

Розрахунково-графічна робота

Варіант: платформа для бронювання та організації весіль

Виконала: Дімова Марія

Група: KB-13

GitHub: https://github.com/txtmf/Weddings_BD

Telegram: [@mnezhyk](https://t.me/mnezhyk)

ER діаграма

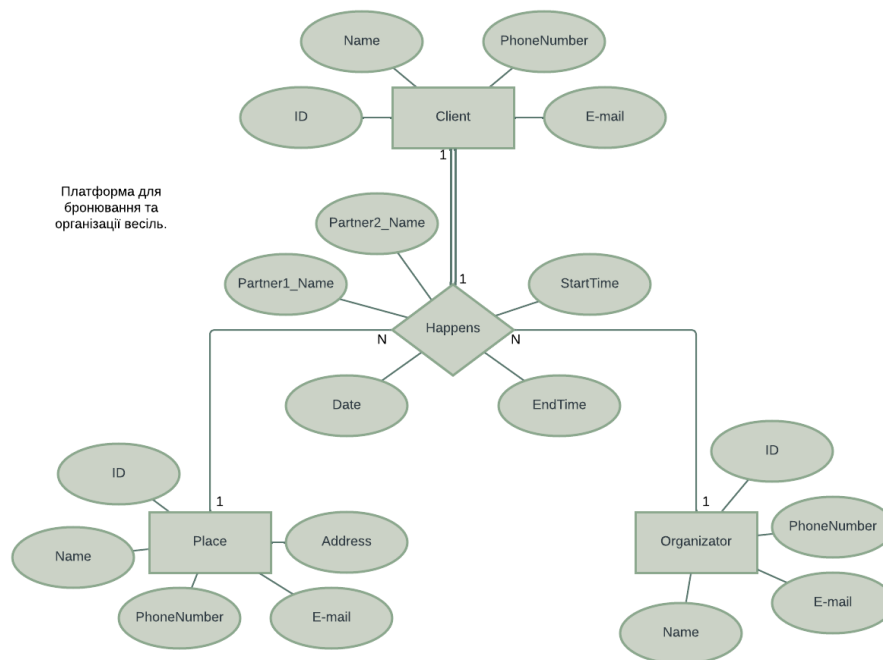
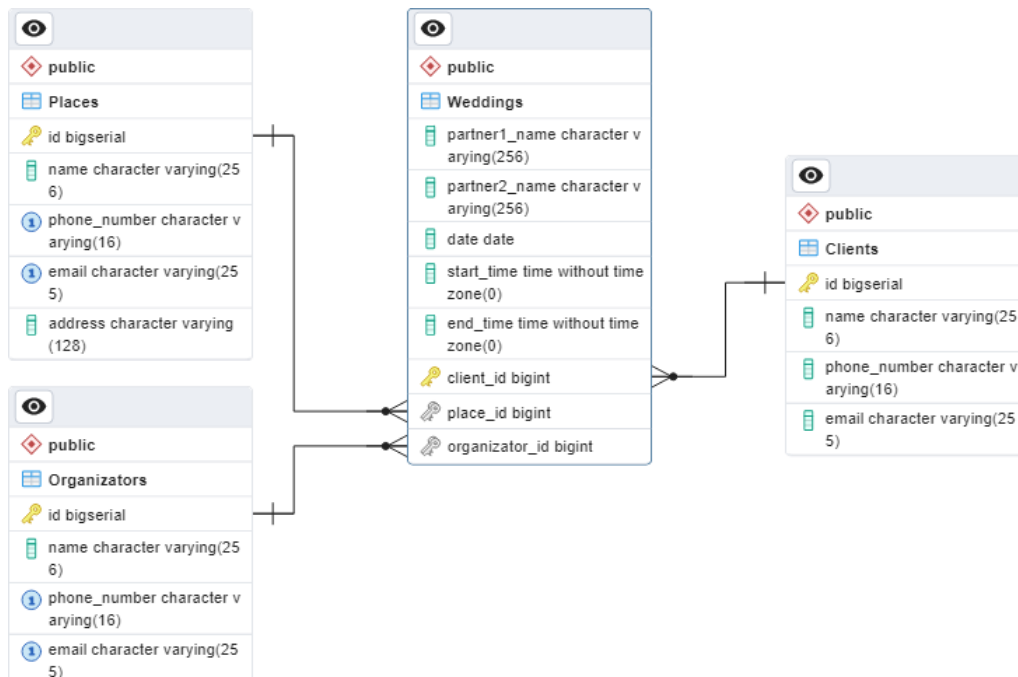


Схема бази даних



Меню програми

```
1. Add client
2. Add organizer
3. Add place
4. Add wedding
5. Show clients
6. Show organizers
7. Show places
8. Show weddings
9. Update client
10. Update organizer
11. Update place
12. Update wedding
13. Delete client
14. Delete organizer
15. Delete place
16. Delete wedding

17. Generate batch

18. Clients, their weddings in date range, and partners
19. Organizer's weddings since
20. Weddings at place since

0. Quit

Option: █
```

Опції 1-4 – це додавання одного рядку до однієї з таблиць.

Опції 5-8 – повний вивід однієї з таблиць.

Опції 9-12 – оновлення даних одного рядку з однієї таблиці за id (для таблиці Weddings – client_id).

Опції 13-16 – видалення одного рядку з однієї таблиці за id (для таблиці Weddings – client_id).

Опція 17 – пакетне генерування даних із вказанням кількості рядків.

Опції 18-20 – запити пошуку у таблицях для пункту 3.

Опція 0 – вихід з програми.

Взаємодія з програмою відбувається за допомогою клавіатури.

Інструментарій

Мова програмування – Python 3.11

Бібліотеки – psycopg2 для взаємодії з базою даних та datetime для валідації дат та часу.

Середовище розробки – VS Code.

CRUD операції

Таблиці Organizers та Places незалежні від інших таблиць, тобто не мають зовнішніх ключів, і всі рядки можуть існувати, навіть якщо на них не посилається жоден рядок з таблиці Weddings.

Батьківсько-дочірній зв'язок наявний тільки між таблицями Clients та Weddings. Кожен рядок з Clients відповідає одному рядку з Weddings. При видаленні рядка Clients, зв'язаний з ним рядок з Weddings автоматично видаляється СУБД.

Видалення рядка з Clients

Спочатку створимо новий рядок у Clients, а також у Weddings, щоб показати, що відбувається при видаленні.

Option: 1

Add new client

Example: "Mykolai Petrenko, +380980012354, mpetrenko@gmail.com"

Arguments: Роман Литвиненко, +380980054809, romlyt@gmail.com

1. Add client

Option: 5

Clients:

ID: 1, Name: Іван Петров, Phone number: +380931432811, Email: ivan.petrov@email.com
ID: 2, Name: Марія Сидорова, Phone number: +380671201500, Email: maria.sidorova@email.com
ID: 3, Name: Олег Коваль, Phone number: +380991241267, Email: oleg.koval@email.com
ID: 4, Name: Анна Мельник, Phone number: +380501214321, Email: anna.melnyk@email.com
ID: 5, Name: Петро Лисенко, Phone number: +380631230060, Email: petro.lisenko@email.com
ID: 16, Name: Mykolai Petrenko, Phone number: +380980012354, Email: mpetrenko@gmail.com
ID: 18, Name: Bebrik, Phone number: +380980010345, Email: realbebrik@gmail.com
ID: 19, Name: Bebrik, Phone number: +380980010345, Email: secret_bebrik@gmail.com
ID: 17, Name: Hryhoriy Kravchenko, Phone number: +380672235540, Email: golekskrav@gmail.com
ID: 20, Name: Petro Schur, Phone number: +380506872233, Email: petro.schur@ukr.net
ID: 21, Name: Артур Павленко, Phone number: +380982640907, Email: pavlenko_artur@gmail.com
ID: 44, Name: Yoas, Phone number: +380438280923, Email: Yoas_302@ukr.net
ID: 45, Name: Xnum, Phone number: +380553908199, Email: Xnum_246@gmail.com
ID: 46, Name: Lceg, Phone number: +38099913567, Email: Lceg_521@ukr.net
ID: 47, Name: Qhpk, Phone number: +380379393993, Email: Qhpk_118@ukr.net
ID: 48, Name: Arur, Phone number: +380967535358, Email: Arur_607@ukr.net
ID: 49, Name: Роман Литвиненко, Phone number: +380980054809, Email: romlyt@gmail.com

1. Add client

Option: 4

Add new wedding

Example: "Mykolai Petrenko, Oksana Kvach, 2024-04-12, 12:30, 23:00, 1, 1, 1"

Arguments: Роман Литвиненко, Марина Кириленко, 2024-03-05, 14:00, 22:00, 49, 4, 1

1. Add client

Partner1 name: Роман Литвиненко, Partner2 name: Марина Кириленко, Date: 2024-03-05, Start time: 14:00:00, End time: 22:00:00, Client ID: 49, Place ID: 4, Organizer ID: 1

1. Add client

Тепер можна видалити Романа Литвиненка з бази, таким чином видалиться і інформація про весілля.

Option: 13

Delete existing client

Example: "42"

ID: 49

1. Add client

ID: 19, Name: Bebrik, Phone number: +380980010345, Email: secret_bebrik@gmail.com
ID: 17, Name: Hryhoriy Kravchenko, Phone number: +380672235540, Email: golekskrav@gmail.com
ID: 20, Name: Petro Schur, Phone number: +380506872233, Email: petro.schur@ukr.net
ID: 21, Name: Артур Павленко, Phone number: +380982640907, Email: pavlenko_artur@gmail.com
ID: 44, Name: Yoas, Phone number: +380438280923, Email: Yoas_302@ukr.net
ID: 45, Name: Xnum, Phone number: +380553908199, Email: Xnum_246@gmail.com
ID: 46, Name: Lceg, Phone number: +38099913567, Email: Lceg_521@ukr.net
ID: 47, Name: Qhpk, Phone number: +380379393993, Email: Qhpk_118@ukr.net
ID: 48, Name: Arur, Phone number: +380967535358, Email: Arur_607@ukr.net

1. Add client

Partner1 name: Sedy, Partner2 name: Przr, Date: 2024-02-16, Start time: 13:56:00, End time: 19:53:00, Client ID: 46, Place ID: 37, Organizer ID: 26
Partner1 name: Utzz, Partner2 name: Hqke, Date: 2024-07-13, Start time: 12:30:00, End time: 19:53:00, Client ID: 47, Place ID: 37, Organizer ID: 25
Partner1 name: Msfz, Partner2 name: Zprz, Date: 2024-12-29, Start time: 14:34:00, End time: 21:19:00, Client ID: 48, Place ID: 38, Organizer ID: 28

1. Add client

Як видно з виводу програми, і клієнт, і весілля – видалені.

Вставка у Weddings

Тепер спробуємо створити нове весілля з неіснуючим id клієнта.

```
Option: 4

Add new wedding
Example: "Mykolai Petrenko, Oksana Kvach, 2024-04-12, 12:30, 23:00, 1, 1, 1"

Arguments: Роман Литвиненко, Марина Кириленко, 2024-3-5, 14:00, 22:00, 49, 1, 1

Client ID doesn't exist

1. Add client
```

Як ми бачимо, створити весілля без замовника неможливо.

Пакетне генерування

При пакетному генеруванні даних, користувач вказує кількість рядків, які він хоче додати до кожної таблиці. При генерації зовнішніх ключів у Weddings, використовуються тільки ті, які були тільки що згенеровані, тобто при генерації 100 рядків, у нових весіллях будуть всі нові клієнти, а також випадково обрані організатори та місця, але обрані вони тільки з останніх 100.

Спочатку виведемо всі таблиці, потім згенеруємо 5 рядків, і подивимось на результат. Після цього можна буде згенерувати 100000 рядків, як було вказано у завданні.

```
Option: 5

Clients:
ID: 1, Name: Іван Петров, Phone number: +380931432811, Email: ivan.petrov@email.com
ID: 2, Name: Марія Сидорова, Phone number: +380671201500, Email: maria.sidorova@email.com
ID: 3, Name: Олег Коваль, Phone number: +380991241267, Email: oleg.koval@email.com
ID: 4, Name: Анна Мельник, Phone number: +380501214321, Email: anna.melnyk@email.com
ID: 5, Name: Петро Лисенко, Phone number: +380631230060, Email: petro.lisenko@email.com
ID: 16, Name: Mykolai Petrenko, Phone number: +380980012354, Email: mpetrenko@gmail.com
ID: 18, Name: Bebrik, Phone number: +380980010345, Email: realbebrik@gmail.com
ID: 19, Name: Bebrik, Phone number: +380980010345, Email: secret_bebrik@gmail.com
ID: 17, Name: Hryhoriy Kravchenko, Phone number: +380672235540, Email: golekskrav@gmail.com
ID: 20, Name: Petro Schur, Phone number: +380506872233, Email: petro.schur@ukr.net
ID: 21, Name: Артур Павленко, Phone number: +380982640907, Email: pavlenko_artur@gmail.com
ID: 44, Name: Yoas, Phone number: +380438280923, Email: Yoas_302@ukr.net
ID: 45, Name: Xnum, Phone number: +380553908199, Email: Xnum_246@gmail.com
ID: 46, Name: Lceg, Phone number: +38099913567, Email: Lceg_521@ukr.net
ID: 47, Name: Qhpk, Phone number: +380379393993, Email: Qhpk_118@ukr.net
ID: 48, Name: Arur, Phone number: +380967535358, Email: Arur_607@ukr.net
```

```
Option: 6

Organizators:
ID: 1, Name: Наталія Білоус, Phone number: +380977772233, Email: natalia.bilous@email.com
ID: 2, Name: Максим Шевченко, Phone number: +380685554422, Email: maxym.shevchenko@email.com
ID: 3, Name: Вікторія Григорович, Phone number: +380502344411, Email: viktorija.grigorovych@email.com
ID: 4, Name: Олександр Коваленко, Phone number: +380933334455, Email: oleksandr.kovalenko@email.com
ID: 5, Name: Юлія Мельничук, Phone number: +380677776666, Email: yulia.melnychuk@email.com
ID: 6, Name: Андрій Стельмах, Phone number: +380987770000, Email: andriy.stelmakh@email.com
ID: 7, Name: Тетяна Петренко, Phone number: +380977772277, Email: tetiana.petrenko@email.com
ID: 8, Name: Григорій Чайка, Phone number: +380980034872, Email: agent_hryhir007@gmail.com
ID: 24, Name: Jtmr, Phone number: +380162730183, Email: Jtmr_500@ukr.net
ID: 25, Name: Xpoj, Phone number: +380514341286, Email: Xpoj_355@ukr.net
ID: 26, Name: Zsaz, Phone number: +380798920588, Email: Zsaz_535@gmail.com
ID: 27, Name: Mitf, Phone number: +380877424044, Email: Mitf_456@gmail.com
ID: 28, Name: Gmvf, Phone number: +380679975378, Email: Gmvf_855@gmail.com
```

Option: 7

Places:

ID: 1, Name: Золотий Замок, Phone number: +380442233311, Email: zoloty.zamok@gmail.com, Address: вул. Гранітна, 12, Київ
ID: 2, Name: Сонячний Сад, Phone number: +380681122334, Email: sonyachnyy.sad@gmail.com, Address: просп. Парковий, 34, Львів
ID: 3, Name: Річковий Ресторан, Phone number: +380573322144, Email: richkovyy.restoran@gmail.com, Address: вул. Берегова, 56, Одеса
ID: 4, Name: Замковий Готель, Phone number: +380671155533, Email: zamkovyy.gotel@gmail.com, Address: вул. Шевченка, 78, Львів
ID: 5, Name: Парковий ОТЕЛЬ, Phone number: +380501288899, Email: parkovyy.otel@gmail.com, Address: просп. Парковий, 18, Київ
ID: 6, Name: Смачна Трапеза, Phone number: +380444555566, Email: smachna.trapeza@gmail.com, Address: вул. Смачна, 5, Одеса
ID: 7, Name: Зелений Сад, Phone number: +380501122333, Email: zelenyy.sad@gmail.com, Address: просп. Парковий, 27, Львів
ID: 8, Name: Закохані Весілля, Phone number: +380732211444, Email: zakohani.vesillia@gmail.com, Address: вул. Любовна, 3, Київ
ID: 9, Name: Сонячний Парк, Phone number: +380682233100, Email: sonyachnyy.park@gmail.com, Address: просп. Сонячний, 8, Одеса
ID: 10, Name: Золотий Пляж, Phone number: +380992299211, Email: zoloty.plyazh@gmail.com, Address: вул. Пляжна, 21, Одеса
ID: 36, Name: Fooz Hteg, Phone number: +380554809600, Email: fooz_hteg_rest_32@gmail.com, Address: 54 Lwenjiir St.
ID: 37, Name: Dvow Vsbc, Phone number: +380740139199, Email: dvow_vsbc_rest_40@gmail.com, Address: 7 Gopippjo St.
ID: 38, Name: Bykx Cgli, Phone number: +380573722937, Email: bykx_cgli_rest_15@ukr.net, Address: 17 Xfinxggp St.
ID: 39, Name: Yjmt Dxmff, Phone number: +380551392571, Email: yjmt_dxmff_rest_987@gmail.com, Address: 24 Xvboxin St.
ID: 40, Name: Tkjx Wnyy, Phone number: +380149348161, Email: tkjx_wnyy_rest_94@gmail.com, Address: 92 Binsziwv St.

Weddings:

Partner1 name: Іван Петров, Partner2 name: Марія Петренко, Date: 2023-05-15, Start time: 15:00:00, End time: 22:00:00, Client ID: 1, Place ID: 10, Organizer ID: 2
Partner1 name: Максим Пропорець, Partner2 name: Марія Сидорова, Date: 2023-06-20, Start time: 14:30:00, End time: 20:30:00, Client ID: 2, Place ID: 4, Organizer ID: 5
Partner1 name: Олег Коваль, Partner2 name: Оксана Лисенко, Date: 2023-07-10, Start time: 16:00:00, End time: 23:00:00, Client ID: 3, Place ID: 6, Organizer ID: 1
Partner1 name: Денис Степанов, Partner2 name: Анна Мельник, Date: 2023-08-05, Start time: 15:30:00, End time: 21:30:00, Client ID: 4, Place ID: 8, Organizer ID: 2
Partner1 name: Петро Лисенко, Partner2 name: Наталя Василенко, Date: 2023-09-15, Start time: 17:00:00, End time: 23:59:59, Client ID: 5, Place ID: 2, Organizer ID: 3
Partner1 name: Миколай Петренко, Partner2 name: Олена Ковальчук, Date: 2023-12-12, Start time: 11:00:00, End time: 19:45:00, Client ID: 16, Place ID: 3, Organizer ID: 4
Partner1 name: Андрій Ігорович, Partner2 name: Ольга Василенко, Date: 2024-05-06, Start time: 09:30:00, End time: 23:30:00, Client ID: 18, Place ID: 8, Organizer ID: 5
Partner1 name: Maksym Volodymyrovych, Partner2 name: Iryna Oleksandrivna, Date: 2024-06-15, Start time: 10:00:00, End time: 22:00:00, Client ID: 19, Place ID: 6, Organizer ID: 4
Partner1 name: Наталія Ігорівна, Partner2 name: Олександр Петрович, Date: 2024-07-02, Start time: 13:00:00, End time: 17:00:00, Client ID: 17, Place ID: 8, Organizer ID: 4
Partner1 name: Петро Щур, Partner2 name: Анна Кузьменко, Date: 2024-07-16, Start time: 13:00:00, End time: 23:59:00, Client ID: 20, Place ID: 3, Organizer ID: 4
Partner1 name: Артур Павленко, Partner2 name: Юлія Остапчук, Date: 2024-03-06, Start time: 13:45:00, End time: 23:30:00, Client ID: 21, Place ID: 7, Organizer ID: 2
Partner1 name: Jсна, Partner2 name: Ljxd, Date: 2024-09-05, Start time: 14:24:00, End time: 21:05:00, Client ID: 44, Place ID: 36, Organizer ID: 28
Partner1 name: Iwgc, Partner2 name: Btbg, Date: 2024-04-30, Start time: 13:20:00, End time: 22:35:00, Client ID: 45, Place ID: 39, Organizer ID: 25
Partner1 name: Sedv, Partner2 name: Przr, Date: 2024-02-16, Start time: 13:56:00, End time: 19:53:00, Client ID: 46, Place ID: 37, Organizer ID: 26
Partner1 name: Utzz, Partner2 name: Hqke, Date: 2024-07-13, Start time: 12:30:00, End time: 19:53:00, Client ID: 47, Place ID: 37, Organizer ID: 25
Partner1 name: Msfz, Partner2 name: Zprr, Date: 2024-12-29, Start time: 14:34:00, End time: 21:19:00, Client ID: 48, Place ID: 38, Organizer ID: 28

Option: 17

Generate batch of clients, organizers, places, and weddings
Example: "100000"

Rows: 5

1. Add client

ID: 47, Name: Qhpk, Phone number: +380379393993, Email: Qhpk_118@ukr.net
ID: 48, Name: Arur, Phone number: +380967535358, Email: Arur_607@ukr.net
ID: 50, Name: Ektv, Phone number: +380267426773, Email: Ektv_712@ukr.net
ID: 51, Name: Dcvz, Phone number: +380400586060, Email: Dcvz_503@ukr.net
ID: 52, Name: Gqjx, Phone number: +380622364151, Email: Gqjx_82@ukr.net
ID: 53, Name: Vugf, Phone number: +380554816221, Email: Vugf_943@ukr.net
ID: 54, Name: Kpsu, Phone number: +380495492961, Email: Kpsu_47@ukr.net

1. Add client

ID: 26, Name: Zsaž, Phone number: +380798920588, Email: Zsaž_535@gmail.com
ID: 27, Name: Mitf, Phone number: +380877424044, Email: Mitf_456@gmail.com
ID: 28, Name: Gmvf, Phone number: +380679975378, Email: Gmvf_855@gmail.com
ID: 29, Name: Kbzu, Phone number: +380863418136, Email: Kbzu_111@gmail.com
ID: 30, Name: Cguq, Phone number: +380902672212, Email: Cguq_241@ukr.net
ID: 31, Name: Lveo, Phone number: +380730068672, Email: Lveo_753@gmail.com
ID: 32, Name: Pxiy, Phone number: +380208005927, Email: Pxiy_876@ukr.net
ID: 33, Name: Fqcs, Phone number: +380991089346, Email: Fqcs_831@gmail.com

1. Add client


```
ID: 38, Name: Bykx Cgli, Phone number: +380573722937, Email: bykx_cgli_rest_15@ukr.net, Address: 17 Xfinxggp St.  
ID: 39, Name: Yjmt Dxmfm, Phone number: +380551392571, Email: yjmt_dxmfm_rest_987@gmail.com, Address: 24 Xvbsoxin St.  
ID: 40, Name: Tkjx Wnyy, Phone number: +380149348161, Email: tkjx_wnyy_rest_94@gmail.com, Address: 92 Binsziw St.  
ID: 41, Name: Svql Ppyx, Phone number: +38097422635, Email: svql_ppyx_rest_625@ukr.net, Address: 38 Jwilsjmc St.  
ID: 42, Name: Sjcc Itwp, Phone number: +380472634023, Email: sjcc_itwp_rest_919@gmail.com, Address: 60 Uykjzfh St.  
ID: 43, Name: Wacq Aqzq, Phone number: +380964570843, Email: wacq_aqzq_rest_154@gmail.com, Address: 75 Iwtklhdw St.  
ID: 44, Name: Curi Vaer, Phone number: +380489335514, Email: curi_vaer_rest_878@ukr.net, Address: 24 Nqnbaqsi St.  
ID: 45, Name: Coqi Tmho, Phone number: +380811060744, Email: coqi_tmho_rest_919@ukr.net, Address: 39 Xdftkxlw St.
```

1. Add client

```
Partner1 name: Sedz, Partner2 name: Przr, Date: 2024-02-16, Start time: 13:56:00, End time: 19:53:00, Client ID: 46, Place ID: 37, Organizer ID: 26  
Partner1 name: Utzz, Partner2 name: Hqke, Date: 2024-07-13, Start time: 12:30:00, End time: 19:53:00, Client ID: 47, Place ID: 37, Organizer ID: 25  
Partner1 name: Msfz, Partner2 name: Zprz, Date: 2024-12-29, Start time: 14:34:00, End time: 21:19:00, Client ID: 48, Place ID: 38, Organizer ID: 28  
Partner1 name: Vhkp, Partner2 name: Medb, Date: 2024-08-25, Start time: 13:56:00, End time: 20:22:00, Client ID: 50, Place ID: 43, Organizer ID: 30  
Partner1 name: Vgwb, Partner2 name: Itlv, Date: 2024-08-27, Start time: 15:46:00, End time: 18:12:00, Client ID: 51, Place ID: 41, Organizer ID: 32  
Partner1 name: Evoj, Partner2 name: Mrfx, Date: 2024-11-08, Start time: 14:04:00, End time: 20:53:00, Client ID: 52, Place ID: 42, Organizer ID: 32  
Partner1 name: Ulqc, Partner2 name: Ulxy, Date: 2024-02-17, Start time: 14:42:00, End time: 22:09:00, Client ID: 53, Place ID: 42, Organizer ID: 32  
Partner1 name: Mcau, Partner2 name: Nycl, Date: 2024-04-16, Start time: 15:44:00, End time: 19:37:00, Client ID: 54, Place ID: 45, Organizer ID: 29
```

1. Add client

Все працює!

Тепер згенеруємо 100000 рядків.

Option: 17

Generate batch of clients, organizers, places, and weddings
Example: "100000"

Rows: 100000

1. Add client

SQL зануму

```
insert into "Clients" (name, phone_number, email) (  
    select name,  
           '+' || rpad((random() * 1000)::int::text, 3, '0') ||  
lpad((random() * 1000000000)::int::text, 9, '0') as phone_number,  
           name || '_' || (random() * 1000)::int::varchar(3) ||  
           '@' || case when random() > 0.5 then 'gmail.com' else  
'ukr.net' end as email  
    from (  
        select (chr((random() * 25)::int + 65)) ||  
               (chr((random() * 25)::int + 97)) ||  
               (chr((random() * 25)::int + 97)) ||  
               (chr((random() * 25)::int + 97)) as name  
        from generate_series(1, %s)  
    )  
);  
  
insert into "Organizers" (name, phone_number, email) (  
    select name,  
           '+' || rpad((random() * 1000)::int::text, 3, '0') ||  
lpad((random() * 1000000000)::int::text, 9, '0') as phone_number,  
           name || '_' || (random() * 1000)::int::varchar(3) ||  
           row::text || '@' || case when random() > 0.5 then 'gmail.com'  
else 'ukr.net' end as email  
    from (  
        select (chr((random() * 25)::int + 65)) ||  
               (chr((random() * 25)::int + 97)) ||  
               (chr((random() * 25)::int + 97)) ||  
               (chr((random() * 25)::int + 97)) as name,  
               row  
        from generate_series(1, %s) as row  
    )  
);
```

```

insert into "Places" (name, phone_number, email, address) (
    select name,
        '+' || rpad((random() * 1000)::int::text, 3, '0') ||
lpad((random() * 1000000000)::int::text, 9, '0') as phone_number,
        replace(lower(name), ' ', '_') || '_rest_' || (random() *
1000)::int::varchar(3) || row::text || '@' || case when random() > 0.5 then
'gmail.com' else 'ukr.net' end as email,
        address
    from (
        select (chr((random() * 25)::int + 65)) ||
            (chr((random() * 25)::int + 97)) ||
            (chr((random() * 25)::int + 97)) ||
            (chr((random() * 25)::int + 97)) || ' ' ||
            (chr((random() * 25)::int + 65)) ||
            (chr((random() * 25)::int + 97)) ||
            (chr((random() * 25)::int + 97)) ||
            (chr((random() * 25)::int + 97)) as name,

            (random() * 99 + 1)::int::varchar(2) || ' ' ||
            (chr((random() * 25)::int + 65)) ||
            (chr((random() * 25)::int + 97)) ||
            (chr((random() * 25)::int + 97)) ||
            (chr((random() * 25)::int + 97)) ||
            (chr((random() * 25)::int + 97)) ||
            (chr((random() * 25)::int + 97)) ||
            (chr((random() * 25)::int + 97)) ||
            (chr((random() * 25)::int + 97)) || ' St.' as address,
            row
        from generate_series(1, %s) as row
    )
);

with new_weddings as (
    select *
    from (
        select (chr((random() * 25)::int + 65)) ||
            (chr((random() * 25)::int + 97)) ||
            (chr((random() * 25)::int + 97)) ||
            (chr((random() * 25)::int + 97)) as partner1_name,
            (chr((random() * 25)::int + 65)) ||
            (chr((random() * 25)::int + 97)) ||
            (chr((random() * 25)::int + 97)) ||
            (chr((random() * 25)::int + 97)) as partner2_name,
            '2024-01-01'::date + (random() * 365)::int as date,
            make_time((random() * 4 + 12)::int, (random() *
59)::int, 0) as start_time,
            make_time((random() * 5 + 18)::int, (random() *
59)::int, 0) as end_time,
            floor(random() * %s + 1)::int + ((select max(id) from
"Organizators") - %s) as organizer_id,
            floor(random() * %s + 1)::int + ((select max(id) from
"Places") - %s) as place_id,
            client_id
        from generate_series((select max(id) - %s + 1 from "Clients"),
(select max(id) from "Clients")) as client_id
    ) insert into "Weddings" (partner1_name, partner2_name, date, start_time,
end_time, client_id, organizer_id, place_id)
(select partner1_name, partner2_name, date, start_time, end_time, client_id,
organizer_id, place_id from new_weddings);

```

%s замінюється на кількість рядків.

Пошук Перший запит

Запит, який шукає клієнтів, які замовили весілля на дату у вказаному проміжку.
Відображає id та ім'я клієнта, дату весілля, та імена партнерів цього весілля.

```
Option: 18

Show clients, date that they requested, and partners of the wedding
Example: "2023-01-01, 2024-01-01"

Date range: 2023-8-1, 2024-1-1

Query execution time is 133.6487 ms

Client id: 4, Client name: Анна Мельник, Date: 2023-08-05, Partner 1 name: Денис Степанов, Partner 2 name: Анна Мельник
Client id: 5, Client name: Петро Лисенко, Date: 2023-09-15, Partner 1 name: Петро Лисенко, Partner 2 name: Наталя Василенко
Client id: 16, Client name: Mykolai Petrenko, Date: 2023-12-12, Partner 1 name: Миколай Петренко, Partner 2 name: Олена Ковальчук
Client id: 11001490, Client name: Jcmg, Date: 2024-01-01, Partner 1 name: Fusp, Partner 2 name: Jeng
Client id: 11001859, Client name: Eveu, Date: 2024-01-01, Partner 1 name: Hjls, Partner 2 name: Sfbk
Client id: 11001989, Client name: Qubd, Date: 2024-01-01, Partner 1 name: Ehkn, Partner 2 name: Bwpb
Client id: 11002445, Client name: Iyrj, Date: 2024-01-01, Partner 1 name: Zuvy, Partner 2 name: Sbrm
Client id: 11003493, Client name: Ohtv, Date: 2024-01-01, Partner 1 name: Oytn, Partner 2 name: Okyz
Client id: 11003911, Client name: Ikbe, Date: 2024-01-01, Partner 1 name: BImj, Partner 2 name: Mnyk
Client id: 11005357, Client name: Bcdk, Date: 2024-01-01, Partner 1 name: Chls, Partner 2 name: Mnyk
```

```
select client_id, "Clients".name, date, partner1_name, partner2_name
from "Weddings"
join "Clients" on client_id = "Clients".id
where date >= '2023-5-1' and date <= '2023-8-1'
group by client_id, "Clients".name, date, partner1_name, partner2_name
order by date asc
```

Другий запит

Запит, який шукає організаторів, які проводили весілля після дати. Виводить інформацію про організатора та кількість весіль, яку вони провели, або будуть проводити, після вказаної дати. Кількість весіль має бути більше або дорівнювати вказаному мінімуму.

```
Option: 19

Show organizers and the count of weddings they are organizing since date
Example: "2023-01-01, 3"

Since date, minimum weddings count: 2024-8-1, 4

Query execution time is 76.976 ms

Organizator id: 389988, Name: Gcyo, Phone number: +822156917444, Email: Gcyo_6698925@gmail.com, Weddings: 5
Organizator id: 320433, Name: Ombd, Phone number: +490419648219, Email: Ombd_46029370@ukr.net, Weddings: 5
Organizator id: 375330, Name: Yyyu, Phone number: +302579637375, Email: Yyyu_16384267@ukr.net, Weddings: 5
Organizator id: 366377, Name: Hslg, Phone number: +134506637783, Email: Hslg_16675314@gmail.com, Weddings: 5
Organizator id: 371812, Name: Fndx, Phone number: +415059982720, Email: Fndx_38180749@ukr.net, Weddings: 5
Organizator id: 299796, Name: Oelk, Phone number: +772582454020, Email: Oelk_3428733@gmail.com, Weddings: 5
Organizator id: 390378, Name: Jzxe, Phone number: +190665221910, Email: Jzxe_17999315@ukr.net, Weddings: 5
Organizator id: 310022, Name: Tcgo, Phone number: +777122547214, Email: Tcgo_84722860@gmail.com, Weddings: 5
```

```
select *
from (
    select id as organizer, name, phone_number, email, count(client_id) as
weddings
    from "Weddings"
    join "Organizators" on id=organizator_id
    where date > '2023-5-1'
    group by id, name, phone_number, email
)
where weddings >= 3
order by weddings desc
```

Третій запит

Запит, що шукає місця, у яких проводились або будуть проводитись весілля після вказаної дати. Виводить кількість весіль, адресу та id місця.

Option: 20

Show weddings since date in a place, and the place itself
Example: "2023-01-01"

Since date: 2024-12-30

Query execution time is 145.9669 ms

Weddings: 1, Address: 53 Ubjefrwj St., Place: 1709
Weddings: 1, Address: 13 Piqzwjon St., Place: 3346
Weddings: 1, Address: 65 Lmpgjdkc St., Place: 3366
Weddings: 1, Address: 9 Ruvuutxu St., Place: 3982
Weddings: 1, Address: 92 Ydjbvrgx St., Place: 4886
Weddings: 1, Address: 32 Afpvexun St., Place: 4944
Weddings: 1, Address: 34 Yvuvzavl St., Place: 6261

```
select count(client_id) as weddings, address, place_id as place
from "Clients"
join "Weddings" on client_id="Clients".id
join "Places" on place_id="Places".id
join "Organizators" on organizer_id="Organizators".id
where date > '2024-5-1'
group by place_id, address
```

Час виконання

Також виводиться час виконання запитів. Як видно, виконуються вони достатньо швидко.

MVC шаблон

```
import psycopg2

class Model:
    def __init__(self) -> None:
        self.connection = psycopg2.connect(
            dbname = "weddings",
            user = "postgres",
            password = "123457896_Qq",
            host = "localhost",
            port = 5432
        )

    def add_client(self, name, phone_number, email) -> None:
        with self.connection.cursor() as cursor:
            cursor.execute('INSERT INTO public."Clients" (name, phone_number, email) VALUES (%s, %s, %s)', (name, phone_number, email))
            self.connection.commit()

    def get_all_clients(self):
        with self.connection.cursor() as cursor:
            cursor.execute('SELECT * FROM public."Clients"')
            return cursor.fetchall()

    def get_client(self, id):
        with self.connection.cursor() as cursor:
            cursor.execute('SELECT * FROM public."Clients" WHERE id=%s', [id])
            return cursor.fetchall()

    def update_client(self, id, name, phone_number, email) -> None:
        with self.connection.cursor() as cursor:
            cursor.execute('UPDATE public."Clients" SET name=%s, phone_number=%s, email=%s WHERE id=%s', (name, phone_number, email, id))
            self.connection.commit()

    def delete_client(self, id):
        with self.connection.cursor() as cursor:
            cursor.execute('DELETE FROM public."Clients" WHERE id=%s', [id])
            self.connection.commit()

    def add_organizator(self, name, phone_number, email) -> None:
```

```

        with self.connection.cursor() as cursor:
            cursor.execute('INSERT INTO public."Organizators" (name, phone_number, email)
VALUES (%s, %s, %s)', (name, phone_number, email))
            self.connection.commit()

    def get_all_organizators(self):
        with self.connection.cursor() as cursor:
            cursor.execute('SELECT * FROM public."Organizators"')
            return cursor.fetchall()

    def get_organizator(self, id):
        with self.connection.cursor() as cursor:
            cursor.execute('SELECT * FROM public."Organizators" WHERE id=%s', [id])
            return cursor.fetchall()

    def update_organizator(self, id, name, phone_number, email) -> None:
        with self.connection.cursor() as cursor:
            cursor.execute('UPDATE public."Organizators" SET name=%s, phone_number=%s,
email=%s WHERE id=%s', (name, phone_number, email, id))
            self.connection.commit()

    def delete_organizator(self, id):
        with self.connection.cursor() as cursor:
            cursor.execute('DELETE FROM public."Organizators" WHERE id=%s', [id])
            self.connection.commit()

    def add_place(self, name, phone_number, email, address) -> None:
        with self.connection.cursor() as cursor:
            cursor.execute('INSERT INTO public."Places" (name, phone_number, email,
address) VALUES (%s, %s, %s, %s)', (name, phone_number, email, address))
            self.connection.commit()

    def get_all_places(self):
        with self.connection.cursor() as cursor:
            cursor.execute('SELECT * FROM public."Places"')
            return cursor.fetchall()

    def get_place(self, id):
        with self.connection.cursor() as cursor:
            cursor.execute('SELECT * FROM public."Places" WHERE id=%s', [id])
            return cursor.fetchall()

    def update_place(self, id, name, phone_number, email, address) -> None:
        with self.connection.cursor() as cursor:
            cursor.execute('UPDATE public."Places" SET name=%s, phone_number=%s, email=%s,
address=%s WHERE id=%s', (name, phone_number, email, address, id))
            self.connection.commit()

    def delete_place(self, id):
        with self.connection.cursor() as cursor:
            cursor.execute('DELETE FROM public."Places" WHERE id=%s', [id])
            self.connection.commit()

    def add_wedding(self, partner1_name, partner2_name,
                    date, start_time, end_time,
                    client_id, place_id, organizer_id) -> None:
        with self.connection.cursor() as cursor:
            cursor.execute('INSERT INTO public."Weddings" (partner1_name, partner2_name, '
                        'date, start_time, end_time, '
                        'client_id, place_id, organizer_id) VALUES (%s, %s, %s, %s,
%s, %s, %s, %s)',
                        (partner1_name, partner2_name,
                        date, start_time, end_time,
                        client_id, place_id, organizer_id))
            self.connection.commit()

    def get_all_weddings(self):
        with self.connection.cursor() as cursor:
            cursor.execute('SELECT * FROM public."Weddings"')
            return cursor.fetchall()

    def get_wedding(self):
        with self.connection.cursor() as cursor:
            cursor.execute('SELECT * FROM public."Weddings" WHERE client_id=%s', [id])
            return cursor.fetchall()

```

```
def update_organizer(self, id, name, phone_number, email) -> None:
    with self.connection.cursor() as cursor:
        cursor.execute('UPDATE public."Organizers" SET name=%s, phone_number=%s,
email=%s WHERE id=%s', (name, phone_number, email, id))
        self.connection.commit()

def update_wedding(self, partner1_name, partner2_name,
date, start_time, end_time,
client_id, place_id, organizer_id) -> None:
    with self.connection.cursor() as cursor:
        cursor.execute('UPDATE public."Weddings" SET partner1_name=%s,
partner2_name=%s, '
                        'date=%s, start_time=%s, end_time=%s, '
                        'place_id=%s, organizer_id=%s WHERE client_id=%s',
(partner1_name, partner2_name,
date, start_time, end_time,
place_id, organizer_id, client_id))
        self.connection.commit()

def delete_wedding(self, client_id):
    with self.connection.cursor() as cursor:
        cursor.execute('DELETE FROM public."Weddings" WHERE client_id=%s', [client_id])
        self.connection.commit()

def generate_batch(self, rows):
    with self.connection.cursor() as cursor:
        try:
            cursor.execute(
                'insert into "Clients" (name, phone_number, email)\n'
                '(select name, \n'
                " ('+380' || (random() * 1000000000)::int) as phone_number, \n"
                " name || '_' || (random() * 1000)::int::varchar(3) || '@' || case
when random() > 0.5 then 'gmail.com' else 'ukr.net' end as email\n"
                'from (select (chr((random() * 25)::int + 65)) || \n'
                '      (chr((random() * 25)::int + 97)) || \n'
                '      (chr((random() * 25)::int + 97)) || \n'
                '      (chr((random() * 25)::int + 97)) as name\n'
                '      from generate_series(1, %s)))',
                [rows])
            cursor.execute(
                'insert into "Organizers" (name, phone_number, email) \n'
                '(select name, \n'
                " ('+380' || (random() * 1000000000)::int) as phone_number, \n"
                " name || '_' || (random() * 1000)::int::varchar(3) || row::text ||
'@' || case when random() > 0.5 then 'gmail.com' else 'ukr.net' end as email\n"
                'from (select (chr((random() * 25)::int + 65)) || \n'
                '      (chr((random() * 25)::int + 97)) || \n'
                '      (chr((random() * 25)::int + 97)) || \n'
                '      (chr((random() * 25)::int + 97)) as name, \n'
                '      row'
                '      from generate_series(1, %s) as row))',
                [rows])
            cursor.execute(
                'insert into "Places" (name, phone_number, email, address) \n'
                '(select name, \n'
                " ('+380' || (random() * 1000000000)::int) as phone_number, \n"
                " replace(lower(name), ' ', '_') || '_rest_' || (random() *
1000)::int::varchar(3) || row::text || '@' || case when random() > 0.5 then 'gmail.com'
else 'ukr.net' end as email, \n"
                'address\n'
                'from (select (chr((random() * 25)::int + 65)) || \n'
                '      (chr((random() * 25)::int + 97)) || \n'
                '      (chr((random() * 25)::int + 97)) || \n'
                '      (chr((random() * 25)::int + 97)) || ' ' ' || \n'
                '      (chr((random() * 25)::int + 65)) || \n'
                '      (chr((random() * 25)::int + 97)) || \n'
                '      (chr((random() * 25)::int + 97)) || \n'
                '      (chr((random() * 25)::int + 97)) as name, \n'
                " (random() * 99 + 1)::int::varchar(2) || ' ' || \n"
                '      (chr((random() * 25)::int + 65)) || \n'
                '      (chr((random() * 25)::int + 97)) || \n'
                '      (chr((random() * 25)::int + 97)) || \n'
                '      (chr((random() * 25)::int + 97)) || \n'
                '      (chr((random() * 25)::int + 97)) || \n'
                '      (chr((random() * 25)::int + 97)) || \n'
```

```

        '                (chr((random() * 25)::int + 97)) || \n'
        '                (chr((random() * 25)::int + 97)) || ' St.' as address,
\n"
        '                row'
        '                from generate_series(1, %s) as row))',
[rows])
cursor.execute(
    "with new_weddings as (\n"
    "select * \n"
    "from (select      (chr((random() * 25)::int + 65)) || \n"
    "                (chr((random() * 25)::int + 97)) || \n"
    "                (chr((random() * 25)::int + 97)) || \n"
    "                (chr((random() * 25)::int + 97)) as partner1_name, \n"
    "                (chr((random() * 25)::int + 65)) || \n"
    "                (chr((random() * 25)::int + 97)) || \n"
    "                (chr((random() * 25)::int + 97)) || \n"
    "                (chr((random() * 25)::int + 97)) as partner2_name, \n"
    "                '2024-01-01'::date + (random() * 365)::int as date, \n"
    "                make_time((random() * 4 + 12)::int, (random() *
59)::int, 0) as start_time, \n"
    "                make_time((random() * 5 + 18)::int, (random() *
59)::int, 0) as end_time, \n"
    "                floor(random() * %s + 1)::int + ((select max(id) from
\n\"Organizers\n\") - %s) as organizer_id, \n"
    "                floor(random() * %s + 1)::int + ((select max(id) from
\n\"Places\n\") - %s) as place_id, \n"
    "                client_id \n"
    "                from generate_series((select max(id) - %s + 1 from
\n\"Clients\n\"), (select max(id) from \n\"Clients\n\")) as client_id) \n"
    ') insert into "Weddings" (partner1_name, partner2_name, date,
start_time, end_time, client_id, organizer_id, place_id) '
    "(select partner1_name, partner2_name, date, start_time, end_time,
client_id, organizer_id, place_id from new_weddings)",
    (rows, rows, rows, rows, rows)
)
except Exception as e:
    self.connection.rollback()
    raise e
self.connection.commit()

def client_date_partners(self, left_date, right_date):
    with self.connection.cursor() as cursor:
        cursor.execute(
            'select client_id, "Clients".name, date, partner1_name, partner2_name\n'
            'from "Weddings"\n'
            'join "Clients" on client_id = "Clients".id\n'
            'where date >= %s and date <= %s\n'
            'group by client_id, "Clients".name, date, partner1_name, partner2_name\n'
            'order by date asc',
            (left_date, right_date)
        )
    return cursor.fetchall()

def organizer_weddings(self, date, min_weddings):
    with self.connection.cursor() as cursor:
        cursor.execute(
            'select * \n'
            'from (\n'
            '    select id as organizer, name, phone_number, email, count(client_id)
as weddings\n'
            '    from "Weddings"\n'
            '    join "Organizers" on id=organizer_id\n'
            '    where date > %s\n'
            '    group by id, name, phone_number, email\n'
            ') \n'
            'where weddings >= %s\n'
            'order by weddings desc',
            (date, min_weddings)
        )
    return cursor.fetchall()

def weddings_place(self, date):
    with self.connection.cursor() as cursor:
        cursor.execute(
            'select count(client_id) as weddings, address, place_id as place\n'
            'from "Clients"\n'

```

```
'join "Weddings" on client_id="Clients".id\n'  
'join "Places" on place_id="Places".id\n'  
'join "Organizers" on organizer_id="Organizers".id\n'  
"where date > %s\n"  
'group by place_id, address\n',  
(date,)  
)  
return cursor.fetchall()
```

Даний модуль імпортується у модулі Controller. Всі виклики цього модуля також знаходяться тільки у Controller.

При ініціалізації об'єкта класу Model, створюється підключення до бази даних, після чого у контролер обробляє ввід користувача який надходить від об'єкта класу View і при необхідності викликає методи з класу Model.

Всі методи класу Model схожі: спочатку метод створює курсор, який потім виконує запит, після чого або закінчує транзакцію, таким чином зберігаючи зміни після запита або повертає отриману таблицю.

Метод які вносять зміни у таблиці (такі які починаються на add_, update_, delete_), комітять зміни. Всі інші методи потрібні для отримання даних, а тому виконують метод cursor.fetchall().

Також тільки у цьому модулі збереженні SQL запити.

add_ методи, а також метод generate_batch роблять запити на вставку у відповідну таблицю рядка (generate_batch генерує вказану кількість випадкових рядків) з переданими даними.

get_, get_all_ методи, а також методи пошуку client_date_partners, organizer_weddings, weddings_place, роблять запити на отримання певних таблиць даних. get_ отримує один рядок за основним ключем.

update_ методи змінюють поля у рядку з вказаним основним ключем.

delete_ методи видаляють рядок з вказаним основним ключем.