls_employee
8       IS
9         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
10        WHERE employee_passport_ID IS NOT NULL
11          AND employee_first_name IS NOT NULL
12          AND employee_last_name IS NOT NULL
13          AND employee_position IS NOT NULL
14          AND employee_email IS NOT NULL
15          AND employee_office_phone IS NOT NULL
16          AND employee_mobile_phone IS NOT NULL
17          AND employee_date_of_hire IS NOT NULL
18          AND employee_date_end_of_contract IS NOT NULL
19        ;
20
21   BEGIN
22     EXECUTE IMMEDIATE 'TRUNCATE TABLE DW_CL.cls_t_employee';
23   FOR i IN cursor_cls_employee LOOP
24     INSERT INTO DW_CL.cls_t_employee(

```

**Script Output:**

```

Session altered.

Package Body PKG_ETL_CLS_EMPLOYEE compiled

```

### 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;

```

```

51 alter session set current_schema = DW_CL;
52 alter user DW_CL QUOTA UNLIMITED ON ts_dw_cl;
53
54 EXEC pkg_etl_cls_employee.load_cls_employee;
55
56
57
58

```

Session altered.

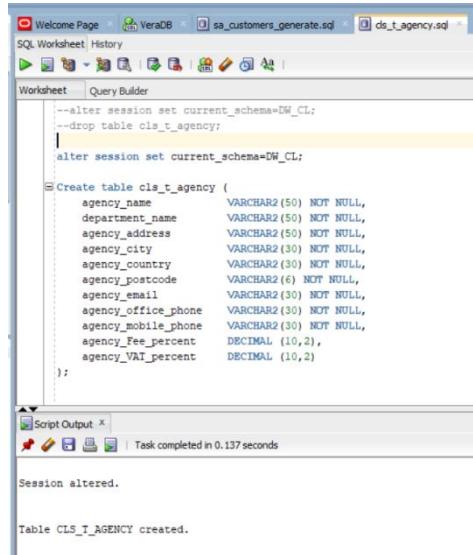
User DW\_CL altered.

PL/SQL procedure successfully completed.

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.

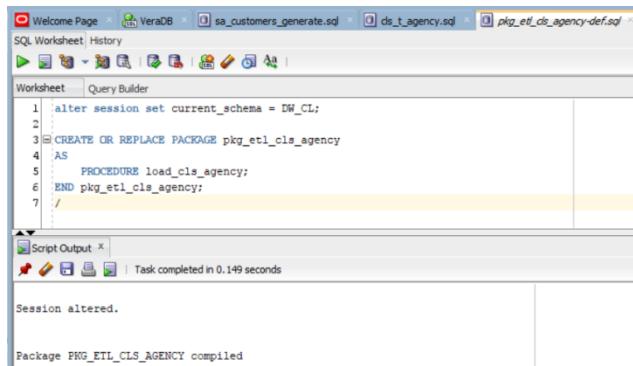


The screenshot shows the Oracle SQL Developer interface with the 'Query Builder' tab selected. The code in the worksheet pane is:

```
--alter session set current_schema=DW_CL;
--drop table cls_t_agency;
|
| alter session set current_schema=DW_CL;
|
| Create table cls_t_agency (
|   agency_name          VARCHAR2(50) NOT NULL,
|   department_name       VARCHAR2(50) NOT NULL,
|   agency_address        VARCHAR2(50) NOT NULL,
|   agency_city           VARCHAR2(30) NOT NULL,
|   agency_country        VARCHAR2(30) NOT NULL,
|   agency_postcode       VARCHAR2(6) NOT NULL,
|   agency_email          VARCHAR2(30) NOT NULL,
|   agency_office_phone   VARCHAR2(30) NOT NULL,
|   agency_mobile_phone   VARCHAR2(30) NOT NULL,
|   agency_fee_percent    DECIMAL (10,2),
|   agency_VAT_percent    DECIMAL (10,2)
| );
|
Session altered.
Table CLS_T_AGENCY created.
```

The 'Script Output' pane shows the message: 'Task completed in 0.137 seconds'. The status bar at the bottom indicates: 'Session altered.' and 'Table CLS\_T\_AGENCY created.'

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

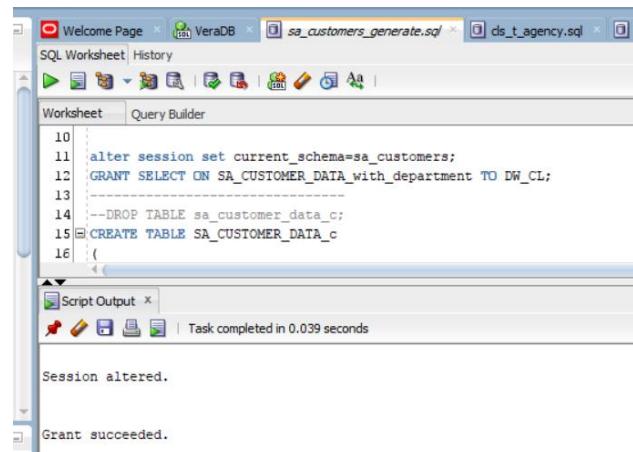


The screenshot shows the Oracle SQL Developer interface with the 'Query Builder' tab selected. The code in the worksheet pane is:

```
1 | alter session set current_schema = DW_CL;
2 |
3 | CREATE OR REPLACE PACKAGE pkg_etl_cls_agency
4 | AS
5 |   PROCEDURE load_cls_agency;
6 | END pkg_etl_cls_agency;
7 |
|
Session altered.
Package PKG_ETL_CLS_AGENCY compiled
```

The 'Script Output' pane shows the message: 'Task completed in 0.149 seconds'. The status bar at the bottom indicates: 'Session altered.' and 'Package PKG\_ETL\_CLS\_AGENCY compiled'.

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.



The screenshot shows the Oracle SQL Developer interface with the 'Query Builder' tab selected. The code in the worksheet pane is:

```
10 |
11 | alter session set current_schema=sa_customers;
12 | GRANT SELECT ON SA_CUSTOMER_DATA_with_department TO DW_CL;
13 |
14 | --DROP TABLE sa_customer_data_c;
15 | CREATE TABLE SA_CUSTOMER_DATA_c
16 | (
| )
|
Session altered.
Grant succeeded.
```

The 'Script Output' pane shows the message: 'Task completed in 0.039 seconds'. The status bar at the bottom indicates: 'Session altered.' and 'Grant succeeded'.

The screenshot shows the Oracle SQL Developer interface with the following details:

- Toolbar:** Welcome Page, VeraDB, sa\_customers\_generate.sql, ds\_t\_agency.sql, pkg\_etl\_ds\_agency-def.sql, pkg\_etl\_ds\_agency-impl.sql, ts\_dv\_d\_sd.
- Worksheet:** History
- Code Area:** A large code editor window titled "Query Builder" containing the following PL/SQL code:

```
1 alter session set current_schema = DW_CL;
2
3 CREATE OR REPLACE PACKAGE body pkg_etl_cls_agency
4 AS
5
6 PROCEDURE load_cls_agency
7 AS
8     CURSOR cursor_cls_agency
9     IS
10        SELECT DISTINCT agency_name, department_name, agency_address, agency_city, agency_country, agency_postcode, agency_email, agency_office_
11        FROM sa_customers.SA_CUSTOMER_DATA_with_department
12        WHERE agency_name IS NOT NULL
13            AND department_name IS NOT NULL
14            AND agency_address IS NOT NULL
15            AND agency_city IS NOT NULL
16            AND agency_postcode IS NOT NULL
17            AND agency_country IS NOT NULL
18            AND agency_email IS NOT NULL
19            AND agency_office_phone IS NOT NULL
20            AND agency_mobile_phone IS NOT NULL
21            AND agency_Fee_percent IS NOT NULL
22            AND agency_VAT_percent IS NOT NULL
23    ;

```
- Script Output:** A panel at the bottom left showing the result of the execution:

```
Session altered.
```
- Status Bar:** Task completed in 0.086 seconds

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_email,
    i.agency_office_phone,
    i.agency_mobile_phone,
    i.agency_Fee_percent,
    i.agency_VAT_percent)
END LOOP;
END;

```

```

    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;

```

The screenshot shows the Oracle SQL Developer interface. The top part displays the PL/SQL code for the `load_cls_agency` procedure. The bottom part shows the `Script Output` window with the following log:

```

57 |-----+
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;
62 |

```

Task completed in 0.361 seconds

Package Body PKG\_ETL\_CLS\_AGENCY compiled

Session altered.

User DW\_CL altered.

PL/SQL procedure successfully completed.

The screenshot shows the Oracle SQL Developer interface with a query result window. The query is:

```

63 | SELECT * FROM cls_t_agency;
64 |
65 |

```

Query Result x | 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.starcommw.com/contact 1-212-	
2 Starcom	Media Planning and Buying	21 Harris Street, Pyrmont	Sydney	Australia	2009	www.starcommw.com/contact 612-86	
3 Starcom	Media Planning and Buying	30-34 chemin Vert Street	Paris	France	75011	www.starcommw.com/contact 33-1-5	
4 Starcom	Media Planning and Buying	Stadelhofstrasse 258	Zurich	Switzerland	8001	www.starcommw.com/contact 41-43-	
5 Starcom	Media Planning and Buying	21 Academias Ave. KEMA Building	Nicosia	Cyprus	2107	www.starcommw.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.starcommw.com/contact 39-051-	
7 Starcom	Media Planning and Buying	Paul Lincke Ufer 39-40	Berlin	Germany	10999	www.starcommw.com/contact 49-211-	
8 Starcom	Media Planning and Buying	21/F Henderson 688	Shanghai	China	12345	www.starcommw.com/contact 39-051-	
9 Starcom	Media Planning and Buying	City Stars 4th Floor	Cairo	Egypt	10521	www.starcommw.com/contact 202-0-	
10 Starcom	Media Planning and Buying	M 3 Damrak 60, 1017	Helsinki	Finland	00-170	www.starcommw.com/contact 358-02	

- `cls_t_product`

Create a `cls_t_product` table at the DW - Cleansing Level.

The screenshot shows the Oracle SQL Developer interface with a query editor window. The code in the editor is:

```

1 --alter session set current_schema=DW_CL;
2 --drop table cls_t_product;
3 |
4 Create table cls_t_product (
5   brand_name      VARCHAR2(50) NOT NULL,
6   product_name    VARCHAR2(50) NOT NULL,
7   category_name   VARCHAR2(50) NOT NULL,
8   subcategory_name VARCHAR2(50) NOT NULL
9 );
10
11 );
12

```

The output pane below shows the results of the execution:

```

Session altered.

Table CLS_T_PRODUCT created.

```

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

The screenshot shows the Oracle SQL Developer interface with a query editor window. The code in the editor is:

```

1 alter session set current_schema = DW_CL;
2
3 CREATE OR REPLACE PACKAGE pkg_etl_cls_product
4 AS
5   PROCEDURE load_cls_product;
6 END pkg_etl_cls_product;
7 /

```

The output pane below shows the results of the execution:

```

Session altered.

Package PRG_ETL_CLS_PRODUCT compiled

```

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.

The screenshot shows the Oracle SQL Developer interface with a query editor window. The code in the editor is:

```

1 --GRANT UNLIMITED TABLESPACE TO VShkrabatovskaya;
2 --GRANT CONNECT,RESOURCE TO sa_customers;
3 --SELECT * from dba_data_files ;
4 --select * from USER tablespaces;
5 alter session set current_schema=sa_customers;
6 alter user sa_customers QUOTA UNLIMITED ON ts_sa_customers_data_01;
7
8 --alter session set current_schema=sa_customers;
9 --GRANT SELECT ON SA_CUSTOMER_DATA_total TO DW_CL;
10
11 alter session set current schema=sa_customers;
12 GRANT SELECT ON SA_CUSTOMER_DATA_with_department TO DW_CL;
13

```

The output pane below shows the results of the execution:

```

Session altered.

Session altered.

Grant succeeded.

```

```

1 alter session set current_schema = DW_CL;
2
3 CREATE OR REPLACE PACKAGE body pkg_etl_cls_product
4 AS
5 PROCEDURE load_cls_product
6 AS
7     CURSOR cursor_cls_product
8     IS
9         SELECT DISTINCT brand_name, product_name, category_name, subcategory_name
10        FROM sa_customers.SA_CUSTOMER_DATA_with_department
11       WHERE brand_name IS NOT NULL
12          AND product_name IS NOT NULL
13          AND category_name IS NOT NULL
14          AND subcategory_name IS NOT NULL
15      ;
16

```

Script Output: X | Task completed in 0.089 seconds

Session altered.

Package Body PKG\_ETL\_CLS\_PRODUCT compiled

### 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;

```

Worksheet    Query Builder

```

36
37 alter session set current_schema = DW_CL;
38 alter user DW_CL QUOTA UNLIMITED ON ts_dw_cl;
39
40 EXEC pkg_etl_cls_product.load_cls_product;

```

Session altered.

Package Body PNG\_ETL\_CLS\_PRODUCT compiled

Session altered.

User DW\_CL altered.

PL/SQL procedure successfully completed.

---

```

42 SELECT * FROM cls_t_product;
43
44

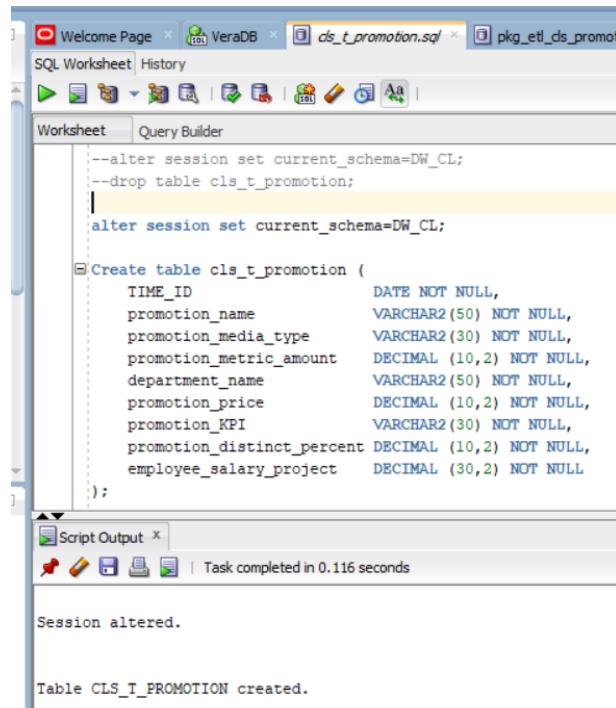
```

Script Output    Query Result | 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.



```
--alter session set current_schema=DW_CL;
--drop table cls_t_promotion;
|
| alter session set current_schema=DW_CL;

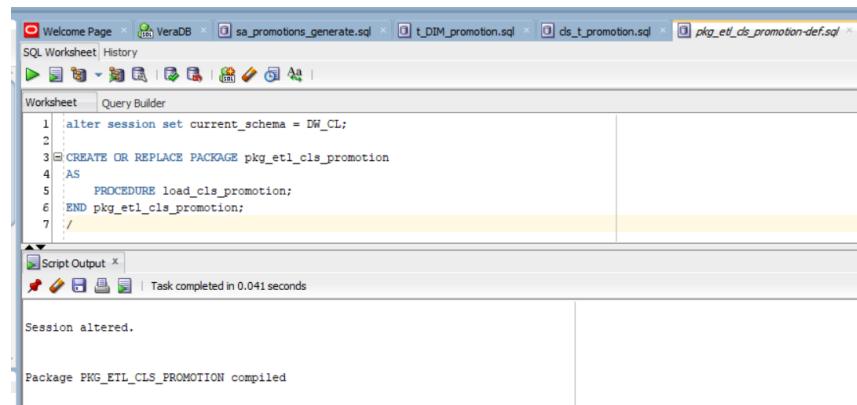
CREATE table cls_t_promotion (
    TIME_ID DATE NOT NULL,
    promotion_name VARCHAR2(50) NOT NULL,
    promotion_media_type VARCHAR2(30) NOT NULL,
    promotion_metric_amount DECIMAL (10,2) NOT NULL,
    department_name VARCHAR2(50) NOT NULL,
    promotion_price DECIMAL (10,2) NOT NULL,
    promotion_KPI VARCHAR2(30) NOT NULL,
    promotion_distinct_percent DECIMAL (10,2) NOT NULL,
    employee_salary_project DECIMAL (30,2) NOT NULL
);
```

Script Output X | Task completed in 0.116 seconds

Session altered.

Table CLS\_T\_PROMOTION created.

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



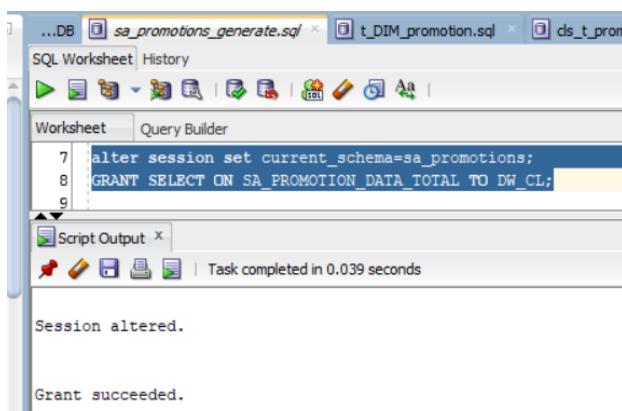
```
1 alter session set current_schema = DW_CL;
2
3 CREATE OR REPLACE PACKAGE pkg_etl_cls_promotion
4 AS
5     PROCEDURE load_cls_promotion;
6 END pkg_etl_cls_promotion;
7 /
```

Script Output X | Task completed in 0.041 seconds

Session altered.

Package PKG\_ETL\_CLS\_PROMOTION compiled

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.



```
7 alter session set current_schema=sa_promotions;
8 GRANT SELECT ON SA_PROMOTION_DATA_TOTAL TO DW_CL;
```

Script Output X | Task completed in 0.039 seconds

Session altered.

Grant succeeded.

```
 1  Welcome Page 2  VerLOB 3  ds_t_promotion.sq 4  pkg_etl.ds_promotion-def.adf 5  pkg_etl.ds_promotion-exp.adf
SQL Worksheet History
File Edit View Insert Run Help
Worksheet Query Builder
Worksheet > Query Builder
Query Builder
ALTER SESSION SET CURRENT_SCHEMA = DW_CL;
--> CREATE OR REPLACE PACKAGE BODY PKG_ETL_CLS_PROMOTION
AS
  PROCEDURE LOAD_CLS_PROMOTION
  AS
    CURSOR CURSOR_CLSPROMOTION
    IS
      SELECT DISTINCT TIME_ID, PROMOTION_NAME, PROMOTION_MEDIA_TYPE, PROMOTION_METRIC_AMOUNT, DEPARTMENT_NAME, PROMOTION_PRICE, PROMOTION_KPI, PROMOTION_DISTINCT_PERCENT
      FROM DS_PROMOTIONS.DS_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';
    END;
  END;
END;
/
```

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;

```

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 x | Query Result x | 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.

The screenshot shows the Oracle SQL Developer interface with a worksheet tab open. The code in the worksheet is:

```

1 --alter session set current_schema=DW_CL;
2 --drop table cls_t_DIM_gen_period;
3
4 alter session set current_schema=DW_CL;
5
6 Create table cls_t_DIM_gen_period (
7     promotion_name          VARCHAR2(50) NOT NULL,
8     promotion_start         Date,
9     promotion_end           Date
10 );
11

```

The script output pane below shows the results of the execution:

```

Session altered.

Table CLS_T_DIM_GEN_PERIOD created.

```

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

The screenshot shows the Oracle SQL Developer interface with a worksheet tab open. The code in the worksheet is:

```

1 alter session set current_schema = DW_CL;
2
3 CREATE OR REPLACE PACKAGE pkg_etl_cls_DIM_gen_period
4 AS
5     PROCEDURE load_cls_DIM_gen_period;
6 END pkg_etl_cls_DIM_gen_period;
7 /

```

The script output pane below shows the results of the execution:

```

Session altered.

Package PKG_ETL_CLS_DIM_GEN_PERIOD compiled

```

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.

The screenshot shows the Oracle SQL Developer interface with a worksheet tab open. The code in the worksheet is:

```

7 alter session set current_schema=sa_promotions;
8 GRANT SELECT ON SA_PROMOTION_DATA_TOTAL TO DW_CL;
9

```

The script output pane below shows the results of the execution:

```

Session altered.

Grant succeeded.

```

```

1 alter session set current_schema = DW_CL;
2 |
3 CREATE OR REPLACE PACKAGE body pkg_etl_cls_DIM_gen_period
4 AS
5 PROCEDURE load_cls_DIM_gen_period
6 AS
7 CURSOR cursor_cls_DIM_gen_period
8 IS
9   SELECT DISTINCT promotion_name, promotion_start, promotion_end
10  FROM sa_promotions.SA_PROMOTION_DATA_TOTAL
11 WHERE promotion_name IS NOT NULL
12   AND promotion_start IS NOT NULL
13   AND promotion_end IS NOT NULL
14 ;
15
16 BEGIN

```

Script Output X | Task completed in 0.049 seconds

Session altered.

Package Body PKG\_ETL\_CLS\_DIM\_GEN\_PERIOD compiled

### 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;

```

```

34 alter session set current_schema = DW_CL;
35 alter user DW_CL QUOTA UNLIMITED ON ts_dw_cl;
36 |
37 EXEC pkg_etl_cls_DIM_gen_period.load_cls_DIM_gen_period;
38

Script Output X | Task completed in 13.598 seconds

Session altered.

Package Body PKG_ETL_CLS_DIM_GEN_PERIOD compiled

Session altered.

User DW_CL altered.

PL/SQL procedure successfully completed.

```

```

1  SELECT * FROM cls_t_promotion;
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41

```

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

- `cls_t_transaction`

```

7  CREATE table cls_t_transaction (
8    time_ID DATE,
9    customer_name VARCHAR2(50) NOT NULL,
10   brand_name VARCHAR2(50) NOT NULL,
11   customer_address VARCHAR2(50) NOT NULL,
12   customer_city VARCHAR2(30) NOT NULL,
13   customer_country VARCHAR2(30) NOT NULL,
14   customer_email VARCHAR2(50) NOT NULL,
15   customer_office_phone VARCHAR2(30) NOT NULL,
16   customer_mobile_phone VARCHAR2(30) NOT NULL,
17   product_name VARCHAR2(50) NOT NULL,
18   agency_name VARCHAR2(50) NOT NULL,
19   agency_city VARCHAR2(30) NOT NULL,
20   agency_country VARCHAR2(30) NOT NULL,
21   agency_address VARCHAR2(50) NOT NULL,
22   agency_postcode VARCHAR2(6) NOT NULL,
23   agency_email VARCHAR2(30) NOT NULL,
24   agency_office_phone VARCHAR2(30) NOT NULL,
25   agency_mobile_phone VARCHAR2(30) NOT NULL,
26   agency_Fee_percent DECIMAL (10,2),
27   agency_VAT_percent DECIMAL (10,2),
28   promotion_media_type VARCHAR2(30) NOT NULL,
29   category_name VARCHAR2(50) NOT NULL,
30   subcategory_name VARCHAR2(50) NOT NULL,
31   department_name VARCHAR2(50) NOT NULL,

```

Table CLS\_T\_TRANSACTION created.

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

```

1  alter session set current_schema = DW_CL;
2
3  CREATE OR REPLACE PACKAGE pkg_etl_cls_transaction
4  AS
5    PROCEDURE load_cls_transaction;
6  END pkg_etl_cls_transaction;
7  /

```

Session altered.

Package PKG\_ETL\_CLS\_TRANSACTION compiled

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.

The image shows two separate Oracle SQL Worksheet windows. Both windows have tabs at the top for various SQL files like ...sql, sa\_transaction\_Report.sql, ds\_t\_transaction.sql, etc. The first window's tab bar also includes 'pkg\_etl\_cls\_transaction-def.sql' and 'pkg\_etl\_cls\_transaction-impl.sql'. The second window's tab bar includes 'VeraDB' and 'Welcome Page'.

**Window 1 (Top):**

```

1 alter session set current_schema=sa_promotions;
2 GRANT SELECT ON SA_TRANSACTION_FACT_DATA TO DW_CL;
3

```

Script Output: Task completed in 0.034 seconds

Session altered.  
Grant succeeded.

**Window 2 (Bottom):**

```

4 alter session set current_schema = DW_CL;
5
6 CREATE OR REPLACE PACKAGE body pkg_etl_cls_transaction
7 AS
8 PROCEDURE load_cls_transaction
9 AS
10 CURSOR cursor_cls_transaction
11 IS
12   SELECT DISTINCT time_ID
13     ,customer_name
14     ,brand_name
15     ,customer_address
16     ,customer_city
17     ,customer_country
18     ,customer_email
19     ,customer_office_phone
20     ,customer_mobile_phone
21     ,product_name
22     ,agency_name
23     ,agency_city
24     ,agency_country

```

Script Output: Task completed in 0.105 seconds

Grant succeeded.  
Session altered.  
Session altered.  
Package Body PKG\_ETL\_CLS\_TRANSACTION compiled

### 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 alter session set current_schema = DW_CL;
191 alter user DW_CL QUOTA UNLIMITED ON ts_dw_cl;
192
193 EXEC pkg_etl_cls_transaction.load_cls_transaction;
194
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 | 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.ds.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.

```

22 | SELECT * FROM DW_CL.cls_t_customer;
23 | SELECT * FROM DW_CL.cls_t_employee;
24 | SELECT * FROM DW_CL.cls_t_agency;
25 | SELECT * FROM DW_CL.cls_t_DIM_gen_period;
26 | SELECT * FROM DW_CL.cls_t_product;
27 | SELECT * FROM DW_CL.cls_t_promotion;
28 | SELECT * FROM cls_t_transaction;

```

Script Output | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4 | Query Result 5 | Query Result 6 | 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	
19	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 ds_t_customer.sql pkg_etl_dls_customer-def.sql pkg_etl_dls_customer-impl.sql t_DIM_customer.sql
SQL Worksheet: History
Worksheet Query Builder
1 --drop table DW_DATA.DIM_customer;
2 --alter session set current_schema = DW_DATA;
3 
4 CREATE TABLE DW_DATA.DIM_customer(
5     customer_ID NUMBER(10),
6     customer_name          VARCHAR2(50),
7     brand_name             VARCHAR2(50),
8     customer_address       VARCHAR2(50),
9     customer_city          VARCHAR2(30),
10    customer_country       VARCHAR2(30),
11    customer_email         VARCHAR2(50),
12    customer_office_phone VARCHAR2(30),
13    customer_mobile_phone VARCHAR2(30),
14
15    CONSTRAINT "PK_T.DIM_customer" PRIMARY KEY(customer_ID)
16 );

```

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
1 --drop sequence DW_DATA.SQ_DIM_customer;
2 --alter session set current_schema = DW_DATA;
3 
4 CREATE SEQUENCE DW_DATA.SQ_DIM_customer
5     START WITH 1
6     INCREMENT BY 1
7     NOCACHE
8     NOCYCLE;
9

```

Script Output | Task completed in 0.032 seconds

Sequence DW\_DATA.SQ\_DIM\_CUSTOMER created.

- DW\_DATA.DIM\_employee

The screenshot shows the Oracle SQL Developer interface with a worksheet titled "Query Builder". The code in the worksheet creates a table named DW\_DATA.DIM\_employee with 16 columns. The columns include employee\_ID (NUMBER(10)), employee\_passport\_ID (VARCHAR2(14)), employee\_first\_name (VARCHAR2(40)), employee\_last\_name (VARCHAR2(50)), employee\_position (VARCHAR2(50)), employee\_email (VARCHAR2(50)), employee\_office\_phone (VARCHAR2(30)), employee\_mobile\_phone (VARCHAR2(30)), employee\_date\_of\_hire (DATE), employee\_date\_end\_of\_contract (DATE), and a primary key constraint "PK\_I.DIM\_employee" on the employee\_ID column.

```

1 --drop table DW_DATA.DIM_employee;
2 --alter session set current_schema = DW_DATA;
3 |
4 CREATE TABLE DW_DATA.DIM_employee(
5   employee_ID NUMBER(10),
6   employee_passport_ID      VARCHAR2(14),
7   employee_first_name       VARCHAR2(40),
8   employee_last_name        VARCHAR2(50),
9   employee_position         VARCHAR2(50),
10  employee_email            VARCHAR2(50),
11  employee_office_phone    VARCHAR2(30),
12  employee_mobile_phone    VARCHAR2(30),
13  employee_date_of_hire    DATE,
14  employee_date_end_of_contract DATE,
15  CONSTRAINT "PK_I.DIM_employee" PRIMARY KEY(employee_ID)
16 );
17

```

Script Output X | Task completed in 0.034 seconds

Table DW\_DATA.DIM\_EMPLOYEE created.

The screenshot shows the Oracle SQL Developer interface with a worksheet titled "Query Builder". The code in the worksheet creates a sequence named DW\_DATA.SQ\_DIM\_employee with a start value of 1, increment by 1, no cache, and no cycle.

```

1 --drop sequence DW_DATA.SQ_DIM_employee;
2 --alter session set current_schema = DW_DATA;
3 |
4 CREATE SEQUENCE DW_DATA.SQ_DIM_employee
5   START WITH      1
6   INCREMENT BY   1
7   NOCACHE
8   NOCYCLE;

```

Script Output X | Task completed in 0.03 seconds

Sequence DW\_DATA.SQ\_DIM\_EMPLOYEE created.

- DW\_DATA.DIM\_agency

The screenshot shows the Oracle SQL Developer interface with a worksheet titled "Query Builder". The code in the worksheet creates a table named DW\_DATA.DIM\_agency with 16 columns. The columns include agency\_ID (NUMBER(10)), agency\_name (VARCHAR2(50)), department\_name (VARCHAR2(50)), agency\_city (VARCHAR2(30)), agency\_country (VARCHAR2(30)), agency\_address (VARCHAR2(50)), agency\_postcode (VARCHAR2(6)), agency\_email (VARCHAR2(30)), agency\_office\_phone (VARCHAR2(30)), agency\_mobile\_phone (VARCHAR2(30)), agency\_fee\_percent (DECIMAL(10,2)), agency\_VAT\_percent (DECIMAL(10,2)), and a primary key constraint "PK\_I.DIM\_agency" on the agency\_ID column.

```

1 --drop table DW_DATA.DIM_agency;
2 --alter session set current_schema = DW_DATA;
3 |
4 CREATE TABLE DW_DATA.DIM_agency(
5   agency_ID NUMBER(10),
6   agency_name      VARCHAR2(50),
7   department_name  VARCHAR2(50),
8   agency_city      VARCHAR2(30),
9   agency_country   VARCHAR2(30),
10  agency_address   VARCHAR2(50),
11  agency_postcode  VARCHAR2(6),
12  agency_email     VARCHAR2(30),
13  agency_office_phone VARCHAR2(30),
14  agency_mobile_phone VARCHAR2(30),
15  agency_fee_percent DECIMAL(10,2),
16  agency_VAT_percent DECIMAL(10,2),
17 |
18  CONSTRAINT "PK_I.DIM_agency" PRIMARY KEY(agency_ID)
19 );
20

```

Script Output X | Task completed in 0.05 seconds

Table DW\_DATA.DIM\_AGENCY created.

The screenshot shows the Oracle SQL Developer interface with a worksheet titled "Query Builder". The code in the worksheet creates a sequence named DW\_DATA.SQ\_DIM\_agency with a start value of 1, increment by 1, no cache, and no cycle.

```

1 --drop sequence DW_DATA.SQ_DIM_agency;
2 --alter session set current_schema = DW_DATA;
3 |
4 CREATE SEQUENCE DW_DATA.SQ_DIM_agency
5   START WITH      1
6   INCREMENT BY   1
7   NOCACHE
8   NOCYCLE;

```

Script Output X | Task completed in 0.037 seconds

Sequence DW\_DATA.SQ\_DIM\_AGENCY created.

- DW\_DATA.DIM\_gen\_period

The screenshot shows two consecutive operations in Oracle SQL Developer:

```

1 --drop table DW_DATA.DIM_gen_period;
2 --alter session set current_schema = DW_DATA;
3
4 CREATE TABLE DW_DATA.DIM_gen_period(
5     gen_period_ID NUMBER(10),
6     promotion_name      VARCHAR2(50),
7     promotion_start    Date,
8     promotion_end      Date,
9
10    CONSTRAINT "PK_T.DIM_gen_period" PRIMARY KEY(gen_period_ID)
11 );

```

Script Output: Table DW\_DATA.DIM\_GEN\_PERIOD created.

```

1 --drop sequence DW_DATA.SQ_DIM_gen_period;
2 --alter session set current_schema = DW_DATA;
3
4 CREATE SEQUENCE DW_DATA.SQ_DIM_gen_period
5     START WITH      1
6     INCREMENT BY   1
7     NOCACHE
8     NOCYCLE;

```

Script Output: Sequence DW\_DATA.SQ\_DIM\_GEN\_PERIOD created.

- DW\_DATA.DIM\_product

The screenshot shows two consecutive operations in Oracle SQL Developer:

```

1 --drop table DW_DATA.DIM_product;
2 --alter session set current_schema = DW_DATA;
3
4 CREATE TABLE DW_DATA.DIM_product(
5     product_ID NUMBER(10),
6     brand_name      VARCHAR2(50),
7     product_name    VARCHAR2(50),
8     category_name   VARCHAR2(50),
9     subcategory_name VARCHAR2(50),
10
11    CONSTRAINT "PK_T.DIM_product" PRIMARY KEY(product_ID)
12 );

```

Script Output: Table DW\_DATA.DIM\_PRODUCT created.

```

1 --drop sequence DW_DATA.SQ_DIM_product;
2 --alter session set current_schema = DW_DATA;
3
4 CREATE SEQUENCE DW_DATA.SQ_DIM_product
5     START WITH      1
6     INCREMENT BY   1
7     NOCACHE
8     NOCYCLE;

```

Script Output: Sequence DW\_DATA.SQ\_DIM\_PRODUCT created.

- DW\_DATA.DIM\_promotion

```

1 alter session set current_schema = DW_DATA;
2
3 CREATE TABLE DW_DATA.DIM_promotion(
4     promotion_ID NUMBER(10),
5     TIME_ID DATE,
6     promotion_name VARCHAR2(50),
7     promotion_media_type VARCHAR2(50),
8     promotion_metric_amount DECIMAL (10,2),
9     department_name VARCHAR2(50),
10    promotion_price DECIMAL (10,2),
11    promotion_KPI VARCHAR2(30),
12    promotion_distinct_percent DECIMAL (10,2),
13    employee_salary_project DECIMAL (30,2),
14    CONSTRAINT "PK_T.DIM_promotion" PRIMARY KEY(promotion_ID)
15 );
16
17

```

Script Output: Task completed in 0.054 seconds

Table DW\_DATA.DIM\_PROMOTION created.

```

1 --drop sequence DW_DATA.SQ_DIM_promotion;
2 alter session set current_schema = DW_DATA;
3
4 CREATE SEQUENCE DW_DATA.SQ_DIM_promotion
5     START WITH      1
6     INCREMENT BY   1
7     NOCACHE
8     NOCYCLE;

```

Script Output: Task completed in 0.021 seconds

Session altered.

Sequence DW\_DATA.SQ\_DIM\_PROMOTION created.

- DW\_DATA.FCT\_business

```

1 ...sg t_FCT_business.sql pkg_etl_dw_FCT_business-def.sql pkg_etl_dw_FCT_business-impl.sql
2 SQL Worksheet | History
3 Worksheet | Query Builder
4
5 CREATE TABLE FCT_business(
6     Business_Fact_ID NUMBER(10),
7     TIME_ID DATE,
8     customer_ID NUMBER(10),
9     employee_ID NUMBER(10),
10    agency_ID NUMBER(10),
11    gen_period_ID NUMBER(10),
12    product_ID NUMBER(10),
13    promotion_ID NUMBER(10),
14    gross_profit_dollar_amount DECIMAL (30,2),
15    net_profit_dollar_amount DECIMAL (30,2),
16    gross_revenue_dollar_amount DECIMAL (30,2),
17    net_revenue_dollar_amount DECIMAL (30,2),
18    gross_cost_dollar_amount DECIMAL (30,2),
19    net_cost_dollar_amount DECIMAL (30,2),
20    gross_salary_employee_dollar_amount DECIMAL (30,2),
21    net_salary_employee_dollar_amount DECIMAL (30,2),
22    gross_profit_margin_percent DECIMAL (10,2),
23    CONSTRAINT "PK_T.FCT_business" PRIMARY KEY(Business_Fact_ID)
24 )
25 PARTITION BY RANGE (TIME_ID)
26     subpartition by hash(agency_ID) subpartitions 4
27 (
28     PARTITION QUARTER_1 VALUES LESS THAN(TO_DATE('01/04/2022', 'DD/MM/YYYY'))
29     ( subpartition QUARTER_1_sub_1,
30       subpartition QUARTER_1_sub_2,
31       subpartition QUARTER_1_sub_3,

```

Script Output: Task completed in 0.075 seconds

Table FCT\_BUSINESS created.

The screenshot shows two sessions in Oracle SQL Developer.

**Session 1 (Top):**

```

55 );
56 |
57 ALTER TABLE FCT_business
58 ADD CONSTRAINT "FK_FCT_business_DIM_customer" FOREIGN KEY (customer_ID )REFERENCES dim_customer (customer_ID);
59 |
60 ALTER TABLE FCT_business
61 ADD CONSTRAINT "FK_FCT_business_DIM_employee" FOREIGN KEY (employee_ID )REFERENCES dim_employee (employee_ID);
62 |
63 ALTER TABLE FCT_business
64 ADD CONSTRAINT "FK_FCT_business_DIM_agency" FOREIGN KEY (agency_ID )REFERENCES dim_agency (agency_ID);
65 |
66 ALTER TABLE FCT_business
67 ADD CONSTRAINT "FK_FCT_business_DIM_gen_period" FOREIGN KEY (gen_period_ID )REFERENCES dim_gen_period (gen_period_ID);
68 |
69 ALTER TABLE FCT_business
70 ADD CONSTRAINT "FK_FCT_business_DIM_product" FOREIGN KEY (product_ID )REFERENCES dim_product (product_ID);
71 |
72 ALTER TABLE FCT_business
73 ADD CONSTRAINT "FK_FCT_business_DIM_promotion" FOREIGN KEY (promotion_ID)REFERENCES dim_promotion (promotion_ID);

```

**Session 2 (Bottom):**

```

Table FCT_BUSINESS created.

Table FCT_BUSINESS altered.

Table FCT_BUSINESS altered.

Table FCT_BUSINESS altered.

```

**Session 3 (Top):**

```

1 --drop sequence DW_DATA.SQ_FCT_business;
2 --alter session set current_schema = DW_DATA;
3 |
4 CREATE SEQUENCE DW_DATA.SQ_FCT_business
5     START WITH      1
6     INCREMENT BY   1
7     NOCACHE
8     NOCYCLE;

```

**Session 4 (Bottom):**

```

Session altered.

Sequence DW_DATA.SQ_FCT_BUSINESS created.

```

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

The screenshot shows a session in Oracle SQL Developer.

```

1 alter session set current_schema = DW_DATA;
2 |
3 CREATE OR REPLACE PACKAGE pkg_etl_dw_customer
4 AS
5     PROCEDURE load_dw_customer;
6 END pkg_etl_dw_customer;
7 /

```

**Script Output:**

```

Session altered.

Package PKG_ETL_DW_CUSTOMER compiled

```

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.

SQL Worksheet History

Worksheet | Query Builder

```

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

```

Script Output X | Task completed in 0.036 seconds

Session altered.

Grant succeeded.

SQL Worksheet History

Worksheet | Query Builder

```

3
4 alter session set current_schema = DW_DATA;
5
6 CREATE OR REPLACE PACKAGE body pkg_etl_dw_customer
7 AS
8 PROCEDURE load_dw_customer
9 AS
10 BEGIN
11 DECLARE
12     TYPE CURSOR_VARCHAR IS TABLE OF varchar2(50);
13     TYPE CURSOR_NUMBER IS TABLE OF number(10);
14
15     TYPE BIG_CURSOR IS REF CURSOR;
16
17     ALL_INF BIG_CURSOR;
18
19     customer_name CURSOR_VARCHAR;
20     brand_name CURSOR_VARCHAR;
21     customer_address CURSOR_VARCHAR;
22     customer_city CURSOR_VARCHAR;
23     customer_country CURSOR_VARCHAR;
24     customer_email CURSOR_VARCHAR;
25     customer_office_phone CURSOR_VARCHAR;
26     customer_mobile_phone CURSOR_VARCHAR;
27     customer_name_stage CURSOR_VARCHAR;
28     customer_ID CURSOR_NUMBER;
29

```

Script Output X | Task completed in 0.021 seconds

Session altered.

Package Body PKG\_ETL\_DW\_CUSTOMER compiled

Let's select data from the DW\_DATA.DIM\_customer table on DW layer.

108 alter session set current\_schema = DW\_DATA;
109 alter user DW\_DATA QUOTA UNLIMITED ON TS\_DW\_DATA\_01;
110
111 EXEC pkg\_etl\_dw\_customer.load\_dw\_customer;

Script Output X | Task completed in 0.042 seconds

Session altered.

User DW\_DATA altered.

PL/SQL procedure successfully completed.

112 | SELECT \* FROM DW\_DATA.DIM\_customer;

Script Output X | Query Result X | 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

```

1 alter session set current_schema = DW_DATA;
2
3 CREATE OR REPLACE PACKAGE pkg_etl_dw_employee
4 AS
5     PROCEDURE load_dw_employee;
6 END pkg_etl_dw_employee;
7 /

```

Session altered.

Package PKG\_ETL\_DW\_EMPLOYEE compiled

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.

```

1 alter session set current_schema = DW_CL;
2
3 GRANT SELECT ON DW_CL.cls_t_employee TO DW_DATA;
4
5

```

Session altered.

Grant succeeded.

```

4 alter session set current_schema = DW_DATA;
5
6 CREATE OR REPLACE PACKAGE body pkg_etl_dw_employee
7 AS
8 PROCEDURE load_dw_employee
9 AS
10 BEGIN
11   DECLARE
12     TYPE CURSOR_VARCHAR IS TABLE OF varchar2(50);
13     TYPE CURSOR_NUMBER IS TABLE OF number(10);
14     TYPE CURSOR_DATE IS TABLE OF DATE;
15
16     TYPE BIG_CURSOR IS REF CURSOR;
17
18     ALL_INF BIG_CURSOR;
19
20     employee_passport_ID CURSOR_VARCHAR;
21     employee_first_name CURSOR_VARCHAR;
22     employee_last_name CURSOR_VARCHAR;
23     employee_position CURSOR_VARCHAR;
24     employee_email CURSOR_VARCHAR;
25     employee_office_phone CURSOR_VARCHAR;
26     employee_mobile_phone CURSOR_VARCHAR;

```

Script Output X | Task completed in 0.047 seconds

Grant succeeded.

Session altered.

Package Body PKG\_ETL\_DW\_EMPLOYEE compiled

```

121
122 alter session set current_schema = DW_DATA;
123 alter user DW_DATA QUOTA UNLIMITED ON TS_DW_DATA_01;
124
125 EXEC pkg_etl_dw_employee.load_dw_employee;

```

Script Output X | Task completed in 0.082 seconds

Session altered.

User DW\_DATA altered.

PL/SQL procedure successfully completed.

126 SELECT \* FROM DW\_DATA.DIM\_employee;

Script Output X | Query Result X | 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	1288060624	Ronnie	Abramson	specialist	Ronnie.Abramson@gmail.com	386-578-822	859-813-747
2	2614315484	Sam	Delon	specialist	Sam.Delon@gmail.com	836-329-404	350-600-222
3	3774952075	Robbie	Joukowski	manager	Robbie.Joukowski@gmail.com	983-570-389	789-665-977
4	4238172456	Steph	Delon	specialist	Steph.Delon@gmail.com	146-448-721	378-828-227
5	5325208747	Billie	Barnes	manager	Billie.Barnes@gmail.com	605-614-814	496-176-799
6	6503351236	Billie	Abramson	manager	Billie.Abramson@gmail.com	480-979-995	444-766-180
7	7804347498	Steph	Enderson	manager	Steph.Enderson@gmail.com	175-223-607	293-852-767
8	8597819409	Frankie	Gauss	specialist	Frankie.Gauss@gmail.com	245-464-208	545-268-268
9	9108955631	Alex	Kendall	specialist	Alex.Kendall@gmail.com	241-324-308	261-571-691
10	102441119453	Frankie	Foster	manager	Frankie.Foster@gmail.com	550-677-649	806-772-115
11	11319638040	Nat	Campbell	specialist	Nat.Campbell@gmail.com	204-587-181	561-422-477
12	12830429493	Frankie	Delon	manager	Frankie.Delon@gmail.com	623-621-530	502-912-303
13	13108955631	Alex	Kendall	manager	Alex.Kendall@gmail.com	241-324-308	261-571-691
14	14288060624	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

```

SQL Worksheet | History
Worksheet | Query Builder
alter session set current_schema = DW_DATA;
CREATE OR REPLACE PACKAGE pkg_etl_dw_agency
AS
    PROCEDURE load_dw_agency;
END pkg_etl_dw_agency;
/

```

Script Output X

Session altered.

Package PKG\_ETL\_DW\_AGENCY compiled

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.

```

SQL Worksheet | History
Worksheet | Query Builder
alter session set current_schema = DW_CL;
GRANT SELECT ON DW_CL.cls_t_agency TO DW_DATA;

```

Script Output X

Session altered.

Grant succeeded.

```

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;

```

Session altered.

Package Body PKG\_ETL\_DW\_AGENCY compiled

```

alter session set current_schema = DW_DATA;
alter user DW_DATA QUOTA UNLIMITED ON TS_DW_DATA_01;
EXEC pkg_etl_dw_agency.load_dw_agency;

```

Session altered.

User DW\_DATA altered.

PL/SQL procedure successfully completed.

### 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_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

```

alter session set current_schema = DW_DATA;
CREATE OR REPLACE PACKAGE pkg_etl_dw_promotion
AS
    PROCEDURE load_dw_promotion;
END pkg_etl_dw_promotion;
/

```

Script Output X | Task completed in 0.067 seconds

Session altered.

Package PKG\_ETL\_DW\_PROMOTION compiled

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.

```

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

```

Script Output X | Task completed in 0.061 seconds

Session altered.

Grant succeeded.

The screenshot shows two panels of Oracle SQL Developer. The top panel is a 'Worksheet' showing the PL/SQL code for creating a package and its body. The bottom panel is a 'Script Output' window showing the results of the execution.

```

128 alter session set current_schema = DW_DATA;
129 |
130 CREATE OR REPLACE PACKAGE body pkg_etl_dw_promotion
131 AS
132 PROCEDURE load_dw_promotion
133 AS
134 BEGIN
135     DECLARE
136         TYPE CURSOR_VARCHAR IS TABLE OF varchar2(50);
137         TYPE CURSOR_DECIMAL IS TABLE OF DECIMAL(30,2);
138         TYPE CURSOR_DATE IS TABLE OF DATE;
139         TYPE CURSOR_NUMBER IS TABLE OF number(10);
140
141         TYPE BIG_CURSOR IS REF CURSOR;
142
143         ALL_INF BIG_CURSOR;
144
145         TIME_ID CURSOR_DATE;
146         promotion_name CURSOR_VARCHAR;
147         promotion_media_type CURSOR_VARCHAR;
148         promotion_metric_amount CURSOR_DECIMAL;
149         department_name CURSOR_VARCHAR;
150         promotion_price CURSOR_DECIMAL;
151         promotion_KPI CURSOR_VARCHAR;
152         promotion_distinct_percent CURSOR_DECIMAL;
153         employee_salary_project CURSOR_DECIMAL;
154
155     END;
156
157     alter session set current_schema = DW_DATA;
158     alter user DW_DATA QUOTA UNLIMITED ON TS_DW_DATA_01;
159
160     EXEC pkg_etl_dw_promotion.load_dw_promotion;

```

Session altered.  
Package Body PKG\_ETL\_DW\_PROMOTION compiled

Session altered.  
User DW\_DATA altered.  
PL/SQL procedure successfully completed.

The screenshot shows a 'Query Result' window displaying the results of a SELECT query on the 'DIM\_promotion' table.

PROMOTION_ID	TIME_ID	PROMOTION_NAME	PROMOTION_MEDIA_TYPE	PROMOTION_METRIC_AMOUNT	DEPARTMENT_NAME	PROMOTION_PRICE	PROMOTION_KPI
1	1 01-JAN-22 5	SAMSUNG_Refrigerators_TV	TV		82 Media Planning and Buying	683 CPL	
2	2 01-JAN-22 26	SAMSUNG_Ovens_TV	TV		99 Media Planning and Buying	198 CPL	
3	3 01-JAN-22 49	SAMSUNG_Washing_machines_TV	TV		40 Media Planning and Buying	164 CPA	
4	4 01-JAN-22 62	SAMSUNG_Diswashers_TV	TV		37 Media Planning and Buying	631 CPA	
5	5 01-JAN-22 83	SAMSUNG_Microwave_ovens_TV	TV		46 Media Planning and Buying	643 CPA	
6	6 01-JAN-22 105	SAMSUNG_Vacuum_Cleaners_TV	TV		52 Media Planning and Buying	150 CPA	
7	7 01-JAN-22 128	SAMSUNG_TVs_TV	TV		44 Media Planning and Buying	624 CPA	
8	8 01-JAN-22 148	SAMSUNG_Smartphones_TV	TV		96 Media Planning and Buying	870 CPA	
9	9 01-JAN-22 166	SAMSUNG_Tablets_TV	TV		66 Media Planning and Buying	346 CPL	
10	10 01-JAN-22 174	SAMSUNG_Tablets_TV	TV		71 Media Planning and Buying	178 CPL	

### 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](#)

```

1 alter session set current_schema = DW_DATA;
2
3 CREATE OR REPLACE PACKAGE pkg_etl_dw_product
4 AS
5     PROCEDURE load_dw_product;
6 END pkg_etl_dw_product;
7 /

```

Session altered.

Package PKG\_ETL\_DW\_PRODUCT compiled

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.

```

1 alter session set current_schema = DW_CL;
2 GRANT SELECT ON DW_CL.cls_t_product TO DW_DATA;
3

```

Session altered.

Grant succeeded.

```

1 alter session set current_schema = DW_DATA;
2 CREATE OR REPLACE PACKAGE body pkg_etl_dw_product
3 AS
4 PROCEDURE load_dw_product
5 AS
6 BEGIN
7 MERGE INTO DW_DATA.dim_product A
8 USING ( SELECT brand_name, product_name, category_name, subcategory_name FROM DW_CL.cls_t_product) B
9 ON (a.brand_name = b.brand_name AND a.product_name = b.product_name)
10 WHEN MATCHED THEN
11 UPDATE SET a.category_name = b.category_name, a.subcategory_name = b.subcategory_name
12 WHEN NOT MATCHED THEN
13 INSERT (a.product_ID, a.brand_name, a.product_name, a.category_name, a.subcategory_name)
14 VALUES (DW_DATA.SQ_DIM_product.NEXTVAL, b.brand_name, b.product_name, b.category_name, b.subcategory_name);
15 COMMIT;
16 END load_dw_product;
17 END pkg_etl_dw_product;
18

```

```

Session altered.

Grant succeeded.

Session altered.

Package Body PKG_ETL_DW_PRODUCT compiled

```

```

24 alter session set current_schema = DW_DATA;
25 alter user DW_DATA QUOTA UNLIMITED ON TS_DW_DATA_01;
26
27 EXEC pkg_etl_dw_product.load_dw_product;

```

```

Session altered.

Package Body PKG_ETL_DW_PRODUCT compiled

Session altered.

User DW_DATA altered.

PL/SQL procedure successfully completed.

```

```

28 SELECT * FROM DW_DATA.DIM_product;
29
30

```

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

The screenshot shows the Oracle SQL Developer interface with three tabs open in the top bar: '...DB', 'ds\_t\_gen\_period.sql', and 'sa\_promotions\_generate.sql'. The current tab is 'pkg\_etl\_dw\_gen\_period-def.sql'. The code in the worksheet is:

```

1 alter session set current_schema = DW_DATA;
2
3 CREATE OR REPLACE PACKAGE pkg_etl_dw_gen_period
4 AS
5   PROCEDURE load_dw_gen_period;
6 END pkg_etl_dw_gen_period;
7 /

```

The script output window below shows the results of the compilation:

```

Session altered.

Package PKG_ETL_DL_GEN_PERIOD compiled

```

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.

The screenshot shows the Oracle SQL Developer interface with four tabs open in the top bar: '...DB', 'ds\_t\_gen\_period.sql', 'sa\_promotions\_generate.sql', and 'pkg\_etl\_dw\_gen\_period-def.sql'. The current tab is 'pkg\_etl\_dw\_gen\_period-impl.sql'. The code in the worksheet is:

```

1 alter session set current_schema = DW_CL;
2 GRANT SELECT ON DW_CL.cls_t_DIM_gen_period TO DW_DATA;
3

```

The script output window shows the grant succeeded:

```

Session altered.

Grant succeeded.

```

The next part of the screenshot shows the creation of the package body 'PKG\_ETL\_DL\_GEN\_PERIOD' in the 't\_DIM\_gen\_pe' tab. The code is:

```

1 --> alter session set current_schema = DW_DATA;
2 alter session set current_schema = DW_DATA;
3
4 CREATE OR REPLACE PACKAGE BODY pkg_etl_dw_gen_period
5 AS
6   PROCEDURE load_dw_gen_period
7   AS
8     BEGIN
9       MERGE INTO DW_DATA.DIM_gen_period A
10      USING ( SELECT promotion_name, promotion_start, promotion_end FROM DW_CL.cls_t_DIM_gen_period) B
11        ON (a.promotion_name = b.promotion_name)
12        WHEN MATCHED THEN
13          UPDATE SET a.promotion_start = b.promotion_start, a.promotion_end = b.promotion_end
14        WHEN NOT MATCHED THEN
15          INSERT (a.gen_period_ID, a.promotion_name, a.promotion_start, a.promotion_end)
16            VALUES (DW_DATA.SQ_DIM_gen_period.NEXTVAL, b.promotion_name, b.promotion_start, b.promotion_end);
17
18        COMMIT;
19      END load_dw_gen_period;
20    END pkg_etl_dw_gen_period;
21
22

```

The script output window shows the package body compiled:

```

Session altered.

Grant succeeded.

Session altered.

Package Body PKG_ETL_DL_GEN_PERIOD compiled

```

The final part of the screenshot shows the execution of the package body in the 't\_DIM\_gen\_pe' tab. The code is:

```

1 --> alter session set current_schema = DW_DATA;
2 alter user DW_DATA QUOTA UNLIMITED ON TS_DL_DATA_01;
3
4 EXEC pkg_etl_dw_gen_period.load_dw_gen_period;
5

```

The script output window shows the procedure completed successfully:

```

Session altered.

User DW_DATA altered.

PL/SQL procedure successfully completed.

```

28 | SELECT \* FROM DW\_DATA.DIM\_gen\_period;

GEN_PERIOD_ID	PROMOTION_NAME	PROMOTION_START	PROMOTION_END
1	19 SAMSUNG_Refrigerators_TV	01-JAN-22	12-FEB-22
2	256 SAMSUNG_Washing_machines_TV	01-JAN-22	07-FEB-22
3	359 SAMSUNG_Washing_machines_TV	01-JAN-22	06-FEB-22
4	477 SAMSUNG_Microwave_ovens_TV	01-JAN-22	04-FEB-22
5	597 SAMSUNG_Vacuum_Cleaners_TV	01-JAN-22	30-JAN-22
6	6108 SAMSUNG_Smart_watches_TV	01-JAN-22	20-MAR-22
7	7114 SAMSUNG_Smart_watches_TV	01-JAN-22	13-MAR-22
8	8117 SAMSUNG_Smart_watches_TV	01-JAN-22	21-JAN-22
9	9127 SAMSUNG_TVs_TV	01-JAN-22	23-FEB-22
10	10142 SAMSUNG_Smartphones_TV	01-JAN-22	25-MAR-22
11	11175 SAMSUNG_Tablets_TV	01-JAN-22	26-JAN-22
12	12211 SAMSUNG_Cooking_ovens_press	01-JAN-22	11-MAR-22
13	13232 SAMSUNG_Washing_machines_press	01-JAN-22	13-JAN-22
14	14235 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.

1 alter session set current\_schema = DW\_DATA;  
2 |  
3 CREATE OR REPLACE PACKAGE pkg\_etl\_dw\_gen\_period  
4 AS  
5 | PROCEDURE load\_dw\_gen\_period;  
6 END pkg\_etl\_dw\_gen\_period;  
7 /

Session altered.

Package PKG\_ETL\_DW\_GEN\_PERIOD compiled

Session altered.

Package PKG\_ETL\_DW\_GEN\_PERIOD compiled

1 END pkg\_etl\_dw\_gen\_period;  
2 |  
3 |  
4 alter session set current\_schema = DW\_DATA;  
5 alter user DW\_DATA quota UNLIMITED on TS\_DW\_DATA\_01;  
6 |  
7 EXEC pkg\_etl\_dw\_gen\_period.load\_dw\_gen\_period;

Session altered.

User DW\_DATA altered.

PL/SQL procedure successfully completed.

Session altered.

Package Body PKG\_ETL\_DW\_GEN\_PERIOD compiled

Session altered.

User DW\_DATA altered.

PL/SQL procedure successfully completed.

28 | SELECT \* FROM DW\_DATA.DIM\_gen\_period;

29 |

30 |-----

Script Output X Query Result X

SQL | Fetched 50 rows in 0.007 seconds

	GEN_PERIOD_ID	PROMOTION_NAME	PROMOTION_START	PROMOTION_END
1	1	SAMSUNG_Refrigerators_TV	01-JAN-22	12-FEB-22
2	2	SAMSUNG_Washing machines_TV	01-JAN-22	07-FEB-22
3	3	SAMSUNG_Washing machines_TV	01-JAN-22	06-FEB-22
4	4	SAMSUNG_Microwave ovens_TV	01-JAN-22	04-FEB-22
5	5	SAMSUNG_Vacuum Cleaners_TV	01-JAN-22	30-JAN-22
6	6	SAMSUNG_Smart watches_TV	01-JAN-22	20-MAR-22
7	7	SAMSUNG_Smart watches_TV	01-JAN-22	13-MAR-22
8	8	SAMSUNG_Smart watches_TV	01-JAN-22	21-JAN-22
9	9	SAMSUNG_TVs_TV	01-JAN-22	23-FEB-22
10	10	SAMSUNG_Smartphones_TV	01-JAN-22	25-MAR-22
11	11	SAMSUNG_Tablets_TV	01-JAN-22	26-JAN-22
12	12	SAMSUNG_Cooking ovens_press	01-JAN-22	11-MAR-22
13	13	SAMSUNG_Washing machines_press	01-JAN-22	13-JAN-22
14	14	SAMSUNG_Washing machines_press	01-JAN-22	17-MAR-22
15	15	SAMSUNG_Diswashers_press	01-JAN-22	28-MAR-22
16	16	SAMSUNG_Diswashers_press	01-JAN-22	06-MAR-22
17	17	SAMSUNG_Diswashers_press	01-JAN-22	16-JAN-22
18	18	SAMSUNG_Microwave ovens_press	01-JAN-22	22-FEB-22
19	19	SAMSUNG_Microwave ovens_press	01-JAN-22	01-FEB-22
20	20	SAMSUNG_Vacuum Cleaners_press	01-JAN-22	27-JAN-22
21	21	SAMSUNG_Smart watches_press	01-JAN-22	21-FEB-22
22	22	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 x pkg\_etl\_dw\_FCT\_business-def.sql x

SQL Worksheet History

Worksheet Query Builder

```

1 alter session set current_schema = DW_DATA;
2 |
3 CREATE OR REPLACE PACKAGE pkg_etl_dw_transaction
4 AS
5   PROCEDURE load_dw_transaction;
6 END pkg_etl_dw_transaction;
7 /

```

Script Output X

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 x ds\_t\_transaction.sql x t\_FCT\_business.sql x pkg\_etl\_dw\_FCT\_business-def.sql x pkg\_etl\_dw\_FCT\_business-impl.sql x

SQL Worksheet History

Worksheet Query Builder

```

1 alter session set current_schema = DW_CL;
2 GRANT SELECT ON DW_CL.cls_t_transaction TO DW_DATA;
3

```

Script Output X

Task completed in 0.035 seconds

Session altered.

Grant succeeded.

...sql **pkg\_etl\_dw\_FCT\_business-def.sql** **pkg\_etl\_dw\_FCT\_business-impl.sql**

SQL Worksheet: History

Worksheet Query Builder

```
3
4 alter session set current_schema = DW_DATA;
5
6 CREATE OR REPLACE PACKAGE body pkg_etl_dw_transaction
7 AS
8   PROCEDURE load_dw_transaction
9   AS
10    BEGIN
11      DECLARE
12        TYPE CURSOR_VARCHAR IS TABLE OF varchar2(50);
13        TYPE CURSOR_DECIMAL IS TABLE OF DECIMAL(30,2);
14        TYPE CURSOR_DATE IS TABLE OF DATE;
15        TYPE CURSOR_NUMBER IS TABLE OF number(10);
16
17        TYPE BIG_CURSOR IS REF CURSOR;
18
19        ALL_INI BIG_CURSOR;
20
21        TIME_ID CURSOR_DATE;
22        customer_ID CURSOR_NUMBER;
23        employee_ID CURSOR_NUMBER;
24        agency_ID CURSOR_NUMBER;
25        gen_period_ID CURSOR_NUMBER;
```

Script Output X

Task completed in 0.118 seconds

Grant succeeded.

Session altered.

Package Body PKG\_ETL\_DW\_TRANSACTION compiled

```
162 alter session set current_schema = DW_DATA;
163 alter user DW_DATA QUOTA UNLIMITED ON TS_DW_DATA_01;
164
165 EXEC pkg_etl_dw_transaction.load_dw_transaction;
```

Script Output X

| Task completed in 132.542 seconds

Grant succeeded.

Session altered.

Package Body PKG\_ETL\_DW\_TRANSACTION compiled

SELECT * from DW_DATA.FCT_business ORDER BY 1;										
Script Output		Query Result								
Fetched 50 rows in 0.346 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_RU
1	1 02-JAN-22	1	63	16	211721	6	783	47.98	41.72	
2	2 02-JAN-22	1	111	3	265625	15	273136	-113.55	-95.42	
3	3 02-JAN-22	1	94	38	265630	16	84181	479.62	393.13	
4	4 02-JAN-22	2	170	1	265696	1	212226	409.63	362.5	
5	5 02-JAN-22	2	115	58	265723	14	190923	-396.62	-330.52	
6	6 03-JAN-22	1	13	34	307616	4	252000	295.81	235.14	
7	7 03-JAN-22	1	1	57	130619	9	252011	653.81	531.55	
8	8 01-JAN-22	1	185	23	259708	13	84070	1185.07	1039.53	
9	9 04-JAN-22	1	138	56	295257	2	252090	1798.64	1591.71	
10	10 05-JAN-22	2	150	58	25958	5	170240	175.58	146.32	
11	11 06-JAN-22	1	86	1	191010	6	84916	-74.24	-65.7	
12	12 06-JAN-22	2	78	12	156272	14	313	-157.24	-136.73	
13	13 07-JAN-22	1	112	10	156322	10	85362	24.92	20.94	

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_DECIMAL;
net_profit_dollar_amount CURSOR_DECIMAL;
gross_revenue_dollar_amount CURSOR_DECIMAL;
net_revenue_dollar_amount CURSOR_DECIMAL;
gross_cost_dollar_amount CURSOR_DECIMAL;
net_cost_dollar_amount CURSOR_DECIMAL;
gross_salary_employee_dollar_amount CURSOR_DECIMAL;
net_salary_employee_dollar_amount CURSOR_DECIMAL;
gross_profit_margin_percent CURSOR_DECIMAL;

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;

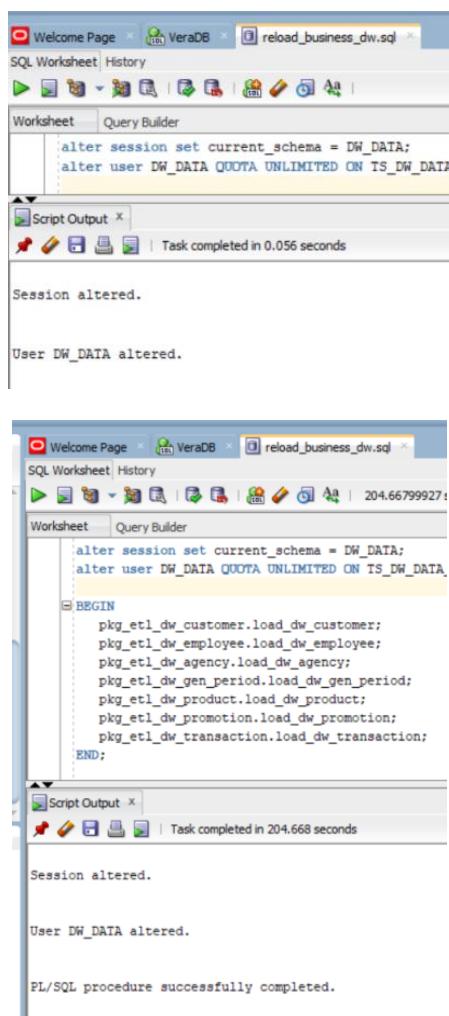
END IF;

END LOOP;

END;
END load_dw_transaction;
END pkg_etl_dw_transaction;

```

Run all procedures at the DW level.



```

alter session set current_schema = DW_DATA;
alter user DW_DATA QUOTA UNLIMITED ON TS_DW_DATA
Session altered.

User DW_DATA altered.

alter session set current_schema = DW_DATA;
alter user DW_DATA QUOTA UNLIMITED ON TS_DW_DATA;
BEGIN
    pkg_etl_dw_customer.load_dw_customer;
    pkg_etl_dw_employee.load_dw_employee;
    pkg_etl_dw_agency.load_dw_agency;
    pkg_etl_dw_gen_period.load_dw_gen_period;
    pkg_etl_dw_product.load_dw_product;
    pkg_etl_dw_promotion.load_dw_promotion;
    pkg_etl_dw_transaction.load_dw_transaction;
END;
Session altered.

User DW_DATA altered.

PL/SQL procedure successfully completed.

```

```

| EXEC pkg_etl_dw_customer.load_dw_customer;
| EXEC pkg_etl_dw_employee.load_dw_employee;
| EXEC pkg_etl_dw_agency.load_dw_agency;
| EXEC pkg_etl_dw_gen_period.load_dw_gen_period;
| EXEC pkg_etl_dw_product.load_dw_product;
| EXEC pkg_etl_dw_promotion.load_dw_promotion;
| EXEC pkg_etl_dw_transaction.load_dw_transaction;

-----[Output]-----
Script Output x | Task completed in 211.833 seconds

PL/SQL procedure successfully completed.

```

	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;		
1	147736 16-JAN-22	1	144	40	297629	15	150365	-203.63	-161.87
2	147739 02-JAN-22	1	60	55	56555	10	190898	284.05	263.74
3	147748 04-JAN-22	1	63	6	34129	10	49064	536.43	439.7
4	147753 06-JAN-22	1	76	27	307945	7	321252	225.64	188.03
5	147757 09-FEB-22	1	142	50	193484	16	47464	307	253.72
6	147771 12-FEB-22	1	85	50	251219	16	147895	1714.49	1416.93
7	147773 13-FEB-22	1	177	34	173486	2	26712	-237.66	-188.92
8	147777 13-FEB-22	2	48	40	46128	1	275672	-610	-484.9
9	147785 16-FEB-22	1	89	11	255537	13	148292	682.36	598.56