

## 1. SQL-аналітика

На основі вибірки за 2 місяці написати SQL-запити, що дозволять отримати:

- ARPU / ARPPU (gross/net) на користувача

**WITH**

*customer\_balance* **AS** (

**SELECT**

customer\_id,

**SUM**(price\_usd) **AS** customer\_revenue\_gross\_after\_refunds,

**SUM**(price\_usd + transaction\_fee) **AS** customer\_revenue\_net\_after\_refunds,

**SUM**(price\_usd) **FILTER** (**WHERE** price\_usd > 0) **AS** customer\_revenue\_gross\_before\_refunds,

**SUM**(price\_usd + transaction\_fee) **FILTER** (**WHERE** price\_usd > 0) **AS** customer\_revenue\_net\_before\_refunds,

**MAX**(**CASE WHEN** order\_status = 'shipped' **THEN** 1 **ELSE** 0 **END**) **AS** has\_shipped\_order

**FROM** subscriptions\_data

**GROUP BY** customer\_id

),

*metrics\_calculation* **as** (

**SELECT**

-- Customers

**COUNT**(customer\_id) **AS** customers,

**COUNT**(customer\_id) **FILTER** (**WHERE** has\_shipped\_order > 0) **AS** customers\_paid\_before\_refunds,

**COUNT**(customer\_id) **FILTER** (**WHERE** customer\_revenue\_gross\_after\_refunds > 0) **AS**

customers\_paid\_after\_refunds,

-- Revenue after refunds Gross / Net

**SUM**(customer\_revenue\_gross\_after\_refunds) **AS** revenue\_after\_refunds\_gross,

**SUM**(customer\_revenue\_net\_after\_refunds) **AS** revenue\_after\_refunds\_net,

-- Revenue before refunds Gross / Net

**SUM**(customer\_revenue\_gross\_before\_refunds) **AS** revenue\_before\_refunds\_gross,

**SUM**(customer\_revenue\_net\_before\_refunds) **AS** revenue\_before\_refunds\_net

**FROM** customer\_balance

)

**select**

customers, customers\_paid\_before\_refunds, customers\_paid\_after\_refunds,

revenue\_after\_refunds\_gross, revenue\_after\_refunds\_net,

-- Gross ARPU / ARPPU

**ROUND**(revenue\_after\_refunds\_gross / **NULLIF**(customers, 0), 2) **AS** arpu\_gross,

**ROUND**(revenue\_after\_refunds\_gross / **NULLIF**(customers\_paid\_after\_refunds, 0), 2) **AS** arppu\_gross,

-- Net ARPU / ARPPU

**ROUND**(revenue\_after\_refunds\_net / **NULLIF**(customers, 0), 2) **AS** arpu\_net,

**ROUND**(revenue\_after\_refunds\_net / **NULLIF**(customers\_paid\_after\_refunds, 0), 2) **AS** arppu\_net

**from** metrics\_calculation

;

customer s	customers_pai d_before_refu nds	customers_p aid_after_ref unds	revenue_a fter_refund s_gross	revenue_aft er_refunds_ net	arpu_gross	arppu_gross	arpu_net	arppu_net
1146	867	642	31336.87	29043.31	27.34	48.81	25.34	45.24

- Топ-3 країни з найбільшим доходом

**WITH**

country\_balance **AS** (

**SELECT**

country,

**SUM**(price\_usd) **AS** revenue\_gross\_after\_refunds,

**SUM**(price\_usd + transaction\_fee) **AS** revenue\_net\_after\_refunds

**FROM** subscriptions\_data

**GROUP BY** country

)

**select** \* **from** country\_balance

**order by** revenue\_gross\_after\_refunds **desc**

**limit** 3;

country	revenue_gross_after_refunds	revenue_net_after_refunds
US	17615.71	16525.54
AU	1634.53	1453.63
GB	1284.65	1152.31

- Топ платників – користувачів, котрі витратили найбільше грошей на продукти

**WITH**

customer\_balance **AS** (

**SELECT**

customer\_id,

**SUM**(price\_usd) **AS** revenue\_gross\_after\_refunds,

**SUM**(price\_usd + transaction\_fee) **AS** revenue\_net\_after\_refunds

**FROM** subscriptions\_data

**GROUP BY** customer\_id

)

**select** \* **from** customer\_balance

**order by** revenue\_gross\_after\_refunds **desc**

**limit** 10;

customer_id	revenue_gross _after_refunds	revenue_net_aft er_refunds
6240901	99.95	96.15
6252859	99.95	96.15
6244732	99.95	96.15
6242533	99.95	94.65
6255944	99.95	96.15
6250204	89.97	85.41
6252992	89.97	85.41
6254805	89.97	86.76
5535203	89.97	85.41
6254228	89.97	85.41

- Найпопулярнішу суму оплати

WITH

prices AS (

SELECT

price\_usd,

count(distinct customer\_id) as customers,

SUM(price\_usd) AS revenue\_gross,

SUM(price\_usd + transaction\_fee) AS revenue\_net

FROM subscriptions\_data

GROUP BY price\_usd

)

select \* from prices

order by revenue\_gross desc

limit 1;

price_usd	customers	revenue_gross	revenue_net
49.99	769	38442.31	36850.56

## 2. Дашборди

Необхідно створити дашборди в BI-системі або іншому інструменті для візуалізації, які наочно відобразять:

- Структура витрат – категорії витрат, їх динаміка та розподіл
- Структура помилок при оплатах. Які помилки отримуємо найчастіше та на яких сегментах.

*Очікуваний результат:* Дашборди у BI-інструменті або скріншоти з поясненням.

У файлі 'Dashboard (Task 2-3).pdf' (або у Dashboard.pbix) показані три Дашборди з аналізом структури Revenue, Transaction Fee та Errors. Також прикріплені сторінки з описом та аналізом bottlenecks та шляхами їх усунення.

## 3. Аналітика та рекомендації

На основі отриманих даних:

- Визначити ключові bottlenecks у продукті
- Запропонувати конкретні покращення для їх усунення та визначити ключові метрики

У файлі 'Dashboard (Task 2-3).pdf' (або у Dashboard.pbix) показані три Дашборди з аналізом структури Revenue, Transaction Fee та Errors. Також прикріплені сторінки з описом та аналізом bottlenecks та шляхами їх усунення.