

Завдання 1

- 1. Вивести 5 останніх записів про оплату, що здійснилися в період з 14 по 16 квітня, та в яких сума оплати складала менше 10\$.

```
SELECT payments.user_id,
       payments.pay_dt,
       payments.amount_usd,
       feature_type_dict.feature_type_name
FROM payments
INNER JOIN feature_type_dict
ON payments.feature_type_id = feature_type_dict.feature_type_id
WHERE (DATE(pay_dt) BETWEEN '2024-04-14' AND '2024-04-16')
AND amount_usd <= 10
# ORDER BY pay_dt DESC (Останніх по даті ?)
LIMIT 5;
```

РЕЗУЛЬТАТ ЗАПИТУ:

	<input type="checkbox"/> user_id ▾	<input type="checkbox"/> pay_dt ▾	<input type="checkbox"/> amount_usd ▾	<input type="checkbox"/> feature_type_name ▾
1	20737	2024-04-16 15:52:00	5.99	premium
2	20885	2024-04-16 01:28:26	5.99	premium
3	20906	2024-04-14 01:50:56	5.99	premium
4	21429	2024-04-14 11:39:55	5.99	premium
5	22467	2024-04-15 18:40:03	5.99	premium

Завдання 2

2. По кожній feature_name порахувати:
- загальну кількість оплат
 - середню сумму оплат
 - загальну суму оплат
 - середню кількість оплат на користувача, що здійснив оплату

```
SELECT feature_type_name,
       COUNT(user_id) AS total_number_of_payments,
       AVG(amount_usd) AS average_pay,
       SUM(amount_usd) AS total_money,
       COUNT(user_id) / COUNT(DISTINCT user_id) AS average_count_per_payer
FROM payments
INNER JOIN feature_type_dict
ON payments.feature_type_id = feature_type_dict.feature_type_id
GROUP BY feature_type_name;
```

РЕЗУЛЬТАТ ЗАПИТУ:

	feature_type_name	total_number_of_payments	average_pay	total_money	average_count_per_payer
1	premium	914	6.816279938758332	6230.079864025116	1.5868
2	vip	246	19.1476014687763	4710.30996131897	1.6078

Завдання 3

3. Порахувати співвідношення лайків до всіх реакцій у розрізі по додатку для користувачів не з США. Реакцією вважаємо одну з двох активностей: лайк або дизлайк.

```
SELECT
  app,
  SUM(CASE WHEN activity_name IN ('like') THEN 1 ELSE 0 END) /
  SUM(CASE WHEN activity_name IN ('like', 'dislike') THEN 1 ELSE 0 END) AS
like_to_reaction_without_usa
FROM activities
INNER JOIN activity_type_dict
ON activities.activity_type = activity_type_dict.activity_type
INNER JOIN registrations
ON registrations.user_id = activities.user_id
INNER JOIN country_info_dict
ON registrations.country_code = country_info_dict.country_code
WHERE country_name != 'United States of America'
GROUP BY app;
```

РЕЗУЛЬТАТ ЗАПИТУ:

	app	like_to_reaction_without_usa
1	ios	0.3079
2	mobile	0.2585
3	desktop	0.4533
4	android	0.2894

Завдання 4

4. Серед тих, у кого назва країни Ukraine або USA, у розрізі гендеру, порахувати долю зареєстрованих користувачів, які не здійснили лайк.

```
SELECT  t1.gender,
        (t2.dont_do_like / t1.total_users) * 100 AS percent_of_users_without_like
#      ,t2.dont_do_like AS users_without_like,
#      t1.total_users AS total_users_ua_us
FROM
(
    SELECT
        COUNT(DISTINCT registrations.user_id) AS total_users,
        registrations.gender
    FROM  registrations
    INNER JOIN  activities ON activities.user_id = registrations.user_id
    LEFT JOIN  country_info_dict ON registrations.country_code = country_info_dict.country_code
    LEFT JOIN  activity_type_dict ON activities.activity_type =
activity_type_dict.activity_type
    WHERE      country_info_dict.country_name IN ('United States of America', 'Ukraine')
    GROUP BY  gender
) AS t1
JOIN
(
    SELECT gender,
        COUNT(DISTINCT registrations.user_id) AS dont_do_like
    FROM registrations
    INNER JOIN activities
    ON activities.user_id = registrations.user_id
    LEFT JOIN country_info_dict
    ON registrations.country_code = country_info_dict.country_code
    LEFT JOIN activity_type_dict
    ON activities.activity_type = activity_type_dict.activity_type
    WHERE registrations.user_id NOT IN (
        SELECT DISTINCT registrations.user_id
        FROM registrations
        INNER JOIN activities
        ON activities.user_id = registrations.user_id
        LEFT JOIN country_info_dict
        ON registrations.country_code = country_info_dict.country_code
        LEFT JOIN activity_type_dict
        ON activities.activity_type = activity_type_dict.activity_type
        WHERE activity_type_dict.activity_name = 'like'
        AND country_info_dict.country_name IN ('United States of America','Ukraine'))

AND country_info_dict.country_name IN ('United States of America', 'Ukraine')
GROUP BY gender
) AS t2
ON
    t1.gender = t2.gender;
```

(Створюємо 2 окремі таблиці для рахунку всіх юзерів з Ukraine та United States of America (t1) та окремо рахуємо таблицю для юзерів з Ukraine та United States of America які не поставили лайк (t2) після цього об'єднуємо ці дві таблиці по гендеру)

РЕЗУЛЬТАТ ЗАПИТУ:

	gender ▾	percent_of_users_without_like ▾
1	f	58.1871
2	m	56.9288

Завдання 5

5. Порахувати скільки лайків та скільки дизлайків здійснив кожен чоловік в першу сесію. Вивести топ-5 чоловіків по дизлайкам.

```
SELECT sessions.user_id,
       SUM(CASE WHEN activity_name = 'like' THEN 1 ELSE 0 END) AS count_like,
       SUM(CASE WHEN activity_name = 'dislike' THEN 1 ELSE 0 END) AS count_dislike
FROM   sessions
RIGHT JOIN activities
ON sessions.user_id = activities.user_id
INNER JOIN activity_type_dict
ON activities.activity_type = activity_type_dict.activity_type
WHERE session_number = 1 AND dt BETWEEN session_start_dt AND DATE_ADD(session_start_dt, INTERVAL
duration SECOND)
GROUP BY sessions.user_id
ORDER BY count_dislike DESC
LIMIT 5;
```

РЕЗУЛЬТАТ ЗАПИТУ:

	user_id	count_like	count_dislike
1	32537	0	10
2	31657	3	10
3	32375	0	9
4	22968	0	9
5	32621	1	9

Завдання 6

6. Навести приклади двох запитів, які можна реалізувати як через віконну функцію, так і без неї. Відповідь на запитання подати у форматі:
- словесний запит
 - реалізація в SQL через віконну функцію
 - реалізація в SQL без віконної функції

Перший запит:

- 1) Словесний запит:
“Вивести інформацію про юзера (реєстраційні дані) та скільки всього він витратив коштів за весь час”
- 2) Реалізація в SQL через віконну функцію

```
SELECT DISTINCT registrations.*,
                SUM(payments.amount_usd) OVER (PARTITION BY registrations.user_id) AS spend_money
FROM registrations
INNER JOIN payments
ON registrations.user_id = payments.user_id
ORDER BY spend_money DESC;
```

РЕЗУЛЬТАТ ЗАПИТУ:

	user_id	reg_dt	gender	age	app	country_code	spend_money
1	30845	2024-04-16 10:03:57	m	32	mobile	US	1223.9699954986572
2	31737	2024-04-17 13:32:01	f	25	mobile	UA	120.12999725341797
3	23733	2024-04-08 12:45:52	f	25	mobile	UA	113.53999710083008
4	30229	2024-04-15 16:42:39	f	25	mobile	UA	113.419997215271
5	30063	2024-04-15 10:36:51	f	25	mobile	UA	107.50999736785889

- 3) Реалізація в SQL без віконної функції

```
SELECT registrations.*,
       spent.spend_money
FROM registrations
INNER JOIN (SELECT user_id,
                  SUM(amount_usd) AS spend_money
            FROM payments
            GROUP BY user_id) spent
ON spent.user_id = registrations.user_id
ORDER BY spend_money DESC;
```

РЕЗУЛЬТАТ ЗАПИТУ:

	user_id	reg_dt	gender	age	app	country_code	spend_money
1	30845	2024-04-16 10:03:57	m	32	mobile	US	1223.9699954986572
2	31737	2024-04-17 13:32:01	f	25	mobile	UA	120.12999725341797
3	23733	2024-04-08 12:45:52	f	25	mobile	UA	113.53999710083008
4	30229	2024-04-15 16:42:39	f	25	mobile	UA	113.419997215271
5	30063	2024-04-15 10:36:51	f	25	mobile	UA	107.50999736785889
6	25017	2024-04-09 20:07:13	m	44	mobile	UA	69.50999927520752
7	25809	2024-04-10 15:59:43	m	64	desktop	GB	68.74999904632568

Другий запит:

- 1) Словесний запит:
“Порахувати кількість юзерів з кожного додатку (app)”
- 2) Реалізація в SQL через віконну функцію

```
SELECT app,
       SUM(COUNT(user_id)) OVER (PARTITION BY app) AS total_users
FROM registrations
GROUP BY app
ORDER BY total_users DESC;
```

РЕЗУЛЬТАТ ЗАПИТУ:

	app	total_users
1	ios	8699
2	mobile	3237
3	android	721
4	desktop	357

- 3) Реалізація в SQL без віконної функції

```
SELECT app,
       COUNT(user_id) AS total_users
FROM registrations
GROUP BY app
ORDER BY total_users DESC;
```

РЕЗУЛЬТАТ ЗАПИТУ:

	app	total_users
1	ios	8699
2	mobile	3237
3	android	721
4	desktop	357

Завдання 7

7. Проведіть дослідження на тему "Що призводить до оплат?". В межах цього завдання важливий не тільки фінальний результат, а й ваш шлях пошуку та декомпозиції основного питання, тому подайте ваші роздуми у структурованому вигляді.

Для початку дізнаємось хто найбільше потратив грошей з юзерів і чи є якась залежність між цим ?

Рахуємо скільки всього витратили, кількість оплат, витрати на кількість всього хто оплачував, кількість витрачених грошей відносно всіх юзерів відносно країн (t1 - всього витратили, t2 - кількість оплат, t3 - кількість юзерів по країнах)

```
SELECT t1.country_name,
       t1.total_spent,
       t2.total_pays,
       t3.total_users,
       t1.total_spent / t2.total_pays AS spent_by_paid,
       t1.total_spent / t3.total_users AS spent_by_user
FROM (SELECT country_name,
      SUM(spend_money) as total_spent
FROM registrations
INNER JOIN (SELECT user_id,
      SUM(amount_usd) AS spend_money
      FROM payments
      GROUP BY user_id) spent
ON spent.user_id = registrations.user_id
INNER JOIN country_info_dict
ON registrations.country_code = country_info_dict.country_code
GROUP BY country_name
ORDER BY total_spent DESC) t1
JOIN (SELECT country_name,
      COUNT(payments.user_id) AS total_pays
FROM payments
INNER JOIN registrations
ON payments.user_id = registrations.user_id
INNER JOIN country_info_dict
ON registrations.country_code = country_info_dict.country_code
GROUP BY country_name) t2
ON t1.country_name = t2.country_name
INNER JOIN
  (SELECT COUNT(user_id) AS total_users,
        country_name
  FROM registrations
  INNER JOIN country_info_dict
  ON registrations.country_code = country_info_dict.country_code
  GROUP BY country_name) t3
ON t1.country_name = t3.country_name
ORDER BY spent_by_user DESC;
```

РЕЗУЛЬТАТ ЗАПИТУ:

	country_name	total_spent	total_pays	total_users	spent_by_paid	spent_by_user
1	United Kingdom of Great Britain and Northern Ireland	227.18999481201172	24	116	9.466249783833822	1.9585344380345837
2	Germany	260.1399955749512	22	145	11.824545253406871	1.7940689349996632
3	Ukraine	3014.6999373435974	358	1924	8.420949545652507	1.566891859326194
4	United States of America	4125.659944057465	315	2781	13.097333155737983	1.4835167004881211
5	Guatemala	20.969999313354492	3	20	6.989999771118164	1.0484999656677245
6	Chile	512.899998664856	64	632	8.014062479138374	0.8115506307988227
7	Spain	239.18999481201172	26	346	9.199615185077374	0.6913005630404963
8	Costa Rica	83.64999961853027	13	143	6.43461535527156	0.5849650322974145
9	Poland	113.36999893188477	14	216	8.097857066563197	0.5248611061661331
10	Brazil	703.9899883270264	106	1382	6.64141498421723	0.5093994126823636

Дуже гарні показники в Україні та США (витрат на всіх юзерів), бо у всіх інших країнах не дуже багато юзерів. Яскравий приклад це Гватемала де всього 20 юзерів, але витратити на 1 юзера рівні 1.04, Британія та Германія також мають відносно мало юзерів, а витрати їх можна пояснити добробутом самих країн. Тому далі зосередемось на юзерах із України та США.

Порахуємо хто найбільше платить в цих країнах, сегментуємо по гендеру також по віку (

```
SELECT gender,
#       age,
       SUM(amount_usd) AS spent_money
FROM registrations
INNER JOIN sessions
ON registrations.user_id = sessions.user_id
INNER JOIN activities
ON registrations.user_id = activities.user_id
INNER JOIN country_info_dict
ON registrations.country_code = country_info_dict.country_code
INNER JOIN payments
ON registrations.user_id = payments.user_id
WHERE country_name IN ('United States of America', 'Ukraine')
GROUP BY gender
#       ,age
ORDER BY spent_money DESC;
```

РЕЗУЛЬТАТ ЗАПИТУ:

	<input type="checkbox"/> gender ▾	÷	<input type="checkbox"/> spent_money ▾	÷
1	m		1424962.2321300507	
2	f		133463.32818508148	

Тобто найбільше оплачують чоловіки в нашому додатку

Можемо тепер проаналізувати коли юзери з цих країн тратили кошти:

Робимо запит в яким номер сесії всього тратили коштів та скільки разів

```
SELECT session_number,
       country_info_dict.country_name,
       SUM(amount_usd),
       COUNT(amount_usd)
FROM registrations
INNER JOIN sessions
ON sessions.user_id = registrations.user_id
INNER JOIN payments
ON payments.user_id = registrations.user_id
INNER JOIN country_info_dict
ON country_info_dict.country_code = registrations.country_code
WHERE country_info_dict.country_name IN ('United States of America', 'Ukraine')
GROUP BY session_number,
       country_info_dict.country_name
```

РЕЗУЛЬТАТ ЗАПИТУ:

	session_number	country_name	`SUM(amount_usd)`	`COUNT(amount_usd)`
1	1	United States of America	3682.1499524116516	278
2	1	Ukraine	1827.6299629211426	208
3	2	United States of America	3320.859960079193	244
4	2	Ukraine	1213.8599710464478	151
5	3	United States of America	3151.1499638557434	225
6	3	Ukraine	1153.9799728393555	142
7	4	United States of America	3087.2099652290344	219
8	4	Ukraine	1047.469974040985	130
9	5	Ukraine	972.1999754905701	121
10	5	United States of America	2985.4099674224854	209
11	6	Ukraine	939.1899762153625	117
12	6	United States of America	2889.229969024658	203
13	7	Ukraine	901.1699771881104	112

Можна зробити висновок що найбільше з України та США робить в свої перші сесії, тобто в перші рази знайомству з додатком

```
SELECT session_number,
        SUM(amount_usd),
        COUNT(amount_usd)
FROM registrations
INNER JOIN sessions
ON sessions.user_id = registrations.user_id
INNER JOIN payments
ON payments.user_id = registrations.user_id
INNER JOIN country_info_dict
ON country_info_dict.country_code = registrations.country_code
# WHERE country_info_dict.country_name IN ('United States of America', 'Ukraine')
GROUP BY session_number
```

РЕЗУЛЬТАТ ЗАПИТУ:

	session_number	`SUM(amount_usd)`	`COUNT(amount_usd)`
1	1	9807.829842567444	1038
2	2	8034.489875793457	848
3	3	7651.329885005951	799
4	4	7280.589891910553	758
5	5	6993.939897060394	723
6	6	6693.3899030685425	691
7	7	6249.1199107170105	645
8	8	6019.839915275574	621
9	9	5851.819916725159	604
10	10	5535.079922199249	572

Тобто по всьому світу така сама тенденція що купують протягом своєї першої сесії

Порахуємо також що юзери роблять в своїй першій сесії до оплати

```
SELECT activity_name,
        COUNT(activity_name) AS numbers_of_action_before_paid
FROM registrations
INNER JOIN sessions
ON sessions.user_id = registrations.user_id
INNER JOIN payments
ON payments.user_id = registrations.user_id
INNER JOIN country_info_dict
ON country_info_dict.country_code = registrations.country_code
```

```
INNER JOIN activities
ON activities.user_id = registrations.user_id
INNER JOIN activity_type_dict
ON activity_type_dict.activity_type = activities.activity_type
WHERE activities.dt BETWEEN session_start_dt AND pay_dt
AND session_number = 1
GROUP by activity_name
```

РЕЗУЛЬТАТ ЗАПИТУ:

	activity_name	numbers_of_action_before_paid
1	dislike	801
2	visit	264
3	like	326

Також перевіримо що вони роблять після оплати в першій:

```
SELECT activity_name,
       COUNT(activity_name) AS numbers_of_action_before_paid
FROM registrations
INNER JOIN sessions
ON sessions.user_id = registrations.user_id
INNER JOIN payments
ON payments.user_id = registrations.user_id
INNER JOIN country_info_dict
ON country_info_dict.country_code = registrations.country_code
INNER JOIN activities
ON activities.user_id = registrations.user_id
INNER JOIN activity_type_dict
ON activity_type_dict.activity_type = activities.activity_type
WHERE activities.dt > pay_dt
AND session_number = 1
GROUP by activity_name
```

РЕЗУЛЬТАТ ЗАПИТУ:

	activity_name	numbers_of_action_before_paid
1	like	1195
2	dislike	3273
3	visit	973

Тобто після оплати в час своєї першої сесії вони роблять на порядок більше дій, можемо припустити що придбання premium або vip, знімає якесь обмеження, яке було без premium або vip.

Також подивимось кількість оплат відносно девайсу

```
SELECT app,
       SUM(amount_usd) AS total_spent,
       COUNT(amount_usd) AS numbers_of_paid
FROM registrations
INNER JOIN sessions
ON sessions.user_id = registrations.user_id
INNER JOIN payments
ON payments.user_id = registrations.user_id
INNER JOIN country_info_dict
ON country_info_dict.country_code = registrations.country_code
INNER JOIN activities
ON activities.user_id = registrations.user_id
INNER JOIN activity_type_dict
ON activity_type_dict.activity_type = activities.activity_type
GROUP BY app
ORDER BY total_spent DESC
```

РЕЗУЛЬТАТ ЗАПИТУ:

	app	total_spent	numbers_of_paid
1	ios	2336145.557466507	325811
2	mobile	540502.4885811806	67625
3	android	461222.13399505615	26256
4	desktop	19208.089462280273	2339

Це може пояснюватись те що оплачують користувачі з більш забезпечених країн, як США або Велика Британія і тд

Висновок:

У висновку можемо підсумувати що перший контакт користувача з додатком має значний вплив на його готовність здійснити оплату. Якщо користувач знайде додаток цікавим та корисним під час своєї першої зустрічі з ним, він вірогідно буде готовий вкластися у покупку premium або вір для зняття обмежень. (які можуть існувати в базовій версії, якщо наш додаток типу freemium). Також немало важливим фактором оплати є саме знаходження користувача відносно країни