вМіністерство освіти і науки України

Харківський національний університет ім. В. Н. Каразіна

Факультет комп’ютерних наук

Кафедра безпеки інформаційних систем і технологій

**КУРСОВА РОБОТА**

з дисципліни «Захист інформації в інформаційно-комунікаційних системах» на тему «Розробка бази даних школи із забезпеченням захисту її даних»

Виконав:

студент групи КБ-31

Кравченко Є.М.

Перевірив:

професор

Єсін В.І

Харків 2020

**ЗМІСТ**

[**ВСТУП** 3](#_Toc42960535)

[**РОЗДІЛ 1** 5](#_Toc42960536)

[**ІНФОЛОГІЧНЕ ПРОЕКТУВАННЯ** 5](#_Toc42960537)

[**1.1 Аналіз предметної області** 5](#_Toc42960538)

[**1.2 Аналіз інформаційних завдань і кола користувачів системи** 8](#_Toc42960539)

[**2. ВИЗНАЧЕННЯ ВИМОГ ДО ОПЕРАЦІЙНОЇ ОБСТАНОВЦІ** 9](#_Toc42960540)

[**2.1 Обсяг роботи інформаційної системи** 9](#_Toc42960541)

[**3. ВИБІР СИСТЕМИ УПРАВЛІННЯ БАЗОЮ ДАНИХ (СУБД)** 11](#_Toc42960542)

[**РОЗДІЛ 4** 14](#_Toc42960543)

[**ЛОГІЧНЕ ПРОЕКТУВАННЯ** 14](#_Toc42960544)

[**4.1 Перетворення ER - діаграми в схему бази даних** 14](#_Toc42960545)

[**4.2 Нормалізація відносин** 17](#_Toc42960546)

[**РОЗДІЛ 5** 19](#_Toc42960547)

[**ФІЗИЧНЕ ПРОЕКТУВАННЯ** 19](#_Toc42960548)

[**5.1 Створення форм, запитів і звітів** 19](#_Toc42960549)

[**5.2 Захист даних** 20](#_Toc42960550)

[**ВИСНОВКИ** 24](#_Toc42960551)

[**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ** 25](#_Toc42960552)

# **ВСТУП**

Останнім часом, через необхідність людині сприймати велику кількость інформації, стає популярною область управління даними. Користувач прагне упорядкувати потрібну йому інформацію за різними ознаками і швидко робити вибірку з довільним поєднанням ознак. Зробити це можливо, тільки якщо дані структуровані. Неструктурованими називаються дані, записані, наприклад, в текстовому файлі. Структуровані дані можна створити в базі даних.

База даних – це організована структура, призначена для зберігання інформації. Зауважу в цьому визначенні: дані та інформація – взаємопов'язані поняття, але не тотожні. Причини цієї невідповідності чисто історичні. У ті роки, коли формувалося поняття баз даних, в них дійсно зберігалися тільки дані. Однак сьогодні більшість систем управління базами даних дозволяють розміщувати в своїх структурах не тільки дані, але і методи, за допомогою яких відбувається взаємодія із користувачем або з іншими програмно-апаратними комплексами. Таким чином, можу говорити, що в сучасних базах даних зберігаються не лише дані, а й інформація.

Переробляти великий обсяг інформації в задані терміни практично неможливо без спеціальних засобів обробки інформації. Хоча більша частина інформації все ще знаходиться поза ЕОМ, однак, вартість запам'ятовуючих пристроїв обчислювальних машин швидко знижується, тому зберігати дані в файлах ЕОМ буде дешевше, ніж на папері.

Використання бази даних значно підвищує продуктивність праці користувача і спрощує використання різних засобів по перетворенню даних. Поява БД стала чи не найважливішою подією для людей, які щодня мають справу з величезною кількістю даних.

Однак створення бази даних кропітке і складне завдання. Правильно створена БД може значно полегшити життя фахівця в конкретній предметній області, не вірно ж створена БД, якщо і буде працювати, може значно ускладнити доступ до інформації і життя користувача.

Метою курсової роботи є створення бази даних для співробітників школи. Це забезпечить легкий доступ до потрібної інформації (Педагогічний стаж, класне керівництво, кабінет, розклад, гуртки і тд.).

# **РОЗДІЛ 1**

# **ІНФОЛОГІЧНЕ ПРОЕКТУВАННЯ**

## **1.1 Аналіз предметної області**

Розробляється база даних призначена для зберігання і роботи з даними в такій предметної області як «Школа». Вона включає в себе повну інформацію про вчителів, класів яких вони вчать. Надає доступ до розкладу занять, а також до розкладу додаткової освіти, тобто гуртках. Для оптимальної взаємодії цих сторін необхідно автоматизувати інформаційні процеси, що призведе до поліпшення якості і підвищення рівня знань.

Розглянувши предметну область, можна виділити наступні сутності:

* Вчителі
* Класи
* Гуртки
* Учні
* Кабінет
* Розклад
* Заняття

Під поняттям «сутність» розуміється будь-який помітний об'єкт про який необхідно зберегати інформацію в базі даних.

Існує чотири основні класи сутностей: стрижневі, асоціативні, характеристичні і позначаюча.

Стрижневими є:

* Сутність «Учитель» буде містити інформацію про вчителя: ПІБ, дата народження, предмет, загальний стаж, педагогічний стаж.
* Сутність «Класи» буде включати номер, кількість осіб, старосту, а також середню успішність. «Гуртки» буде міститися інформація про його назву, ПІБ вчителя, кількість осіб що займаються в ньому, час і день тижня.
* Сутності «Учні» буде містити інформацію про ПІБ учня, дата народження і номер класу.

Всі наведені сутності мають незалежне існування. Асоціаціями в проектованій базі даних будуть суті:

* Сутність «Розклад», в якій будуть знаходитися ПІБ вчителя, номер класу, день тижня, а також номер уроку.
* Сутність «Заняття», вона буде містити назву гуртка і ПІБ вчителя.

Також в базі даних є позначаюча сутність «Кабінет», що містить номер кабінету, паспорт і поверх.

Рис. 1.1 – ER - діаграма база даних «Школа»

Вчителі

Розклад

Гуртки

Класи

Заняття

Учні

Кабінет

## **1.2 Аналіз інформаційних завдань і кола користувачів системи**

Дана база розрахована для співробітників школи, яким доводиться працювати з навчальною програмою, що включає в себе розклад предметів, гуртків, а також класи і учні що в них навчаються.

Завданням даної бази даних є надання інформації про співробітників школи, здійснення зручного введення і поновлення даних в БД, а також знаходження одних даних по іншим.

# **2. ВИЗНАЧЕННЯ ВИМОГ ДО ОПЕРАЦІЙНОЇ ОБСТАНОВЦІ**

## **2.1 Обсяг роботи інформаційної системи**

Періодичність оновлення даних раз на рік.

База даних буде містити інформацію про:

* 20 вчителів;
* 12 кабінетів;
* 9 гуртків;
* 11 класів;

Всього кількість записів в БД 79.

**2.2 Об'єм пам'яті, що відводиться під дані**

Обсяг пам'яті необхідний для зберігання даних, можна наближено порахувати за формулою: ,

Де li – довжина запису в i - й таблиці;

Ni – зразкову (максимально можливе) кількість записів у таблиці;

Nai – кількість записів в архіві i таблиці.

Коефіцієнт 2 перед сумою потрібен для того, щоб виділити пам'ять для зберігання індексів, проміжних даних, для виконання об'ємних операцій (наприклад, сортування) і т.п.

Проведу розрахунок на прикладі таблиці «Вчителі» (таблица1)

Таблиця 1 Розмір полів в таблиці «Вчителі»

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ПІБ | Дата народження | Загальний стаж | Педагогічний стаж | Кслас керівн | К-ть годин | категорія | Кабінет | Предмет |
| 225 байт | 8 байт | 8 байт | 8 байт | 8 байт | 4 байта | 4 байта | 8 байт | 4 байта |

Довжина запису: 255

Тоді загальний обсяг пам'яті буде дорівнює:

МД = 2 \* 20 \* (225 + 8 + 8 + 8 + 8 + 4 + 4 + 8 + 8) = 11080байт

Наведу підсумкові значення обсягу пам'яті таблиць (Таблиця 2)

Таблиця 2 Обсяг пам'яті таблиць

|  |  |
| --- | --- |
| Назва таблиці | Объема памяти (байт) |
| Гуртки | 4608 |
| Класи | 5214 |
| Учні | 14220 |
| Кабінет | 5592 |
| Заняття | 12165 |
| Розклад | 15424 |

Обсяг пам'яті необхідний для зберігання даних, становить 68303 байт.

# **3. ВИБІР СИСТЕМИ УПРАВЛІННЯ БАЗОЮ ДАНИХ (СУБД)**

Серед найбільш яскравих представників систем управління базами даних можна відзначити: Lotus Approach, Microsoft Access, Borland dBase, Borland Paradox, Microsoft Visual FoxPro, Microsoft Visual Basic, а також баз даних Microsoft SQL Server і Oracle, використовувані в додатках, побудованих за технологією «клієнт-сервер». Фактично, у будь-якої сучасної СУБД існує аналог, що випускається іншою компанією, що має аналогічну область застосування і можливості, будь-який додаток здатний працювати з багатьма форматами представлення даних, здійснювати експорт та імпорт даних завдяки наявності великої кількості конвертерів. Загальноприйнятими, також, є технологи, що дозволяють використовувати можливості інших додатків, наприклад, текстових процесорів, пакетів побудови графіків і т.п., і вбудовані версії мов високого рівня (частіше - діалекти SQL і / або VBA) і засоби візуального програмування інтерфейсів розробляються . Тому вже не має істотного значення якою мовою і на основі якого пакету написано конкретний додаток, і який формат даних в ньому використовується. Більш того, стандартом «де-факто» стала «швидка розробка додатків» або RAD (від англійського Rapid Application Development), заснована на широко декларованої в літературі «відкритому підході», тобто необхідність і можливість використання різних прикладних програм і технологій для розробки більш гнучких і потужних систем обробки даних. Тому в одному ряду з «класичними» СУБД все частіше згадуються мови програмування Visual Basic 4.0 і Visual C ++, які дозволяють швидко створювати необхідні компоненти додатків, критичні за швидкістю роботи, які важко, а іноді неможливо розробити засобами «класичних» СУБД. Сучасний підхід до управління базами даних має на увазі також широке використання технології «клієнт-сервер».

Таким чином, на сьогоднішній день розробник не пов'язаний рамками якого-небудь конкретного пакету, а в залежності від поставленої задачі може використовувати самі різні програми. Тому, важливішим є загальний напрям розвитку СУБД та інших засобів розробки додатків в даний час.

Visual FoxPro відрізняється високою швидкістю, має вбудований об'єктно-орієнтована мова програмування з використанням xBase і SQL, діалекти яких вбудовані в багато СУБД. Має високий рівень об'єктної моделі. При використанні в обчислювальних мережах забезпечує як монопольний, так і роздільний доступ користувачів до даних. Застосовується для додатків масштабу підприємства для роботи на різних платформах: Windows 3.x, Windows 95, Macintosh ... Мінімальні ресурси ПК: для Visual FoxPro версії 3.0 - процесор 468DX, Windows 3.1, 95, NT, обсяг оперативної пам'яті 8 (12) Мб, займаний обсяг на ЖМД 15-80 Мб, а для Visual FoxPro версії 5.0 (випущена в 1997 році) - Windows 95 або NT, 486 з тактовою частотою 50 МГц, 10 Мб ОЗУ, від 15 до 240 Мб на ЖМД.

Visual Basic - це універсальний об'єктно-орієнтована мова програмування, діалекти якого вбудовані в Access, Visual FoxPro. Переваги: ​​універсальність, можливість створення компонентів OLE, невисокі вимоги до апаратних ресурсів ЕОМ. Застосовується для створення додатків середньої потужності, не пов'язаних з великою інтенсивністю обробки даних, розробки компонентів OLE, інтеграція компонентів Mi-crosoft Office. Мінімальні ресурси ПК: процесор 368DX, Windows 3.1, 95, NT, обсяг оперативної пам'яті 6 (16) Мб, займаний обсяг на ЖМД 8-36 Мб.

Visual C ++ - найбільш потужний об'єктно-орієнтована мова програмування, має необмеженої функціональністю. Призначений для створення компонентів додатків для виконання операцій, критичних за швидкістю.

SQL Server - сервер баз даних, реалізує підхід «клієнт-сервер» і взаємодіє з зазначеними пакетами. Головні переваги: високий ступінь захисту даних, потужні засоби для обробки даних, висока продуктивність. Область застосування: зберігання великих обсягів даних, зберігання високоцінних даних або даних, що вимагають дотримання режиму секретності. Мінімальні ресурси ПК: процесор 468DX-33МГц, Windows NT, обсяг оперативної пам'яті 16 (32) Мб, займаний обсяг на ЖМД 80 Мб.

Зазначені програмні продукти мають можливості візуального проектування інтерфейсу користувача, тобто розробник з готових фрагментів створює елементи інтерфейсу, програмує тільки їх зміни у відповідь на будь-які події.

У цій роботі використовувалася СУБД Access, що входить до складу популярного пакету Microsoft Office. Основні переваги: знайомий багатьом кінцевим користувачам і має високу стійкість даних, простий в освоєнні, може використовуватися непрофесійним програмістом, дозволяє готувати звіти з баз даних різних форматів. Призначений для створення звітів довільної форми на підставі різних даних і розробки некомерційних додатків.

# **РОЗДІЛ 4**

# **ЛОГІЧНЕ ПРОЕКТУВАННЯ**

## **4.1 Перетворення ER - діаграми в схему бази даних**

Перетворимо ЕR - діаграму в схему даних, використовуючи СУБД Access. Для цього сутності представляються таблицями, атрибути – полями. Зв'язки виставляються у відповідність з діаграмою.

Таблиця 3 «Вчителі»

|  |  |
| --- | --- |
| Назва поля | Тип даних |
| ПІБ вчителя | Текстовий  Розмір поля – 255  Обов'язкове поле – так |
| Дата народження | Дата  Формат поля – короткий формат дати  Обов'язкове поле – так |
| Звгальний стаж | Числовий  Розмір поля – довге ціле  Обов'язкове поле – так |
| Педагогічний стаж | Числовий  Розмір поля – 255  Обов'язкове поле – так |
| Категорія | Текстовий  Розмір поля – 255  Обов'язкове поле – так |
| Класне курівництво | Текстовий  Розмір поля – 255  Обов'язкове поле – так |
| Кабінет | Числовий  Розмір поля – 255  Обов'язкове поле – ні |
| Кількість годин | Числовий  Розмір поля – 255  Обов'язкове поле – так |
| Предмет | Текстовий  Розмір поля – 255  Обов'язкове поле – так |

Таблиця 5 «Кабінет»

|  |  |
| --- | --- |
| Назва поля | Тип даних |
| Поверх | Числовий  Розмір поля – довге ціле  Обов'язкове поле – так |
| Відповідальний вчитель | Текстовий  Розмір поля – 255  Обов'язкове поле – так |
| Паспорт | Текстовий  Розмір поля – 255  Обов'язкове поле – так |

Таблиця 5 «Гуртки»

|  |  |
| --- | --- |
| Назва поля | Тип даних |
| Назва | Текстовий  Розмір поля – 255  Обов'язкове поле – так |
| День тижня | Текстовий  Розмір поля – 255  Обов'язкове поле – так |
| ПІБ вчителя | Текстовий  Розмір поля – 255  Обов'язкове поле – так |
| Час проведення | Дата/Час  Обов'язкове поле – так |

Таблиця 6 «Заняття»

|  |  |
| --- | --- |
| Назава поля | Тип даних |
| Назва | Текстовий  Розмір поля – 255  Обов'язкове поле – так |
| ПІБ учня | Текстовий  Розмір поля – 255  Обов'язкове поле – так |

Таблиця 7 «Учні»

|  |  |
| --- | --- |
| Назва поля | Тип даних |
| ПІБ | Текстовий  Розмір поля – 255  Обов'язкове поле – так |
| Дата народження | Дата  Формат поля – короткий формат дати  Обов'язкове поле – так |
| № класу | Числовий |

Таблиця 8 «Розклад»

|  |  |
| --- | --- |
| Назва поля | Тип даних |
| ПІБ Вчителя | Текстовий  Розмір поля – 255  Обов'язкове поле – так |
| День тижня | Текстовий  Розмір поля – 255  Обов'язкове поле – так |
| Номер заняття | Числовий  Розмір поля – довге ціле  Обов'язкове поле – так |

Таблиця 9 «Класи»

|  |  |
| --- | --- |
| Назва поля | Тип даних |
| № класу | Числовий  Розмір поля – довге ціле  Обов'язкове поле – так |
| Кількість учнів | Числовий  Розмір поля – довге ціле  Обов'язкове поле – так |
| Староста | Текстовий  Розмір поля – довге ціле  Обов'язкове поле – так |

Об'єднуючи все таблиці, отримаємо схему бази даних «Школа» в форматі, в якому вона виглядає у вікні схеми даних програми Microsoft Access.

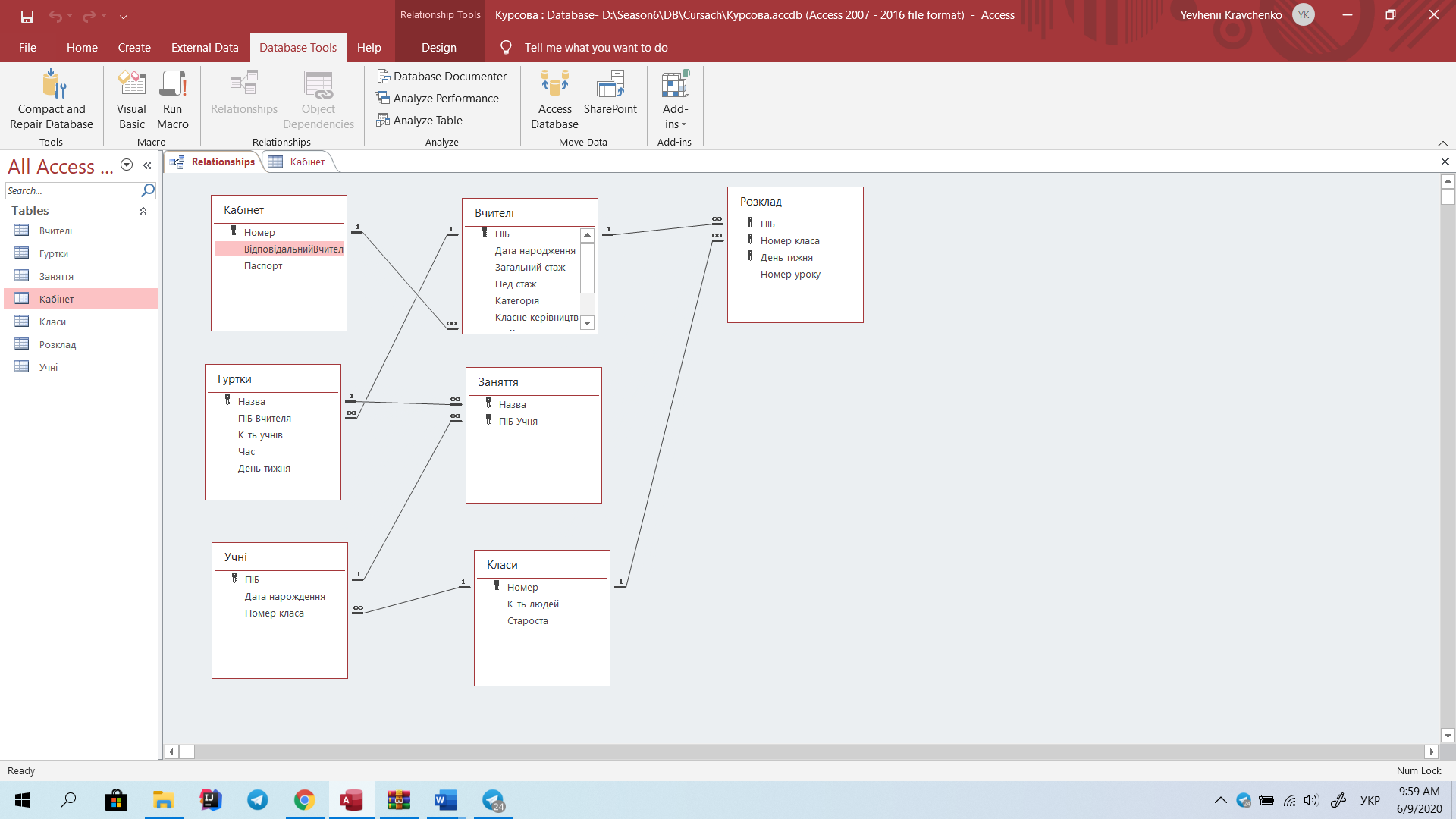


Рис. 4.1 – Схема бази даних «Школа»

## **4.2 Нормалізація відносин**

Проведу нормалізацію отриманих відносин. Перевірю відносини на першу нормальну форму (1НФ). Відношення знаходиться в 1НФ, якщо всі його атрибути прості. Розглядаючи на прикладі таблиці «Вчителі», видно, що атрибут «ПІБ вчителя» є складовим, так як він складається з трьох компонентів, тобто його можна розділити на 3 незалежних атрибути, так як в даній БД ці компоненти окремо не використовуються, то ми приймаємо цей атрибут за простий. Всі інші атрибути, є простими.

Перевірю відносини на другу нормальну форму (2НФ). Відношення знаходиться в 2НФ, якщо воно знаходиться в 1НФ і кожен неключових атрибут функціонально повно залежить від складеного первинного ключа. Наприклад, в таблиці «Розклад», неключових атрибут «Номер уроку» не залежить від частини складного ключа в цілому, а залежить від нього в цілому.

Перевірю відносини на третю нормальну форму (3НФ). Відношення знаходиться в 3НФ, якщо воно знаходиться в 2НФ і кожен не ключовий атрибут нетранзитивно залежить від первинного ключа, тобто не залежить функціонально від будь-якого іншого не ключового поля. Транзитивная залежність відсутня, так як неключових атрибути не залежать одне від одного.

# **РОЗДІЛ 5**

# **ФІЗИЧНЕ ПРОЕКТУВАННЯ**

## **5.1 Створення форм, запитів і звітів**

Доступ до всіх компонентів, а саме, форм і звітів здійснюється через вікно головної кнопкової форми, яка запускається після відкриття бази даних.

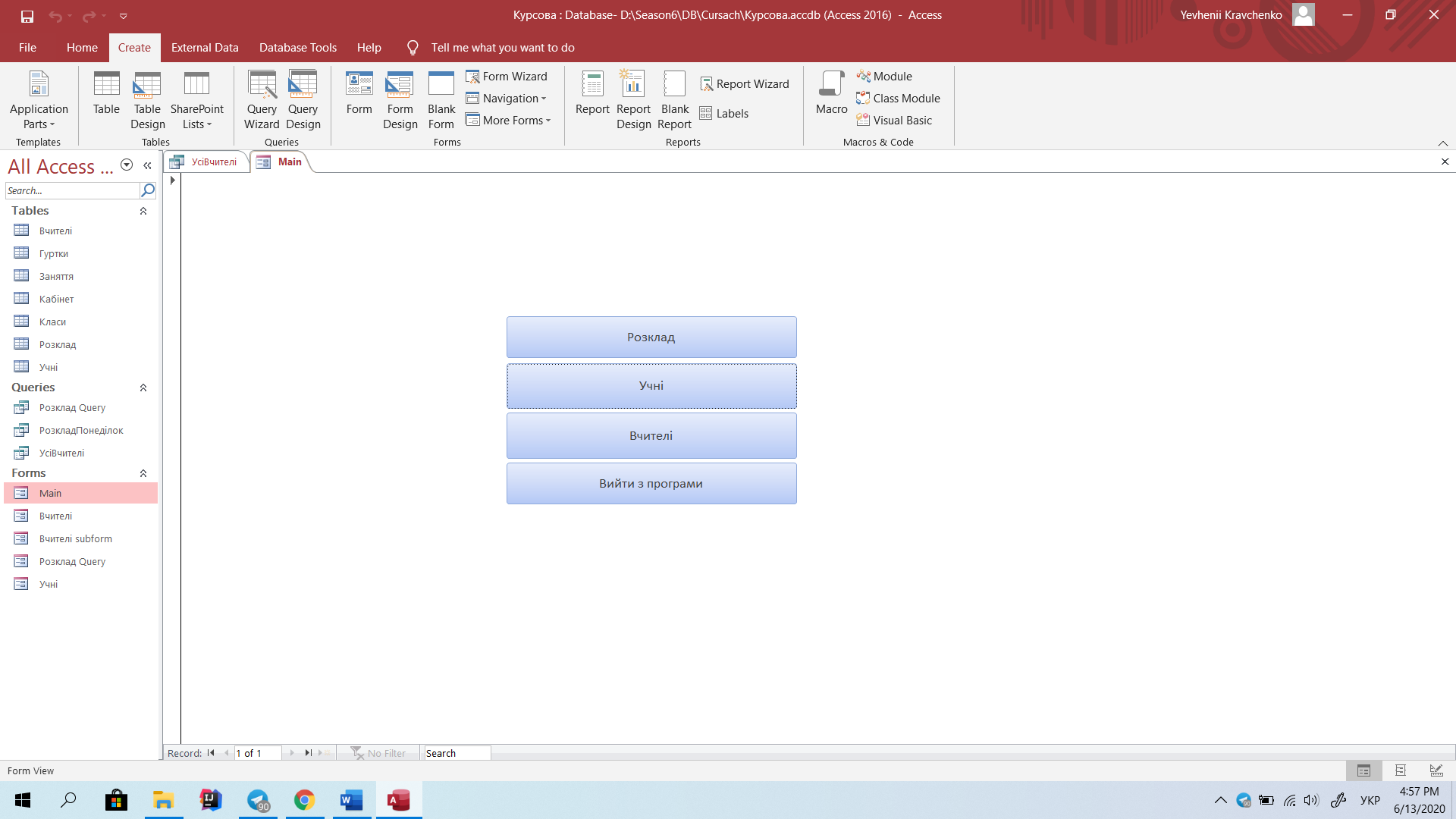


Рис. 5.1 – Головна форма пронрами

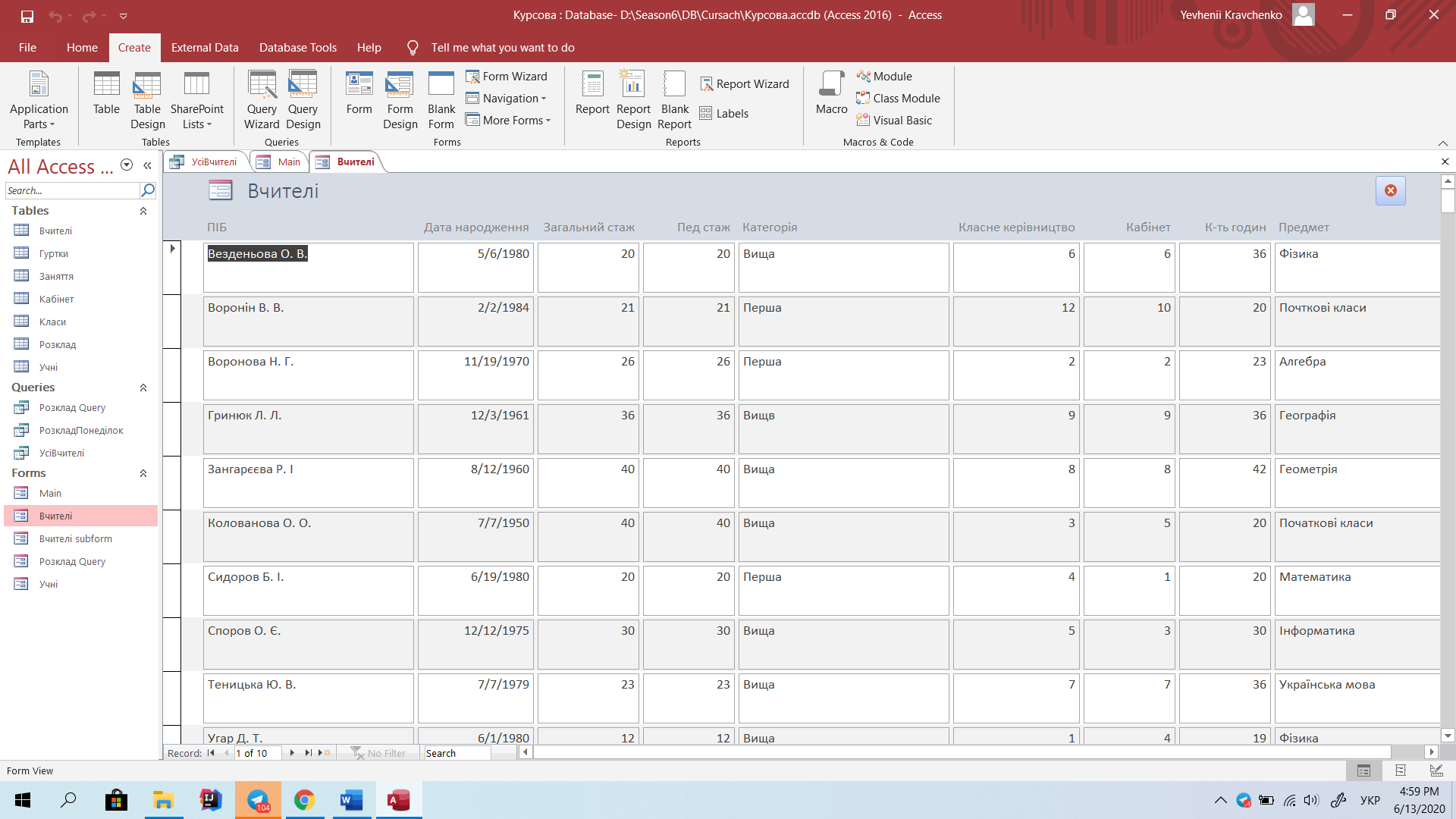


Рис. 5.2 – Форма відображення вчителів

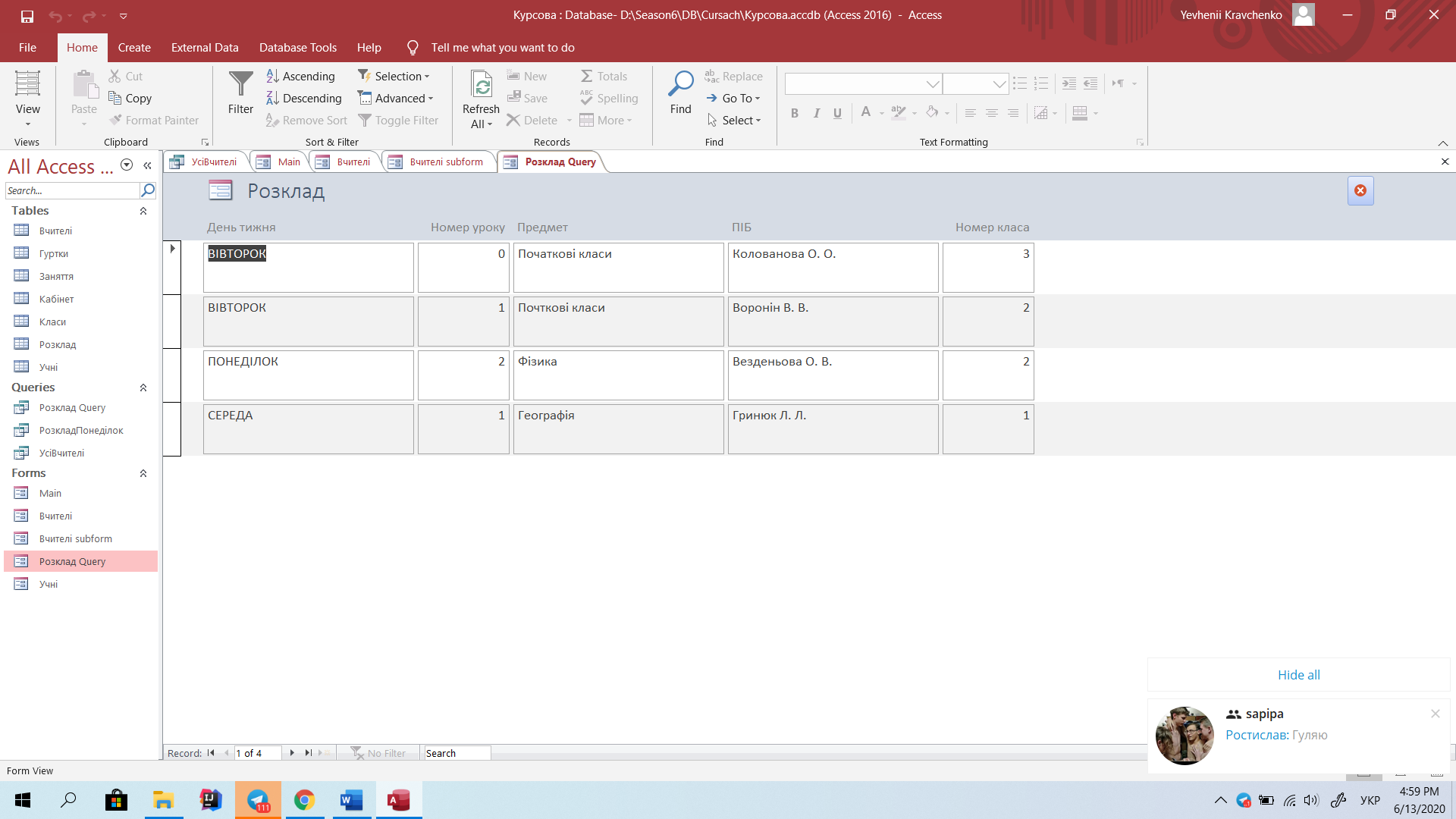


Рис. 5.3 – Форма для відображення розкладу занять

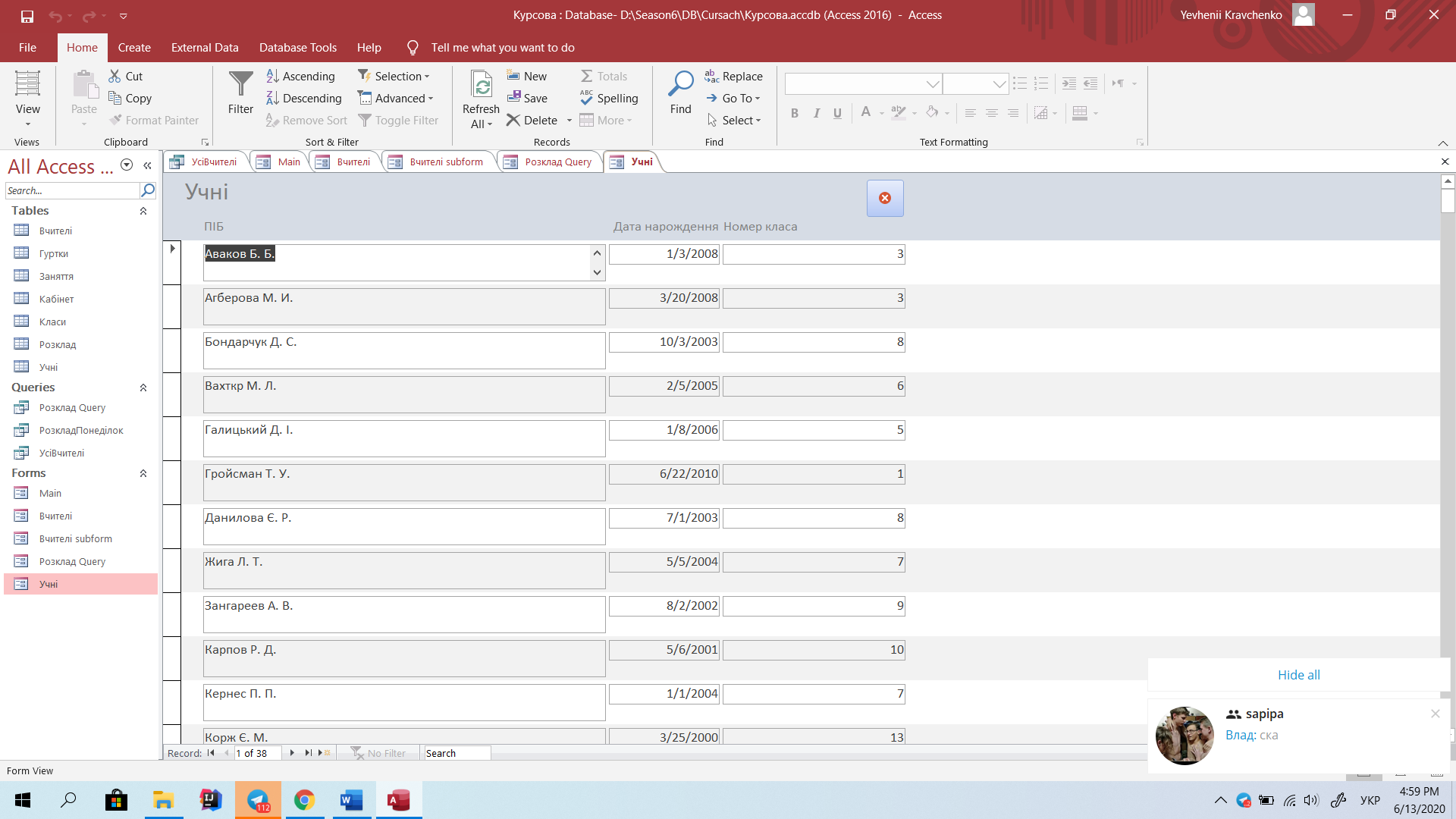


Рис. 5.4 – Форма для відображення інформації про учнів

## **5.2 Захист даних**

Як тільки дані структуровані і зведені в базу даних, виникає проблема організації доступу до них безлічі користувачів. Очевидно, що не можна дозволити всім без винятку користувачам безперешкодний доступ до всіх елементів бази даних. У будь-якій базі даних існує конфіденційна інформація, доступ до якої може бути дозволений лише обмеженому колу осіб. Це – один з аспектів проблеми безпеки в СУБД. Досить зазначити, що в найзагальнішому вигляді вимоги до безпеки реляційних СУБД формулюються так:

• По-перше, дані в будь-якій таблиці повинні бути доступні не всім користувачам, а лише деяким з них.

• По-друге, деяким користувачам має бути дозволено оновлювати дані в таблицях, в той час як для інших допускається лише вибір даних з цих же таблиць.

• По-третє, для деяких таблиць необхідно забезпечити вибірковий доступ до її стовпцями.

• По-четверте, деяким користувачам повинен бути заборонений безпосередній (через запити) доступ до таблиць, але дозволений доступ до цих же таблицях в діалозі з прикладної програмою.

Схема доступу до даних у всіх реляційних СУБД виглядає приблизно однаково і базується на трьох принципах:

• Користувачі СУБД розглядаються як основні діючі особи, які бажають отримати доступ до даних. СУБД від імені конкретного користувача виконує операції над базою даних, тобто додає рядки в таблиці (INSERT), видаляє рядки (DELETE), оновлює дані в рядках таблиці (UPDATE). Вона робить це в залежності від того, чи має конкретний користувач правами на виконання конкретних операцій над конкретним об'єктом бази даних.

• Об'єкти доступу - це елементи бази даних, доступом до яких можна управляти (дозволяти доступ або захищати від доступу). Зазвичай об'єктами доступу є таблиці, проте ними можуть бути і інші об'єкти бази даних - форми, звіти, прикладні програми і т.д. Конкретний користувач має конкретними правами доступу до конкретного об'єкта.

• Привілеї (priveleges) - це операції, які дозволено виконувати користувачеві над конкретними об'єктами. Наприклад, користувачеві може бути дозволено виконання над таблицею операцій SELECT (ВИБРАТИ) і INSERT (ВКЛЮЧИТИ). Таким чином, в СУБД авторизація доступу здійснюється за допомогою привілеїв. Встановлення та контроль привілеїв - прерогатива адміністратора бази даних. Привілеї встановлюються і скасовуються спеціальними операторами мови SQL - GRANT (ВИРІШИТИ) і REVOKE (СКАСУВАТИ). Оператор GRANT вказує конкретного користувача, який отримує конкретні привілеї доступу до зазначеної таблиці. Наприклад, оператор

GRANT SELECT, INSERT ОN Вчителі TO Сидоров

встановлює привілеї користувачеві Бітов на виконання операцій вибору і включення над таблицею Клієнти. Як видно з прикладу, оператор GRANT встановлює відповідність між операціями, користувачем і об'єктом бази даних (таблицею в даному випадку).

Привілеї легко встановити, але легко і скасувати. Скасування привілеїв виконується оператором REVOKE. Нехай, наприклад, користувач Сидоров втратив довіру адміністратора бази даних і останній вирішив позбавити його привілеїв на включення рядків в таблицю Клієнти. Він зробить це, виконавши оператор

REVOKE INSERT ON Вчителі FROM Сидоров

Конкретний користувач СУБД орієнтується по унікальному ідентифікатору (user-id). Будь-яка дія над базою даних, будь-який оператор мови SQL виконує не анонімно, але від імені конкретного користувача. Ідентифікатор користувача визначає набір доступних об'єктів бази даних для конкретної фізичної особи або групи осіб. Однак він нічого не повідомляє про механізм його зв'язку з конкретним оператором SQL. Наприклад, коли запускається інтерактивний SQL, як СУБД дізнається, від імені якого користувача здійснюється доступ до даних? Для цього в більшості СУБД використовується сеанс роботи з базою даних. Для запуску на комп'ютері-клієнті програми переднього плану (наприклад, інтерактивного SQL) користувач повинен повідомити СУБД свій ідентифікатор і пароль. Всі операції над базою даних, які будуть виконані після цього, СУБД зв'яже з конкретним користувачем, який запустив програму.

Для сучасних баз даних з великою кількістю користувачів актуальна проблема їх об'єднання в групи. Традиційно застосовуються два способу визначення груп користувачів. Відповідно до першого, один і той же ідентифікатор використовується для доступу до бази даних цілої групи фізичних осіб (наприклад, співробітників одного відділу). Це спрощує завдання адміністратора бази даних, так як досить один раз встановити привілеї для цього "узагальненого" користувача. Однак такий спосіб в основному передбачає дозвіл на перегляд, можливо, на включення, але ні в якому разі - на видалення та оновлення. Як тільки ідентифікатор (і пароль) стає відомий великому числу людей, виникає небезпека несанкціонованого доступу до даних сторонніх осіб.

Інший спосіб полягає в тому, що конкретній фізичній особі присвоюється унікальний ідентифікатор. У цьому випадку адміністратор бази даних повинен подбати про те, щоб кожен користувач отримав власні привілеї.

# **ВИСНОВКИ**

Метою створення будь-якої БД є спрощення використання великих масивів інформації. БД дозволяють збирати, зберігати, оновлювати і виводити інформацію в зрозумілій користувачеві формі.

В ході курсової роботи була розроблена і створена база даних "Школа", яка значно спрощує роботу з даними і багато в чому економить час співробітників. Створена БД може бути вдосконалена і доповнена новими даними, в неї можуть бути введені додаткові кошти формування даних. Наприклад, можуть значно допомогти кошти VBA і мови SQL.

Таким чином, можна говорити про можливість створення більш потужної системи, заснованої на використанні засобів самодіагностики, які будуть в автоматичному режимі прибирати надмірність даних і виробляти операції пов'язані з формуванням правильної структури даних.

# **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Карпов Б.В Microsoft Access 2000 Справочник.-1-е изд. –М.: Питер, 2000.-416 с.

2. Синева Н.Ф. Создание реляционных баз данных в MS Access. -1-е изд. –Саратов: Копипринтер СГТУ, 1996.-40 с.

3. Информатика: Учебник. – 3-е перераб. изд. / Под ред. проф. Н.В. Макаровой. – М.: Финансы и статистика,2001. – 768 с.: ил.

4. Диго С.М. Проектирование и использование баз данных. – М.: Финансы и статистика, 1995.

5. Каратыгин С., Тихонов А., Долголаптев В. Базы данных. В 2-х т. – М.: ЭКОМ, 1995.

6. Штайнер Г. Access 2000. – М.: Лаборатория базовых знаний, 2000. – 480 с.: ил. – (Справочник).