

U2M4.LW.Core PL/SQL

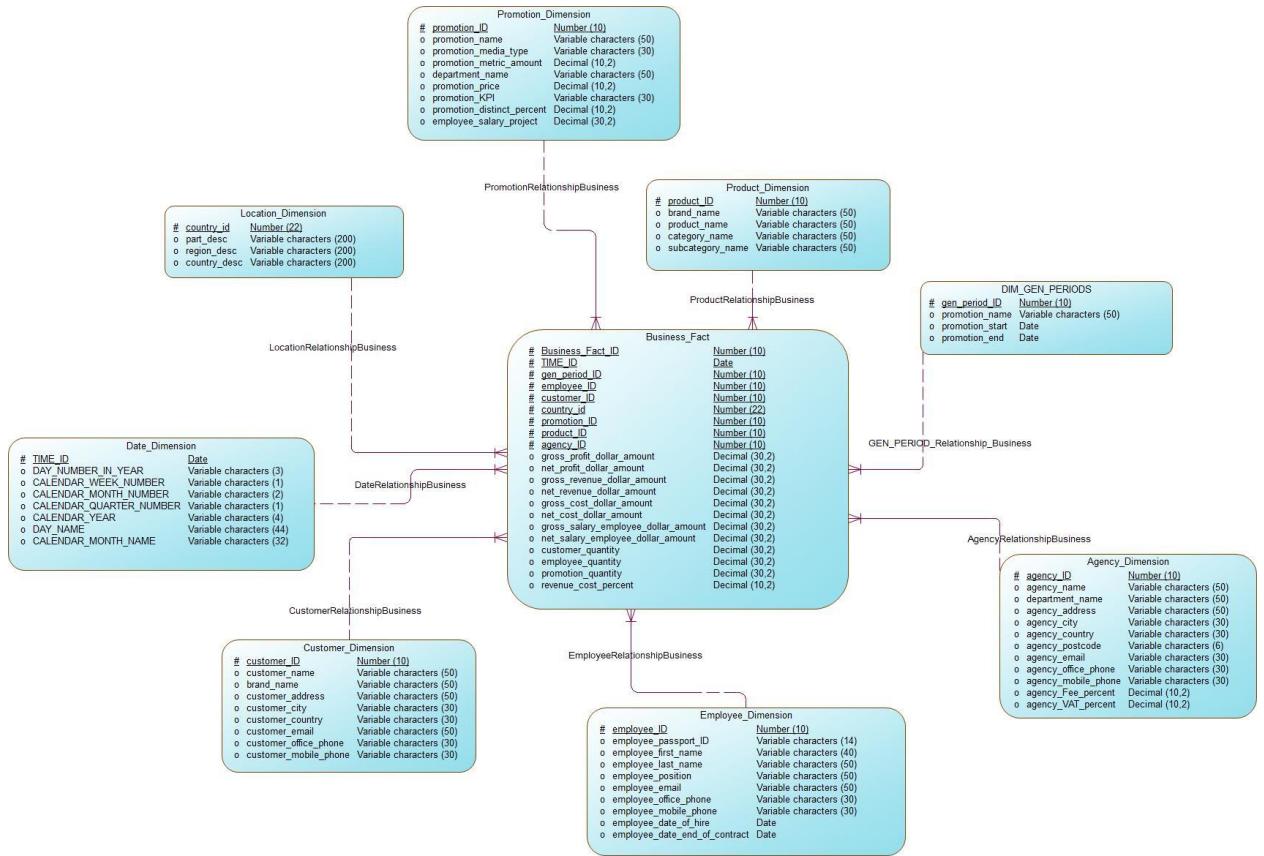
Shkrabatouskaya Vera

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

2. Business analyses tasks – Reports

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 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. | 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.

```

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 script. The bottom pane shows the resulting data from a query.

Script Output (Top Pane):

```
48| alter session set current_schema = DW_CL;
49| alter user DW_CL QUOTA UNLIMITED ON ts_dw_cl;
50|
51|
52| EXEC pkg_etl_cls_customer.load_cls_customer;
53| 
```

Session altered.
User DW_CL altered.
PL/SQL procedure successfully completed.

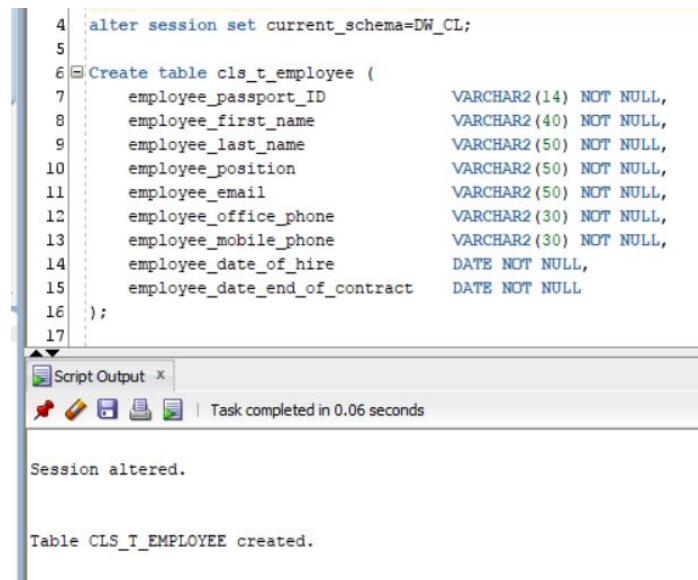
Query Result (Bottom Pane):

```
54| SELECT * FROM cls_t_customer;
55|
56| 
```

| CUSTOMER_NAME | BRAND_NAME | CUSTOMER_ADDRESS | CUSTOMER_CITY | CUSTOMER_COUNTRY | CUSTOMER_EMAIL | CUSTOMER_PHONE |
|--|------------|-----------------------------------|---------------|------------------|--------------------------------------|----------------|
| 1 Samsung Electronics | SAMSUNG | 85 Challenger Rd. Ridgefield Park | New Jersey | United States | www.facebook.com/SamsungUS | 1-201-229 |
| 2 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

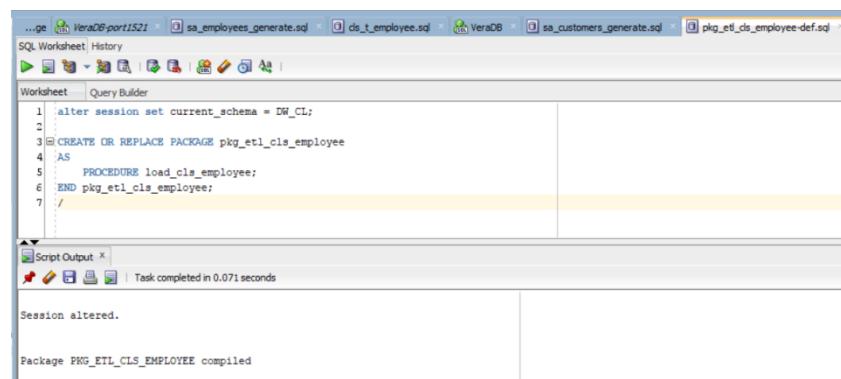
```

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 /

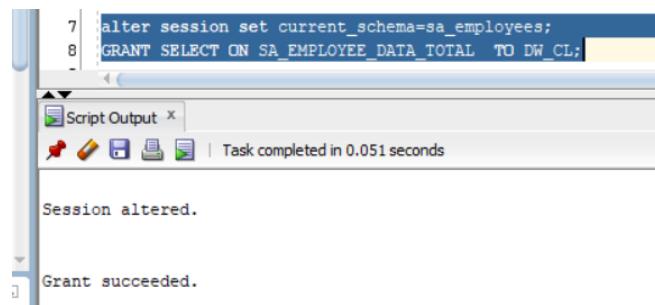
```

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)

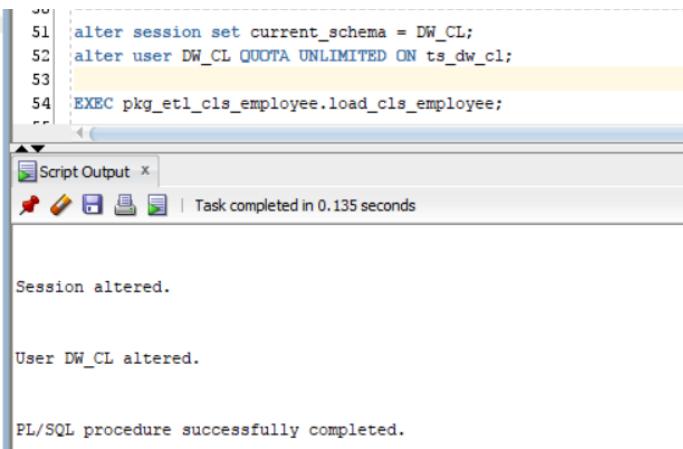
```

```

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;

```



```

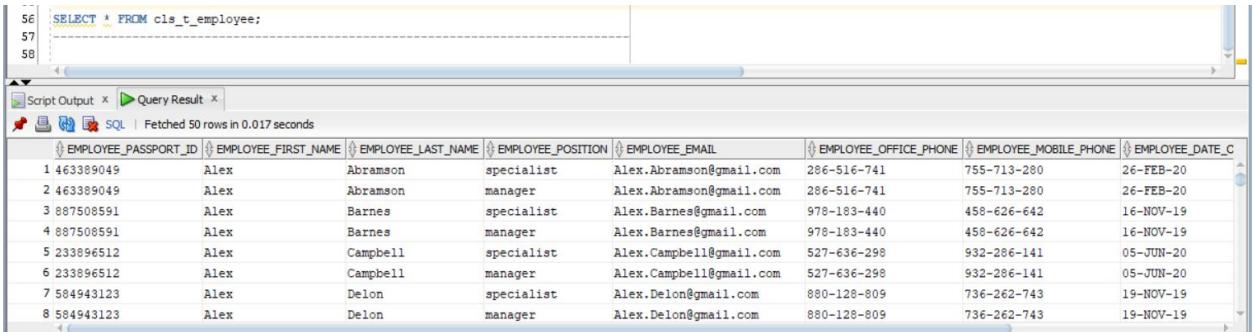
51 alter session set current_schema = DW_CL;
52 alter user DW_CL QUOTA UNLIMITED ON ts_dw_cl;
53
54 EXEC pkg_etl_cls_employee.load_cls_employee;
55
56
57
58

```

Session altered.

User DW_CL altered.

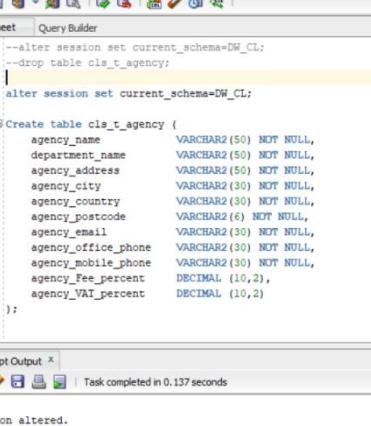
PL/SQL procedure successfully completed.



| 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 a Microsoft SQL Server Management Studio (SSMS) window. The title bar displays the connection name "Welcome Page" and the database "VeraDB". The tabs at the top include "sa_customers_generate.sq" and "ds_t_agency.sq". Below the tabs is a toolbar with icons for file operations, copy, paste, and search. The main area is divided into two panes: "Worksheet" on the left and "Query Builder" on the right. The "Worksheet" pane contains the following T-SQL code:

```
--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)
};

The "Query Builder" pane is visible on the right, showing the structure of the "cls_t_agency" table with columns: agency_name, department_name, agency_address, agency_city, agency_country, agency_postcode, agency_email, agency_office_phone, agency_mobile_phone, agency_fee_percent, and agency_VAT_percent.
```

At the bottom of the screen, there is a "Script Output" pane with a progress bar and the message "Task completed in 0.137 seconds". The status bar at the bottom of the window shows the message "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 Worksheet interface. The title bar has tabs for 'Welcome Page', 'VeraDB', 'sa_customers_generate.sql', 'cls_t_agency.sql', and 'plq_etl_cls_agency-def.sql'. The main area is titled 'Worksheet' and contains the following PL/SQL code:

```
1 alter session set current_schema = DW_CLS;
2
3 CREATE OR REPLACE PACKAGE pkg_etl_cls_agency
4 AS
5     PROCEDURE load_cls_agency;
6 END pkg_etl_cls_agency;
7 /
```

Below the code, a 'Script Output' window shows the message: '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 VeraDB SQL Worksheet interface. The top menu bar includes tabs for 'Welcome Page', 'VeraDB', 'sa_customers_generate.sql', and 'ds_t_agency.sql'. Below the menu is a toolbar with icons for file operations, database management, and code navigation. The main workspace is divided into two panes: 'Worksheet' and 'Query Builder'. The 'Worksheet' pane contains the following SQL script:

```
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 :(
```

Below the worksheet is a 'Script Output' pane which displays the results of the executed command:

Session altered.
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 |

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 |

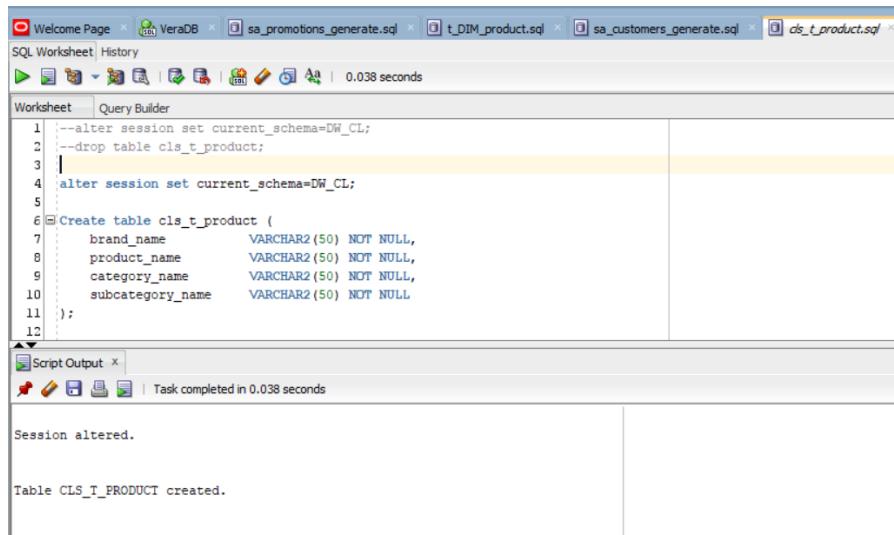
```

The results show 50 rows of data from the `cls_t_agency` table. The columns are:

| 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 60-62 | Helsinki | Finland | 00-177 | 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 being run 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 shows the results of the execution:

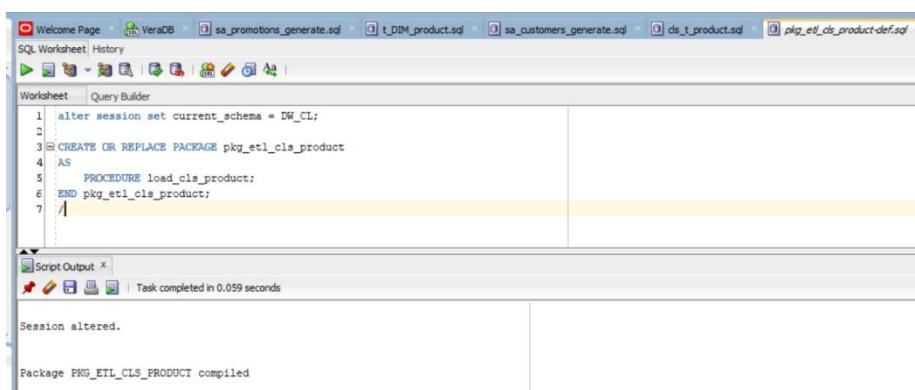
```

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 being run 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 shows the results of the execution:

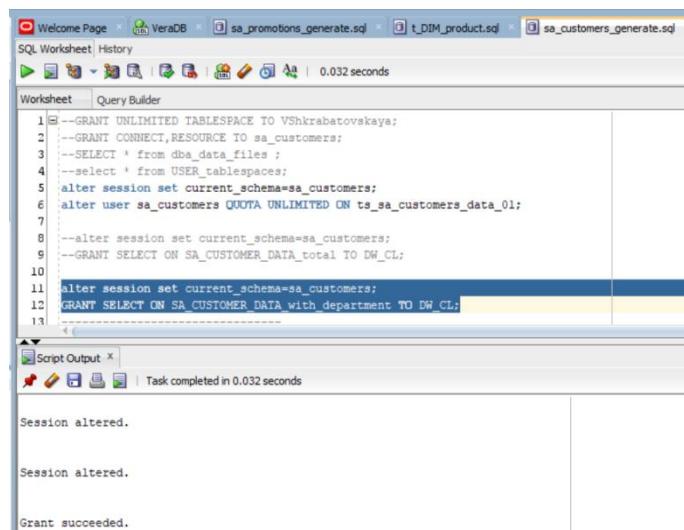
```

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 being run 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 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;

```

SQL Worksheet: History

```

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;

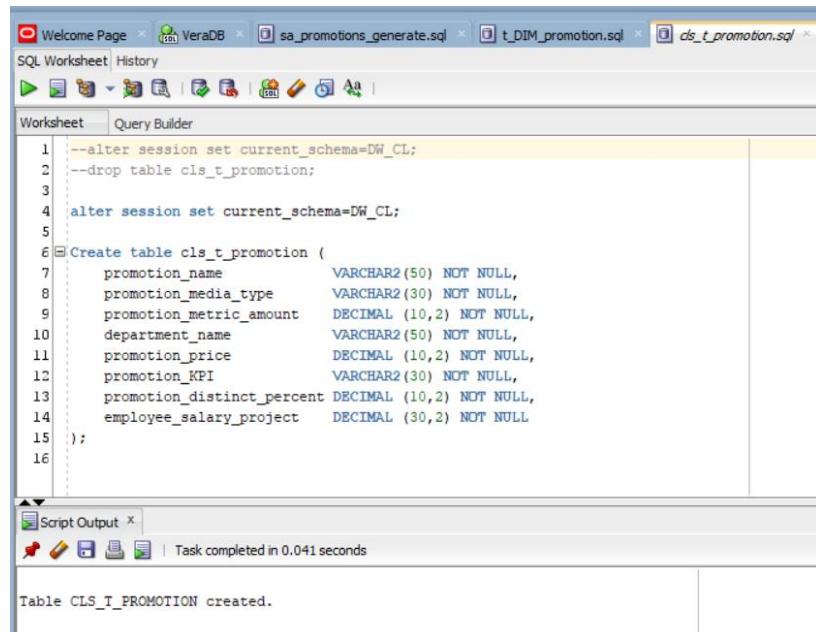
```

All Rows Fetched: 16 in 0.021 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 | OLED |
| 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 |
| 15 VISA | Benefit | Services | Retirement Savings |
| 16 VISA | Service | Services | Internet banks |

- `cls_t_promotion`

Create a `cls_t_promotion` 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:

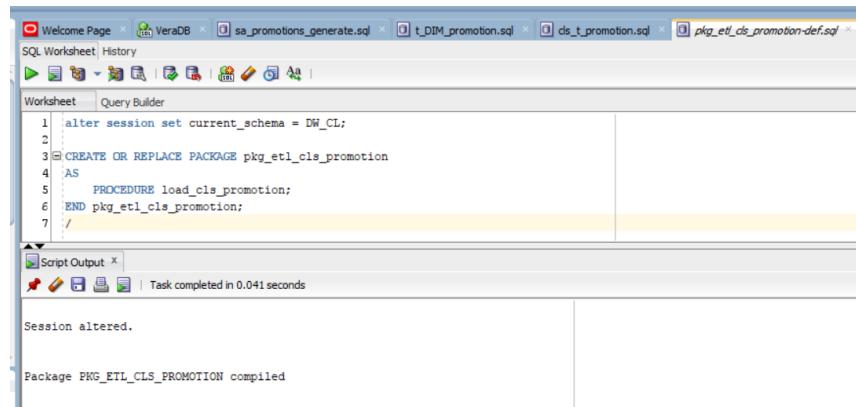
```

1 --alter session set current_schema=DW_CL;
2 --drop table cls_t_promotion;
3
4 alter session set current_schema=DW_CL;
5
6 Create table cls_t_promotion (
7     promotion_name          VARCHAR2(50) NOT NULL,
8     promotion_media_type    VARCHAR2(30) NOT NULL,
9     promotion_metric_amount DECIMAL (10,2) NOT NULL,
10    department_name         VARCHAR2(50) NOT NULL,
11    promotion_price         DECIMAL (10,2) NOT NULL,
12    promotion_KPI           VARCHAR2(30) NOT NULL,
13    promotion_distinct_percent DECIMAL (10,2) NOT NULL,
14    employee_salary_project DECIMAL (30,2) NOT NULL
15 );
16

```

The script output window below shows the message: "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`.



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

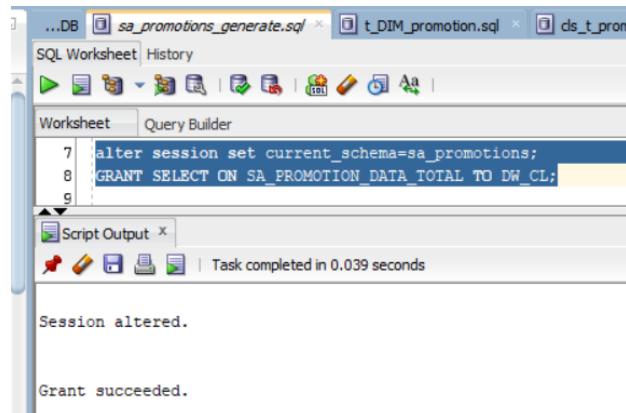
```

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

```

The script output window below shows the messages: "Session altered." and "Package PNG_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.



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 window below shows the messages: "Session altered." and "Grant succeeded."

The screenshot shows the Oracle SQL Worksheet interface. The 'Worksheet' tab is active, displaying a code editor with the following SQL script:

```

1 alter session set current_schema = DW_CLS;
2
3 CREATE OR REPLACE PACKAGE body pkg_etl_cls_promotion
4 AS
5 PROCEDURE load_cls_promotion
6   AS
7     CURSOR cursor_cls_promotion
8   IS
9     SELECT DISTINCT promotion_name, promotion_media_type, promotion_metric_amount, department_name, promotion_price, promotion_KPI, promotion_distinct_percent, employee_salary_project
10    FROM sa_promotions.SA_PROMOTION_DATA_TOTAL
11   WHERE promotion_name IS NOT NULL
12     AND promotion_media_type IS NOT NULL
13     AND promotion_metric_amount IS NOT NULL
14     AND department_name IS NOT NULL
15     AND promotion_price IS NOT NULL
16     AND promotion_KPI IS NOT NULL
17     AND promotion_distinct_percent IS NOT NULL
18     AND employee_salary_project IS NOT NULL
19   ;
20
21 BEGIN
22   EXECUTE IMMEDIATE 'TRUNCATE TABLE DW_CLS.cls_t_promotion';
23   FOR i IN cursor_cls_promotion LOOP
24     INSERT INTO DW_CLS.cls_t_promotion(
25       promotion_name,
26       promotion_media_type,
27       promotion_metric_amount,
28       department_name,
29       promotion_price,
30       promotion_KPI,
31       promotion_distinct_percent,
32       employee_salary_project)
33     VALUES (
34       i.promotion_name,
35       i.promotion_media_type,
36       i.promotion_metric_amount,
37       i.department_name,
38       i.promotion_price,
39       i.promotion_KPI,
40       i.promotion_distinct_percent,
41       i.employee_salary_project);
42   EXIT WHEN cursor_cls_promotion%NOTFOUND;
43   END LOOP;
44
45 COMMIT;
46 END load_cls_promotion;
47 END pkg_etl_cls_promotion;

```

The 'Script Output' pane at the bottom shows the results of the execution:

```

Session altered.

Package Body PKG_ETL_CLS_PROMOTION compiled

```

Code:

```

CREATE OR REPLACE PACKAGE body pkg_etl_cls_promotion
AS
PROCEDURE load_cls_promotion
AS
CURSOR cursor_cls_promotion
IS
  SELECT DISTINCT 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 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_CLS.cls_t_promotion';
  FOR i IN cursor_cls_promotion LOOP
    INSERT INTO DW_CLS.cls_t_promotion(
      promotion_name,
      promotion_media_type,
      promotion_metric_amount,
      department_name,
      promotion_price,
      promotion_KPI,
      promotion_distinct_percent,
      employee_salary_project)
    VALUES (
      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.

```

53 | SELECT * FROM cls_t_promotion;
54 |
55 |
56 |

```

Query Result | Fetched 50 rows in 0.006 seconds

| PROMOTION_NAME | PROMOTION_MEDIA_TYPE | PROMOTION_METRIC_AMOUNT | DEPARTMENT_NAME | PROMOTION_PRICE | PROMOTION_KPI | PROMOTION_DISTINCT_PERCE |
|--------------------------------------|----------------------|-------------------------|------------------------------|-----------------|---------------|--------------------------|
| 1 68_SAMSUNG_Dishwashers_TV | TV | | 3 Media Planning and Buying | 784 CPL | | |
| 2 73_SAMSUNG_Dishwashers_TV | TV | | 28 Media Planning and Buying | 849 CPL | | |
| 3 133_SAMSUNG_TVs_TV | TV | | 97 Media Planning and Buying | 493 CPL | | |
| 4 135_SAMSUNG_TVs_TV | TV | | 56 Media Planning and Buying | 286 CPL | | |
| 5 230_SAMSUNG_Washing machines_press | press | | 39 Media Planning and Buying | 688 CPL | | |
| 6 245_SAMSUNG_Dishwashers_press | press | | 85 Media Planning and Buying | 653 CPL | | |
| 7 250_SAMSUNG_Dishwashers_press | press | | 44 Media Planning and Buying | 316 CPL | | |
| 8 262_SAMSUNG_Microwave ovens_press | press | | 82 Media Planning and Buying | 693 CPL | | |
| 9 266_SAMSUNG_Microwave ovens_press | press | | 19 Media Planning and Buying | 843 CPL | | |
| 10 269_SAMSUNG_Microwave ovens_press | press | | 74 Media Planning and Buying | 813 CPL | | |
| 11 283_SAMSUNG_Vacuum Cleaners_press | press | | 48 Media Planning and Buying | 756 CPL | | |
| 12 288_SAMSUNG_Smart watches_press | press | | 71 Media Planning and Buying | 533 CPL | | |
| 13 289_SAMSUNG_Smart watches_press | press | | 97 Media Planning and Buying | 154 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.

```

The screenshot shows an Oracle SQL Worksheet interface. The main area displays the PL/SQL code for creating a package body. The code defines a package named 'pkg_etl_cls_DIM_gen_period' with a single procedure 'load_cls_DIM_gen_period'. This procedure uses a cursor to select distinct promotion names, start, and end dates from the 'sa_promotions.SA_PROMOTION_DATA_TOTAL' table where promotion names are not null. It then truncates a table named 'DW_CL.cls_t_DIM_gen_period', loops through the cursor to insert each row into the table, and commits the changes. The script output window at the bottom shows the session was altered and the package body was compiled successfully.

```

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;

COMMIT;
END load_cls_DIM_gen_period;
END pkg_etl_cls_DIM_gen_period;

```

```

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.

39 | SELECT * FROM cls_t_DIM_gen_period;
40 |
41 |

[Script Output X] | [Query Result X] | Fetched 50 rows in 0.017 seconds
SQL | PROMOTION_NAME | PROMOTION_START | PROMOTION_END
---|---|---|---
1 62_SAMSUNG_Dishwashers_TV | 01-JAN-22 | 24-MAR-22
2 64_SAMSUNG_Dishwashers_TV | 01-JAN-22 | 10-JAN-22
3 73_SAMSUNG_Dishwashers_TV | 01-JAN-22 | 11-JAN-22
4 88_SAMSUNG_Microwave ovens_TV | 01-JAN-22 | 03-FEB-22
5 109_SAMSUNG_Smart watches_TV | 01-JAN-22 | 09-MAR-22
6 118_SAMSUNG_Smart watches_TV | 01-JAN-22 | 25-MAR-22
7 137_SAMSUNG_Smartphones_TV | 01-JAN-22 | 28-JAN-22
8 146_SAMSUNG_Smartphones_TV | 01-JAN-22 | 10-JAN-22
9 173_SAMSUNG_Tablets_TV | 01-JAN-22 | 09-JAN-22
10 180_SAMSUNG_Tablets_TV | 01-JAN-22 | 19-FEB-22
11 205_SAMSUNG_Ovens_press | 01-JAN-22 | 16-FEB-22
12 232_SAMSUNG_Washing machines_press | 01-JAN-22 | 20-MAR-22
13 234_SAMSUNG_Washing machines_press | 01-JAN-22 | 27-MAR-22
14 254_SAMSUNG_Dishwashers_press | 01-JAN-22 | 15-MAR-22
15 264_SAMSUNG_Microwave ovens_press | 01-JAN-22 | 06-FEB-22

```

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_period;
9 |   pkg_etl_cls_product.load_cls_product;
10 |  pkg_etl_cls_promotion.load_cls_promotion;
11 |
12 | END;
13 |

[Script Output X] | Task completed in 30.684 seconds
Session altered.

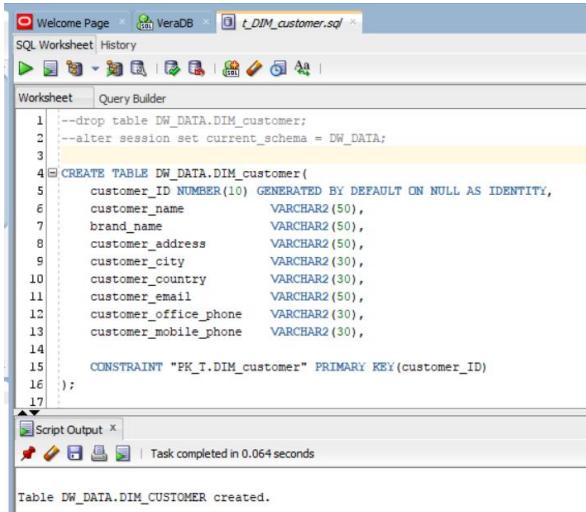
User DW_CL altered.

PL/SQL procedure successfully completed.

```

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

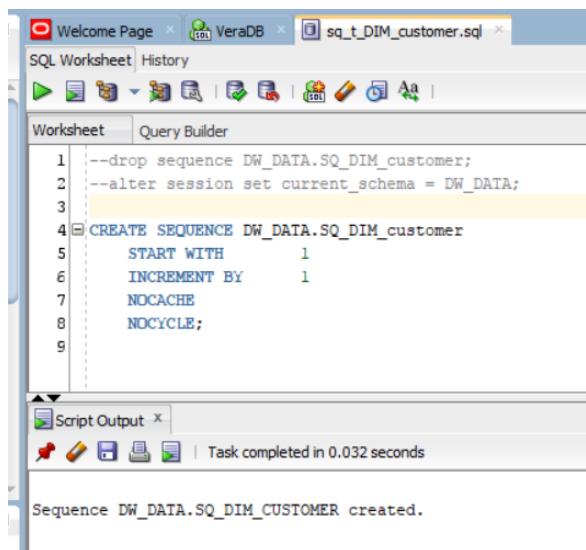


```

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) GENERATED BY DEFAULT ON NULL AS IDENTITY,
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 );
17

```

Table DW_DATA.DIM_CUSTOMER created.



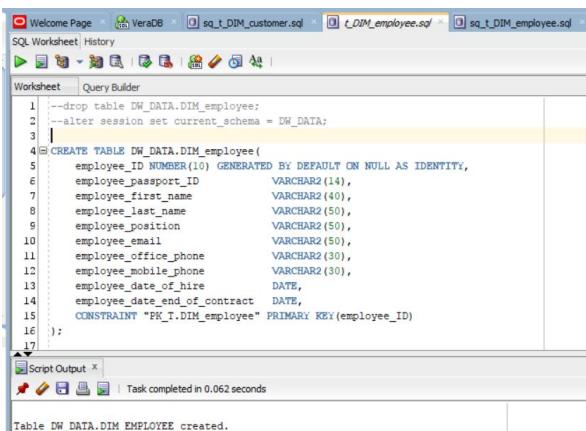
```

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

```

Sequence DW_DATA.SQ_DIM_CUSTOMER created.

- DW_DATA.DIM_employee



```

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) GENERATED BY DEFAULT ON NULL AS IDENTITY,
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_T.DIM_employee" PRIMARY KEY(employee_ID)
16 );
17

```

Table DW_DATA.DIM_EMPLOYEE created.

SQL Worksheet History

```

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

SQL Worksheet History

```

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) GENERATED BY DEFAULT ON NULL AS IDENTITY,
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(10),
14  agency_mobile_phone  VARCHAR2(10),
15  agency_fee_percent  DECIMAL(10,2),
16  agency_VAT_percent  DECIMAL(10,2),
17
18  CONSTRAINT "PK_T.DIM_agency" PRIMARY KEY(agency_ID)
19 );

```

Script Output X | Task completed in 0.046 seconds

Table DW_DATA.DIM_AGENCY created.

SQL Worksheet History

```

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

SQL Worksheet History

```

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) GENERATED BY DEFAULT ON NULL AS IDENTITY,
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 X | Task completed in 0.047 seconds

Table DW_DATA.DIM_GEN_PERIOD created.

SQL Worksheet History

```

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 X | Task completed in 0.027 seconds

Sequence DW_DATA.SQ_DIM_GEN_PERIOD created.

- DW_DATA.DIM_product

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

```

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) GENERATED BY DEFAULT ON NULL AS IDENTITY,
6     brand_name        VARCHAR2(50),
7     product_name      VARCHAR2(50),
8     category_name    VARCHAR2(50),
9     subcategory_name VARCHAR2(50),
10    CONSTRAINT "PK_T.DIM_product" PRIMARY KEY(product_ID)
11 );
12

```

The script output pane below shows the message: "Table DW_DATA.DIM_PRODUCT created." and "Task completed in 0.041 seconds".

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

```

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;

```

The script output pane below shows the message: "Session altered." and "Sequence DW_DATA.SQ_DIM_PRODUCT created." and "Task completed in 0.028 seconds".

- DW_DATA.DIM_promotion

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

```

1 --drop table DW_DATA.DIM_promotion;
2 --alter session set current_schema = DW_DATA;
3
4 CREATE TABLE DW_DATA.DIM_promotion(
5     promotion_ID NUMBER(10) GENERATED BY DEFAULT ON NULL AS IDENTITY,
6     promotion_name        VARCHAR2(50),
7     promotion_media_type VARCHAR2(30),
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
15    CONSTRAINT "PK_T.DIM_promotion" PRIMARY KEY(promotion_ID)
16 );

```

The script output pane below shows the message: "Table DW_DATA.DIM_PROMOTION created." and "Task completed in 0.047 seconds".

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

```

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;

```

The script output pane below shows the message: "Session altered." and "Sequence DW_DATA.SQ_DIM_PROMOTION created." and "Task completed in 0.021 seconds".

Moving the tables from the cleansing layer to the DW layer will be done later.