

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УНІВЕРСИТЕТ ЕКОНОМІКИ І ПРАВА «КРОК»

Індивідуальна робота №6

Індивідуальна робота по спеціальності 122 - комп'ютерні науки
(Освітня програма - Комп'ютерні науки)

Виконав: студенти 3 курсу
групи КН-19
Вергун Дмитрій
Грязнов Данил
Ластівка Тимофій
Симоновський Євгеній

м. Київ – 2021 рік

Зміст

Завдання	2
План робіт	3
Знайти інформацію	3
База даних	3
Unit Tests	3
GUI	3
Back-End	4
Інше	4
GUI	4
База даних	4
Висновок	8
Додатки	9

Завдання

Спроектувати реляційну базу даних, що містить інформацію із наведеної таблиці. Проектування здійснюється групами по 10 осіб за домовленістю. Кожен із студентів за допомогою Entity Framework створює по 3 або 4 таблиці.

Проект повинен містити класи, що відповідають таблицям бази та файли міграції. Після проектування створюється проект у репозиторій GitHub/GitLab, у якому кожен із студентів робить свій внесок. Викладачеві призначається право перегляду проекту та коментування.

Варіант: №1 "Аптеки"

План робіт

Знайти інформацію

- ☒ Знайти як правильно підключитися до БД за допомогою SQL коду Entity Framework;
- ☒ Знайти як створити сервер на SQLite;
- ☒ Знайти як створити сервер на PostgreSQL;
- ☒ Знайти як створити сервер на Heroku;
- ☒ Знайти як створити сервер на MS SQL Server;
- ☒ Знайти, як писати стилі через WPF. Саме як писати в XAML і CS (мається на увазі C#).

База даних

- ☒ По 3 таблиці для кожного (Діма, Тимофій, Даня, Женья);
- ☒ Зв'язати кожне складання таблиць (прим. Товари, Персонал, Клієнти, Постачальники);
- ☒ Наповнити БД;
- ☒ Розмістити у панелі софту.

Unit Tests

- ☐ Часткове покриття;
- ☐ Покрити запити у БД;
- ☐ Перевірити, чи вводить користувач;
- ☐ Перевірити коректність роботи БД;

GUI

- ☒ Розробити основну сторінку з функціями:
 - ☒ Кнопка "Подивитись";
 - ☒ Кнопка "Додати";
 - ☒ Кнопка "Редагувати";
 - ☒ Кнопка "Вилучити";
- ☒ Кожна кнопка п.1 відкриває нове вікно (можна одне) зі списком всіх таблиць;

- ☒ Візуалізація полів таблиці;
- ☒ Створити 12 кнопок на кожну таблицю;
- ☒ використовувати конструктори для відображення таблиць;
- ☐ Застосувати стилі (розмальовувати).

Back-End

- ☐ Надсилати інформацію в GUI;
- ☐ Редагування БД;
- ☐ Написання коду:
 - ☒ Використовувати код, який самодокументується;
 - ☒ Дотримуватися стилю;
 - ☐ Коментувати всі функції, класи, формули та івенти та запити;
 - ☐ Написати документацію;
- ☒ Підключити БД через Entity Framework.

Інше

- ☒ Створити проект на GitHub;
- ☒ Додати учасників;
- ☒ Розмістити проект;
- ☒ Написати записку до проекту.

GUI

Разработано GUI помощью WPF. Созданы интерфейсы с пониманием как вообще можно и нужно создавать грамотно составленный интерфейс с кнопками и вкладками.

Оптимизировали передачу информации из базы данных в GUI с помощью Entity Framework.

База данных

Для перевірки бази даних і візуального представлення спочатку розробили на базі продукту Access. Актуальна на даний момент схема бази даних, показана на малюнку 1.

Подальше створення бази даних ґрунтується на MS SQL Server. Як очевидно з малюнка 1 ми створили 12 таблиць.

1. Positions - Посада;
 - a. назва посади **VARCHAR(55)** не порожнє;
 - b. зарплату **INT** не порожнє;
2. Clients - Клієнти;
 - a. ім'я **VARCHAR(55)** не порожнє;
 - b. прізвище **VARCHAR(55)** не порожнє;
3. Suppliers - Постачальники;
 - a. назва компанії **VARCHAR(55)** не порожнє;
 - b. країна **VARCHAR(55)** не порожнє;
4. Pharmacological_Group - Фармацевтична група;
 - a. ім'я **VARCHAR(55)** не порожнє;
5. Instructions - Інструкції;
 - a. вік із якого можна використовувати продукт **INT** не порожнє;
 - b. кількість обов'язкових днів **INT** не порожнє;
 - c. детальний опис інструкції із застосування **VARCHAR(2048)** не порожнє;
6. Employees - Співробітники;
 - a. ім'я **VARCHAR(55)** не порожнє;
 - b. прізвище **VARCHAR(55)** не порожнє;
 - c. по батькові **VARCHAR(55)** не порожнє;
 - d. посада **INT** не порожнє;
7. Drugs - Товар;
 - a. виробник **INT** не порожнє;

- b. фармацевтична група **INT** не порожнє;
- c. інструкція для продукту **INT** не порожнє;
- d. назва продукту **VARCHAR(55)** не порожнє;
- e. наявність рецепту **BIT** не порожнє;
- f. ціна продукту **INT** не порожнє;

8. Pharmacy - Аптека;

- a. Директор аптеки **INT** не порожнє;
- b. Телефон **VARCHAR(55)** не порожнє;
- c. Назва **VARCHAR(55)** не порожнє;
- d. Адреса **VARCHAR(55)** не порожнє;
- e. Час роботи **VARCHAR(55)** не порожнє;

9. Phones - Телефон;

- a. Номер телефона **VARCHAR(55)** не порожнє;

10. Wishes - Відгуки клієнтів;

- a. назва аптеки **INT** не порожнє;
- b. клієнт (хто саме звертається) **INT** не порожнє;
- c. пропозиції, побажання та скарги **VARCHAR(2048)** не порожнє;

11. Receipts - чек;

- a. продала товар аптека **INT** не порожнє;
- b. клієнт купивший товар **INT** не порожнє;
- c. клієнт товар, що купив **INT** не порожнє;
- d. дата продажу **DATE** не порожнє;
- e. сума куплених товарів **INT** не порожнє;

12. Discounts - знижка;

- a. аптека в якій проводиться акція **INT** не порожнє;
- b. акційний продукт **INT** не порожнє;
- c. відсоток знижки **INT** не порожнє;
- d. дата початку акції **DATE** не порожнє;
- e. дата кінця акції **DATE** не порожнє.

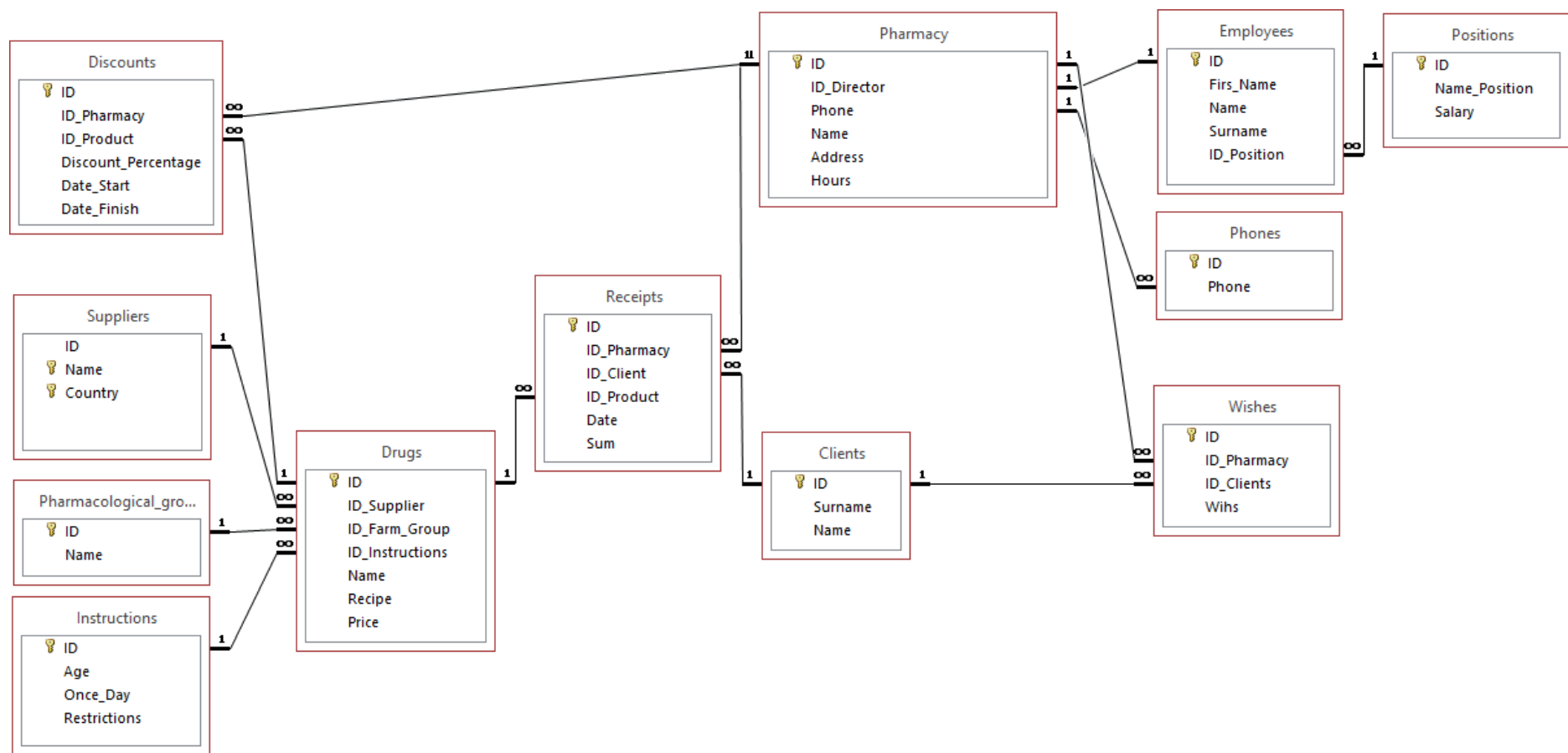


Рис.1 Схема бази даних

Висновок

В цьому проєкті, ми навчилися:

1. створювати віртуальні машини;
2. налаштовувати віртуальні машини;
3. працювати віддалено;
4. працювати з базами даних;
5. писати SQL команди;
6. розгортати базу даних на сервері heroku;
7. піднімати MS SQL Server;
8. створювати базу даних третього рівня нормалізації;
9. працювати з GUI за допомогою WPF;
10. створювати дивні інтерфейси;
11. розуміти, як взагалі можна створити грамотно складений інтерфейс із кнопками та вкладками;
12. передавати інформацію з бази даних GUI за допомогою Entity Framework;
13. використовувати ООП (невпевнено);
14. використовувати Visual Studio на початковому рівні;
15. отримувати задоволення від написання коду;
16. отримувати задоволення від реалізації складного завдання;
17. працювати в команді.

Додатки

Скріншоти на згадку

Свойства подключения

Введите данные для подключения к выбранному источнику данных или нажмите кнопку "Изменить", чтобы выбрать другой источник данных и (или) поставщик.

Источник данных:
Microsoft SQL Server (SqlClient) Изменить...

Имя сервера:
TESTPC\KN19 Обновить

Вход на сервер

Проверка подлинности: Проверка подлинности Windows

Имя пользователя:

Пароль:

☐ Сохранить пароль

Подключение к базе данных

☒ Выберите или введите имя базы данных:
PharmacyDB

☐ Прикрепить файл базы данных:
 Обзор...

Логическое имя:

Дополнительно...

Проверить подключение ОК Отмена

9 годин підключення до бази даних

```
ID INT NOT NULL PRIMARY KEY,  
ID_Pharmacy INT NOT NULL REFERENCES "Pharmacy" (ID) ON DELETE CASCADE ON UPDATE CASCADE,  
ID_Clients INT NOT NULL REFERENCES "Clients" (ID) ON DELETE CASCADE ON UPDATE CASCADE,
```

100 %

Сообщения

Выполнение команд успешно завершено.

Время выполнения: 2021-12-10T09:26:18.5909575+02:00

Велика перемога!

```
// Test Data.  
List<ClientsTalbeModel> tablesList = new();  
  
using (PharmacyDBEntities entities = new PharmacyDBEntities())  
{  
    List<Clients> ClientsList = entities.Clients.ToList();  
    foreach (Clients client in ClientsList)  
    {  
        tablesList.Add(new ClientsTalbeModel(client));  
    }  
    entities.Dispose();  
}  
if(tablesList.Count > 0)  
    TableGrid.ItemsSource = tablesList;  
else  
    TableGrid.Columns.Add(new DataGridColumn() { Header = });  
  
// Mode of interacting with data  
switch (Mode)  
{  
    case 0:  
        TableGrid.IsReadOnly = true;  
        // add.Visibility = Visibility.Collapsed;  
        break;  
    case 1:  
        TableGrid.IsReadOnly = false;  
}
```

Dispatcher
DisplayIndex
DragIndicatorStyle
Header
HeaderStringFormat
HeaderStyle
HeaderTemplate
HeaderTemplateSelector
IsReadOnly

object DataGridColumn.Header { get; set; }
Gets or sets the content of the column header.

Коли мозок вже не працює