# Виконання

## представлення у схемі Students:

SELECT \* FROM `data-analytics-mate.Students.v\_Bozhko\_SQL Advanced Module Task`

## створення представлення

CREATE VIEW `Students.v\_Bozhko\_SQL Advanced Module Task`

as

“основний код SQL-запиту”

## основний код SQL-запиту :

WITH account AS

-- інформація по акаунтах

--CTE account У цьому етапі ми об'єднуємо дані з таблиць DA.account, DA.account\_session, DA.session та DA.session\_params, щоб отримати інформацію про акаунти за датою, країною, інтервалом розсилки та статусами верифікації та відписки. Вихідні дані групуються за комбінаціями полів s.date, sp.country, ac.send\_interval, ac.is\_verified та ac.is\_unsubscribed, і підраховується кількість унікальних акаунтів (account\_cnt).

(

SELECT

s.date,

sp.country,

COUNT(DISTINCT a.id) AS account\_cnt,

a.send\_interval AS send\_interval,

a.is\_verified AS is\_verified,

a.is\_unsubscribed AS is\_unsubscribed

FROM `data-analytics-mate.DA.account` a

JOIN `DA.account\_session` acs

on a.id = acs.account\_id

join `DA.session` s

on acs.ga\_session\_id = s.ga\_session\_id

join `DA.session\_params` sp

ON acs.ga\_session\_id = sp.ga\_session\_id

GROUP BY s.date,sp.country, send\_interval, is\_verified, is\_unsubscribed

),

message AS

--інформація по листах

--CTE message (email\_metrics) На цьому етапі збирається інформація про активність електронної пошти (кількість надісланих, відкритих та переглянутих повідомлень) за країною, датою, інтервалом розсилки та іншими параметрами. Ми об’єднуємо таблиці DA.email\_sent, DA.email\_open, DA.email\_visit, DA.account\_session, DA.session, DA.session\_params та DA.account, щоб сформувати ці метрики. Результат групується за комбінацією полів, таких як sent\_date, sp.country, ac.send\_interval, ac.is\_verified та ac.is\_unsubscribed.

(

SELECT

DATE\_ADD(s.date, INTERVAL ems.sent\_date DAY) AS sent\_date,

sp.country,

ems.sent\_date as send\_interval,

a.is\_verified AS is\_verified,

a.is\_unsubscribed AS is\_unsubscribed,

COUNT(DISTINCT ems.id\_message) AS sent\_msg,

COUNT(DISTINCT eo.id\_message) AS open\_msg,

COUNT(DISTINCT ev.id\_message) AS visit\_msg

FROM `data-analytics-mate.DA.account` a

JOIN `DA.account\_session` acs

on a.id = acs.account\_id

join `DA.session\_params` sp

ON acs.ga\_session\_id = sp.ga\_session\_id

JOIN `DA.session` s

ON acs.ga\_session\_id = s.ga\_session\_id

join `DA.email\_sent` ems

ON ems.id\_account = acs.account\_id

LEFT JOIN `DA.email\_open` eo

ON ems.id\_message = eo.id\_message

LEFT JOIN `DA.email\_visit` ev

ON ems.id\_message = ev.id\_message

GROUP BY sent\_date, sp.country, send\_interval, is\_verified, is\_unsubscribed

),

unions as

--дві частини (по акаунтах і листах) поєднані UNION

--CTE unions Цей етап об'єднує результати з попередніх етапів account та email\_metrics. Вихідні дані об’єднуються за допомогою оператора UNION ALL, щоб отримати повну таблицю з усіма метриками та параметрами.

(

SELECT

account.date AS date,

account.country as country,

account.account\_cnt,

account.send\_interval AS send\_interval,

account.is\_verified AS is\_verified,

account.is\_unsubscribed AS is\_unsubscribed,

0 AS sent\_msg,

0 AS open\_msg,

0 AS visit\_msg

FROM account

UNION ALL

SELECT

sent\_date AS date,

message.country AS country,

0 AS account\_cnt,

message.send\_interval AS send\_interval,

message.is\_verified AS is\_verified,

message.is\_unsubscribed AS is\_unsubscribed,

message.sent\_msg,

message.open\_msg,

message.visit\_msg

FROM message

),

final\_groups as

--загальна кількість створених акаунтів та відправлених листів по країні;

--CTE final\_groups У цьому етапі ми підсумовуємо об'єднані дані за датою, країною, інтервалом розсилки та іншими параметрами. Результатом є зведена таблиця, в якій обчислюються загальні кількості для кожної комбінації параметрів.

(

SELECT

date,

country,

send\_interval,

is\_verified,

is\_unsubscribed,

SUM(account\_cnt) AS account\_cnt,

SUM(sent\_msg) AS sent\_msg,

SUM(open\_msg) AS open\_msg,

SUM(visit\_msg) AS visit\_msg

FROM unions

GROUP BY date,

country,

send\_interval,

is\_verified,

is\_unsubscribed

),

sums AS

-- використання віконний функцій для ранжування країн

-- CTE sums Цей етап обчислює загальну кількість акаунтів (total\_country\_account\_cnt) і надісланих повідомлень (total\_country\_sent\_cnt) для кожної країни, використовуючи аналітичні функції SUM OVER. Після цього застосовується DENSE\_RANK, щоб кожній країні присвоїти рейтинг за кількістю акаунтів і кількістю надісланих повідомлень.

(

SELECT

date,

country,

send\_interval,

is\_verified,

is\_unsubscribed,

account\_cnt,

sent\_msg,

open\_msg,

visit\_msg,

SUM(account\_cnt) OVER(PARTITION BY country ) AS total\_country\_account\_cnt,

SUM(sent\_msg) OVER(PARTITION BY country ) AS total\_country\_sent\_cnt,

DENSE\_RANK() OVER( ORDER BY account\_cnt DESC) AS rank\_total\_country\_account\_cnt,

DENSE\_RANK() OVER( ORDER BY sent\_msg DESC) AS rank\_total\_country\_sent\_cnt

FROM final\_groups

)

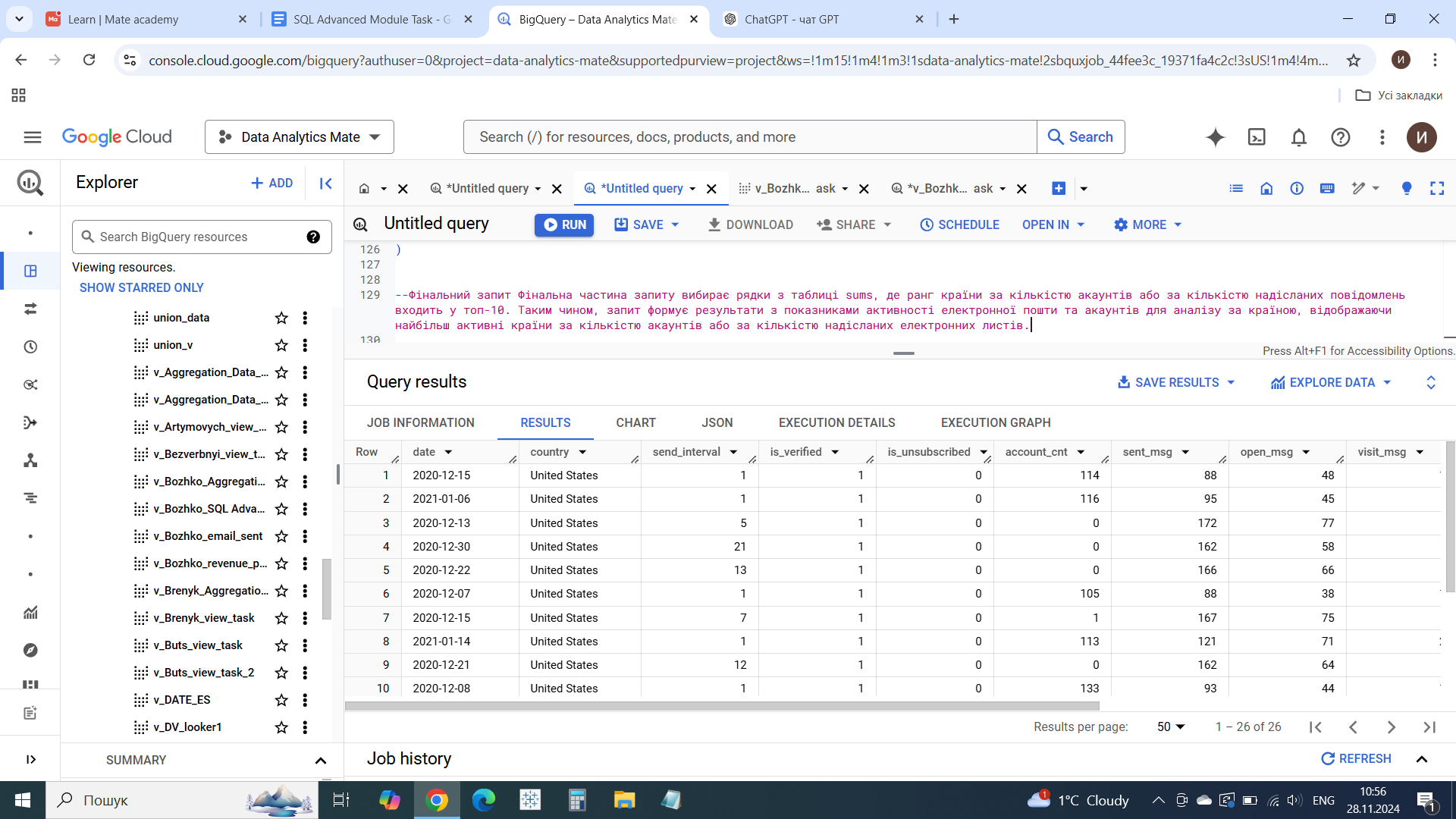
--Фінальний запит Фінальна частина запиту вибирає рядки з таблиці sums, де ранг країни за кількістю акаунтів або за кількістю надісланих повідомлень входить у топ-10. Таким чином, запит формує результати з показниками активності електронної пошти та акаунтів для аналізу за країною, відображаючи найбільш активні країни за кількістю акаунтів або за кількістю надісланих електронних листів.

SELECT \*

FROM sums

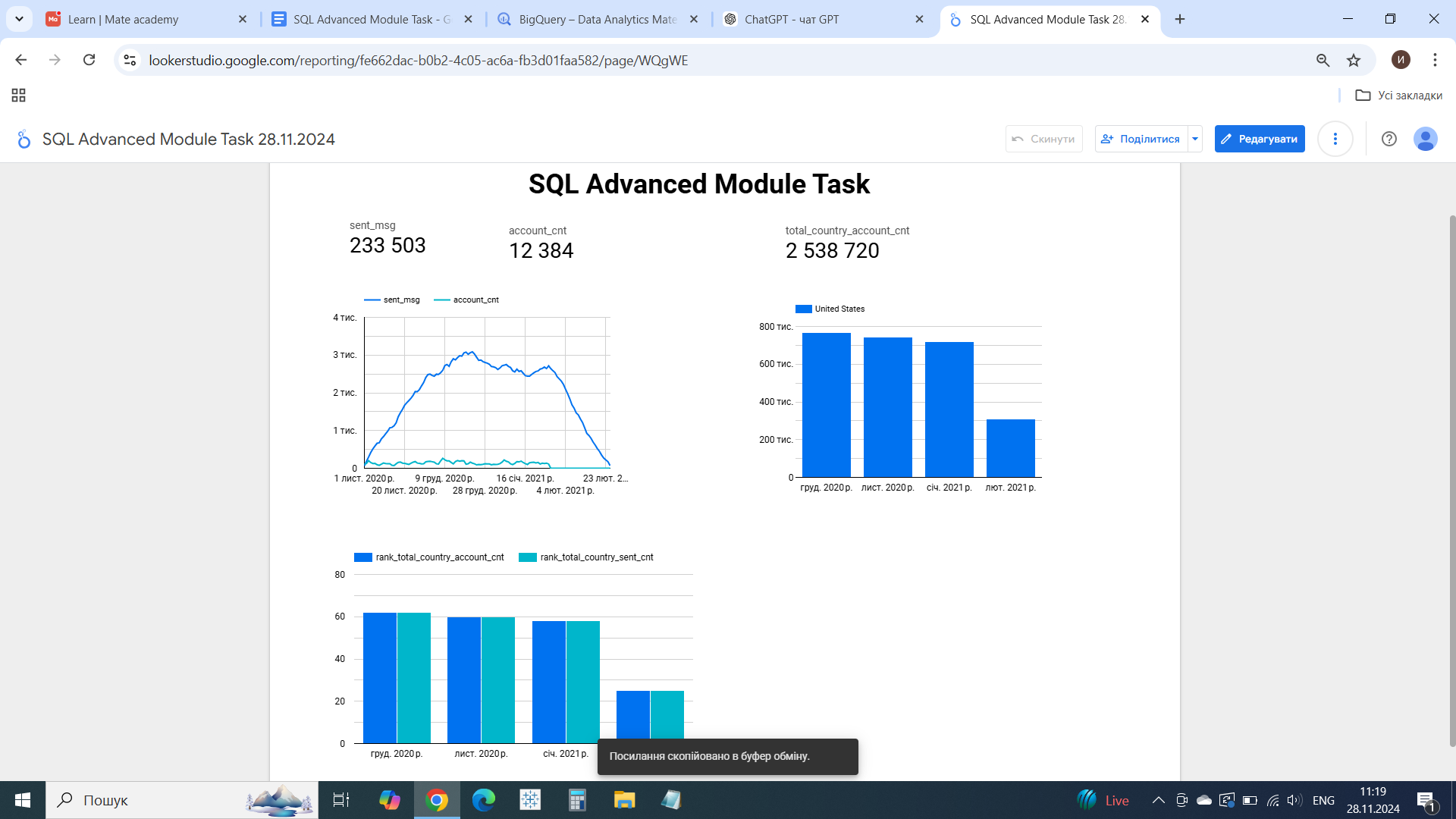
where rank\_total\_country\_account\_cnt <= 10 or rank\_total\_country\_sent\_cnt <=10

## скріншот результату:



## **Скріншот** створеної візуалізації в Looker Studio.

<https://lookerstudio.google.com/reporting/fe662dac-b0b2-4c05-ac6a-fb3d01faa582>



# Завдання модуля SQL Advanced

Для виконання завдання використовуй e-commerce базу даних, з якою ми працювали в BigQuery. Напиши один SQL-запит для створення набору даних. Додатково створи візуалізацію в Looker Studio на основі отриманих даних.

### Опис

Необхідно зібрати дані, які допоможуть аналізувати динаміку створення акаунтів, активність користувачів за листами (відправлення, відкриття, переходи), а також оцінювати поведінку в категоріях, таких як інтервал відправлення, верифікація акаунтів і статус підписки. Дані дозволять порівнювати активність між країнами, визначати ключові ринки, сегментувати користувачів за різними параметрами.

### Вимоги до SQL-запиту

Результат запиту повинен містити певний перелік полів групування - тобто категоріальних значень (з якими не проводиться ніяких обчислень). Основні метрики по акаунтам, а також основні метрики по емейлам, повинні розраховуватися в цих розрізах:

* date — дата (для акаунтів - дата створення акаунта, для емейлів - дата відправки емейла);
* country — країна;
* send\_interval — інтервал відправлення, встановлений акаунтом;
* is\_verified — перевірено акаунт чи ні;
* is\_unsubscribed — підписник відписався чи ні.

**Важливо**! Ці розрізи стосуються як метрик акаунтів, так і метрик емейлів. Потрібна інформація є в таблиці account.

Твій запит має розраховувати інформацію (в розрізі перелічених полів) по наступним основним метриках:

* account\_cnt — кількість створених акаунтів;
* sent\_msg — кількість відправлених листів;
* open\_msg — кількість відкритих листів;
* visit\_msg — кількість переходів по листах;

А також по додатковим метриках (які розраховуються на базі основних метрик):

* total\_country\_account\_cnt — загальна кількість створених підписників в цілому по країні;
* total\_country\_sent\_cnt — загальна кількість відправлених листів в цілому по країні;
* rank\_total\_country\_account\_cnt — рейтинг країн за кількістю створених підписників в цілому по країні;
* rank\_total\_country\_sent\_cnt — рейтинг країн за кількістю відправлених листів в цілому по країні.

Метрики для акаунтів і емейлів потрібно розраховувати окремо, щоб зберегти унікальні розрізи для кожного та уникнути конфліктів через різну логіку використання поля date. Для об’єднання результатів використовуй UNION. У фіналі залиш лише ті записи, де rank\_total\_country\_account\_cnt або rank\_total\_country\_sent\_cnt менше або дорівнює 10.

Обов'язково використай хоча б одне CTE, винеси туди логічні частини запиту. Для розрахунку рангу використовуй функції-вікна.

### **Структура набору**

Запит повинен виводити такі стовпчики:

* date — дата;
* country — країна;
* send\_interval — інтервал відправлення;
* is\_verified — перевірено акаунт чи ні;
* is\_unsubscribed — підписник відписався;
* account\_cnt — кількість створених акаунтів;
* sent\_msg — кількість відправлених листів;
* open\_msg — кількість відкритих листів;
* visit\_msg — кількість переходів по листах;
* total\_country\_account\_cnt — загальна кількість створених підписників по країні;
* total\_country\_sent\_cnt — загальна кількість відправлених листів по країні;
* rank\_total\_country\_account\_cnt — рейтинг країн за кількістю створених підписників;
* rank\_total\_country\_sent\_cnt — рейтинг країн за кількістю відправлених листів.

### **Візуалізація**

* Створи візуалізацію в Looker Studio, яка показуватиме загальні значення в розрізі країн для таких полів:
  + account\_cnt;
  + total\_country\_sent\_cnt;
  + rank\_total\_country\_account\_cnt;
  + rank\_total\_country\_sent\_cnt.
* Покажи динаміку по даті для поля sent\_msg.

### **Формат оформлення результатів**

1. **Документ**: додай код SQL-запиту в документ. Додай коментарі до основних частин коду для пояснення логіки.
2. **Скріншот**: додай в документ скріншот створеної візуалізації в Looker Studio.

❗️ **Посилання на документ з описом результатів прикріпи в завданні SQL Advanced Module Task**.