**Partitioning task documentation (Yauheniya Kavalchuk)**

**Explanation of the partitioning strategy**

Range partitioning by date has been used because of the following reasons:

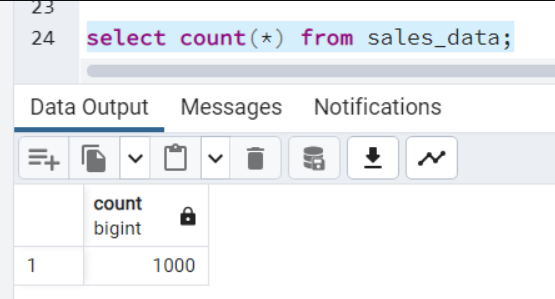
* easy to know when to archive old data
* etl process can be automatically scheduled on the first day of the month
* collected data for monthly reports
* optimized perfomance, because the data will not accumulate.

**Step-by-step documentation**

1. A table with the following structure was created:

* **sale\_id** (Primary Key, Integer)
* **product\_id** (Integer)
* **region\_id** (Integer)
* **salesperson\_id** (Integer)
* **sale\_amount** (Numeric)
* **sale\_date** (Date)

1. Table partitions for the the past 12 months were created.
2. A script, which generates 1000 random rows for the table, was created

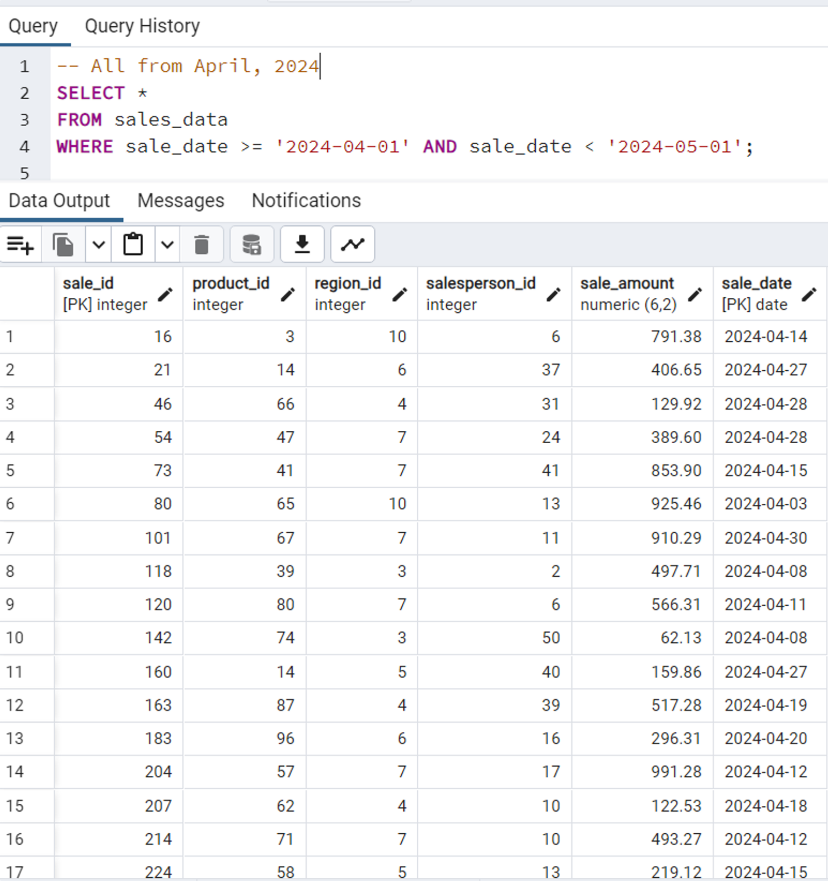


The data was inserted

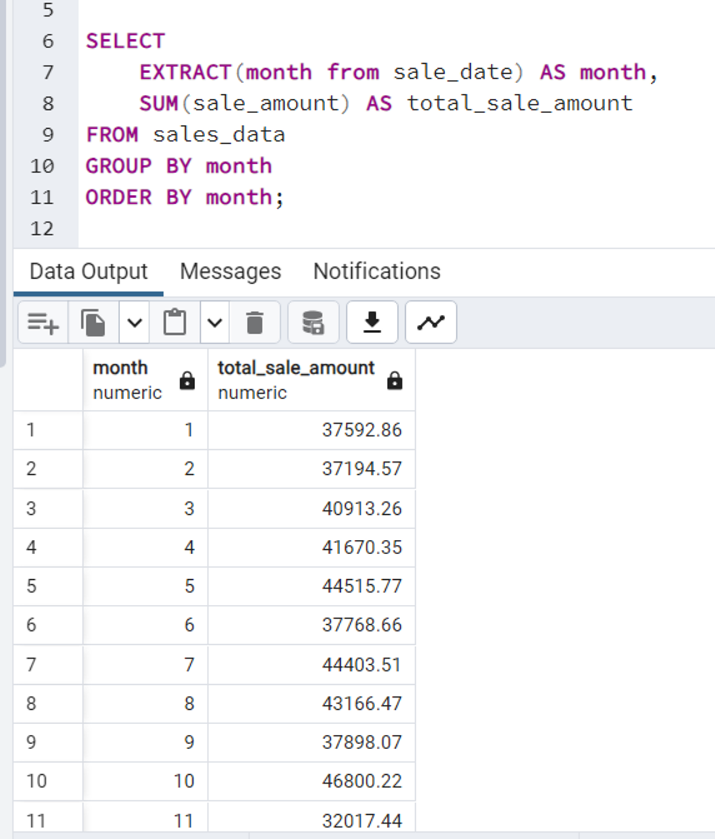
The code for this part can be found in files 01\_Table\_Creation\_Yauheniya\_Kavalchuk.sql and in 02\_Data\_Insertion\_Yauheniya\_Kavalchuk.sql.

**Query results**

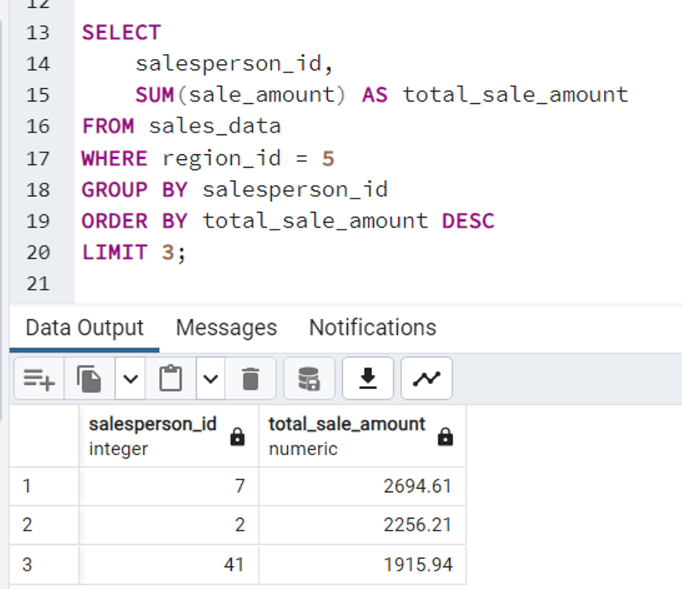
1. Retrieve all sales in a specific month



1. Calculate the total **sale\_amount** for each month



* Identify the top three **salesperson\_id** values by **sale\_amount** within a specific region across all partitions.



The original of the script can be found in 03\_Query\_Partitioning\_Yauheniya\_Kavalchuk.sql.

**Maintenance strategy**

For maintenance strategy automatic partition management was used. It was chosen for several reasons.

1. Simplifies management: With automatic partition creation and deletion, you don't need to manually create and manage partitions as your data grows and changes. This can simplify management and reduce the risk of errors.
2. Improves performance: Automatic partition creation and deletion can help to improve query performance by ensuring that data is stored in the appropriate partition. This can reduce the amount of data that needs to be scanned for a given query, which can improve query performance.
3. Reduces storage costs: Automatic partition deletion can help to reduce storage costs by allowing you to easily delete old data that is no longer needed. This can help to free up storage space and reduce costs.
4. Supports data retention policies: Automatic partition deletion can also help to support data retention policies by allowing you to easily delete data that is no longer needed after a certain period of time.

For implementing automation pg\_cron extension was used. The pg\_cron extension is a PostgreSQL extension that allows you to schedule and run cron-like jobs directly from within the database. It provides a simple and convenient way to schedule tasks such as backups, data exports, and other maintenance tasks.

The code of the pg\_cron script can be found in file 04\_Maintenance\_Yauheniya\_Kavalchuk.sql.

**Personal reflection**

During this task I revised what are the main types of table partitioning, their pros and cons. Also it was beneficial to ‘touch it with my own hands’ and learn how to do partitioned tables in postgresql and also how to maintain them.

For me the most challenging parts were inserting data, because I had to come up with a way of generating data instead of writing it by hand and also thinking of a more efficient way to maintain them.