**Lab report #10**

**Sadovskaya Veronika**

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

**Task 1 - Transformation Description**

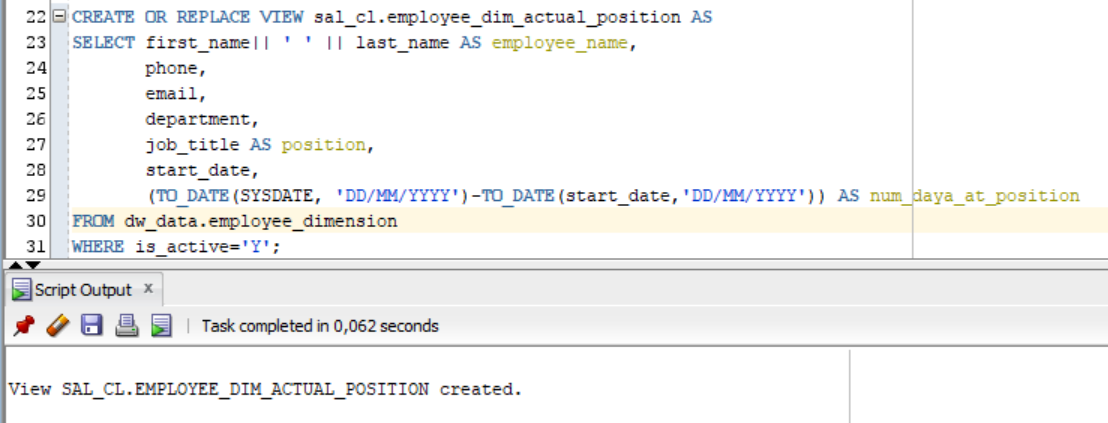
Transforming Data Using PL/SQL

In a data warehouse environment, you can use procedural languages such as PL/SQL to implement complex transformations in the Oracle Database. Whereas CTAS operates on entire tables and emphasizes parallelism, PL/SQL provides a row-based approached and can accommodate very sophisticated transformation rules. For example, a PL/SQL procedure could open multiple cursors and read data from multiple source tables, combine this data using complex business rules, and finally insert the transformed data into one or more target table. It would be difficult or impossible to express the same sequence of operations using standard SQL statements.

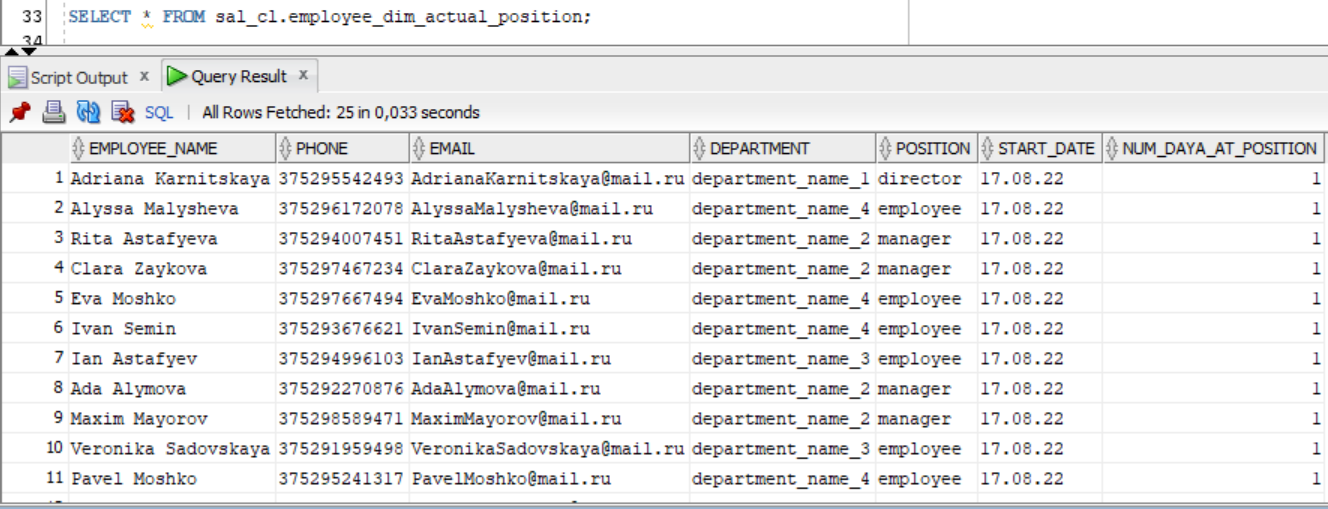
Using a procedural language, a specific transformation (or number of transformation steps) within a complex ETL processing can be encapsulated, reading data from an intermediate staging area and generating a new table object as output. A previously generated transformation input table and a subsequent transformation will consume the table generated by this specific transformation. Alternatively, these encapsulated transformation steps within the complete ETL process can be integrated seamlessly, thus streaming sets of rows between each other without the necessity of intermediate staging. You can use table functions to implement such behavior.

**Task 2 - Loading to SAL Layer Data**

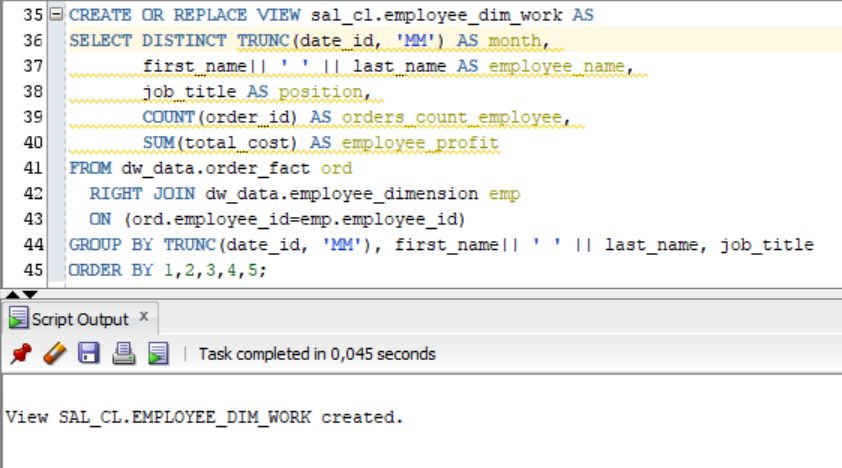
Let's create a view that contains information about the duration of work of all employees in the current position:



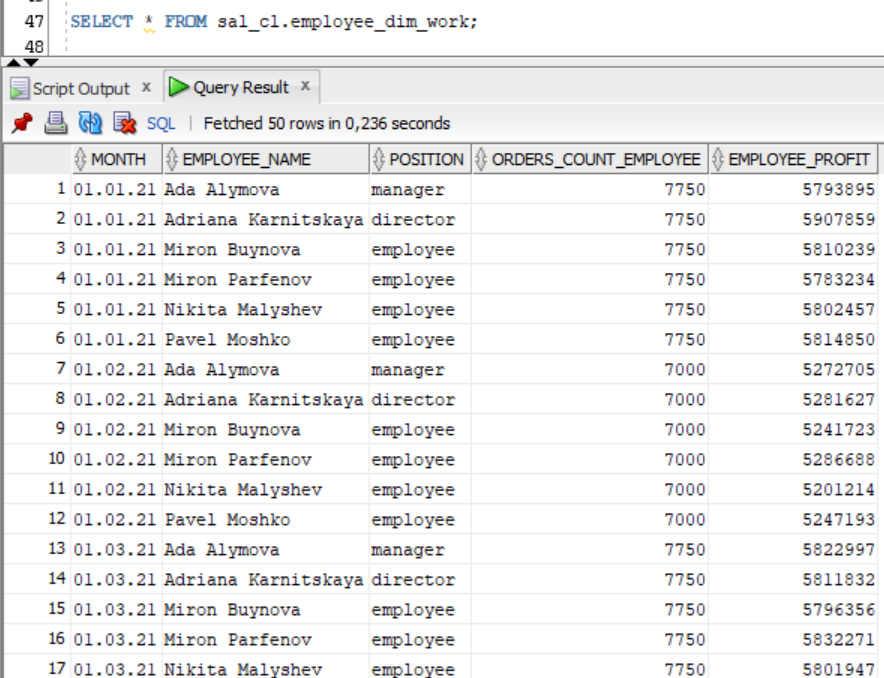
Select from this view:



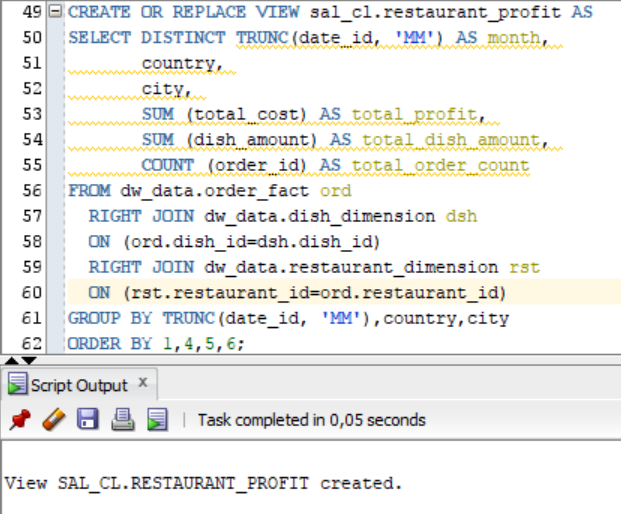
Let's create a view that contains information about the amount of work of each employee per month (number of orders and total profit):



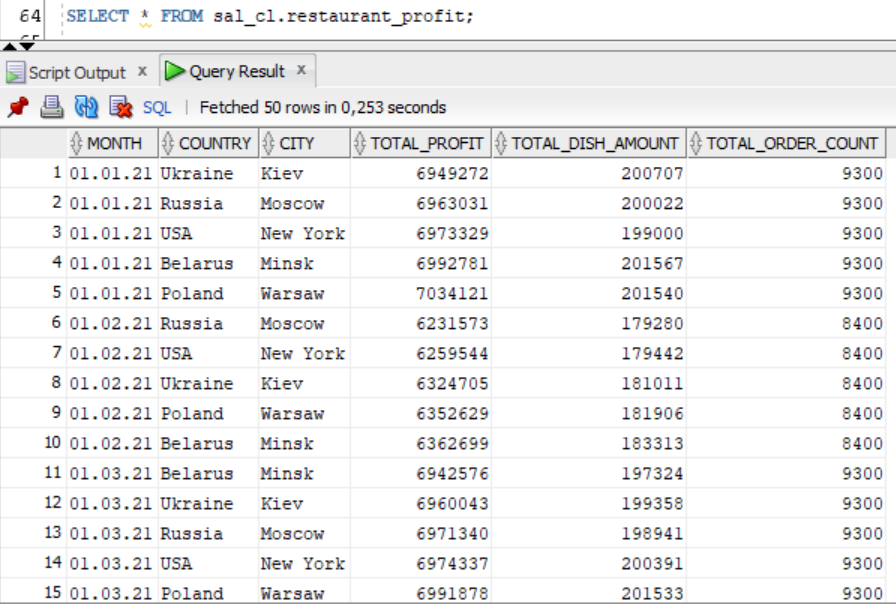
Select from this view:



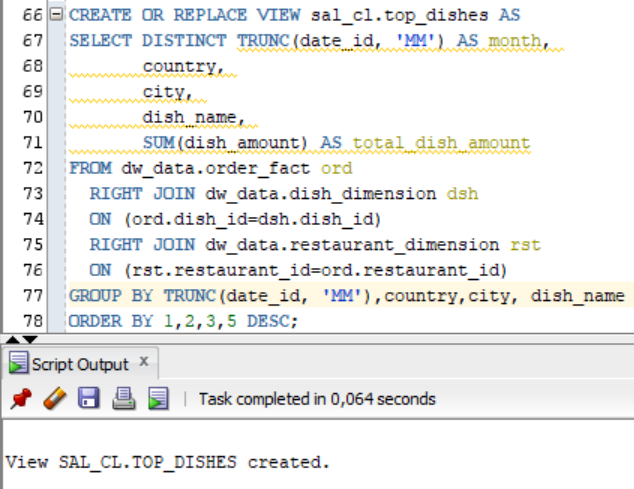
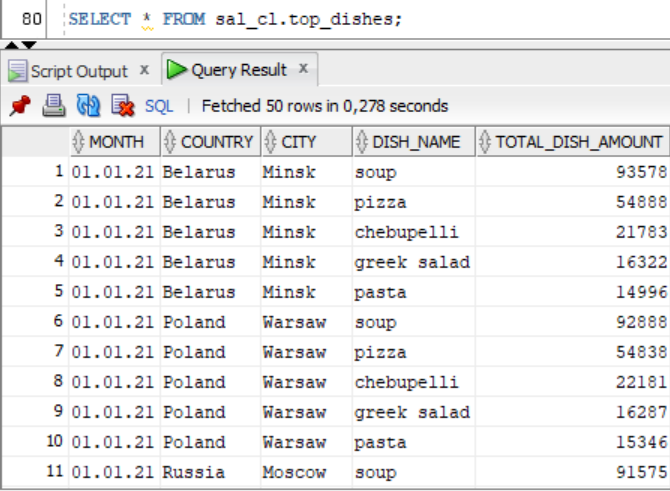
Let's create a view that contains information about the profit for the month for each restaurant, the number of dishes sold and the total number of orders:



Select from this view:

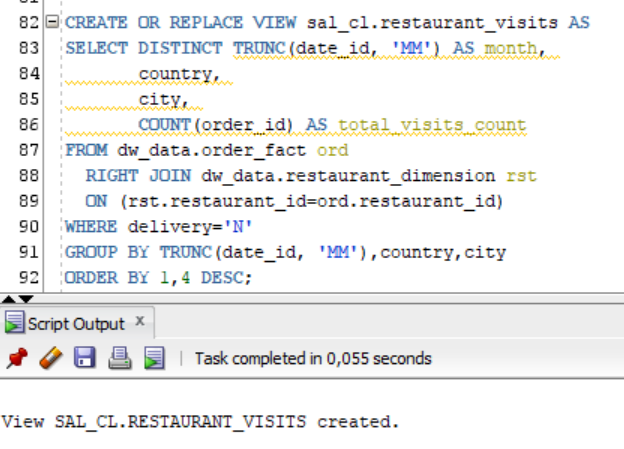


Let's create a view that contains information about the popularity of each dish in each restaurant for a month:

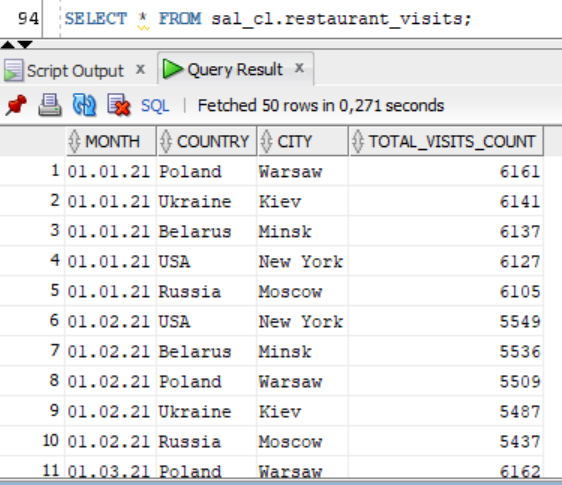


Select from this view:

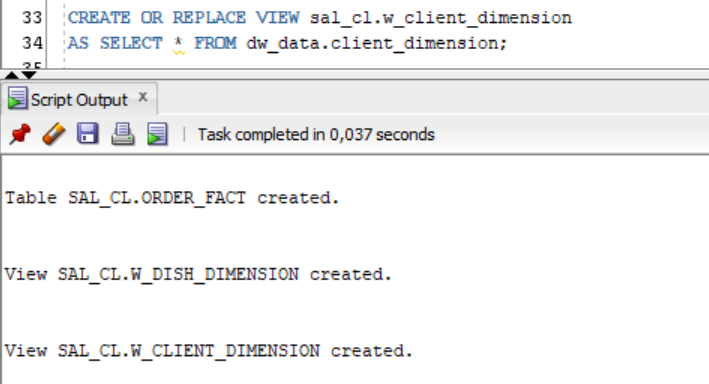
Let's create a view that contains information about the attendance of each restaurant for a month:



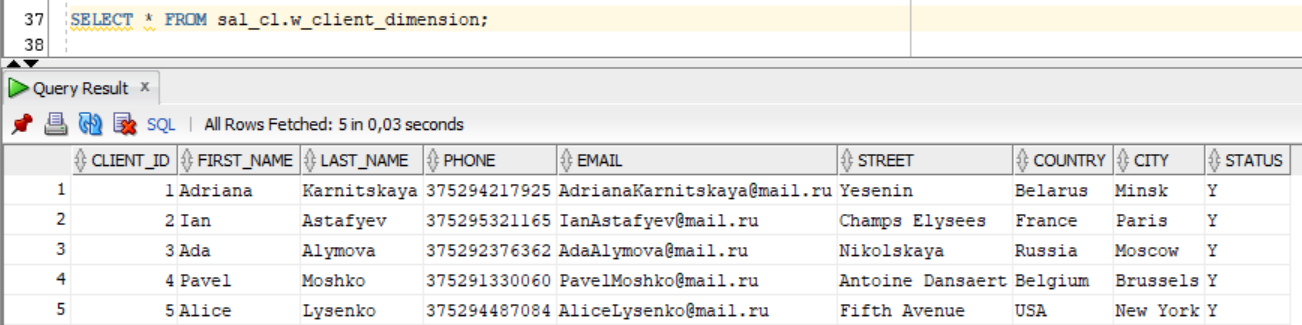
Select from this view:



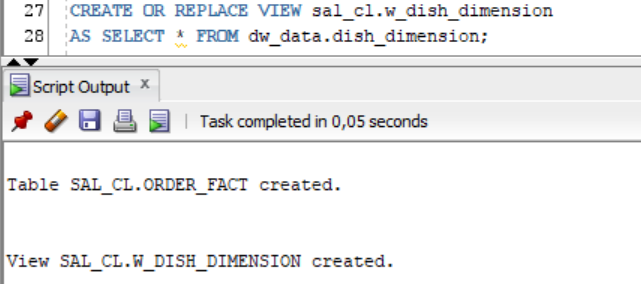
Let's create a view that stores all information about clients:



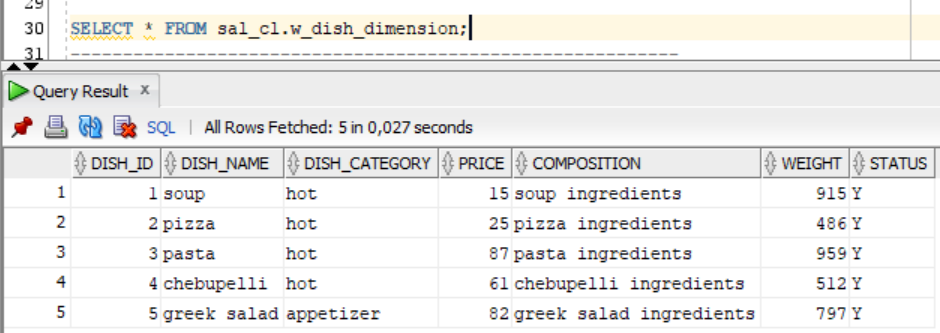
Select from this view:



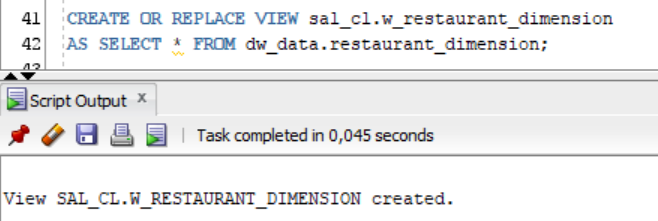
Let's create a view that stores all information about dishes:



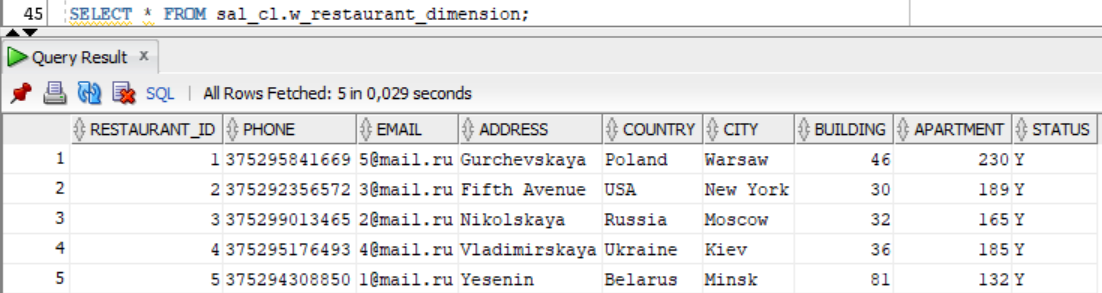
Select from this view:



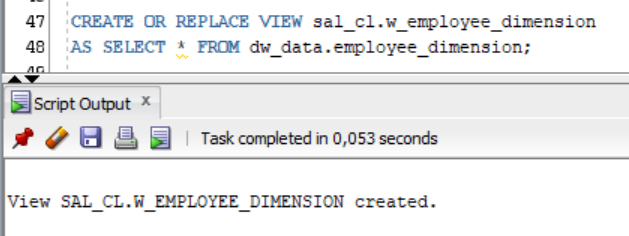
Let's create a view that stores all information about restaurants:



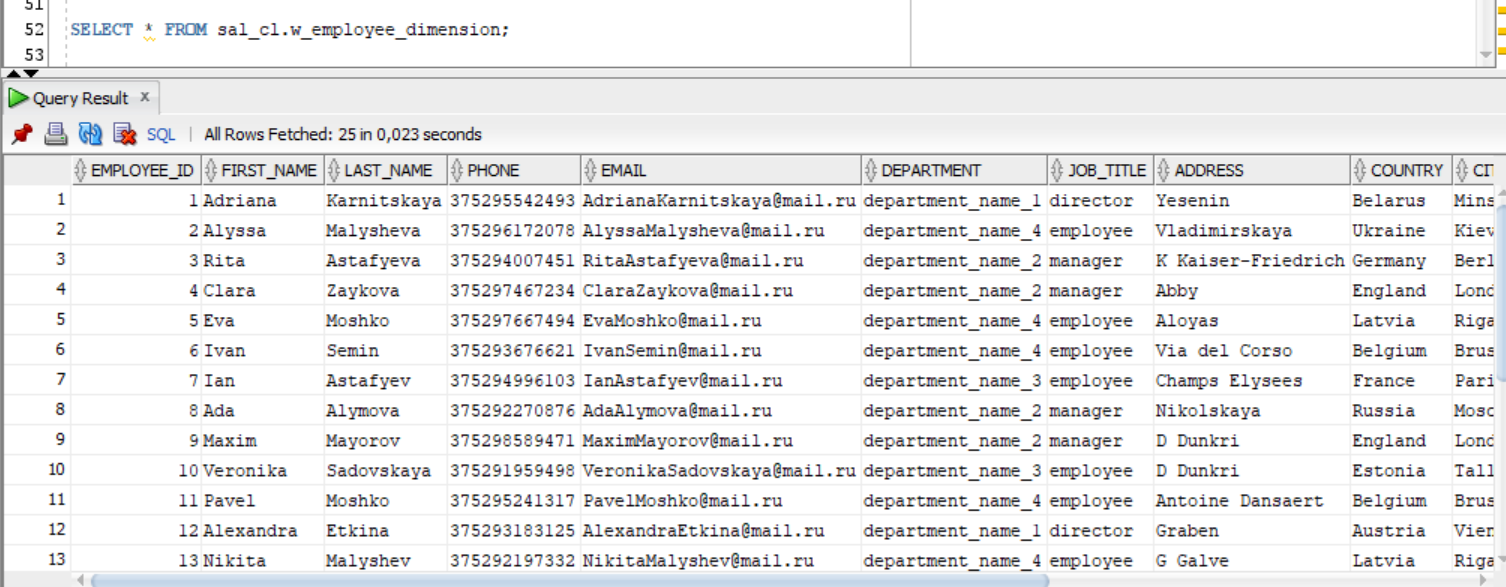
Select from this view:



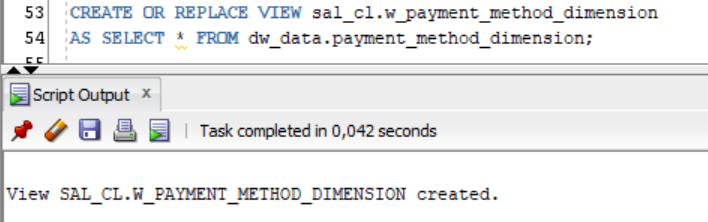
Let's create a view that stores all information about employees:



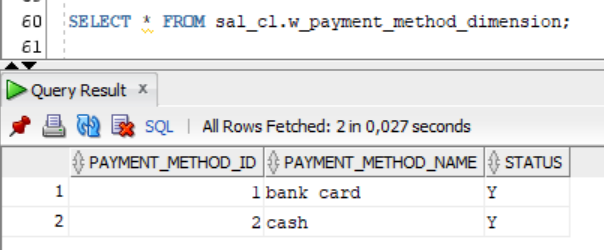
Select from this view:



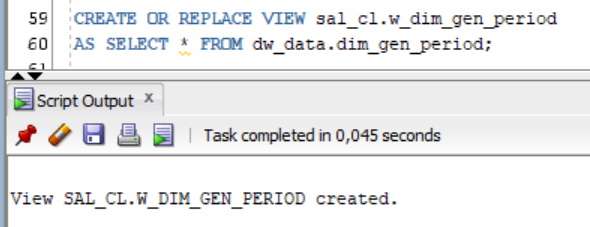
Let's create a view that stores all information about payment methods:



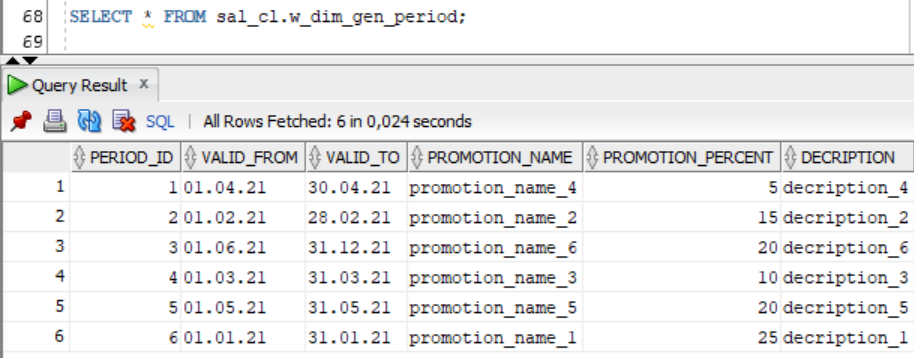
Select from this view:



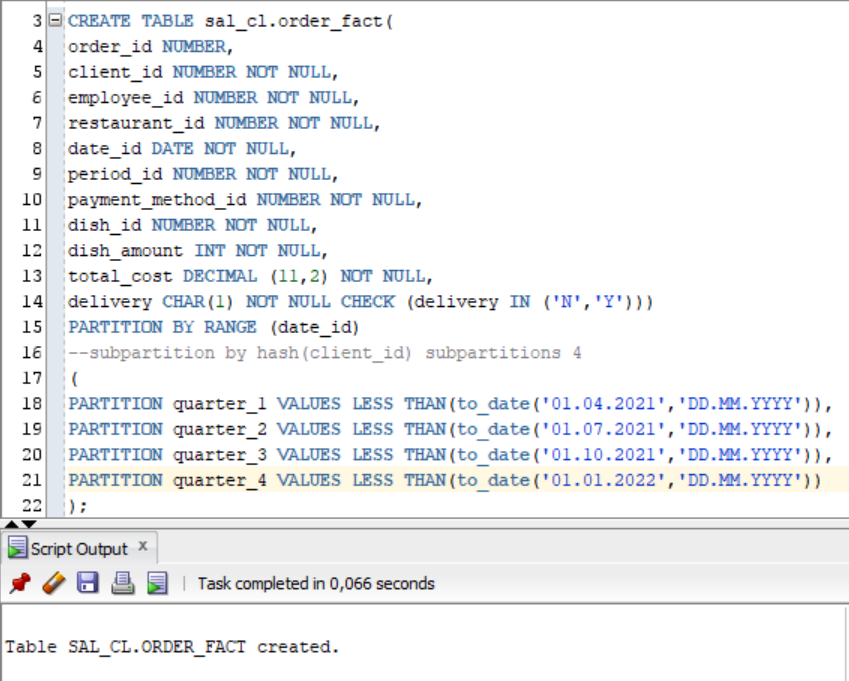
Let's create a view that stores all information about gen periods:



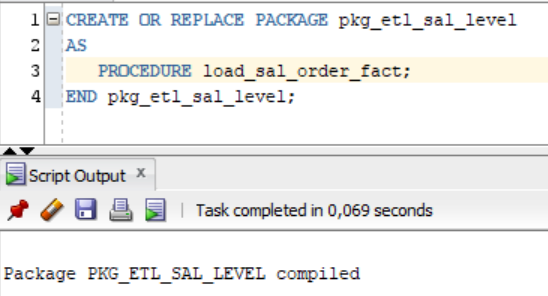
Select from this view:

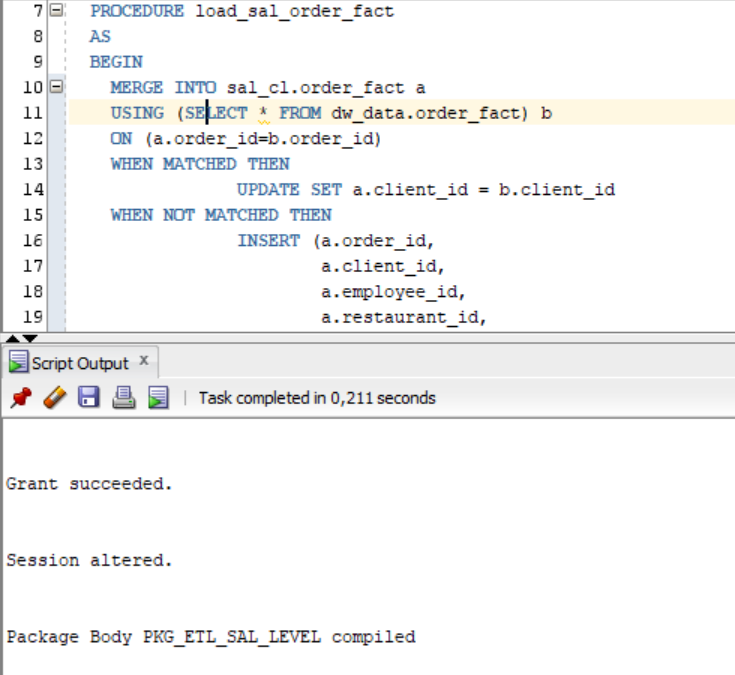


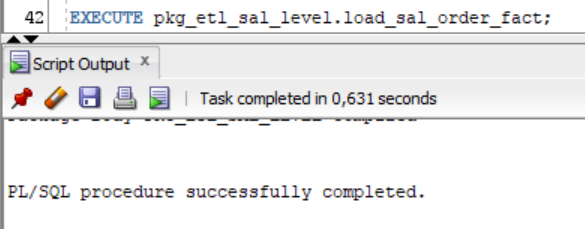
Let's create a table that stores all information about orders:



Let’s create package and procedure to load data to this table:







Select from this table:

