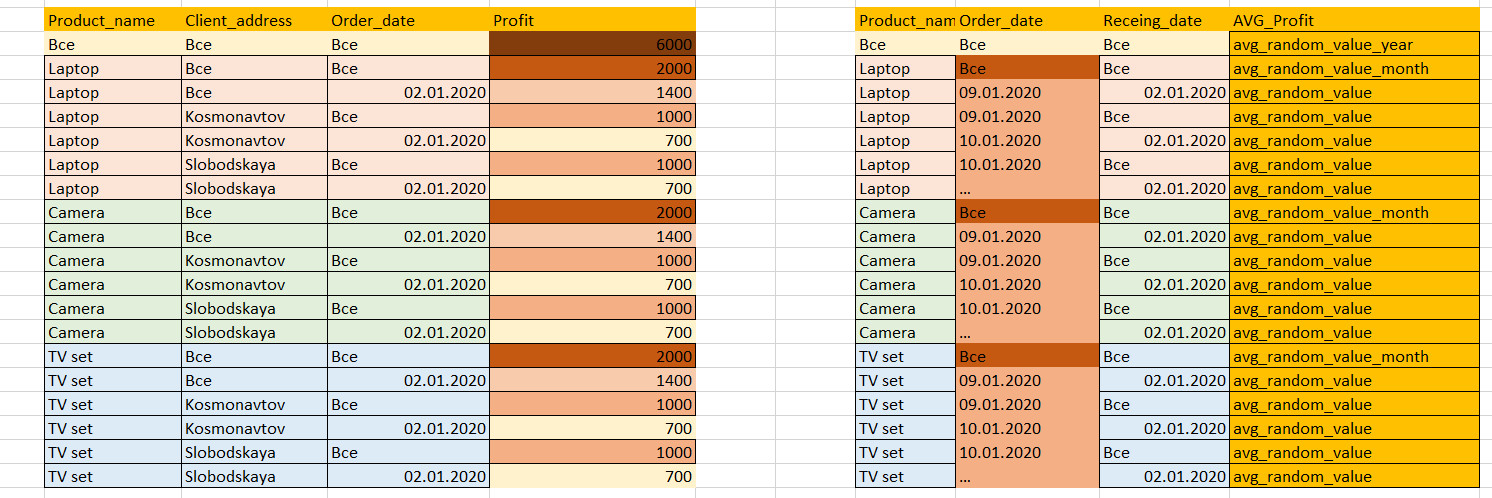
**Лабораторная работа №2**

**Невейков Андрей, 2022**

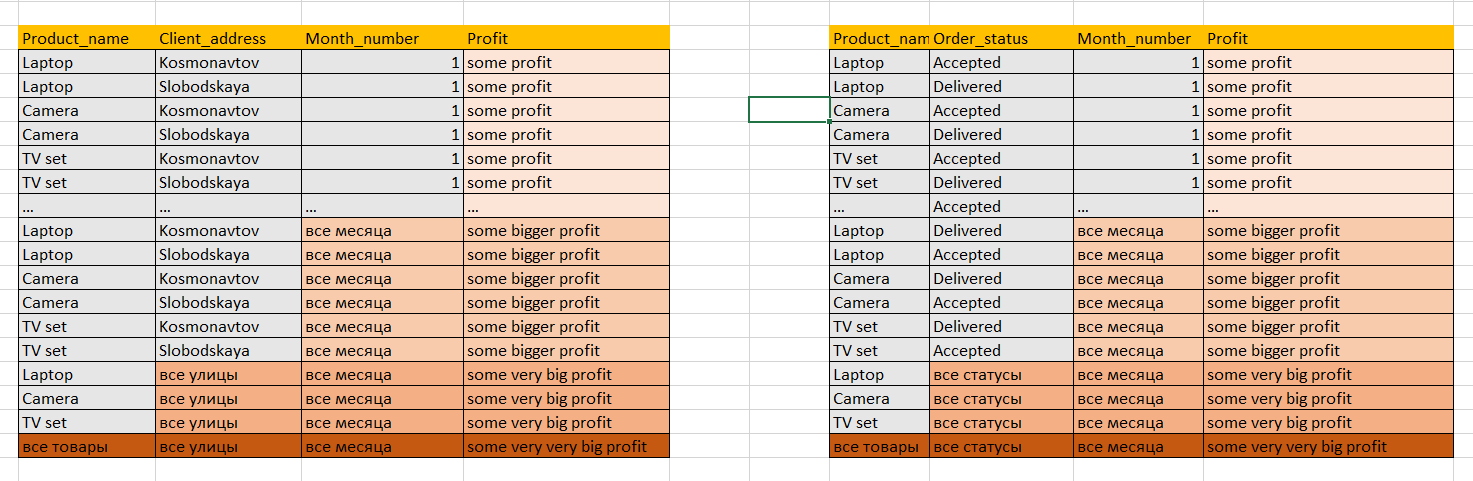
**All scripts here:**

**https://github.com/AndreyNeveikov/DataMola\_Labs/blob/main/Modul\_2/lab2/Reports\_task3-5.sql**

Task 01: CREATE Daily Reports Layouts



Task 02: CREATE Monthly Reports Layouts



**SUMMARY “TASK\_1-2”**

1. **Я научился** предугадывать какой результат я получу, используя GROUPING

BY (CUBE, POLLUP, GROUPING SET).

1. **Я узнал**, что очень сложно с первого раза получить результат, который

задумал и нужно полное понимание процесса работы группировок, а для

полного понимания процесса нужна и теория, и практика;

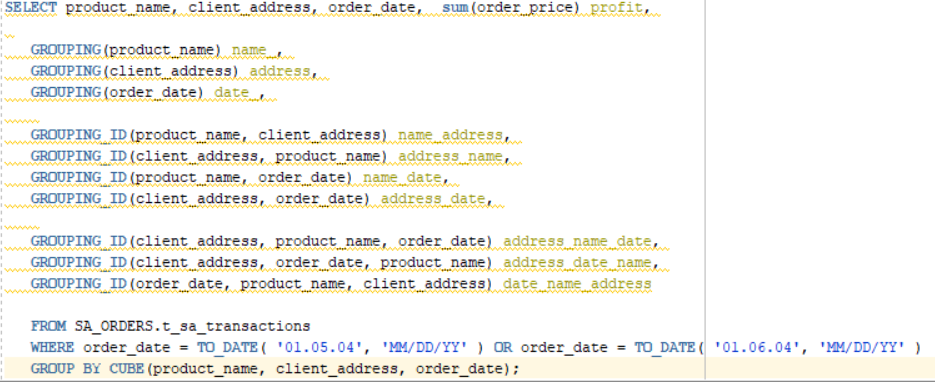
1. **Я понял**, принцип работы различных типов GROUP BY.

Task 03: CREATE Test AdHoc SQL - Daily Reports (CUBE)

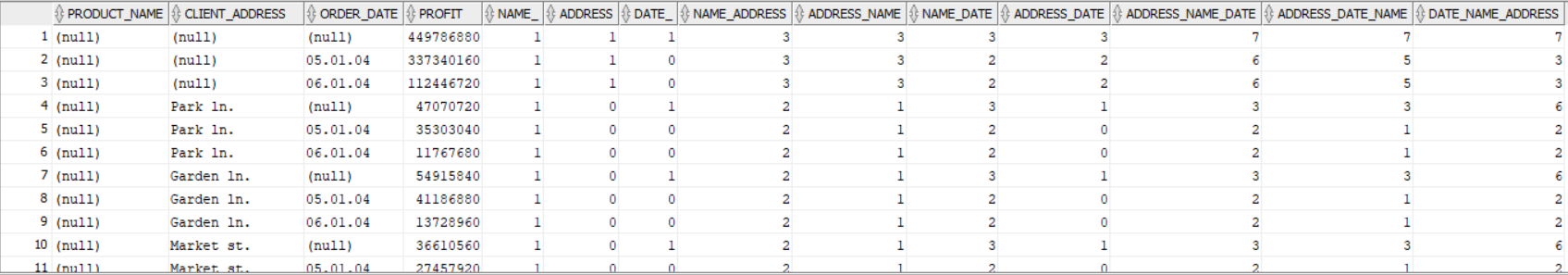
**CUBE** — оператор PL/SQL, который формирует результаты для всех возможных перекрестных вычислений.

**GROUPING** – функция PL/SQL, которая возвращает истину (1), если указанное выражение является статистическим, и ложь (0), если выражение нестатистическое (сами данные).

**GROUPING\_ID** возвращает число, соответствующее **GROUPING** битовому вектору, связанному со строкой. С помощью **GROUPING\_ID** вы можете избежать необходимости в нескольких **GROUPING** функциях и упростить выражение условий фильтрации строк. С фильтрацией строк проще, **GROUPING\_ID** потому что нужные строки можно идентифицировать с помощью одного условия **GROUPING\_ID** = ***n***. Эта функция особенно полезна при хранении нескольких уровней агрегации в одной таблице.



Скрипт выше делает выборку по названию товара, адресу клиента, дате заказа и выводит выручку за день.



1-ая строка – это вычисление по всем товарам, доставленным по всем адресам, за все дни.

Т.к. все выражения статистические, получаем 1 во всех столбцах **GROUPING**-а. **GROUPING\_ID – для двух параметров возвращает 3 т.к. существует 4 возможных варианта:**

|  |  |  |
| --- | --- | --- |
| Null | Null | 3 |
| Null | Знач | 2 |
| Знач | Null | 1 |
| Знач | Знач | 0 |

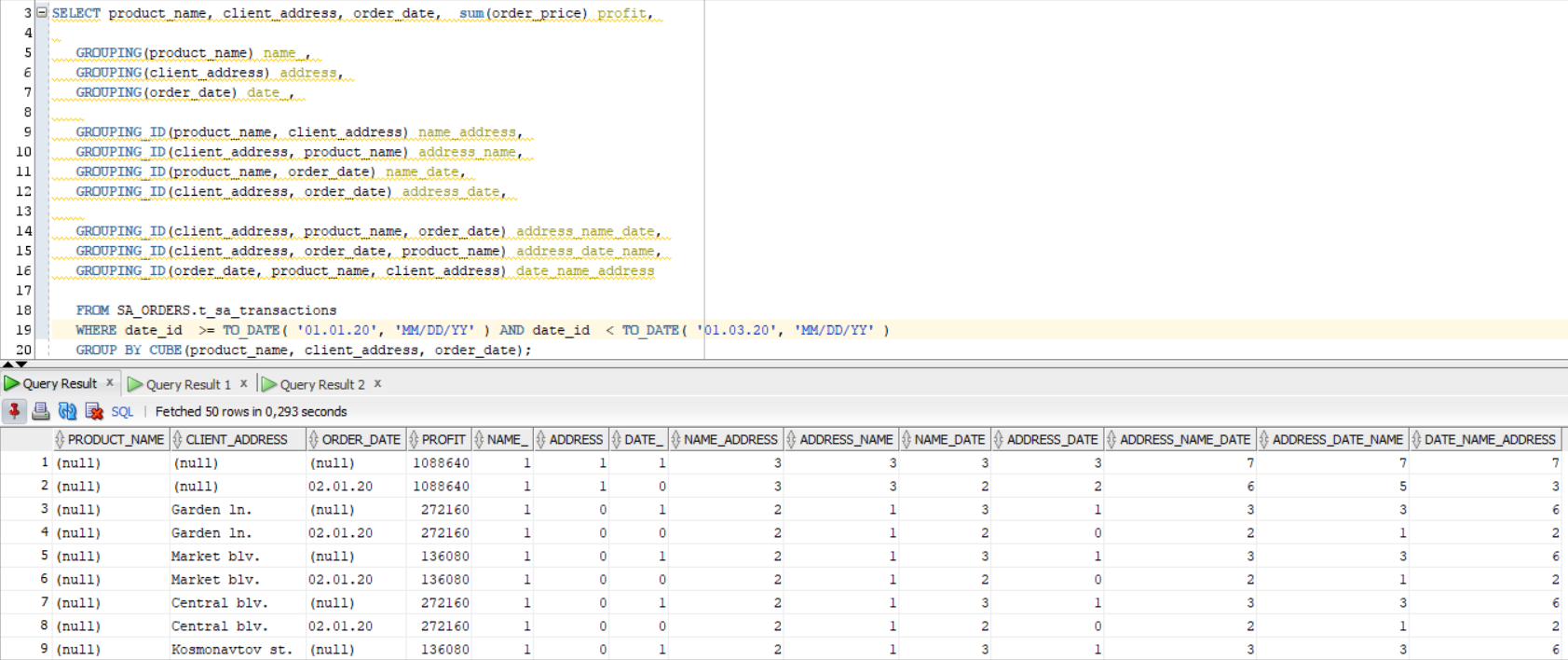
Аналогичный принцип работает и для трех параметров, только возможных вариантов 8.

2^(кол-во парамтров) = 2^3 =8. Для **GROUPING\_ID в порядке 1,2,3**

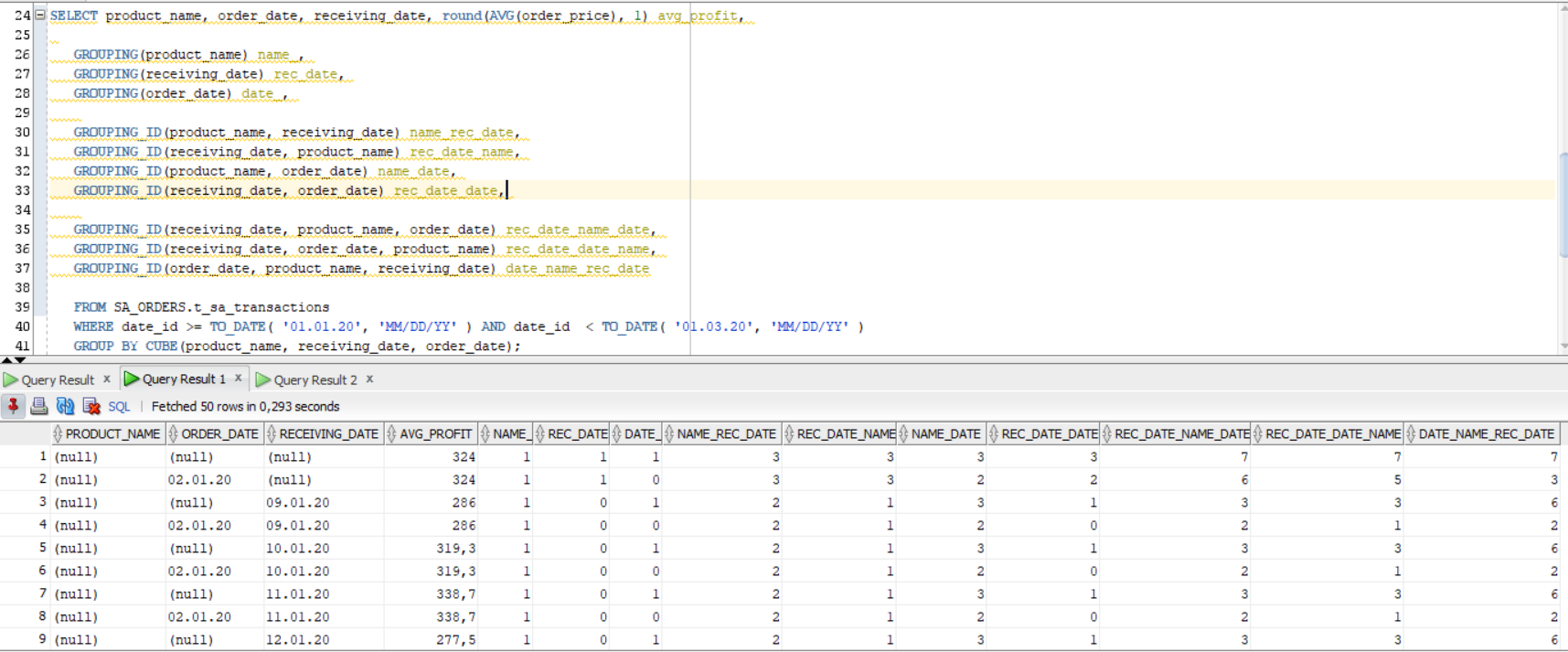
|  |  |  |  |
| --- | --- | --- | --- |
| Null | Null | Null | 7 |
| Null | Null | Знач | 6 |
| Null | Знач | Null | 5 |
| Null | Знач | Знач | 4 |
| Знач | Null | Null | 3 |
| Знач | Null | Знач | 2 |
| Знач | Знач | Null | 1 |
| Знач | Знач | Знач | 0 |

***Daily reports:***

Выручка с каждого вида товара, доставленная по каждому из адресов за 02.01.2020



Выручка с каждого вида товара, заказанного 02.01.2020 на другие дни.



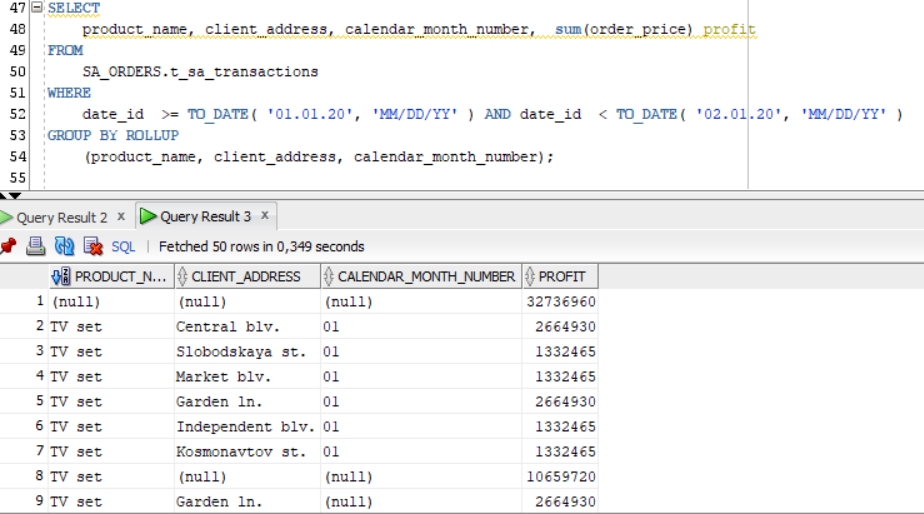
Task 04: CREATE Test AdHoc SQL - Monthly Reports (ROLLUP & GROUPING SETS)

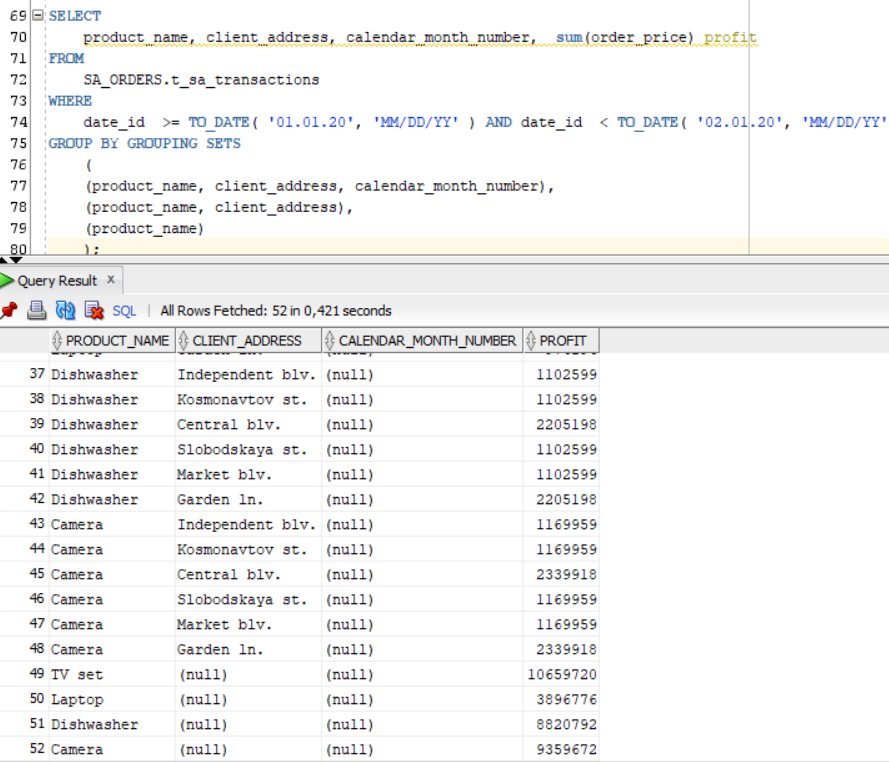
**ROLLUP** – оператор PL/SQL, который формирует промежуточные итоги для каждого указанного элемента и общий итог.

**GROUPING SETS** – оператор PL/SQL, который формирует результаты нескольких группировок в один набор данных, другими словами, он эквивалентен конструкции UNION ALL к указанным группам.

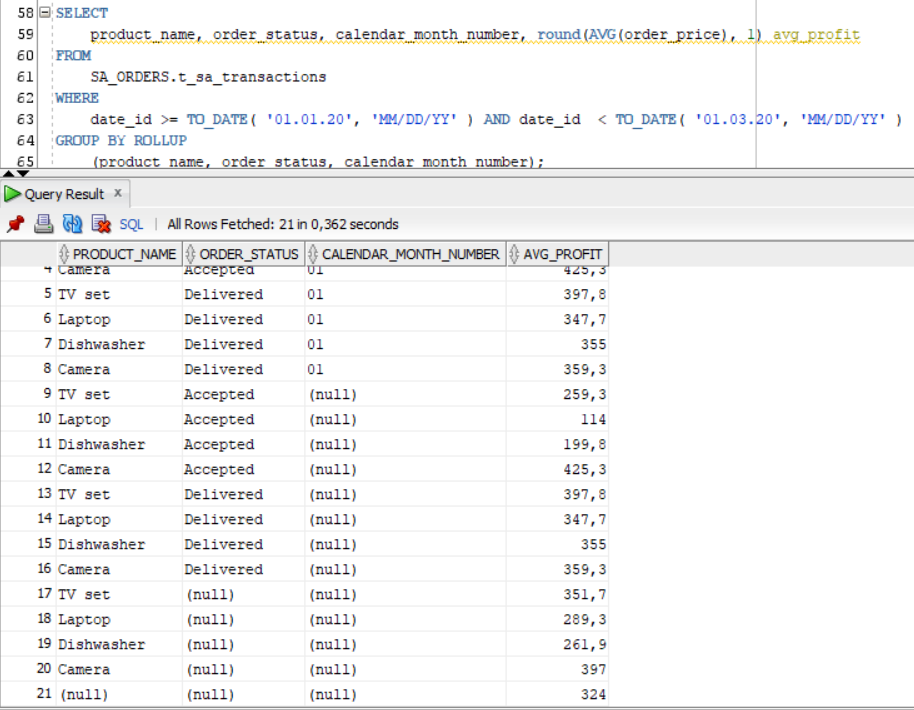
***Monthly reports:***

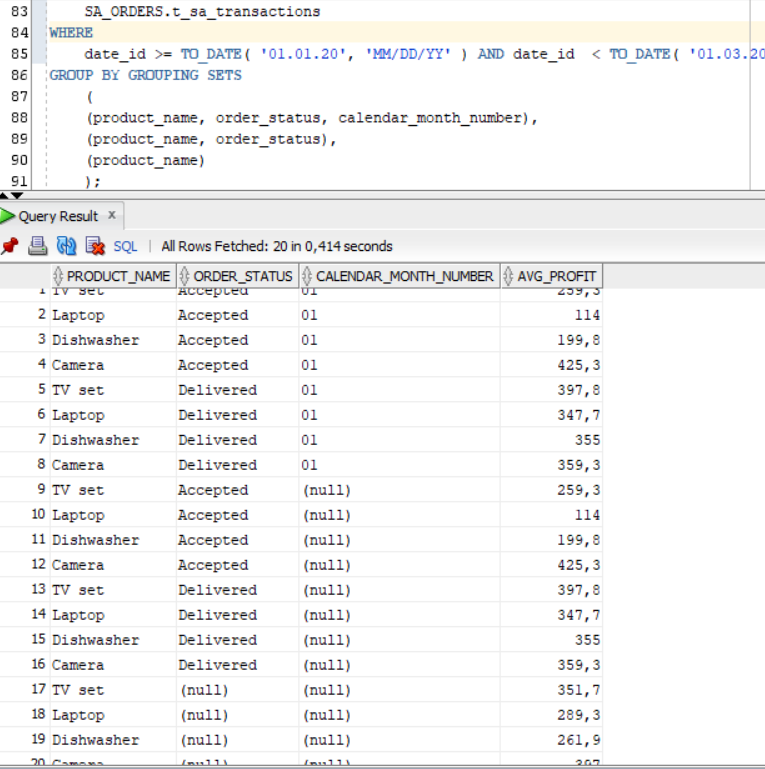
Выручка с каждого вида товара, доставленная по каждому из адресов за январь 2020-го.





Средняя выручка с каждого вида товара, за январь 2020-го, согласно статусу заказа.





**SUMMARY “TASK\_3-4”**

1. **Я научился** получать необходимый результат использования GROUPING

BY (CUBE, POLLUP, GROUPING SET).

1. **Я узнал,** *что очень сложно с первого раза получить результат, который*

*задумал и нужно полное понимание процесса работы группировок,* ***а для***

***полного понимания*** *процесса работы группировок* ***нужна и теория, и***

***практика и много!!!!!!!!;***

1. **Я понял**, принцип работы GROUPING и GROUPING\_ID и описал его выше.

Task 05: CREATE Test AdHoc SQL – ROLLUP by Time

Принцип работы ROLLUP: ROLLUP (d1, d2, d3) создает только четыре набора группировок, предполагая иерархию d1 > d2 > d3 следующим образом:

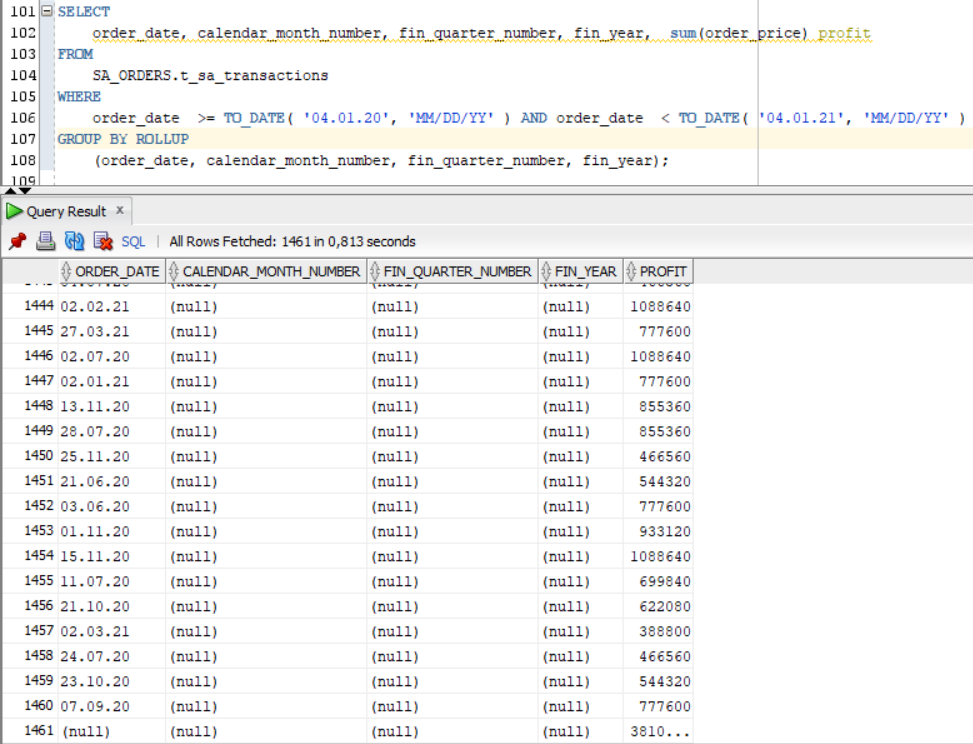
(d1, d2, d3)

(d1, d)

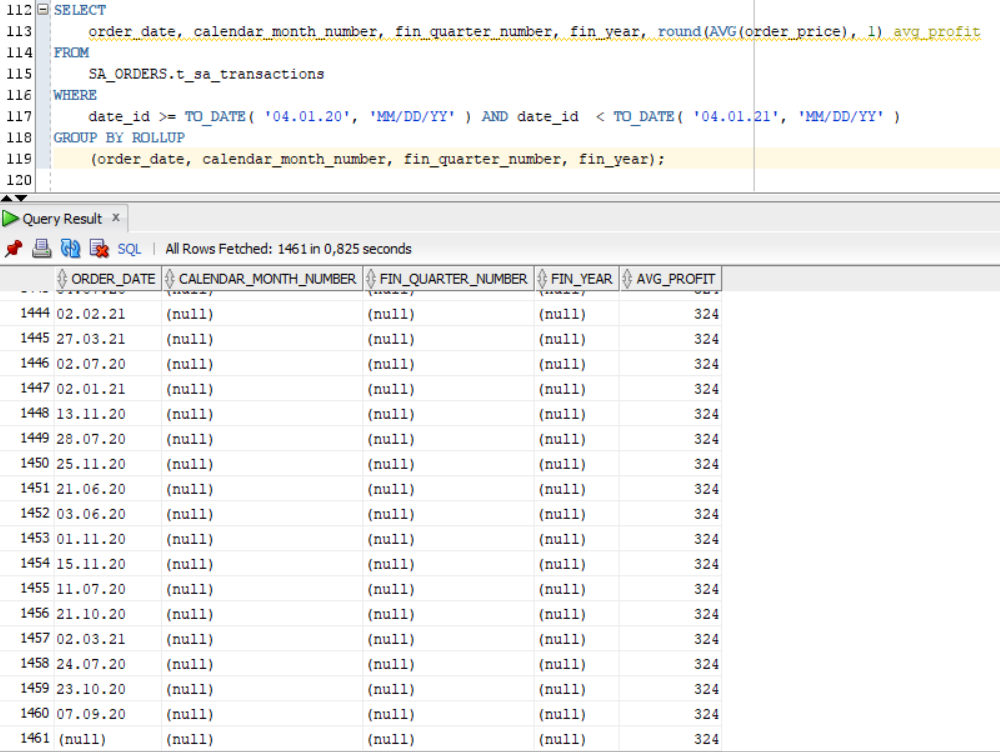
(d1)

()

***Report: суммарная выручка за 2020 финансовый год***



***Report: средняя выручка за 2020 финансовый год***



**SUMMARY “TASK\_5”**

1. **Я научился** делать отчет по иерархии времени.
2. **Я узнал**, что для более наглядных результатов нужно больше параметров,

которые можно агрегировать;

1. **Я понял**, принцип работы ROLLUP BY TIME.

**FINAL SUMMARY**

Данная лабораторная работа …

1. … учит получать отчеты, которые ты изначально хотел получить.
2. … показывает принцип работы GROUP BY - функций.
3. … показывает принцип работы GROUPING\_ID и GROUPING.
4. … учит генерировать отчеты с иерархией.