**ДНІПРОВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ОЛЕСЯ ГОНЧАРА**

**Факультет прикладної математики**

**Кафедра математичного забезпечення ЕОМ**

**Звіт**

з практичної роботи № 3

з курсу «Бази даних»

Виконав:

студент 3-го курсу

групи ПЗ-21у-1

Войцехов М.О.

м. Дніпро

2023 р.

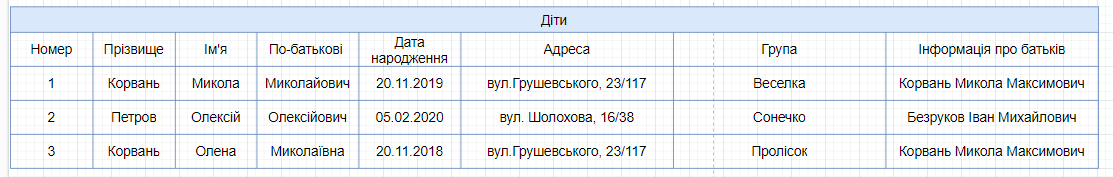
1. Нормалізація таблиці

Таблиці приведені до **1 нормальної форми**, так як всі значення є неділимі та в якості значення атрибутів таблиці не використовуються групи та масиви даних, що повторюються.

Таблиці приведені до **2 нормальної форми**, так як таблиця приведена до 1НФ та у таблиці відсутні часткові залежності. Тобто кожен не ключовий атрибут таблиці не залежить від первинного ключа таблиці.

Для приведення до **3 нормальної форми**, необхідно привести таблицю до 2НФ та позбавлення транзитивної залежності.

1. Таблиці "Діти" і "Батьки" потребують аналізу та приведення до третьої нормальної форми (3NF).



Таблиця «Діти» до приведення

Таблиця "Діти":

• ID дитини (первинний ключ)

• Прізвище

• Ім'я

• По батькові

• Дата народження

• Адреса

• Група

• Інформація про батьків

Аналіз:

У таблиці "Діти" відсутня повторювана інформація, що вказує на відповідність першій нормальній формі (1NF).

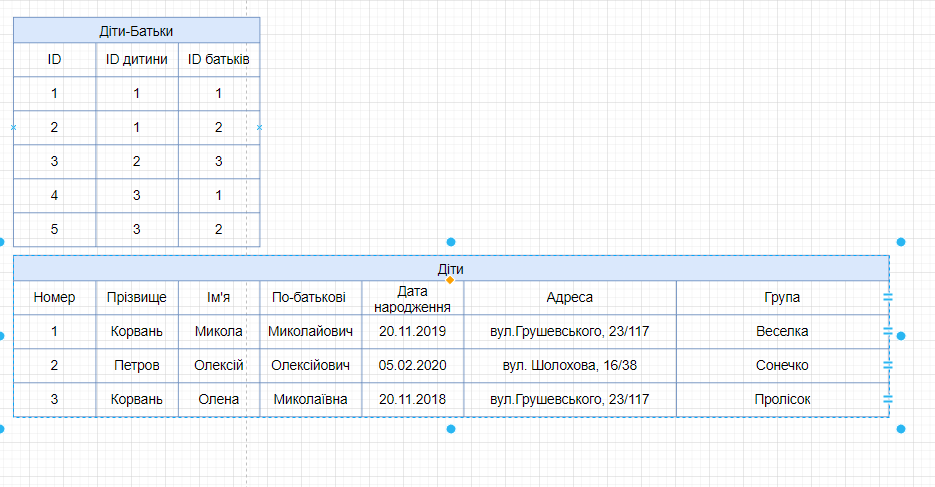
Поле "Група" є пов'язаним з таблицею "Групи", за допомогою зовнішнього ключа.

Приведення до 3NF:

1. Розділимо таблицю "Діти" на дві окремі таблиці: "Діти1" і "Діти\_Батьки".

Таблиця "Діти\_Батьки":

* + - ID зв'язку (первинний ключ)
    - ID дитини (зовнішній ключ до таблиці "Діти")
    - ID батьків (зовнішній ключ до таблиці "Батьки")



Таблиця після приведення

Таблиця "Батьки":

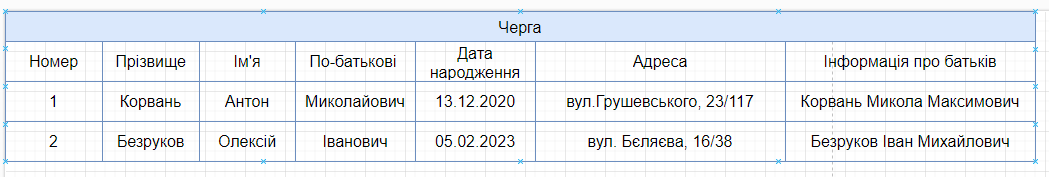
* + ID батьків (первинний ключ)
  + Ім'я
  + Прізвище
  + Адреса
  + Телефон
  + Електронна пошта

Аналіз:

У таблиці "Батьки" відсутня повторювана інформація, що вказує на відповідність першій нормальній формі (1NF).

Приведення до 3NF: Таблиця "Батьки" вже знаходиться в третій нормальній формі (3NF), оскільки у неї відсутні повторювані групи полів та всі дані атомарні.

Таблиці "Черга" потребує аналізу та приведення до третьої нормальної форми (3NF).



Таблиця «Черга» до приведення

Таблиця "Черга":

• Номер

• Прізвище

• Ім'я

• По батькові

• Дата народження

• Адреса

• Інформація про батьків

Аналіз:

* У таблиці "Черга" відсутня повторювана інформація, що вказує на відповідність першій нормальній формі (1NF).
* Поле "Інформація про батьків" може містити додаткову інформацію про батьків, яка не відноситься безпосередньо до даної таблиці. Це може призвести до повторювання даних та зайвої ненормалізації.

Приведення до 3NF:

1. Розділимо таблицю "Черга" на дві окремі таблиці: "Черга1" і "Черга-батьки".

Таблиця "Черга1":

• Номер

• Прізвище

• Ім'я

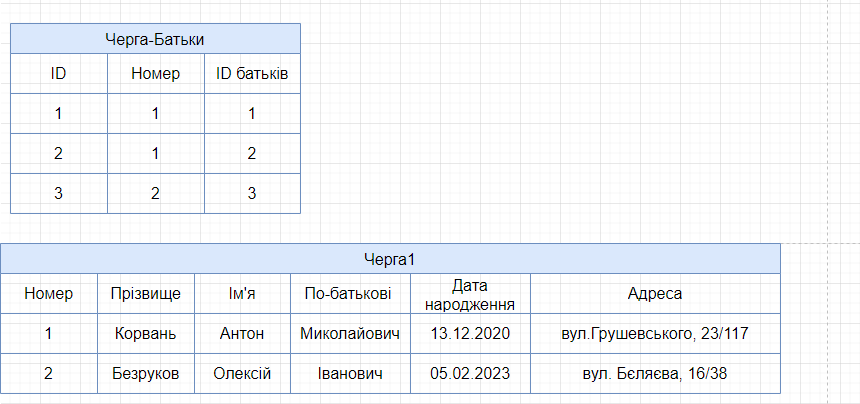
• По батькові

• Дата народження

• Адреса

Таблиця "Черга\_батьки":

* + - Номер (первинний ключ)
    - ID дитини (первинний ключ)
    - ID батьків (зовнішній ключ до таблиці "Батьки")



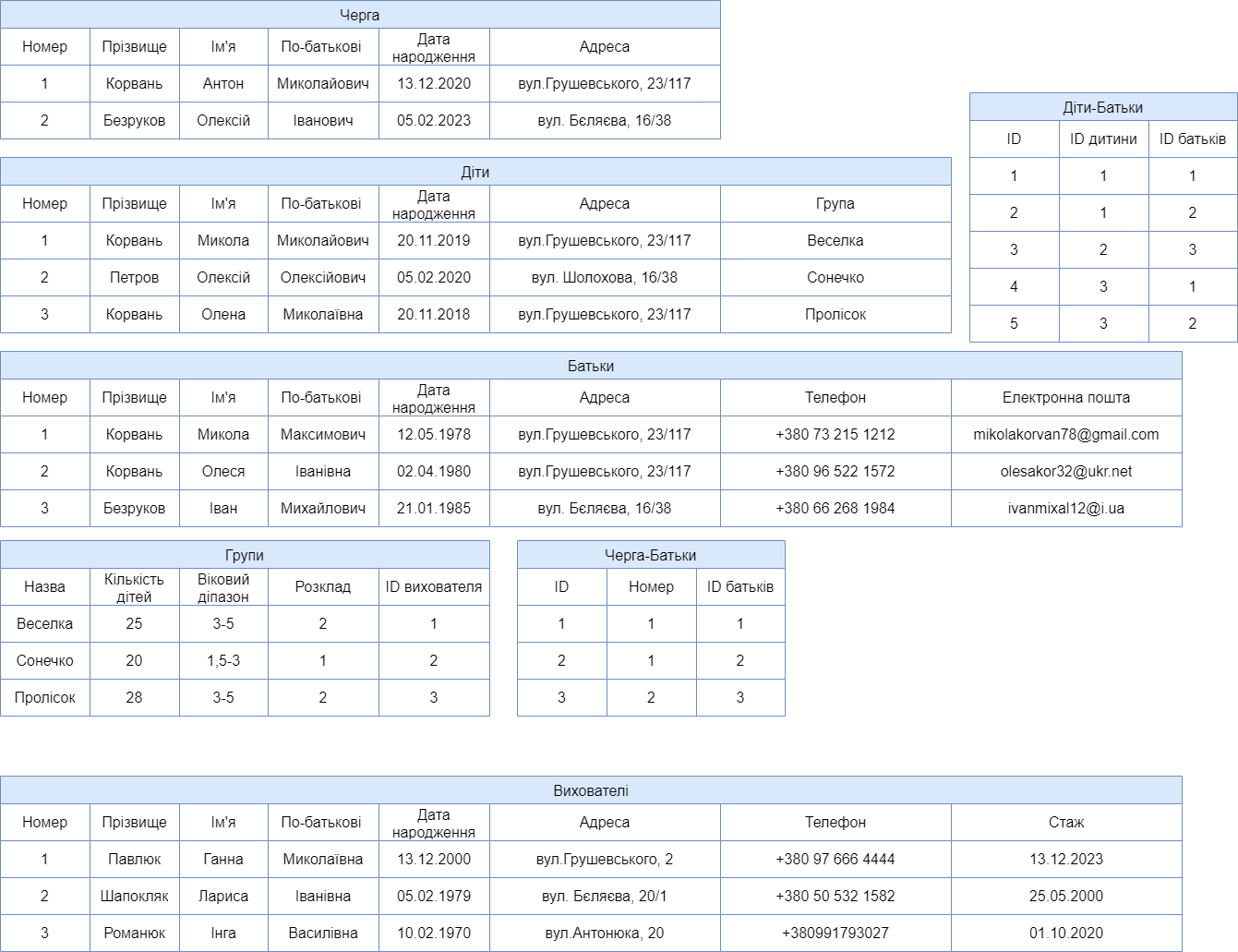
Таблиця «Черга» після приведення

У таблиці "Групи" відсутня повторювана інформація, що вказує на відповідність першій нормальній формі (1NF).

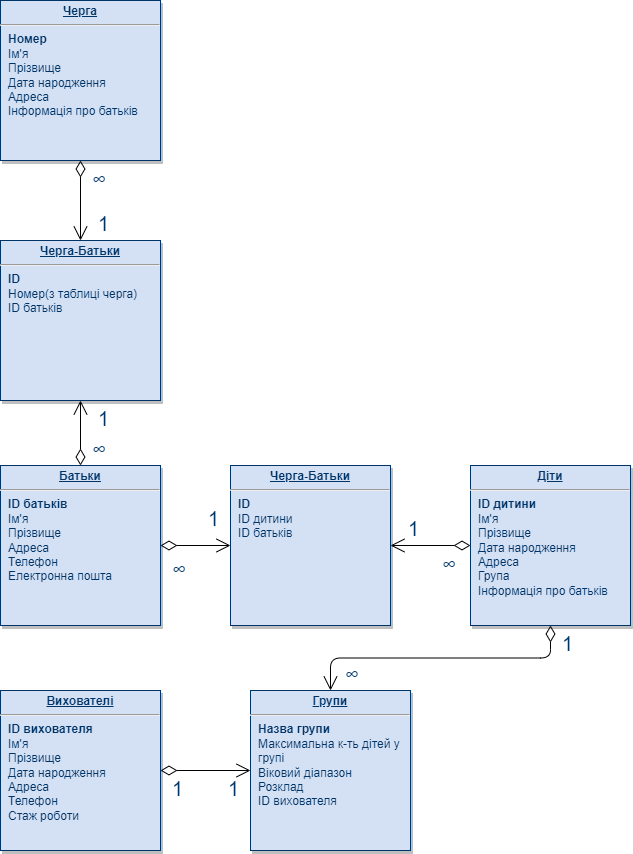
Таблиця " Групи " вже знаходиться в третій нормальній формі (3NF), оскільки у неї відсутні повторювані групи полів та всі дані атомарні.

У таблиці "Вихователі" відсутня повторювана інформація, що вказує на відповідність першій нормальній формі (1NF).

Таблиця "Вихователі" вже знаходиться в третій нормальній формі (3NF), оскільки у неї відсутні повторювані групи полів та всі дані атомарні.



2. ER діаграма



Запит 1.1: Додавання нової дитини до бази даних (адміністратор, вихователь)

Оптимальність і швидкість роботи запиту залежить від кількості полів, які потрібно заповнити, а також від складності алгоритму для зберігання даних в базі даних. Оптимальний запит має бути швидким та надійним. Запит має бути зрозумілим для користувачів і забезпечувати необхідну безпеку даних.

Запит 1.2: Додавання батьків дитини (адміністратор, вихователь)

Оптимальність і швидкість роботи запиту залежить від кількості полів, які потрібно заповнити, а також від складності алгоритму для зберігання даних в базі даних. Оптимальний запит має бути швидким та надійним. Запит має бути зрозумілим для користувачів і забезпечувати необхідну безпеку даних.

Запит 1.3: Виведення інформації певної дитини (адміністратор, вихователь)

Оптимальність і швидкість роботи запиту залежить від кількості даних, які потрібно вивести, і від кількості записів у базі даних. Запит має бути зрозумілим для користувачів та забезпечувати точну інформацію про дитину.

Запит 1.4 виводить кількість дітей в кожній групі, що може бути корисним для адміністратора та вихователів при плануванні роботи та забезпеченні належного догляду за дітьми.

Запит 1.5 дозволяє переглянути інформацію про батьків певної дитини, що є важливою інформацією для вихователів та адміністратора при взаємодії з батьками та розв'язанні питань, пов'язаних з дітьми.

Запит 1.7 Надання списку дітей, які в цьому році закінчують дитячий садок: цей запит дозволяє адміністратору та вихователям знайти дітей, які готуються до виходу з дитячого садка. Це може бути корисно для планування подальшої роботи зі студентами, організації випускних заходів і т.д.

Запит 2.1 Перегляд інформації про батьків дитини: цей запит дозволяє вихователям отримати інформацію про батьків конкретної дитини. Це може бути корисно для зв'язку з батьками, розуміння сімейної ситуації дитини, а також для забезпечення безпеки та добробуту дітей.

Запит 2.2 Виведення інформації про дітей певних батьків (адміністратор, вихователь): Цей запит виконує необхідну дію, а саме виводить інформацію про дітей, які належать певним батькам. Проте, оптимальність і швидкість роботи залежить від того, як саме виконується запит. Якщо в базі даних зберігається інформація про батьків та дітей у відповідних таблицях, то запит можна виконати швидко і ефективно, використовуючи JOIN-операції.

Запит 2.3 Виведення інформації про усіх батьків (адміністратор, вихователь): Цей запит виконує необхідну дію, а саме виводить інформацію про всіх батьків. Проте, якщо база даних містить велику кількість записів про батьків, то виконання запиту може зайняти багато часу. Оптимальність запиту можна покращити, якщо використовувати оптимізацію запиту, наприклад, обмеження вибірки або використання індексів.

Запит 2.4 Визначити кількість батьків в кожній групи (адміністратор): Цей запит виконує необхідну дію, а саме визначає кількість батьків в кожній групі. Якщо в базі даних зберігається інформація про батьків та групи у відповідних таблицях, то запит можна виконати швидко і ефективно, використовуючи GROUP BY-операції.

Запит 3.2 - Перегляд даних про всіх вихователів (адміністратор): Цей запит виконує необхідну дію, а саме - дозволяє адміністратору переглядати інформацію про всіх вихователів, що знаходяться у базі даних. Проте оптимальність і швидкість роботи цього запиту може залежати від кількості записів про вихователів, що містяться в базі даних.

Запит 3.3 - Виведення вихователів, що працюють в 2 або більше групах (адміністратор): Цей запит також виконує необхідну дію, а саме - дозволяє адміністратору переглядати інформацію про вихователів, які працюють в двох або більше групах. Оптимальність і швидкість роботи цього запиту може залежати від кількості записів про вихователів, що містяться в базі даних, а також кількості груп, в яких вони працюють.

Запит 3.3 - Виведення загальної кількості вихователів (адміністратор): Цей запит виконує необхідну дію, а саме - дозволяє адміністратору переглядати загальну кількість вихователів, що знаходяться у базі даних. Оптимальність і швидкість роботи цього запиту зазвичай буде високою, оскільки він не потребує складних обчислень.

Запит 3.4 "Перегляд вихователів, що працюють в групах певного віку" є доцільним і необхідним для адміністратора, оскільки дозволяє швидко знайти вихователів, які працюють з дітьми відповідного віку. Швидкість роботи запиту може залежати від того, яким чином зберігається вік дітей і вихователів в базі даних.

Запит 3.5 "Пошук вихователя за прізвищем" також є корисним і необхідним, оскільки дозволяє швидко знайти вихователя за його прізвищем. Швидкість роботи запиту може залежати від того, яким чином зберігається інформація про прізвища вихователів.

Запит 3.6 "Виведення вихователів, вік яких наближається до пенсійного" також може бути корисним для адміністратора, оскільки дозволяє вчасно знайти вихователів, яким потрібно буде замінити через наближення до пенсійного віку. Швидкість роботи запиту може залежати від того, яким чином зберігається вік вихователів в базі даних.

Запит 4.1 Перегляд інформації про групу (всі): Цей запит виконує необхідну дію і є досить оптимальним та швидким, оскільки він просто виводить інформацію про конкретну групу з бази даних без будь-яких додаткових операцій.

Запит 4.2 Перегляд груп, певної вікової категорії(адміністратор): Цей запит також виконує необхідну дію, але може бути менш оптимальним, якщо база даних містить велику кількість записів, оскільки потребує фільтрації за віковим критерієм. Швидкість роботи буде залежати від ефективності операції фільтрації.

Запит 4.3 Виведення кількості дітей, які виховуються в першій зміні та в другій зміні (адміністратор, вихователь): Цей запит виглядає досить простим і має потенціал бути виконаним досить швидко, якщо база даних вірно побудована. Однак, важливо врахувати, що залежно від структури бази даних, цей запит може потребувати складних запитів та з'єднань таблиць.

Запит 4.4 Виведення інформації про дітей, що навчаються в певній групі (адміністратор): Цей запит також має потенціал бути виконаним досить швидко, якщо база даних вірно побудована. Проте, якщо в базі даних є багато груп та дітей, цей запит може потребувати досить багато часу для виконання.

Запит 4.5 Виведення кількості груп та загальної кількості дітей(адміністратор, вихователь): Цей запит є простим і може бути виконаним досить швидко, якщо дані зберігаються в окремих полях. Проте, якщо дані потрібно отримати з декількох таблиць, то цей запит може потребувати додаткового часу на виконання.

Запит 4.6 Виведення інформації про батьків певної групи(адміністратор, вихователь): Цей запит може бути складним, особливо якщо потрібно вибрати інформацію з декількох таблиць. Проте, якщо база даних вірно побудована, то цей запит може бути виконаним досить швидко.