

U2M1.LW.Core SQL

Shkrabatouskaya Vera

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

2. Data Warehouse Architecture – Storage Layers

2.1. Task 01: CREATE Storage Objects

Physical Objects according Solution Proposal were created on Module 8.

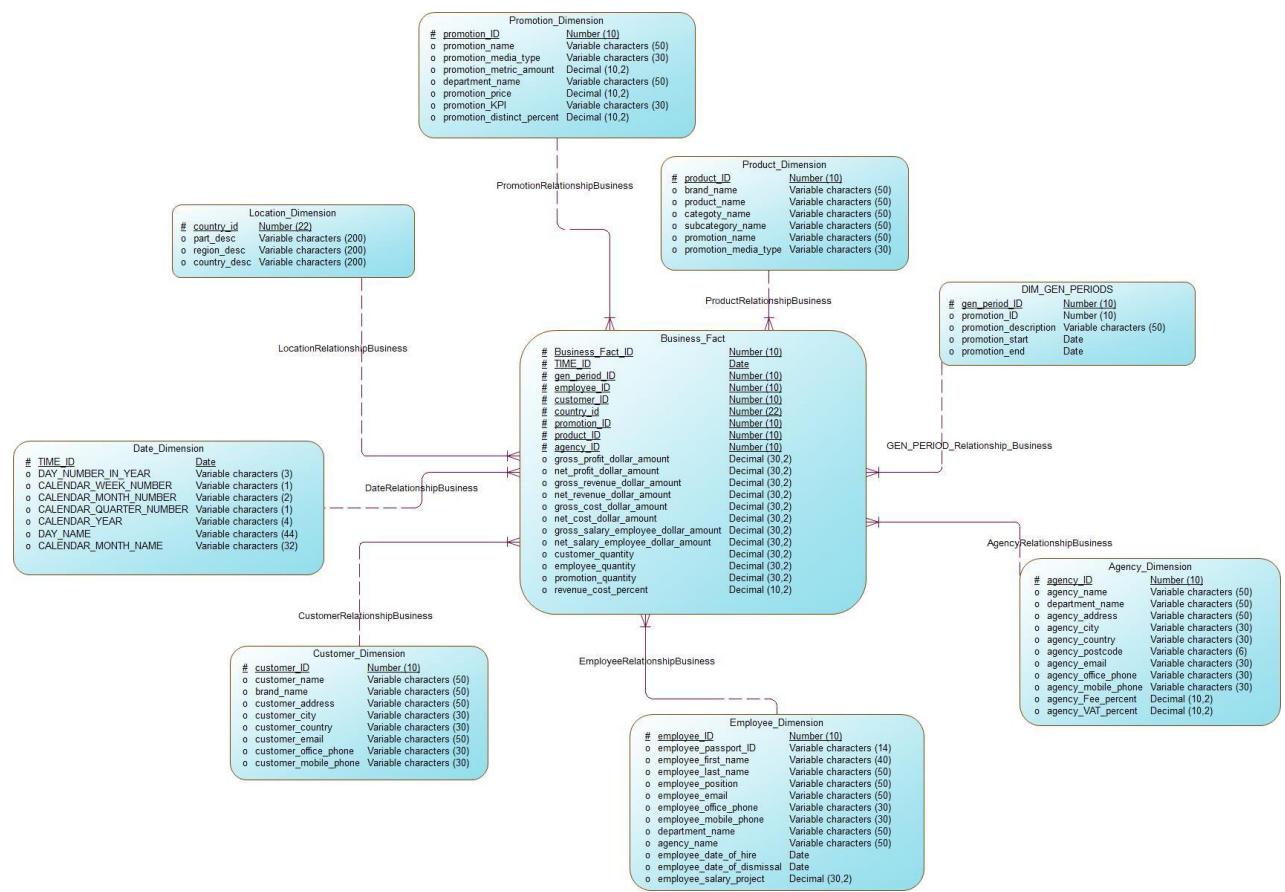
Name Conversation table

Level Type	Object Name	Tablespace	Desctiption
Storage level SA_*	SA_CUSTOMERS	ts_sa_customers_data_01	Loading from structured files. Contains Customer information.
	SA_EMPLOYEES	ts_sa_employees_data_01	Loading from structured files. Contains Employee information.
	SA_PROMOTIONS	ts_sa_promotions_data_01	Loading from structured files. Contains Promotion information.
DW - Cleansing Level	DW_CL	ts_dw_cl	Loading from a scene-level system. Contains information about preparation for subsequent use (cleaning).
DW – Level	DW_DATA	ts_dw_data_01	Loading data from cleansing tables. Contains clean information tending to the 3rd normal form ready to prepare a star schema.
DW– Prepare Star Cleansing Level	SAL_DW_CL	ts_dw_str_cls	Loading data from DW system. Contains views merging objects from DW level.

STAR - Cleansing	SAL_CL	ts_sal_cl	Loading data from DW_CL system. Contains views from previous level but clean any redundancy.
STAR – Level	DM_FCT_KPI	ts_sa_fct_kpi_01	Store information about fact KPI of ad promotions.
	DM_FCT_BUDGET	ts_sa_fct_budget_01	Store information about fact budget.
	DM_FCT_SALARY	ts_sa_fct_salary_01	Store information about fact salary of employees.
	DM_FCT_QUANTITY	ts_sa_fct_quantity_01	Store information about fact quantity of agency customers, employees and promotions.

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



2.2. Task 02: Generate Test Data in Storage Layers

Let's generate test data on Storage layers objects.

Object Name	Tablespace
SA_CUSTOMERS	ts_sa_customers_data_01

The screenshot shows a SQL Worksheet window with the following content:

```
--GRANT UNLIMITED TABLESPACE TO VShkrabatovskaya;
--SELECT * from dba_data_files ;
--select * from USER tablespaces;
alter session set current_schema=sa_customers;
alter user sa_customers QUOTA UNLIMITED ON ts_sa_customers_data_01;
```

Script Output: Task completed in 0.045 seconds

Session altered.

User SA_CUSTOMERS altered.

Create a table SA_CUSTOMER_DATA_c about the Samsung Electronics customer.

The screenshot shows a SQL Worksheet window with the following content:

```
48
49 CREATE TABLE SA_CUSTOMER_DATA_c
50 (
51     customer_name        VARCHAR2(50),
52     brand_name          VARCHAR2(50),
53     customer_address    VARCHAR2(50),
54     customer_city       VARCHAR2(30),
55     customer_country    VARCHAR2(30),
56     customer_email      VARCHAR2(50),
57     customer_office_phone VARCHAR2(30),
58     customer_mobile_phone VARCHAR2(30)
59 )
60 TABLESPACE ts_sa_customers_data_01;
```

Script Output: Task completed in 0.042 seconds

Table SA_CUSTOMER_DATA_C created.


```
62 INSERT INTO sa_customer_data_c (
63     customer_name,
64     brand_name,
65     customer_address,
66     customer_city,
67     customer_country,
68     customer_email,
69     customer_office_phone,
70     customer_mobile_phone
71 )
72 VALUES (
73     'Samsung Electronics',
74     'SAMSUNG',
75     '85 Challenger Rd. Ridgefield Park',
76     'New Jersey',
77     'United States',
78     'www.facebook.com/SamsungUS',
79     '1-201-229-4000',
80     '1-201-229-4029');
```

Script Output: Task completed in 0.053 seconds

1 row inserted.

```

82 select * from sa_customer_data_c;

```

Script Output X | Query Result X | SQL | All Rows Fetched: 1 in 0.024 seconds

CUSTOMER_NAME	BRAND_NAME	CUSTOMER_ADDRESS	CUSTOMER_CITY	CUSTOMER_COUNTRY	CUSTOMER_EMAIL	CUSTOMER_PHONE	CUSTOMER_MOBILE
1 Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	1-201-229-4029

Create a table SA_CUSTOMER_DATA_p about the Samsung Electronics products for an advertising campaign.

```

84 CREATE TABLE SA_CUSTOMER_DATA_P
85 (
86     product_name          VARCHAR2(50)
87 )
88 TABLESPACE ts_sa_customers_data_01;

```

Script Output X | Task completed in 0.048 seconds

Table SA_CUSTOMER_DATA_P created.

```

104 select * from sa_customer_data_p;

```

Script Output X | Query Result X | SQL | All Rows Fetched: 12 in 0.014 seconds

PRODUCT_NAME
1 Refrigerators
2 Ovens
3 Cooking ovens
4 Washing machines
5 Dishwashers
6 Microwave ovens
7 Vacuum Cleaners
8 Smart watches
9 TVs
10 Smartphones
11 Headphones
12 Tablets

Cross join of SA_CUSTOMER_DATA_c with SA_CUSTOMER_DATA_p.

```

106 select sa_customer_data_c.*, sa_customer_data_p.*
107   from sa_customer_data_c, sa_customer_data_p;
108

```

Script Output X | Query Result X | SQL | All Rows Fetched: 12 in 0.012 seconds

CUSTOMER_NAME	BRAND_NAME	CUSTOMER_ADDRESS	CUSTOMER_CITY	CUSTOMER_COUNTRY	CUSTOMER_EMAIL	CUSTOMER_PHONE	CUSTOMER_MOBILE
1 Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	1-201-229-4029
2 Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	1-201-229-4029
3 Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	1-201-229-4029
4 Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	1-201-229-4029
5 Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	1-201-229-4029
6 Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	1-201-229-4029
7 Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	1-201-229-4029
8 Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	1-201-229-4029
9 Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	1-201-229-4029
10 Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	1-201-229-4029
11 Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	1-201-229-4029
12 Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park	New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	1-201-229-4029

Create a table SA_CUSTOMER_DATA_c2 about the Visa International Service Association customer.

The screenshot shows the Oracle SQL Developer interface. It consists of three main sections: a code editor at the top, a script output window in the middle, and a query result window at the bottom.

Code Editor:

```
109 CREATE TABLE SA_CUSTOMER_DATA_C2
110 (
111     customer_name      VARCHAR2(50),
112     brand_name         VARCHAR2(50),
113     customer_address   VARCHAR2(50),
114     customer_city      VARCHAR2(30),
115     customer_country   VARCHAR2(30),
116     customer_email     VARCHAR2(50),
117     customer_office_phone VARCHAR2(30),
118     customer_mobile_phone VARCHAR2(30)
119 )
120 TABLESPACE ts_sa_customers_data_01;
```

Script Output:

1 row inserted.
1 row inserted.
1 row inserted.
2 rows deleted.
1 row inserted.
Table SA_CUSTOMER_DATA_C2 created.

Code Editor:

```
122 INSERT INTO sa_customer_data_c2 (
123     customer_name,
124     brand_name,
125     customer_address,
126     customer_city,
127     customer_country,
128     customer_email,
129     customer_office_phone,
130     customer_mobile_phone
131 )
132 VALUES (
133     'Visa International Service Association',
134     'VISA',
135     '900 Metro Center Blvd.',
136     'Foster City',
137     'United States',
138     'https://usa.visa.com/contact-us.html',
139     '1-800-847-2911',
140     '1-800-847-2922');
```

Script Output:

1 row inserted.

Code Editor:

```
102 select * from sa_customer_data_c2;
103
```

Query Result:

CUSTOMER_NAME	BRAND_NAME	CUSTOMER_ADDRESS	CUSTOMER_CITY	CUSTOMER_COUNTRY	CUSTOMER_EMAIL	CUSTOMER_PHONE
Visa International Service Association	VISA	900 Metro Center Blvd.	Foster City	United States	https://usa.visa.com/contact-us.html	1-800-847-2911

Create a table SA_CUSTOMER_DATA_p2 about the Visa International Service Association for an advertising campaign.

The screenshot shows the Oracle SQL Developer interface. It consists of a code editor at the top, a script output window in the middle, and a query result window at the bottom.

Code Editor:

```
144 CREATE TABLE SA_CUSTOMER_DATA_P2
145 (
146     product_name      VARCHAR2(50)
147 )
148 TABLESPACE ts_sa_customers_data_01;
```

Script Output:

Table SA_CUSTOMER_DATA_P2 created.

```

151 INSERT INTO sa_customer_data_p2 (product_name) VALUES ('Sponsorship');
152 INSERT INTO sa_customer_data_p2 (product_name) VALUES ('Card');
153 INSERT INTO sa_customer_data_p2 (product_name) VALUES ('Benefit');
154 INSERT INTO sa_customer_data_p2 (product_name) VALUES ('Service');

155
156
157
158
159
160

```

Script Output x | Query Result x
Task completed in 0.051 seconds

Table SA_CUSTOMER_DATA_P2 created.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

PRODUCT_NAME
1 Sponsorship
2 Card
3 Benefit
4 Service

Cross join of SA_CUSTOMER_DATA_c2 with SA_CUSTOMER_DATA_p2.

```

157
158 select sa_customer_data_c2.*, sa_customer_data_p2.*
159   from sa_customer_data_c2, sa_customer_data_p2;
160

```

Script Output x | Query Result x
SQL | All Rows Fetched: 4 in 0.021

CUSTOMER_NAME	BRAND_NAME	CUSTOMER_ADDRESS	CUSTOMER_CITY	CUSTOMER_COUNTRY	CUSTOMER_EMAIL	CUSTOMER_PHONE
1 Visa International Service Association	VISA	900 Metro Center Blvd. Foster City	United States	United States	https://usa.visa.com/contact-us.html	1-800-847-2911
2 Visa International Service Association	VISA	900 Metro Center Blvd. Foster City	United States	United States	https://usa.visa.com/contact-us.html	1-800-847-2911
3 Visa International Service Association	VISA	900 Metro Center Blvd. Foster City	United States	United States	https://usa.visa.com/contact-us.html	1-800-847-2911
4 Visa International Service Association	VISA	900 Metro Center Blvd. Foster City	United States	United States	https://usa.visa.com/contact-us.html	1-800-847-2911

Create a table SA_CUSTOMER_DATA_a about the Starcom network agencies serving product promotion for the customer.

```

122 CREATE TABLE SA_CUSTOMER_DATA_a
123 (
124     agency_name          VARCHAR2(50),
125     agency_city          VARCHAR2(30),
126     agency_country       VARCHAR2(30),
127     agency_address        VARCHAR2(50),
128     agency_postcode       VARCHAR2(6),
129     agency_email          VARCHAR2(30),
130     agency_office_phone  VARCHAR2(30),
131     agency_mobile_phone  VARCHAR2(30),
132     agency_Fee_percent    DECIMAL (10,2),
133     agency_VAT_percent    DECIMAL (10,2)
134 )
135 TABLESPACE ts_sa_customers_data_01;

136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160

```

Script Output x | Task completed in 0.048 seconds

Table SA_CUSTOMER_DATA_A created.

```

347 |   348 | INSERT INTO SA_CUSTOMER_DATA_a
349 |   350 | VALUES (
351 |   352 |   'Starcom',
353 |   354 |   'London',
355 |   356 |   'United Kingdom',
357 |   358 |   '2 Television Centre, 101 Wood Lane',
359 |   360 |   '19721',
361 |   362 |   'www.starcomww.com/contact',
363 |   364 |   '44-204-577-43-17',
365 |   366 |   '44-204-577-43-17',
367 |   368 |   25,
369 |   370 |   20
371 | );

```

Script Output x | Task completed in 0.153 seconds

1 row inserted.

363 | select * from sa_customer_data_a;

Script Output x | Query Result x | All Rows Fetched: 15 in 0.012 seconds

AGENCY_NAME	AGENCY_CITY	AGENCY_COUNTRY	AGENCY_ADDRESS	AGENCY_POSTCODE	AGENCY_EMAIL	AGENCY_PHONE	AGENCY_MOBI
1 Starcom	New York	United States	375 Hudson St. Floor 12	10014	www.starcomww.com/contact 1-212-468-3888	1-212-468-35	
2 Starcom	Sydney	Australia	21 Harris Street, Pyrmont	2009	www.starcomww.com/contact 612-8666-8000	612-8666-801	
3 Starcom	Paris	France	30-34 chemin Vert Street	75011	www.starcomww.com/contact 33-1-58-74-08-68	33-1-58-74-7	
4 Starcom	Zurich	Switzerland	Stadelhoferstrasse 258	8001	www.starcomww.com/contact 41-43-366-61-61	41-43-366-66	
5 Starcom	Nicosia	Cyprus	21 Academias Ave. KEMA Building	2107	www.starcomww.com/contact 357-22378828	357-22378833	
6 Starcom	Hong Kong	China	29/F, Paul Y. Centre 51 Hung To Road Kwun Tong	12345	www.starcomww.com/contact 39-051-77-44-90	34-911-99-37	
7 Starcom	Berlin	Germany	Paul Lincke Ufer 39-40	10999	www.starcomww.com/contact 49-211-5684962	49-211-56849	
8 Starcom	Shanghai	China	21/F Henderson 688	12345	www.starcomww.com/contact 39-051-77-55-90	39-051-77-99	
9 Starcom	Cairo	Egypt	City Stars 4th Floor	10521	www.starcomww.com/contact 202-0-2-4800-941	202-0-5-4800	
10 Starcom	Warsaw	Poland	44 A Domanska Street	02-672	www.starcomww.com/contact 48-22-493-99-99	48-22-493-09	
11 Starcom	Barcelona	Spain	385 Aragon Street	08013	www.starcomww.com/contact 34-933-96-77-00	34-933-96-88	
12 Starcom	Milan	Italy	C.R. Darwin Street, 20	20143	www.starcomww.com/contact 39-02-66-798-1	39-02-66-798	
13 Starcom	Dubai	UAE	King Salman Bin Abdulaziz Al Saud St.	7534	www.starcomww.com/contact 971-0-4-367-6400	971-0-4-367-	
14 Starcom	Moscow	Russia	15 Leningradsky prospekt, bld 1	194021	www.starcomww.com/contact 7-495-969-2010	7-495-969-20	
15 Starcom	London	United Kingdom	2 Television Centre, 101 Wood Lane	19721	www.starcomww.com/contact 44-204-577-43-17	44-204-577-4	

Create a table SA_CUSTOMER_DATA_pm about the type of promotion the customer chooses for the products.

```

542 | CREATE TABLE SA_CUSTOMER_DATA_pm
543 | (
544 |     promotion_media_type    VARCHAR2(50)
545 | )
546 | TABLESPACE ts_sa_customers_data_01;

```

Script Output x | Task completed in 0.047 seconds

Table SA_CUSTOMER_DATA_PM created.

The screenshot shows the Oracle SQL Developer interface. At the top, a script window displays several `INSERT INTO` statements into the `SA_CUSTOMER_DATA_pm` table, each inserting a value for the `PROMOTION_MEDIA_TYPE` column. The output pane below shows the results: "1 row inserted." repeated six times. A message at the bottom indicates the task completed in 0.057 seconds.

Below this, another script window shows a `SELECT * FROM SA_CUSTOMER_DATA_pm;` query. The output pane displays a table titled "PROMOTION_MEDIA_TYPE" with seven rows, each containing a value from 1 to 7 corresponding to the media types: TV, press, digital, radio, OOH, production, and design.

Using the Cross join tables and CTE subqueries defined in the WITH clause, as well as generating total_ID and TIME_ID to promote the order from the customer, we got a common table `SA_CUSTOMER_DATA` on the `SA_CUSTOMERS` storage level in the `ts_sa_customers_data_01` tablespace for customers.

The screenshot shows the Oracle SQL Developer interface with a large query window containing complex SQL code. The code uses Common Table Expressions (CTEs) to join multiple tables (`sa_customer_data_c`, `sa_customer_data_p`, `sa_customer_data_a`, `SA_CUSTOMER_DATA_pm`) and generate `total_ID` and `TIME_ID`. It also includes a subquery to generate a sequence of numbers from 1 to 200. The final `SELECT * FROM SA_CUSTOMER_DATA;` statement is highlighted.

The output pane at the bottom shows the results of the query. A table titled "SA_CUSTOMER_DATA" is displayed with columns: `TOTAL_ID`, `TIME_ID`, `CUSTOMER_NAME`, `BRAND_NAME`, `CUSTOMER_ADDRESS`, `CUSTOMER_CITY`, `CUSTOMER_COUNTRY`, `CUSTOMER_EMAIL`, and `CUSTOMER_PHONE`. The data shows five rows for Samsung Electronics, all located at 85 Challenger Rd., Ridgefield Park, New Jersey, United States, with their respective customer IDs and email addresses.

```

385 with cte_customer_agency AS (
386     select sa_customer_data_c.*, sa_customer_data_p.*, sa_customer_data_a., SA_CUSTOMER_DATA_pm.*
387     from sa_customer_data_c, sa_customer_data_p, sa_customer_data_a, SA_CUSTOMER_DATA_pm
388     union all
389     select sa_customer_data_c2.*, sa_customer_data_p2., sa_customer_data_a., SA_CUSTOMER_DATA_pm.*
390     from sa_customer_data_c2, sa_customer_data_p2, sa_customer_data_a, SA_CUSTOMER_DATA_pm
391 ),
392 SA_CUSTOMER_DATA AS (
393     select ROW_NUMBER() OVER (ORDER BY TIME_ID) total_ID, t. *, cte_customer_agency. *
394     from (
395         SELECT
396             TRUNC( sd + rn ) time_id
397             FROM
398             (
399                 SELECT
400                     TO_DATE( '12/31/2021', 'MM/DD/YYYY' ) sd,
401                     rounum rn
402                     FROM dual
403                     CONNECT BY level <= 200
404             )
405         ) t,
406         cte_customer_agency
407     order by TOTAL_ID DESC)
408 select * from SA_CUSTOMER_DATA;

```

Script Output x | Query Result x | SQL | Fetched 50 rows in 2.202 seconds

	TOTAL_ID	TIME_ID	CUSTOMER_NAME	BRAND_NAME	CUSTOMER_ADDRESS	CUSTOMER_CITY	CUSTOMER_COUNTRY	CUSTOMER_EMAIL
1	33600019-JUL-22	Visa International Service Association	VISA	900 Metro Center Blvd. Foster City	United States	https://usa.visa.com/contact-us.html	1	
2	33599919-JUL-22	Visa International Service Association	VISA	900 Metro Center Blvd. Foster City	United States	https://usa.visa.com/contact-us.html	1	
3	33599819-JUL-22	Visa International Service Association	VISA	900 Metro Center Blvd. Foster City	United States	https://usa.visa.com/contact-us.html	1	
4	33599719-JUL-22	Visa International Service Association	VISA	900 Metro Center Blvd. Foster City	United States	https://usa.visa.com/contact-us.html	1	
5	33599619-JUL-22	Visa International Service Association	VISA	900 Metro Center Blvd. Foster City	United States	https://usa.visa.com/contact-us.html	1	

336,000 rows were generated.

```

409 with cte_customer_agency AS (
410     select sa_customer_data_c.*, sa_customer_data_p.*, sa_customer_data_a., SA_CUSTOMER_DATA_pm. *
411     from sa_customer_data_c, sa_customer_data_p, sa_customer_data_a, SA_CUSTOMER_DATA_pm
412     union all
413     select sa_customer_data_c2.*, sa_customer_data_p2., sa_customer_data_a., SA_CUSTOMER_DATA_pm. *
414     from sa_customer_data_c2, sa_customer_data_p2, sa_customer_data_a, SA_CUSTOMER_DATA_pm
415 ),
416 CUSTOMER_DATA AS (
417     select ROW_NUMBER() OVER (ORDER BY TIME_ID) total_ID, t. *, cte_customer_agency. *
418     from (
419         SELECT
420             TRUNC( sd + rn ) time_id
421             FROM
422             (
423                 SELECT
424                     TO_DATE( '12/31/2021', 'MM/DD/YYYY' ) sd,
425                     rounum rn
426                     FROM dual
427                     CONNECT BY level <= 200
428             )
429         )
430     ) t,
431     cte_customer_agency
432     order by TOTAL_ID ASC)
433 select count(*) from CUSTOMER_DATA;

```

Script Output x | Query Result x | SQL | All Rows Fetched: 1 in 0.93 seconds

	COUNT(*)
1	336000

Let's form this data into the table SA_CUSTOMER_DATA_total.

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor with a script containing the creation of a table named SA_CUSTOMER_DATA_total. The table has 41 columns of various data types, including NUMBER, DATE, and VARCHAR2. It also includes constraints and a tablespace assignment. In the bottom-right pane, there is a 'Script Output' tab showing the message 'Table SA_CUSTOMER_DATA_TOTAL created.' and a note indicating the task completed in 1.835 seconds.

```
385 | -----  
386 | create table SA_CUSTOMER_DATA_total  
387 | (  
388 |     TOTAL_ID NUMBER (10),  
389 |     TIME_ID DATE,  
390 |     CUSTOMER_NAME varchar2 (50),  
391 |     BRAND_NAME VARCHAR2(50),  
392 |     CUSTOMER_ADDRESS VARCHAR2(50),  
393 |     CUSTOMER_CITY VARCHAR2(30),  
394 |     CUSTOMER_COUNTRY VARCHAR2(30),  
395 |     CUSTOMER_EMAIL VARCHAR2(50),  
396 |     CUSTOMER_OFFICE_PHONE VARCHAR2(30),  
397 |     CUSTOMER_MOBILE_PHONE VARCHAR2(30),  
398 |     PRODUCT_NAME VARCHAR2(50),  
399 |     AGENCY_NAME VARCHAR2 (50),  
400 |     AGENCY_CITY VARCHAR2(30),  
401 |     AGENCY_COUNTRY VARCHAR2(30),  
402 |     AGENCY_ADDRESS VARCHAR2(50),  
403 |     AGENCY_POSTCODE VARCHAR2(6),  
404 |     AGENCY_EMAIL VARCHAR2(30),  
405 |     AGENCY_OFFICE_PHONE VARCHAR2(30),  
406 |     AGENCY_MOBILE_PHONE VARCHAR2(30),  
407 |     AGENCY_FEE_PERCENT DECIMAL (10,2),  
408 |     AGENCY_VAT_PERCENT DECIMAL (10,2),  
409 |     PROMOTION_MEDIA_TYPE VARCHAR2(30)  
410 | )  
411 | TABLESPACE ts_sa_customers_data_01;  
...  
Table SA_CUSTOMER_DATA_TOTAL created.  
Task completed in 1.835 seconds
```

Insert values into the

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor with a complex INSERT INTO statement for the SA_CUSTOMER_DATA_total table. The statement uses a Common Table Expression (CTE) named cte_customer_agency to select data from multiple source tables (sa_customer_data_c, sa_customer_data_p, sa_customer_data_a, SA_CUSTOMER_DATA_pm) and generates a time_id column based on a sequence of dates and row numbers. The final query inserts the generated rows into the target table. In the bottom-right pane, there is a 'Script Output' tab showing the message '336,000 rows inserted.' and a note indicating the task completed in 1.835 seconds.

```
413 | INSERT INTO SA_CUSTOMER_DATA_total  
414 | with cte_customer_agency AS (  
415 |     select sa_customer_data_c.* , sa_customer_data_p.* , sa_customer_data_a.* , SA_CUSTOMER_DATA_pm.*  
416 |     from sa_customer_data_c , sa_customer_data_p , sa_customer_data_a , SA_CUSTOMER_DATA_pm  
417 |     union all  
418 |     select sa_customer_data_c2.* , sa_customer_data_p2.* , sa_customer_data_a.* , SA_CUSTOMER_DATA_pm.*  
419 |     from sa_customer_data_c2 , sa_customer_data_p2 , sa_customer_data_a , SA_CUSTOMER_DATA_pm  
420 | ),  
421 |     SA_CUSTOMER_DATA AS (  
422 |     select ROW_NUMBER() OVER (ORDER BY TIME_ID) total_ID , t.* , cte_customer_agency.*  
423 |     from (  
424 |         SELECT  
425 |             TRUNC( sd + rn ) time_id  
426 |             FROM  
427 |                 (  
428 |                     SELECT  
429 |                         TO_DATE( '12/31/2021' , 'MM/DD/YYYY' ) sd,  
430 |                         rounum rn  
431 |                         FROM dual  
432 |                         CONNECT BY level <= 200  
433 |                 )  
434 |             ) t,  
435 |             cte_customer_agency  
436 |             order by TOTAL_ID DESC  
437 |     select * from SA_CUSTOMER_DATA;
```

336,000 rows inserted.

```

438 | select * from SA_CUSTOMER_DATA_total
439 |
440 | order by 1;
441 |

```

Script Output x Query Result x

Fetched 50 rows in 0.383 seconds

TOTAL_ID	TIME_ID	CUSTOMER_NAME	BRAND_NAME	CUSTOMER_ADDRESS	CUSTOMER_CITY	CUSTOMER_COUNTRY	CUSTOMER_EMAIL	CUSTOMER_PHONE
1	01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
2	2 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
3	3 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
4	4 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
5	5 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
6	6 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
7	7 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
8	8 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
9	9 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
10	10 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
11	11 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
12	12 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
13	13 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	

As a result, we can explore the SA_CUSTOMER_DATA_total table in terms of metrics.

```

441 |
442 | select DISTINCT (agency_city) from SA_CUSTOMER_DATA_total
443 | order by 1;
444 |

```

Script Output x Query Result x

All Rows Fetched: 15 in 0.075 seconds

AGENCY_CITY
1 Barcelona
2 Berlin
3 Cairo
4 Dubai
5 Hong Kong
6 London
7 Milan
8 Moscow
9 New York
10 Nicosia
11 Paris
12 Shanghai
13 Sydney
14 Warsaw
15 Zurich

Object Name	Tablespace
SA_EMPLOYEES	ts_sa_employees_data_01

```

| alter session set current_schema=sa_employees;
| alter user sa_employees QUOTA UNLIMITED ON ts_sa_employees_data_01;

```

Script Output x

Task completed in 0.023 seconds

Grant succeeded.

Session altered.

User SA_EMPLOYEES altered.

Create a table SA_EMPLOYEE_DATA about Starcom employees.

```
28 CREATE TABLE SA_EMPLOYEE_DATA
29   ( employee_ID          NUMBER (10),
30     employee_first_name  VARCHAR2 (40),
31     employee_last_name   VARCHAR2 (40),
32     employee_position    VARCHAR2 (40),
33     employee_salary_project DECIMAL (30,2)
34   )
35 TABLESPACE ts_sa_employees_data_01;
36
```

Table SA_EMPLOYEE_DATA created.

```
192 , trunc(dbms_random.value(1, 11))      AS id_fn
193 , trunc(dbms_random.value(1, 11))      AS id_ln
194 , trunc(dbms_random.value(1, 3))       AS id_p
195 , trunc(dbms_random.value(50, 500))    AS id_email
196
197   (
198     SELECT
199       ROWNUM rn
200     FROM
201       dual
202     CONNECT BY
203       level <= 336000
204   ) a
205 )
206
207   SELECT
208     rn
209   , fn.a
210   , ln.a
211   , p.a
212   , id_email
213
214   FROM
215     cte_gen g
216   LEFT OUTER JOIN cte_fn fn ON g.id_fn = fn.id
217   LEFT OUTER JOIN cte_ln ln ON g.id_ln = ln.id
218   LEFT OUTER JOIN cte_p p ON g.id_p = p.id
219
220
221
222
223
224
```

336,000 rows inserted.

```
254 SELECT count(*)
255 FROM SA_EMPLOYEE_DATA
256 ORDER BY 1;
257
```

COUNT(*)

1	336000
---	--------

```
219
220
221
222
223
224
```

Script Output | Query Result | Fetched 50 rows in 0.084 seconds

EMPLOYEE_ID	EMPLOYEE_FIRST_NAME	EMPLOYEE_LAST_NAME	EMPLOYEE_POSITION	EMPLOYEE_SALARY_PROJECT
1	1 Robbie	Delon	specialist	251
2	2 Jackie	Foster	specialist	475
3	3 Jackie	Delon	manager	322
4	4 Sam	Foster	manager	211
5	5 Ronnie	Hasse	specialist	478
6	6 Frankie	Delon	manager	451
7	7 Alex	Foster	manager	439
8	8 Nat	Barnes	specialist	337
9	9 Robbie	Delon	specialist	483
10	10 Steph	Hasse	specialist	251
11	11 Alex	Barnes	specialist	342
12	12 Jackie	Joukowski	manager	267
13	13 Robbie	Kendall	manager	388
14	14 Robbie	Joukowski	specialist	362
15	15 Sam	Campbell	specialist	479
16	16 Robbie	Hasse	manager	86
17	17 Robbie	Campbell	manager	442
18	18 Chris	Abramson	manager	396
19	19 Steph	Abramson	manager	418
20	20 Chris	Delon	specialist	356

Let's join table SA_CUSTOMER_DATA_total from TABLESPACE ts_sa_customers_data_01 with table SA_EMPLOYEE_DATA from TABLESPACE ts_sa_employees_data_01 using the left join by key fields TOTAL_ID and EMPLOYEE_ID. We will consider the TOTAL_ID field as a unique number of the promotion whose services are provided by the advertising agency. The employee_salary_project field contains the cost of the agency employee's services per project.

```

17 |
18 | select * from sa_customers.SA_CUSTOMER_DATA_total left outer join sa_employees.SA_EMPLOYEE_DATA
19 | on sa_customers.SA_CUSTOMER_DATA_total.TOTAL_ID = sa_employees.SA_EMPLOYEE_DATA.EMPLOYEE_ID
20 | order by 1;

```

Script Output x Query Result x

Fetched 50 rows in 0.368 seconds

TOTAL_ID	TIME_ID	CUSTOMER_NAME	BRAND_NAME	CUSTOMER_ADDRESS	CUSTOMER_CITY	CUSTOMER_COUNTRY	CUSTOMER_EMAIL	CUSTOMER_OFFICE_
1	1 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
2	2 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
3	3 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
4	4 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
5	5 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
6	6 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
7	7 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
8	8 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
9	9 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
10	10 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	
11	11 01-JAN-22	Samsung Electronics	SAMSUNG	85 Challenger Rd. Ridgefield Park New Jersey	United States	www.facebook.com/SamsungUS	1-201-229-4000	


```

18 |
19 | select * from sa_customers.SA_CUSTOMER_DATA_total left outer join sa_employees.SA_EMPLOYEE_DATA
20 | on sa_customers.SA_CUSTOMER_DATA_total.TOTAL_ID = sa_employees.SA_EMPLOYEE_DATA.EMPLOYEE_ID
21 | order by 1;

```

Script Output x Query Result x

Fetched 50 rows in 0.368 seconds

FILE_PHONE	AGENCY_FEE_PERCENT	AGENCY_VAT_PERCENT	PROMOTION_MEDIA_TYPE	EMPLOYEE_ID	EMPLOYEE_FIRST_NAME	EMPLOYEE_LAST_NAME	EMPLOYEE_POSITION	EMPLOYEE_SALARY_PROJECT
1 588	15	15 TV		1 Robbie	Delon	specialist	251	
2 10	10	25.8 TV		2 Jackie	Foster	specialist	475	
3 78-66	20	20 TV		3 Jackie	Delon	manager	322	
4 6-66	20	7.7 TV		4 Sam	Foster	manager	211	
5 3	7	19 TV		5 Ronnie	Hasse	specialist	478	
6 7-11	5	13 TV		6 Frankie	Delon	manager	451	
7 962	22	19 TV		7 Alex	Foster	manager	439	
8 9-90	5	13 TV		8 Nat	Barnes	specialist	337	
9 0-941	5	14 TV		9 Robbie	Delon	specialist	483	
10 9-99	18	23 TV		10 Steph	Hasse	specialist	251	
11 8-00	9	21 TV		11 Alex	Barnes	specialist	342	


```

21 |
22 | select count(*) from sa_customers.SA_CUSTOMER_DATA_total left outer join sa_employees.SA_EMPLOYEE_DATA
23 | on sa_customers.SA_CUSTOMER_DATA_total.TOTAL_ID = sa_employees.SA_EMPLOYEE_DATA.EMPLOYEE_ID
24 | order by 1;

```

Script Output x Query Result x

All Rows Fetched: 1 in 0.119 seconds

COUNT(*)
1 336000

Object Name	Tablespace
SA_PROMOTIONS	ts_sa_promotions_data_01

Create a table SA_PROMOTION_DATA in the tablespace ts_sa_promotions_data_01 and generate a date for it.

```

4 alter session set current_schema=sa_promotions;
5 alter user sa_promotions QUOTA UNLIMITED ON ts_sa_promotions_data_01;
6

```

Script Output X
 Task completed in 0.051 seconds

Session altered.

User SA_PROMOTIONS altered.

```

-- -----
29 | CREATE TABLE SA_PROMOTION_DATA
30 | ( promotion_ID           NUMBER (10),
31 |   promotion_metric_amount DECIMAL (10,2),
32 |   promotion_price         DECIMAL (10,2),
33 |   promotion_KPI           VARCHAR2 (30),
34 |   promotion_distinct_percent DECIMAL (10,2)
35 |
36 | TABLESPACE ts_sa_promotions_data_01;
37

```

Script Output X
 Task completed in 0.041 seconds

Table SA_PROMOTION_DATA created.

SQL Worksheet | History

Worksheet | Query Builder

```

55      , 'CPS' AS a
56      FROM
57      dual
58  ), cte_gen AS (
59      SELECT
60          a.*
61          , trunc(dbms_random.value(1,100))           AS id_ma
62          , trunc(dbms_random.value(100, 1000))        AS id_pr
63          , trunc(dbms_random.value(1, 3))            AS id_kpi
64          , trunc(dbms_random.value(3, 20))           AS id_dp
65      FROM
66      (
67          SELECT
68              ROWNUM rn
69          FROM
70              dual
71          CONNECT BY
72              level <= 336000
73      ) a
74  )
75  SELECT
76      g.rn
77      , id_ma
78      , id_pr
79      , kpi.a
80      , id_dp
81  FROM
82      cte_gen g
83  LEFT OUTER JOIN cte_kpi    kpi ON g.id_kpi = kpi.id
84 ;

```

Script Output | Task completed in 2.585 seconds

336,000 rows inserted.

```

86 |
87 | SELECT *
88 | FROM SA_PROMOTION_DATA
89 | ORDER BY 1;

```

Script Output | Query Result | Fetched 50 rows in 0.091 seconds

	PROMOTION_ID	PROMOTION_METRIC_AMOUNT	PROMOTION_PRICE	PROMOTION_KPI	PROMOTION_DISTINCT_PERCENT
1	1	58	154 CPA		11
2	2	93	363 CPL		10
3	3	17	288 CPL		14
4	4	14	151 CPL		15
5	5	82	683 CPL		3
6	6	14	466 CPA		12
7	7	74	677 CPL		9
8	8	95	200 CPL		19
9	9	7	329 CPA		4
10	10	56	675 CPA		14
11	11	52	688 CPL		7
12	12	83	724 CPL		10

```

90 |
91 | SELECT count(*)
92 | FROM SA_PROMOTION_DATA
93 | ORDER BY 1;

```

Script Output | Query Result | All Rows Fetched: 1 in 0.017 seconds

COUNT(*)
1 336000

Get a total table of customers, agency staff, and promotions from three different tables.

```

22 |
23 @select * from sa_customers.SA_CUSTOMER_DATA_total
24 left outer join sa_employees.SA_EMPLOYEE_DATA
25 on sa_customers.SA_CUSTOMER_DATA_total.TOTAL_ID = sa_employees.SA_EMPLOYEE_DATA.EMPLOYEE_ID
26 left outer join sa_promotions.SA_PROMOTION_DATA
27 on sa_customers.SA_CUSTOMER_DATA_total.TOTAL_ID = sa_promotions.SA_PROMOTION_DATA.PROMOTION_ID
28 order by 1;
29 |

Script Output x Query Result x
SQL | Fetched 50 rows in 0.403 seconds


| TOTAL_ID | TIME_ID     | CUSTOMER_NAME       | BRAND_NAME | CUSTOMER_ADDRESS                             | CUSTOMER_CITY | CUSTOMER_COUNTRY           | CUSTOMER_EMAIL | CUSTOMER_OFFICE |
|----------|-------------|---------------------|------------|----------------------------------------------|---------------|----------------------------|----------------|-----------------|
| 1        | 101-JAN-22  | Samsung Electronics | SAMSUNG    | 85 Challenger Rd. Ridgefield Park New Jersey | United States | www.facebook.com/SamsungUS | 1-201-229-4000 |                 |
| 2        | 201-JAN-22  | Samsung Electronics | SAMSUNG    | 85 Challenger Rd. Ridgefield Park New Jersey | United States | www.facebook.com/SamsungUS | 1-201-229-4000 |                 |
| 3        | 301-JAN-22  | Samsung Electronics | SAMSUNG    | 85 Challenger Rd. Ridgefield Park New Jersey | United States | www.facebook.com/SamsungUS | 1-201-229-4000 |                 |
| 4        | 401-JAN-22  | Samsung Electronics | SAMSUNG    | 85 Challenger Rd. Ridgefield Park New Jersey | United States | www.facebook.com/SamsungUS | 1-201-229-4000 |                 |
| 5        | 501-JAN-22  | Samsung Electronics | SAMSUNG    | 85 Challenger Rd. Ridgefield Park New Jersey | United States | www.facebook.com/SamsungUS | 1-201-229-4000 |                 |
| 6        | 601-JAN-22  | Samsung Electronics | SAMSUNG    | 85 Challenger Rd. Ridgefield Park New Jersey | United States | www.facebook.com/SamsungUS | 1-201-229-4000 |                 |
| 7        | 701-JAN-22  | Samsung Electronics | SAMSUNG    | 85 Challenger Rd. Ridgefield Park New Jersey | United States | www.facebook.com/SamsungUS | 1-201-229-4000 |                 |
| 8        | 801-JAN-22  | Samsung Electronics | SAMSUNG    | 85 Challenger Rd. Ridgefield Park New Jersey | United States | www.facebook.com/SamsungUS | 1-201-229-4000 |                 |
| 9        | 901-JAN-22  | Samsung Electronics | SAMSUNG    | 85 Challenger Rd. Ridgefield Park New Jersey | United States | www.facebook.com/SamsungUS | 1-201-229-4000 |                 |
| 10       | 1001-JAN-22 | Samsung Electronics | SAMSUNG    | 85 Challenger Rd. Ridgefield Park New Jersey | United States | www.facebook.com/SamsungUS | 1-201-229-4000 |                 |
| 11       | 1101-JAN-22 | Samsung Electronics | SAMSUNG    | 85 Challenger Rd. Ridgefield Park New Jersey | United States | www.facebook.com/SamsungUS | 1-201-229-4000 |                 |



22 |
23 @select * from sa_customers.SA_CUSTOMER_DATA_total
24 left outer join sa_employees.SA_EMPLOYEE_DATA
25 on sa_customers.SA_CUSTOMER_DATA_total.TOTAL_ID = sa_employees.SA_EMPLOYEE_DATA.EMPLOYEE_ID
26 left outer join sa_promotions.SA_PROMOTION_DATA
27 on sa_customers.SA_CUSTOMER_DATA_total.TOTAL_ID = sa_promotions.SA_PROMOTION_DATA.PROMOTION_ID
28 order by 1;
29 |

Script Output x Query Result x
SQL | Fetched 50 rows in 0.403 seconds


| TYPE | EMPLOYEE_ID | EMPLOYEE_FIRST_NAME | EMPLOYEE_LAST_NAME | EMPLOYEE_POSITION | EMPLOYEE_SALARY_PROJECT | PROMOTION_ID | PROMOTION_METRIC_AMOUNT | PROMOTION_PRICE | PROMOTION |
|------|-------------|---------------------|--------------------|-------------------|-------------------------|--------------|-------------------------|-----------------|-----------|
| 1    | 1           | Robbie              | Delon              | specialist        | 251                     | 1            | 50                      | 154             | CPLA      |
| 2    | 2           | Jackie              | Foster             | specialist        | 475                     | 2            | 93                      | 363             | CPL       |
| 3    | 3           | Jackie              | Delon              | manager           | 322                     | 3            | 17                      | 288             | CPL       |
| 4    | 4           | Sam                 | Foster             | manager           | 211                     | 4            | 14                      | 151             | CPL       |
| 5    | 5           | Ronnie              | Hasse              | specialist        | 478                     | 5            | 82                      | 683             | CPL       |
| 6    | 6           | Frankie             | Delon              | manager           | 451                     | 6            | 14                      | 466             | CPL       |
| 7    | 7           | Alex                | Foster             | manager           | 439                     | 7            | 74                      | 677             | CPL       |
| 8    | 8           | Nat                 | Barnes             | specialist        | 337                     | 8            | 95                      | 200             | CPL       |
| 9    | 9           | Robbie              | Delon              | specialist        | 483                     | 9            | 7                       | 329             | CPLA      |
| 10   | 10          | Steph               | Hasse              | specialist        | 251                     | 10           | 56                      | 675             | CPLA      |
| 11   | 11          | Alex                | Barnes             | specialist        | 342                     | 11           | 52                      | 688             | CPL       |



30 |
31 @select count(*) from sa_customers.SA_CUSTOMER_DATA_total
32 left outer join sa_employees.SA_EMPLOYEE_DATA
33 on sa_customers.SA_CUSTOMER_DATA_total.TOTAL_ID = sa_employees.SA_EMPLOYEE_DATA.EMPLOYEE_ID
34 left outer join sa_promotions.SA_PROMOTION_DATA
35 on sa_customers.SA_CUSTOMER_DATA_total.TOTAL_ID = sa_promotions.SA_PROMOTION_DATA.PROMOTION_ID
36 order by 1;

Script Output x Query Result x
SQL | All Rows Fetched: 1 in 0.181 seconds


| COUNT(*) |
|----------|
| 1        |
| 336000   |


```

If necessary, data for other DIMENTIONS TABLES can be generated in the same way.

2.3. Task 03: Create Group by Plan

Prepare adhoc SQL for segregate data view on SA_* objects (Storage).

Operations:

- Select
 - Group by

The screenshot shows a SQL worksheet titled "sa_customers_generate.sql" with the following code:

```
470    SELECT
471        TRUNC( sd + rn ) time_id
472        FROM
473        (
474            SELECT
475                TO_DATE( '12/31/2021', 'MM/DD/YYYY' ) sd,
476                rownum rn
477                FROM dual
478                CONNECT BY level <= 200
479            )
480        ) t,
481        cte_customer_agency
482        order by TOTAL_ID ASC)
483 select CUSTOMER_DATA.brand_name, CUSTOMER_DATA.promotion_media_type from CUSTOMER_DATA
484 GROUP BY CUSTOMER_DATA.brand_name, CUSTOMER_DATA.promotion_media_type;
```

The query result table has two columns: BRAND_NAME and PROMOTION_MEDIA_TYPE. The data is as follows:

BRAND_NAME	PROMOTION_MEDIA_TYPE
1 SAMSUNG	TV
2 SAMSUNG	press
3 SAMSUNG	digital
4 SAMSUNG	radio
5 SAMSUNG	OOH
6 SAMSUNG	production
7 SAMSUNG	design
8 VISA	TV
9 VISA	press
10 VISA	digital
11 VISA	radio
12 VISA	OOH
13 VISA	production
14 VISA	design

In this selection, we can see which services and for which brands have been ordered through the Starcom network of agencies.

The following selection shows that Starcom provides services in different parts of the world.

The screenshot shows a SQL worksheet titled "sa_customers_generate.sql" with the following code:

```
476        rownum rn
477        FROM dual
478        CONNECT BY level <= 200
479    )
480    ) t,
481    cte_customer_agency
482    order by TOTAL_ID ASC)
483 select CUSTOMER_DATA.agency_country, CUSTOMER_DATA.agency_city, CUSTOMER_DATA.brand_name, CUSTOMER_DATA.promotion_media_type from CUSTOMER_DATA
484 GROUP BY CUSTOMER_DATA.agency_country, CUSTOMER_DATA.agency_city, CUSTOMER_DATA.brand_name, CUSTOMER_DATA.promotion_media_type;
```

The query result table has four columns: AGENCY_COUNTRY, AGENCY_CITY, BRAND_NAME, and PROMOTION_MEDIA_TYPE. The data is as follows:

AGENCY_COUNTRY	AGENCY_CITY	BRAND_NAME	PROMOTION_MEDIA_TYPE
1 United States	New York	SAMSUNG	TV
2 Australia	Sydney	SAMSUNG	TV
3 France	Paris	SAMSUNG	TV
4 Switzerland	Zurich	SAMSUNG	TV
5 Cyprus	Nicosia	SAMSUNG	TV
6 China	Hong Kong	SAMSUNG	TV
7 Germany	Berlin	SAMSUNG	TV
8 China	Shanghai	SAMSUNG	TV
9 Egypt	Cairo	SAMSUNG	TV
10 Poland	Warsaw	SAMSUNG	TV
11 Spain	Barcelona	SAMSUNG	TV
12 Italy	Milan	SAMSUNG	TV
13 UAE	Dubai	SAMSUNG	TV
14 Russia	Moscow	SAMSUNG	TV
15 United Kingdom	London	SAMSUNG	TV
16 United States	New York	SAMSUNG	press
17 Australia	Sydney	SAMSUNG	press
18 France	Paris	SAMSUNG	press
19 Switzerland	Zurich	SAMSUNG	press

- Merge
- Group by

The MERGE statement is a single command that combines the ability to update or insert rows into a table by conditionally deriving the rows to be updated or inserted from one or more sources.

Consider two customer tables with the same set of columns. Let's consider the use of the MERGE operator.

```

517 | select * from sa_customer_data_c;
518 | select * from sa_customer_data_c2;
519 |
520 |
521 | MERGE INTO sa_customer_data_c t1
522 | USING sa_customer_data_c2 t2
523 | ON (t1.customer_name = t2.customer_name)
524 | WHEN MATCHED THEN
525 | UPDATE SET
526 | t1.brand_name = t2.brand_name,
527 | t1.customer_address = t2.customer_address,
528 | t1.customer_city = t2.customer_city,
529 | t1.customer_country = t2.customer_country,
530 | t1.customer_email = t2.customer_email,
531 | t1.customer_office_phone = t2.customer_office_phone,
532 | t1.customer_mobile_phone = t2.customer_mobile_phone
533 | WHEN NOT MATCHED THEN
534 | INSERT (t1.customer_name, t1.brand_name, t1.customer_address, t1.customer_city, t1.customer_country, t1.customer_email, t1.customer_office_phone, t1.customer_mobile_phone)
535 | VALUES (t2.customer_name, t2.brand_name, t2.customer_address, t2.customer_city, t2.customer_country, t2.customer_email, t2.customer_office_phone, t2.customer_mobile_phone)
536 |
537 |
538 |
539 |
540 |
541 |
542 |
543 |
544 |

```

Script Output: Task completed in 0.046 seconds

1 row merged.

It may be noted that the MERGE operator is suitable for updating and merging data in tables.

Updated table:

```

535 | select * from sa_customer_data_c t1;
536 |
537 |
538 |
539 |
540 |
541 |
542 |
543 |
544 |

```

Query Result: All Rows Fetched: 2 in 0.036 seconds

CUSTOMER_NAME	BRAND_NAME	CUSTOMER_ADDRESS	CUSTOMER_CITY	CUSTOMER_COUNTRY	CUSTOMER_EMAIL	CUSTOMER_MOBILE_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	1-800-847-2911

Additional table:

```

537 | select * from sa_customer_data_c2 t2;
538 |
539 |
540 |
541 |
542 |
543 |
544 |

```

Query Result: All Rows Fetched: 1 in 0.011 seconds

CUSTOMER_NAME	BRAND_NAME	CUSTOMER_ADDRESS	CUSTOMER_CITY	CUSTOMER_COUNTRY	CUSTOMER_EMAIL	CUSTOMER_MOBILE_PHONE
1 Visa International Service Association	VISA	900 Metro Center Blvd.	Foster City	United States	https://usa.visa.com/contact-us.html	1-800-847-2911