

Ім'я користувача:
Порубльов Ілля Миколайович

ID перевірки:
1015696010

Дата перевірки:
27.06.2023 11:07:29 EEST

Тип перевірки:
Doc vs Library

Дата звіту:
27.06.2023 11:11:14 EEST

ID користувача:
95833

Назва документа: ОБДЗ_KC20_ЦаренкоРМ_безДодатків

Кількість сторінок: 35 Кількість слів: 4041 Кількість символів: 30079 Розмір файлу: 1.31 MB ID файлу: 1015339788

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

7.99%
Схожість

Найбільша схожість: 4.08% з джерелом з Бібліотеки (ID файлу: 1014690184)

Пошук збігів з Інтернетом не проводився

7.99% Джерела з Бібліотеки

201

Сторінка 37

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Підозріле форматування

16
сторінок

Міністерство освіти і науки України

Черкаський національний університет імені Богдана Хмельницького

Факультет обчислювальної техніки, інтелектуальних та управляючих систем

Кафедра програмного забезпечення автоматизованих систем

КУРСОВА РОБОТА

з дисципліни

ОРГАНІЗАЦІЯ БАЗ ДАНИХ І ЗНАНЬ

на тему: «Інформаційна система фермерського господарства»

Студента(ки) 3 курсу групи КС-20
спеціальності 21 – Інженерія

програмного забезпечення»

Царенко Р.М.

(прізвище та ініціали)

Керівник

СТ. ВИКЛ.

Гребенович Ю.Є.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Оцінка:

за університетською шкалою

за шкалою ECTS

за національною шкалою

Члени комісії

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

м. Черкаси – 2023 рік

Зміст

Вступ.....	3
1. Аналіз предметної області.....	4
1.1. Огляд та порівняння аналогів об'єкта розробки.	4
1.2. Постановка задач	6
2. Проектування бази даних	8
2.1. Інфологічне та даталогічне проектування.....	8
2.1.1. Інфологічне проектування.....	8
2.1.2. Даталогічне проектування.....	10
2.2. Проектування серверної частини	15
2.2.1. Таблиці та об'єкти бази даних	15
2.2.2. Виконання запитів.....	23
3. Опис клієнтського додатку.....	30
Висновок.....	34
Список інформаційних джерел	35
Додатки.....	36

Вступ

Інформаційні технології розвиваються дуже швидко. Вони допомагають влюбій сфері життя людини. Починаючи від простого використання як бази знань і примітивних задач запису, запам'ятовування, і прискореного вирішення примітивних обрахунків, до, в нинішній час, обробки великої кількості інформації, обрахунку всіх можливих фізичних явищ природи, космічних тіл, а наданий момент нашумілу технологію нейронних мереж, яка майже миттєво генерує текст, картинку, музику, може керувати автомобілем, заводом, будь чим, дивлячись чому буде вона навчена.

Так само і сфера сільського господарства в нинішній час впроваджує нові технології, щоб полегшити собі роботу, оптимізувати процеси, мати потрібну інформацію в одному місці, і до якої є швидкий доступ, і зручний інтерфейс.

Метою, а слідом і темою моєї курсової є: «Інформаційна система фермерського господарства», потенційно, це програма, загального обліку фермерського господарства.

А отже під час виконання курсової роботи потрібно буде проаналізувати предметну область, пояснити, чому ця тема має актуальність, проявити навички, які здобув під час вивчення цього курсу, щоб побудувати згодом правильну БД.

Застосувати навички попередніх курсів, щоб мав логічність інтерфейс, і реалізувати повноцінну програму, з підтримкою БД.

1. Аналіз предметної області

Отже першим пунктом є аналіз предметної області, в даному випадку в нас «Інформаційна система фермерського господарства». Для такої структури, потрібно враховувати велику кількість даних, оскільки фермерські господарства в нинішній час, це структури хоч і різного масштабу, але мають під собою різні підрозділи, структури, різний набір компаній партнерів. Це складна система з великою кількістю процесів.

У кожній ферми також і різні пріоритети. Якщо взяти в загальному, то фермерське господарство займається як насадженнями різної продукції, так і вирощуванням тварин, облік продукції яку має ферма для подальшого експорту, звітність цього експорту, що вона імпортувала до себе, дані з ким, співпрацюють.

Цю велику кількість даних, є потреба структурувати, щоб потім мати, майже завжди, актуальні дані, швидкий доступ до цих даних, і зручність керування цими даними.

1.1. Огляд та порівняння аналогів об'єкта розробки.

Дивлячись на ситуацію, зазвичай фермерські господарства для зберігання даних використовували, великі зошити з розміром листа А4, або спеціальні альбоми, вже з зарані розкресленими, або роздрукованими, таблицями, це те, з чого розпочинали, і навіть місцями використовують і зараз, для локальних, невеликих господарств. Через певний час, передові ферми, з появою перших доступних комп'ютерів і інтуїтивного приємного інтерфейсу, почали користуватися звичайними текстовими документами, а згодом і таблицями. Це звісно спростило задачу деякого обрахунку, збереження даних, але не сильно, оскільки більшість даних могли бути в різних документах, що забирало час на пошук потрібної інформації, або її зміну в різних документах, їх можна було легко втратити.

Зараз основними програмами обліку так і залишаються програми типу Word, для збереження загальної інформації. Також Excel – це вже більш професійна програма, яка в загальному задовольняє потреби невеликих ферм (Приклади на рис. 1.1.1 та 1.1.2). Більші ж структури замовляють собі кастомне

ПО, або більш «програмовану» Excel таблицю за вимогами ферми, або використовують 1С.

Дякуючи одному фермерському господарству, можна побачити, приблизно які дані, вони зберігають, для рослинництва, які фінансові дані вони зберігають. Справді існуючої програми, розглянути неможливо, оскільки кожна структура замовляє, таблицю, ПО, під свої потреби, а надавати **ДЕТАЛЬНИЙ** приклад якась структура навряд-чи буде, а якщо і буде, то нажалть виходів на неї немає, **ОСКІЛЬКИ** в більшості це може бути корпоративною таємницею.

А отже написання програми, і створення зручної БД, є актуальною.

The screenshot shows an Excel spreadsheet with the following data:

№	Назва (номер) ділянки	Культура/сорт	Рік посадки	Площа, га	Схема садіння	Кількість рослин на 1 га	Кількість рослин на майдані						
9	ТО-9	Черешня високоросла, сорт Bluecrop	2016	1,25	3,5 x 1	2837	3 571						
10	ТО-10	Лозина сорт Duke BB001	2021	2,7	3,5 x 1	2837	7 714						
11	ТО-11	Лозина сорт Duke BB001	2021	3,35	3,5 x 1	2837	9 371						
12	ТО-12	Лозина сорт Duke BB001	2021	3,91	3,5 x 1	2837	11 171						
13	ТО-13	Лозина сорт Duke BB001	2021	3,62	3,5 x 1	2837	10 342						
14	ТО-14	Лозина сорт Duke BB001	2021	3,74	3,5 x 1	2837	10 685						
15	ТО-15	Експериментальна ділянка (абрикос, груша, черешня, горіхи)	2020	6,7	3,5 x 2								

Рис 1.1.1 – приклад Excel таблиці фермерського господарства

№№	Культура/Вид	Площа, га	Валовий збір зерна, тн	Вартість за 1 тн	Сума, грн	Загальна урожайність, тн з 1 га	Осередкована врожайність, тн з 1 га
1	Пшениця	53	73,600		484 313,00	1,37	2,5-4,5
2	Соняшник	17	49,120		651 402,85	2,89	2-3
ВСЬОГО		70	122,720		1 055 715,85		
№№	Сумовні суми витрат	Пшениця	Соняшник	ВСЬОГО витрат П.Д.В.	% в структурі витрат		
1	Посівний матеріал під урожай 2022 року	26 303,02	31 563,62	42 992,11	53 154,16	6,13	
2	Оренда ґрунтового трактора 2022 року	30 000,00	38 000,00	17 000,00	20 400,00	4,18	
3	Добрива під урожай 2022 року	116 556,00	139 903,20	39 678,00	47 613,80	13,82	
4	Матеріальні витрати	419 217,81	428 861,46	80 124,26	96 159,11	45,84	
5	Витрати на внесення матеріальних добрив	25 970,00	31 184,00	4 080,00	4 899,00	2,69	
6	Порівняння зазначених витрат з витратами на внесення добрив	14 875,00	17 490,00	2 975,00	3 573,00	1,55	
7	Витрати на		0,00	10 230,00	12 240,00	0,99	
8	Витрати на	14 875,00	17 490,00		17 490,00	1,29	
9	Витрати на внесення добрив (по матеріалу)	34 450,00	41 340,00	11 050,00	13 260,00	4,03	
10	Витрати на експлуатацію та зношення засобів в експлуатації	18 550,00	22 380,00	5 950,00	7 340,00	2,17	
11	Паливо	34 450,00	41 340,00	11 050,00	13 260,00	4,03	
12	Обслуговування	190 700,00	120 840,00	32 330,00	38 760,00	13,78	
13	Витрати на порівняння продукції	19 000,00	22 800,00		22 800,00	1,08	
Всього:		873 376,83	1 048 052,28	247 430,37	306 883,44	1 064 932,72	100,00

Рис 1.1.2 – приклад Excel таблиці фермерського господарства

1.2. Постановка задач

І все ж, для початку, постає питання, що потрібно враховувати на фермерському господарстві? Фермерів які б проінструктували, на жаль немає, тому виокремлено загально потрібні розділи даних, яким потрібен облік, до яких потрібен швидкий доступ та структурність, ці дані будуть розбиті на декілька таблиць, а також для них буде розроблено графічний інтерфейс, відповідно до пунктів.

Розглянемо пункти:

1. Провівши аналіз предметної області можна виділити основні категорії, яким потрібен облік та БД: облік адрес компаній партнерів, та облік адрес підрозділів ферми, облік продуктів експорту та імпорту, робітників, процесу посіву вирощування і тд.

Виходячи з вище написаного, під наші цілі, підходить SQLite – оскільки це реляційна база даних, яку і потрібно використати, а також зберігає БД (дані)

локально, в окремому файлі, що не потребує додаткових ресурсів людей які будуть використовувати програму, а також це спростить трішки розробку, щоб уникнути непотрібної тяганини з окремим сервером.

2. Розробити програмний додаток та його інтерфейс. Для його розробки буде використано мову програмування C#, та фреймворк WPF, обрані дані технології були по причині, що вони вже використовувалися і знайомі, що дозволить швидше виконати поставленні задачі, а також ці технології дадуть нам пов'язати БД з програмним додатком (інтерфейсом), і головний плюс, що це універсальні технології для написання додатків для Windows.

Далі перейдемо до більш детальнішого розгляду зовнішнього вигляду програмного додатку, **потрібно** буде реалізувати 3 вкладки:

2.1. Перша та друга вкладки **ПОВИННІ** мати бокову панель, яка дозволить нам переключатися між таблицями, збоку від цієї панелі, блок, де будуть відображатися самі таблиці. Також за задумом, біля записів повинна розміщатися кнопка видалення запису. А також кнопка додавання запису, скоріш всього над таблицею. Чому дві вкладки, пояснення доволі просте, потрібно буде розділити таблиці, які стосуються тільки ферми, і таблиці які відносяться до фінансової частини ферми та співробітництва з компаніями партнерами.

2.2. Третя вкладка, це просто функціональна вкладка з рядком для користувацького запиту, якщо користувачу буде потрібно щось нестандартне.

На цьому етапі загальна постановка задач завершена, переходимо до більш поглибленого проектування БД і розширення задач, які буде потрібно виконати в наступному пункті.

2. Проектування бази даних

2.1. Інфологічне та даталогічне проектування

2.1.1. Інфологічне проектування

Логічним завершенням аналізу предметної області, є те, що тепер ми можемо детально розглянути максимально потрібні сутності для майбутньої БД, для початку потрібно спроектувати ER-діаграму яка зобразить взаємодію сутностей, полів і допоможе нам створити БД.

Головні, а отже наймісткіші сутності можна виділити такі:

1. Тварини (Animals);

Ця сутність повинна вміщати тип тварини (наприклад корова, куриця, індик, і тп.), стать, кількість, де розміщені, споживання. З цього переліку, випливає, що нам потрібна ще одна сутність *Animal Type*, щоб потім, у випадку, коли ферма не буде працювати з певними типами тварин, просто видалити цей тип, і він автоматично видалився з основної сутності *Animals*.

2. Насадження (Plantation);

Ця сутність повинна вміщати, назву насадження, де посажене, кількість насаджень, у випадку якщо це кущ або саджанець, а якщо порахувати не можливо, то вказувати площу насаджень, і дату посадки (посіву і тп.)

Тепер, що стосується розміщення, було прийнято рішення, зробити 4 сутності. Загальну, під назвою *Location Type*, для загальних назв територій: магазин, оптова база, ділянка землі, склад. Оскільки всі вище перераховані назви можуть бути використані, як і в частинах землі які відносяться тільки до ферми, і можуть бути використані як для експорту так і для імпорту. Тому, щоб було зручно, і не було зайвої інформації в інших сутностях (таблицях), створено 3 сутності *Land Addresses*, *Export Addresses*, *Import Addresses*, зробивши це, ми уникаємо ситуації, коли всі адреси, були б записані в одній таблиці, щоб створило плутанину. У всіх трьох сутностях, зберігатиметься адреса, id загальної назви локації з таблиці *Location Type*. А тепер відмінності: в *Land Addresses* буде зберігатися назва підрозділу ферми, в *Export Addresses*,

Import Addresses будуть зберігатися назви компаній, магазинів і тп., з якими співпрацює ферма.

3. Експорт (Export);

Ця сутність, повинна зберігати велику кількість інформації, я то лот який продано, тип експорту (в нашому випадку продукту нехай то, м'ясо, фрукти, овочі, ягоди і тп.), куди експортувалося, назва що продали (свинина, огірок, помідор, яловичина), кількість в кілограмах, ціна за весь лот, і дата продажу. Потенційно ці всі поля потрібні, щоб в майбутньому була можливість проаналізувати продажі за рік до прикладу, або підбити різні підсумки які захоче ферма самостійно. Також варто зазначити, що така сутність не повинна бути прив'язана до інших сутностей, оскільки вона є загальною історією (обліком), всього експорту ферми.

З ОПИСАНОГО, випливає, що потрібно ще 2 сутності — це *Export Type* та *Export Warehouse*. Для першої сутності, причина **створення** така сама як і для сутності *Animal Type*, щоб у випадку, коли ферма перестане виробляти той чи інший тип продукції, її просто можна було б видалити, і видалення в одній таблиці видалить пов'язані записи з інших таблиць. Що до другої сутності, це загальний склад, який буде зберігати, всю можливу для продажу продукцію. Ця сутність буде зберігати майже всю ту саму інформацію, але з деякими поправками, ціна буде вказана за кілограм, щоб замовник, міг оцінювати, що йому вигідніше, кількість буде змінною, оскільки продукт може псуватися.

4. Імпорт (Import);

Сутність імпорту, як і експорту має той самий сенс, зберігати дані, для подальшого звіту, аналізу, або що тому ця сутність теж відрізнена від зав'язків. Поля ті самі, тип імпорту (деталі, інструменти, насіння, саджанці, яди, добрива і тп.), назва імпорту (лопата «Копатор 3000», мінеральне добриво N, і тп.), хто імпортував, скільки заплатили, і коли імпортовано. Також додатково створено *Import Type*, з тих самих причин що і *Export Type*, і для додаткової зручності.

5. Робітники (Employees);

І остання сутність, як же не вести облік тих, хто працює у тебе на фермі. Вона вміщує такі дані: повне ім'я, адреса проживання робітника, номер телефону, на якому підрозділі (ділянці) ферми працює, і також тип його роботи.

Як наслідок, має ще одну сутність під назвою Type Of Work, ця сутність матиме назву (тип, професія) роботи, а також буде вказуватися заробітна плата.

Виходячи з аналізу вище, маємо таку ER-діаграму або ж інфологічну модель БД, яка розміщена в [додатку А](#).

2.1.2. Даталогічне проектування

Основну частину я описав при інфологічному проектуванні, тому під час даталогічної, в основному буде сухий опис, з усіма полями, типами даних для цих полів, і назвами самих таблиць **ВІДПОВІДНО**.

1. Таблиця Location_Type:

- Поле id – тип даних INTEGER, має додаткову умову автоінкременту та NOT NULL, тобто поле не може бути пустим, первинний ключ (PK);
- Поле name – тип даних TEXT, має додаткову умову NOT NULL;

2. Таблиця Land_Addresses:

- Поле id – тип даних INTEGER, має додаткову умову автоінкременту та NOT NULL, тобто поле не може бути пустим, первинний ключ (PK);
- Поле location_type_id – тип даних INTEGER, має додаткову умову NOT NULL, зовнішній ключ (FK) пов'язаний з таблицею Location_Type;
- Поле branch_name – тип даних TEXT, має додаткову умову NOT NULL;
- Поле address – тип даних TEXT, має додаткову умову NOT NULL;

3. Таблиця Type_Of_Work:

- Поле id – тип даних INTEGER, має додаткову умову автоінкременту та NOT NULL, тобто поле не може бути пустим, первинний ключ (PK);

- Поле name – тип даних TEXT, має додаткову умову NOT NULL;
- Поле salary – тип даних INTEGER, має додаткову умову NOT NULL;

4. Таблиця Employees:

- Поле id – тип даних INTEGER, має додаткову умову автоінкременту та NOT NULL, тобто поле не може бути пустим, первинний ключ (PK);
- Поле type_of_work_id – тип даних INTEGER, має додаткову умову NOT NULL, зовнішній ключ (FK) пов'язаний з таблицею Type_Of_Work;
- Поле land_addresses_id – тип даних INTEGER, має додаткову умову NOT NULL, зовнішній ключ (FK) пов'язаний з таблицею Land_Addresses;
- Поле full_name – тип даних TEXT, має додаткову умову NOT NULL;
- Поле address_of_residence – тип даних TEXT, має додаткову умову NOT NULL;
- Поле phone_number – тип даних TEXT, має додаткову умову NOT NULL;

5. Таблиця Animal_Type:

- Поле id – тип даних INTEGER, має додаткову умову автоінкременту та NOT NULL, тобто поле не може бути пустим, первинний ключ (PK);
- Поле name – тип даних TEXT, має додаткову умову NOT NULL;

6. Таблиця Animals:

- Поле id – тип даних INTEGER, має додаткову умову автоінкременту та NOT NULL, тобто поле не може бути пустим, первинний ключ (PK);
- Поле land_addresses_id – тип даних INTEGER, має додаткову умову NOT NULL, зовнішній ключ (FK) пов'язаний з таблицею Land_Addresses;
- Поле animal_type_id – тип даних INTEGER, має додаткову умову NOT NULL, зовнішній ключ (FK) пов'язаний з таблицею Animal_Type;

- Поле gender – тип даних TEXT, має додаткову умову NOT NULL;
- Поле quantity – тип даних INTEGER, має додаткову умову NOT NULL;
- Поле consumption – тип даних INTEGER, має додаткову умову NOT NULL;

7. Таблиця Plantation:

- Поле id – тип даних INTEGER, має додаткову умову автоінкременту та NOT NULL, тобто поле не може бути пустим, первинний ключ (PK);
- Поле land_addresses_id – тип даних INTEGER, має додаткову умову NOT NULL, зовнішній ключ (FK) пов'язаний з таблицею Land_Addresses;
- Поле plantation_type_name – тип даних TEXT, має додаткову умову NOT NULL;
- Поле quantity – тип даних INTEGER, має додаткову умову NOT NULL;
- Поле planting_date – тип даних TEXT, має додаткову умову NOT NULL;

8. Таблиця Export_Addresses:

- Поле id – тип даних INTEGER, має додаткову умову автоінкременту та NOT NULL, тобто поле не може бути пустим, первинний ключ (PK);
- Поле location_type_id – тип даних INTEGER, має додаткову умову NOT NULL, зовнішній ключ (FK) пов'язаний з таблицею Location_Type;
- Поле name_companies – тип даних TEXT, має додаткову умову NOT NULL;
- Поле address – тип даних TEXT, має додаткову умову NOT NULL;

9. Таблиця Export_Type:

- Поле id – тип даних INTEGER, має додаткову умову автоінкременту та NOT NULL, тобто поле не може бути пустим, первинний ключ (PK);
- Поле name – тип даних TEXT, має додаткову умову NOT NULL;

10. Таблиця Export_Warehouse

- Поле id – тип даних INTEGER, має додаткову умову автоінкременту та NOT NULL, тобто поле не може бути пустим, первинний ключ (PK);
- Поле export_type_id – тип даних INTEGER, має додаткову умову NOT NULL зовнішній ключ (FK) пов'язаний з таблицею Export_Type;
- Поле name – тип даних TEXT, має додаткову умову NOT NULL;
- Поле quantity – тип даних INTEGER, має додаткову умову NOT NULL;
- Поле price_per_kilo – тип даних INTEGER, має додаткову умову NOT NULL;

11. Таблиця Export:

- Поле id – тип даних INTEGER, має додаткову умову автоінкременту та NOT NULL, тобто поле не може бути пустим, первинний ключ (PK);
- Поле export_addresses_id – тип даних INTEGER, має додаткову умову NOT NULL;
- Поле export_warehouse_id – тип даних INTEGER, має додаткову умову NOT NULL;
- Поле export_type_id – тип даних INTEGER, має додаткову умову NOT NULL;
- Поле name – тип даних TEXT, має додаткову умову NOT NULL;
- Поле quantity – тип даних INTEGER, має додаткову умову NOT NULL;
- Поле price – тип даних INTEGER, має додаткову умову NOT NULL;
- Поле sale_date – тип даних TEXT, має додаткову умову NOT NULL;

12. Таблиця Import_Addresses:

- Поле id – тип даних INTEGER, має додаткову умову автоінкременту та NOT NULL, тобто поле не може бути пустим, первинний ключ (PK);
- Поле location_type_id – тип даних INTEGER, має додаткову умову NOT NULL, зовнішній ключ (FK) пов'язаний з таблицею Location_Type;
- Поле company_name (різниця в назвах між name_companies ті цією, можливо через те, що створення самої БД було в різні дні) – тип даних TEXT, має додаткову умову NOT NULL;
- Поле address – тип даних TEXT, має додаткову умову NOT NULL;

13. Таблиця Import_Type:

- Поле id – тип даних INTEGER, має додаткову умову автоінкременту та NOT NULL, тобто поле не може бути пустим, первинний ключ (PK);
- Поле name – тип даних TEXT, має додаткову умову NOT NULL;

14. Таблиця Import:

- Поле id – тип даних INTEGER, має додаткову умову автоінкременту та NOT NULL, тобто поле не може бути пустим, первинний ключ (PK);
- Поле import_addresses_id – тип даних INTEGER, має додаткову умову NOT NULL;
- Поле import_type_id – тип даних INTEGER, має додаткову умову NOT NULL;
- Поле name – тип даних TEXT, має додаткову умову NOT NULL;
- Поле quantity – тип даних INTEGER, має додаткову умову NOT NULL;
- Поле price – тип даних INTEGER, має додаткову умову NOT NULL;
- Поле import_date – тип даних TEXT, має додаткову умову NOT NULL;

Розписавши всі пункти, які потрібні, тепер ми можемо поглянути на даталогічну модель виконання (Додаток Б), і перейти до проектування БД.

2.2. Проектування серверної частини

В даному розділі, буде створимо БД по діаграмам вище, а також заповнено інформацією ці таблиці, щоб виконати запити, які вказані як одне із завдань до курсової роботи.

2.2.1. Таблиці та об'єкти бази даних

Почну показувати таблиці в порядку, який був розписаний вище.

id	Location type name
1	Магазин
2	Оптова база
4	Сільськогосподарська ділянка
6	Ринок
7	Торговий центр
8	Супермаркет
10	Гараж
11	Ферма
12	Склад
13	Дилер
14	Завод
15	Офіс
16	Адміністративна будівля
17	Лабораторія

Рис. 2.2.1.1. Таблиця Location_Type

id	Location type id	Branch name	Address
2	4	Посівна площа №1	м. Черкаси, вул. Зарічна 91
3	10	Гараж ремонтний	м. Черкаси, вул. Коцюбинського 129
4	12	Склад запчастин №1	м. Черкаси, вул. Шевченка 57
5	12	Склад продуктовий №1	м. Черкаси, вул. Кобиланської 186
6	4	Посівна площа №2	м. Черкаси, вул. Лесі Українки 123
7	4	Оранжерея №1	м. Черкаси, вул. Хрещатик 92
8	11	Курник №1	м. Черкаси, вул. Київська 179
9	11	Корівник №1	м. Черкаси, вул. Героїв Дніпра 176
10	10	Тракторний гараж	м. Черкаси, вул. Бандери 46
11	10	Гараж ремонтний №2	м. Черкаси, вул. Котляревського 142
12	12	Склад зерновий №1	м. Черкаси, вул. Карпенка-Карого 12
13	4	Город №1	м. Черкаси, вул. Гайдара 29
14	12	Склад запчастин №2	м. Черкаси, вул. Шевченка 98
15	11	Курник №2	м. Черкаси, вул. Соборна 174
17	10	Гараж з навісним...	м. Черкаси, вул. Жовтнева 173
18	11	Ферма страусів	м. Черкаси, вул. Господарська 33
19	11	Ферма молочна	м. Черкаси, вул. Молодіжна 50

Рис. 2.2.1.2. Таблиця Land_Addresses

Id	Type of work name	Salary(UAH)
1	Юрист	10000
2	Бухгалтер	12000
3	Сільськогосподарський інженер	16000
4	Економіст	20500
5	Логіст	14500
6	Менеджер з якості	14250
7	Робітник на фермі	10000
8	Механік	13000
9	Лаборант	13500
10	Тваринник	15500
11	Садівник	13250
12	Ветеринар	15250
13	Машиніст	16500
14	Агроном	16250
15	Агротехнолог	17750

Рис. 2.2.1.3. Таблиця Type_Of_Work

Id	Type of work id	Land address id	Full name	Address of residence	Phone number
1	1	21	Олексій Іваненко	м. Черкаси, вул. Зарічна 80	0671234567
2	4	21	Олександр Ковальчук	м. Черкаси, вул. Котляревського 141	0999876543
3	8	11	Володимир Григоренко	м. Черкаси, вул. Котляревського 140	0502468135
4	8	11	Михайло Коваленко	м. Черкаси, вул. Котляревського 139	0632349876
5	13	10	Сергій Литвиненко	м. Черкаси, вул. Соборна 45	0662345678
12	2	23	Іван Біленький	м. Черкаси, вул. Зарічна 91	0681111111
13	3	22	Марія Коваленко	м. Черкаси, вул. Коцюбинського 129	0952222222
14	4	23	Ольга Петренко	м. Черкаси, вул. Шевченка 57	0673333333
15	5	12	Андрій Лисенко	м. Черкаси, вул. Кобилянської 186	0994444444

Рис. 2.2.1.4. Таблиця Employees

Id	Animal type name
1	Корова
2	Страус
3	Куриця
4	Перепілка
5	Свиня
6	Кролики
7	Кози
8	Вівці
9	Індюки

Рис. 2.2.1.5. Таблиця Animal_Type

Id	Land Address Id	Animal Type Id	Gender	Quantity	Consumption(kg)
2	25	4	Самка	50	25
3	8	3	Самець	10	25
4	25	4	Самець	13	9
5	9	1	Самець	20	250
6	9	1	Самка	40	400
7	18	2	Самка	18	180
8	18	2	Самець	9	90
9	15	3	Самка	55	55
10	15	3	Самець	20	25
11	19	1	Самка	60	500
12	24	5	Самець	66	660

Рис. 2.2.1.6. Таблица Animals

Id	Plantation name	Land Address Id	Quantity(or m2)	Planting Date
3	Соняшник	2	2000	2023-03-10
4	Пшениця	6	5000	2023-03-15
5	Гремка	2	2350	2023-03-25
6	Яблуня	27	15	2023-04-25
7	Помідори	30	45	2023-05-15
8	Жито	6	5500	2023-03-18
11	Редька	7	56	2023-05-17
12	Кукурудза	6	550	2023-03-30

Рис. 2.2.1.7. Таблица Plantation

Id	Location type Id	Company name	Address
3	1	Делікат №2	м. Черкаси, вул. Київська 17
4	1	Делікат №1	м. Черкаси, вул. Житнева 5
5	6	Великий Смак	м. Черкаси, вул. Зелена 10
6	6	Мета Оптовка	м. Черкаси, вул. Садова 15
7	18	Ароматна Долина Біла	м. Черкаси, вул. Володимирська 30
8	18	Гастро Палац	м. Черкаси, вул. Шенченка 65
9	18	Смакова Спекуса	м. Черкаси, вул. Лесі Українки 8
10	8	АТБ №1	м. Черкаси, вул. Молодіжна 40
11	8	АТБ №2	м. Черкаси, вул. Соборна 25
12	8	Сільпо №1	м. Черкаси, вул. Героїв Дніпра 14
13	8	Сільпо №2	м. Черкаси, вул. Мирна 11
14	18	Міський Рай	м. Черкаси, вул. Мирна 11
15	2	Оптова Крайниці Смаку	м. Черкаси, вул. Незалежності 28

Рис. 2.2.1.8. Таблиця Export_Addresses

Id	Export Type Name
1	Фрукти
2	Овочі
3	Яйця
4	Молоко
5	Зернові
6	Ягоди
7	Зелень
8	Гриби
9	Боби

Рис. 2.2.1.9. Таблиця Export_Type

↑ Id	Export type Id	Product name	Quantity	Price per kilo
8	8	Шампінйони	20	100
9	1	Груші	10	150
10	2	Перець болгарський	5	140
11	6	Вишні	10	130
12	6	Полуниця	15	125
13	7	Петрушка	1	80
14	7	Кріп	3	55
15	1	Абрикос	10	170
16	2	Помідори	15	100
17	2	Огірки	20	105
18	9	Квасоля	30	75
19	10	Свинина	45	200

Рис. 2.2.1.10. Таблиця Export_Warehouse

Id	Export address Id	Export warehouse Id	Export type Id	Name	Quantity	Price	Sale date
4	1	4	1	Вишні	20	2000	2023-06-23
5	1	5	1	Яблука	100	2000	2023-06-24
6	4	8	8	Шампінйони	20	2000	2023-06-26
7	14	20	10	Яловичина	25	6250	2023-06-09
8	14	19	10	Свинина	45	9000	2023-06-09
9	10	17	2	Огірки	20	2100	2023-06-11
10	11	16	2	Помідори	15	1500	2023-06-12
11	13	12	6	Полуниця	15	1875	2023-06-12

Рис. 2.2.1.11. Таблиця Export

ID	Location type ID	Company name	Address
2	1	Фермерський Джерело	м. Київ, вул. Коперника 183
3	13	ЗАЗ	м. Запоріжжя, вул. Сиваганського 149
4	13	Агро Інжиніринг	м. Київ, вул. Дорогомицька 83
5	14	Фермерські Технології	м. Київ, вул. Костянтинівська 70
6	14	Фермерські Машини	м. Київ, вул. Драгоманова 92
7	1	АгроПродукт	м. Київ, вул. Житомирська 175
8	17	Агрохімія	м. Київ, вул. Борщагівська 81

Рис. 2.2.1.12. Таблиця Import_Addresses

ID	Import Type Name
1	Інструменти
2	Деталі
3	Зерно
4	Яд
5	Саджанці
6	Добриво
7	Кондиционери
8	Навіжки
9	Медикаменти
10	Техніка

Рис. 2.2.1.13. Таблиця Import_Type

ID	Import address ID	Import type ID	Import name	Quantity	PriceID...	Import Date
1	1	2	Колеса	4	5000	2023-06-24
2	3	10	Трактор	3	400000	2023-06-09
3	8	4	Яд проти колорадських жуків	50	15000	2023-06-08
4	6	10	Машини для доїння	5	700000	2023-06-25
5	8	6	Міндобрива	500	40000	2023-06-05
6	9	9	Таблетки для корів	15	15000	2023-06-01
7	10	8	Помідори	150	3000	2023-06-02
8	10	5	Вичіні	35	9000	2023-04-02

Рис. 2.2.1.14. Таблиця Import

2.2.2. Виконання запитів

Перейдемо до виконання запитів. Вигляд буде як і в попередньому розділі, рисунок і результат виконання на скріншоті.

Запит який виведе всі дані таблиці Employees;

SELECT * FROM Employees

id	current_work_id	last_address_id	last_name	address_id	phone_number
1	1	21	Олександр Іваненко	м. Черкаси, вул. Зарічна 80	0671234567
2	4	21	Олександр Коваленко	м. Черкаси, вул. Катерининська 141	099876543
3	8	11	Володимир Григоренко	м. Черкаси, вул. Катерининська 148	050456789
4	8	11	Миколай Коваленко	м. Черкаси, вул. Катерининська 139	0432345678
5	12	10	Сергій Литвиненко	м. Черкаси, вул. Сабова 43	0662345678
12	2	23	Іван Білоусов	м. Черкаси, вул. Зарічна 91	0481111111
13	3	22	Марія Коваленко	м. Черкаси, вул. Катерининська 129	0952222222
14	4	23	Олена Петренко	м. Черкаси, вул. Шевченка 57	0673333333
15	5	12	Андрій Лисенко	м. Черкаси, вул. Катерининська 186	0994444444
16	6	23	Наталія Савченко	м. Черкаси, вул. Лесі Українки 123	0105555555
17	7	13	Олександр Мельник	м. Черкаси, вул. Зарічна 92	0636666666
18	8	17	Ірина Попренко	м. Черкаси, вул. Катерининська 179	0687777777
19	9	20	Сергій Волошенко	м. Черкаси, вул. Героїв Дніпра 176	0478888888
20	10	8	Марина Соколенко	м. Черкаси, вул. Бандери 45	0609999999
21	11	13	Олег Шевченко	м. Черкаси, вул. Катерининська 142	0500000000
22	12	22	Наталія Іваненко	м. Черкаси, вул. Катерининська 12	0201111111
23	13	10	Марія Гайдар	м. Черкаси, вул. Гайдара 28	0422222222
24	14	13	Володимир Шевченко	м. Черкаси, вул. Шевченка 98	0463333333
25	15	20	Олена Голоса	м. Черкаси, вул. Сабова 174	0674444444
26	16	22	Ігор Петров	м. Черкаси, вул. Сабова 172	0985555555
27	11	7	Марина Мельниченко	м. Черкаси, вул. Катерининська 74	0982345678

Рис. 2.2.2.1. Простий запит на вибірку

Цей запит вибирає всі професії, де заробітна плата знаходиться в діапазоні від 12000 до 15000.

SELECT * FROM Type_Of_Work WHERE salary BETWEEN 12000 AND 15000;

id	name	salary
2	Бухгалтер	12000
5	Ліній	14500
6	Менеджер з продажів	14250
8	Механік	13000
9	Лаборант	13500
11	Садівник	13250

Рис. 2.2.2.2. Запит на вибірку з використанням «between....and»

Запит який виводить робітників, у яких type_of_work_id 1,8,5.

SELECT * FROM Employees WHERE type_of_work_id IN (1, 8, 5);

id	type_of_work_id	landaddresses_id	fullname	address_of_residence	phonenumber
1	1	21	Олексій Іваненко	м. Черкаси, вул. Зарічна 80	0671234567
3	8	11	Володимир Григоренко	м. Черкаси, вул. Котляревського 140	0502468135
4	8	11	Михайло Коваленко	м. Черкаси, вул. Котляревського 139	0632349876
15	5	12	Андрій Лисенко	м. Черкаси, вул. Кобиланської 186	0994444444
18	8	17	Ірина Петренко	м. Черкаси, вул. Київська 179	0667777777

Рис. 2.2.2.3. Запит на вибірку з використанням «in»

Запит який виводить робітників, які проживають на вулиці «Котляревського»

SELECT * FROM Employees WHERE address_of_residence LIKE '%Котляревського%';

id	type_of_work_id	landaddresses_id	fullname	address_of_residence	phonenumber
2	4	21	Олександр Ковальчук	м. Черкаси, вул. Котляревського 141	0998876543
3	8	11	Володимир Григоренко	м. Черкаси, вул. Котляревського 140	0502468135
4	8	11	Михайло Коваленко	м. Черкаси, вул. Котляревського 139	0632349876
21	11	13	Олег Шевченко	м. Черкаси, вул. Котляревського 142	0990000000
27	11	7	Марина Мельничук	м. Черкаси, вул. Котляревського 76	0992345678

Рис. 2.2.2.4. Запит на вибірку з використанням «like»

SELECT * FROM Employees WHERE type_of_work_id = 8 AND land_addresses_id = 11;

id	type_of_work_id	landaddresses_id	fullname	address_of_residence	phonenumber
3	8	11	Володимир Григоренко	м. Черкаси, вул. Котляревського 140	0502468135
4	8	11	Михайло Коваленко	м. Черкаси, вул. Котляревського 139	0632349876

Рис. 2.2.2.5. Запит на вибірку з двома умовами через «and»

SELECT * FROM Employees WHERE type_of_work_id = 8 OR land_addresses_id = 11;

id	type_of_work_id	landaddresses_id	fullname	address_of_residence	phonenumber
3	8	11	Володимир Григоренко	м. Черкаси, вул. Котляревського 140	0502468135
4	8	11	Михайло Коваленко	м. Черкаси, вул. Котляревського 139	0632349876
18	8	17	Ірина Петренко	м. Черкаси, вул. Київська 179	0667777777

Рис. 2.2.2.6. Запит на вибірку з двома умовами через «or»

Цей запит виводить кількість унікальних значень `land_addresses_id`, в таблиці `employees`;

```
SELECT Count(*) AS DistinctLandAddressesId
FROM (SELECT DISTINCT land_addresses_id FROM Employees);
```

DistinctLandAddressesId
11

Рис. 2.2.2.7. Запит на вибірку з використанням «DISTINCT»

У цьому запиті ми об'єднали таблиці `Land_Addresses` і `Employees` за допомогою умови `Land_Addresses.id = Employees.land_addresses_id`. Потім ми вибрали поле `branch_name` з таблиці `Land_Addresses` і використали агрегатну функцію `COUNT(*)`, щоб підрахувати кількість співробітників для кожного `branch_name`. Далі, за допомогою умови `HAVING employees_count > 2`, ми обмежили результати вибірки, вибираючи тільки ті записи, для яких кількість співробітників більше 2.

```
1 SELECT branch_name, COUNT(*) AS employees_count
2 FROM Land_Addresses
3 JOIN Employees ON Land_Addresses.id = Employees.land_addresses_id
4 GROUP BY branch_name
5 HAVING employees_count > 2;
```

	branch_name	employees_count
1	Адміністративна будівля №2	3
2	Город №1	3
3	Лабораторія №2	3

Рис. 2.2.2.8. Запит з функцією «count»

```

1 SELECT Employees.id, Employees.full_name, Type_Of_Work.name
2 FROM Employees
3 INNER JOIN Type_Of_Work ON Employees.type_of_work_id = Type_Of_Work.id

```

	id	full_name	name
1	1	Олексій Іваненко	Юрист
2	2	Олександр Ковальчук	Економіст
3	3	Володимир Григоренко	Механік
4	4	Михайло Коваленко	Механік
5	5	Сергій Литвиненко	Машиніст
6	12	Іван Біленький	Бухгалтер
7	13	Марія Коваленко	Сільськогосподарський інженер
8	14	Ольга Петренко	Економіст
9	15	Андрій Лисенко	Ліпист
10	16	Наталія Савченко	Менеджер з якості
11	17	Олексій Мельник	Робітник на фермі
12	18	Ірина Петренко	Механік
13	19	Сергій Василенко	Лаборант
14	20	Марина Бондаренко	Тваринник
15	21	Олег Шевченко	Садник
16	22	Наталія Іванченко	Ветеринар
17	23	Марія Гайдар	Машиніст
18	24	Володимир Шевченко	Агроном
19	25	Оксана Полова	Агрономолог
20	26	Ігор Петров	Агрономолог
21	27	Марина Мельниченко	Садник

Рис. 2.2.2.9. Запит на вибірку з використанням агрегатної функції і умовою на вибірку поля

У цьому запиті вибираються імена працівників та назви типів робіт, які вони виконують. Використовується зв'язок INNER JOIN між таблицями "Employees" і "Type_Of_Work" на основі спільного поля "type_of_work_id". Підзапит використовується для обмеження вибірки працівників, які працюють у "Адміністративна будівля №1" у таблиці "Land_Addresses".

```

1 SELECT Employees.full_name, Type_Of_Work.name
2 FROM Employees
3 INNER JOIN Type_Of_Work ON Employees.type_of_work_id = Type_Of_Work.id
4 WHERE Employees.land_addresses_id IN (
5     SELECT id FROM Land_Addresses
6     WHERE Land_Addresses.branch_name = 'Адміністративна будівля №1'
7 )

```

	full_name	name
1	Олексій Іваненко	Юрист
2	Олександр Ковальчук	Економіст

Рис. 2.2.2.10. Запит з використанням підзапита і зв'язку INNER JOIN

У цьому запиті вибираються всі записи з таблиці "Import", якщо значення поля "import_addresses_id" знаходиться серед значень поля "id" у таблиці "Import_Addresses", де значення поля "location_type_id" рівне 17.

```

1  SELECT * FROM Import
2  WHERE import_addresses_id IN (
3      SELECT id FROM Import_Addresses
4      WHERE Import_Addresses.location_type_id = 17
5  );

```

	id	import_addresses_id	import_type_id	name	quantity	price	import_date
1	3	8	4	Яд проти колорадських жуків	50	15000	2023-06-08
2	5	8	6	Міндобрива	500	40000	2023-06-05
3	6	9	9	Таблетки для корів	15	15000	2023-06-01

Рис. 2.2.2.11. Запит з використанням підзапита і зв'язку INNER JOIN

У цьому запиті вибираються всі записи з таблиці "Animals", для яких значення поля "quantity" більше ніж найбільше значення поля "quantity" для тварин з типом "animal_type_id" рівним 1.

```

1  SELECT * FROM Animals
2  WHERE quantity > (
3      SELECT quantity FROM Animals
4      WHERE animal_type_id = 1
5      ORDER BY quantity DESC
6      LIMIT 1
7  );

```

	id	land_addresses_id	animal_type_id	gender	quantity	consumption
1	12	24	5	Самець	66	660
2	15	26	6	Самки	70	140

Рис. 2.2.2.12. Запит з використанням підзапита з використанням оператора (=, <, >):

У цьому запиті обирається середня ціна, для кожного типу експорту, де кількість більше 15, а також застосовується умова (HAVING), що обмежує результати, вибираючи лише ті записи, де середня ціна менше 5000. Результати сортуються за спаданням середньої ціни.

```

1  SELECT export_type_id, AVG(price) AS average_price
2  FROM Export
3  WHERE quantity > 15
4  GROUP BY export_type_id
5  HAVING AVG(price) < 5000
6  ORDER BY average_price DESC;

```

	export_type_id	average_price
1	2	2100.0
2	8	2000.0
3	1	2000.0

Рис. 2.2.2.13. Запит на вибірку з використанням агрегатної функції, умовою на агрегатну функцію, умовою на вибірку поля з сортуванням даних:

У цьому запиті вибирається максимальна кількість ("quantity") для кожного типу тварин ("animal_type_id") з таблиці "Animals". Застосовується умова (HAVING), що обмежує результати, вибираючи лише ті записи, де максимальна кількість перевищує 50.

```

1  SELECT animal_type_id, MAX(quantity) AS max_quantity
2  FROM Animals
3  GROUP BY animal_type_id
4  HAVING MAX(quantity) > 50;

```

	animal_type_id	max_quantity
1	1	60
2	3	55
3	5	66
4	6	70

Рис. 2.2.2.14. Запит на вибірку з використанням агрегатної функції і умовою на агрегатну функцію

У цьому запиті використовується INNER JOIN для з'єднання таблиці "Export" з таблицею "Export_Addresses" за допомогою умови E.export_addresses_id = EA.id. Запит вибирає поля E.id, E.name та EA.name_companies. Умова WHERE EA.name_companies LIKE '%АТБ%' фільтрує результати і вибирає лише ті записи, в яких поле EA.name_companies містить підрядок "АТБ".

```

1 SELECT E.id, E.name, EA.name_companies
2 FROM Export E
3 INNER JOIN Export_Addresses EA ON E.export_addresses_id = EA.id
4 WHERE EA.name_companies LIKE '%АТБ%';

```

	id	name	ame_companie
1	9	Огірки	АТБ №1
2	10	Помі...	АТБ №2

Рис. 2.2.2.15. Запит на вибірку з використанням агрегатної функції і умовою на агрегатну функцію

У цьому запиті використовується підзапит з оператором EXISTS. Зовнішній запит вибирає всі записи з таблиці "Export", для яких існує хоча б один запис в таблиці "Export_Addresses", де поле export_addresses_id в таблиці "Export" співпадає з полем id в таблиці "Export_Addresses" і поле name_companies в таблиці "Export_Addresses" має значення 'АТБ №1'. Таким чином, запит поверне всі записи з таблиці "Export", які мають принаймні один відповідний запис у таблиці "Export_Addresses" зі значенням 'АТБ №1'.

```

1 SELECT * FROM Export
2 WHERE EXISTS (
3     SELECT * FROM Export_Addresses
4     WHERE Export.export_addresses_id = Export_Addresses.id
5     AND Export_Addresses.name_companies = 'АТБ №1'
6 );

```

	id	export_addresses_id	export_warehouse_id	export_type_id	name	quantity	price	sale_date
1	9	10	17	2	Огірки	20	2100	2023-06-11

Рис. 2.2.2.16. Запит з використанням підзапита з використанням оператора EXIST.

3. Опис клієнтського додатку

Основна частина опису, можна вважати була написана в **Розділі 1, пункті 1.2. Постановка задач^[5]**. Відповідно в цьому пункті, будуть прикріплені скріншоти, і короткий опис до них.

Під час запуску, нас зустрічає головний екран (Рис. 3.1.). Як вже писалося, три вкладки, між якими можна переключатися та бокова панель, в перших 2 вкладках для перемикавання, між таблицями.

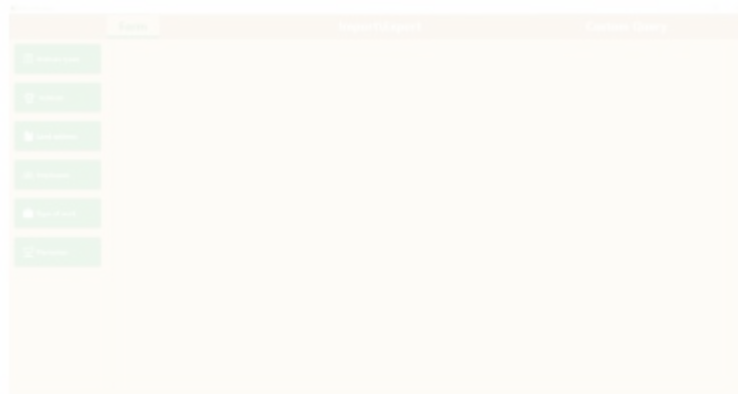


Рис 3.1. Головний екран

Такий вигляд має сторінка таблиці, в більшості, вони схожі, різниця тільки в кількості стовпчиків самої таблиці. А також зверху з'являється кнопка, для створення запису, для відповідної таблиці.

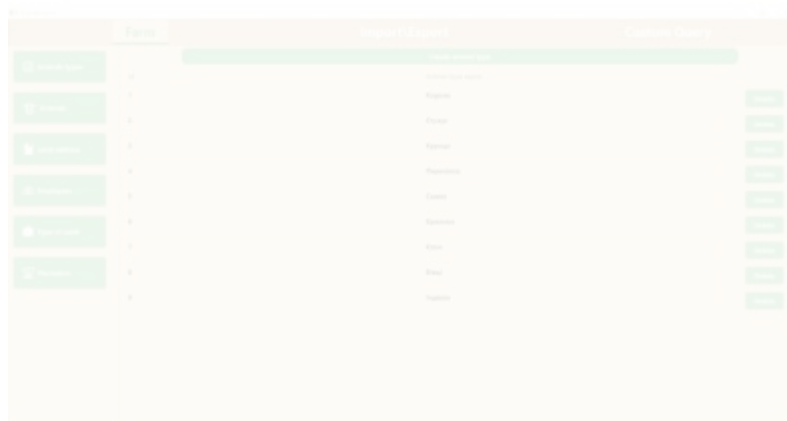


Рис 3.2. Вигляд таблиці Animals_Type

Також варто згадати, що можна було бачити вигляд таблиць, раніше в Розділі 2, пункті 2.2.2. Таблиці та об'єкти бази даних^[6]

	Export animal ID	Export location	Export type ID	Name	Quantity	Price	Date time
1	1	4	1	Bovine	20	2000	2023-06-23
2	1	3	1	Bovine	100	2000	2023-06-24
3	4	4	2	Equine	20	2000	2023-06-26
4	14	20	30	Equine	20	2000	2023-06-26
5	14	10	30	Equine	40	2000	2023-06-26
6	14	11	3	Equine	20	2000	2023-06-26
7	14	14	2	Equine	10	2000	2023-06-26
8	14	14	4	Equine	10	2000	2023-06-26
9	14	14	5	Equine	10	2000	2023-06-26
10	14	14	6	Equine	10	2000	2023-06-26
11	14	14	7	Equine	10	2000	2023-06-26
12	14	14	8	Equine	10	2000	2023-06-26

Рис 3.3. Вигляд таблиці Export

Меню створення запису, має поля для вводу (TextBox), а де є зовнішні ключі в таблицях, то під час створення є спадаючого меню (ComboBox), в те меню підтягуються дані, для зручності.

Рис 3.4. Вигляд сторінки створення запису для таблиці Animals_Type

Form titled "Import/Export" with tabs: Farm, Import/Export, Custom Query.

Left sidebar buttons: Export, Export address, Export type, Export Warehouse, Import.

Main form fields:

- Export address
- Export warehouse
- Sale date: 26.06.2023

Create button.

Рис 3.5. Вигляд сторінки створення запису для таблиці Export

Form titled "Import/Export" with tabs: Farm, Import/Export, Custom Query.

Left sidebar buttons: Export, Export address, Export type, Export Warehouse, Import.

Main form fields:

- Export address (dropdown menu open)
- Export warehouse
- Sale date

Dropdown menu items:

- Делікат №2
- Делікат №1
- Великий Смак
- Мега Оптовка
- Ароматна Дольче Віта
- Гастро Палац
- Смакова Спокуса
- АТБ №1
- АТБ №2
- Сільпо №1

Create button.

Рис 3.6. Вигляд спадаючого меню в таблиці Export, яке підтягує дані з таблиці Export_Addresses

Ну і вигляд третьої вкладки, для користувацького запиту, яка має три частини, кнопка щоб виконати запит, поле куди вводити запит, і нижнє поле, де виведеться результат.

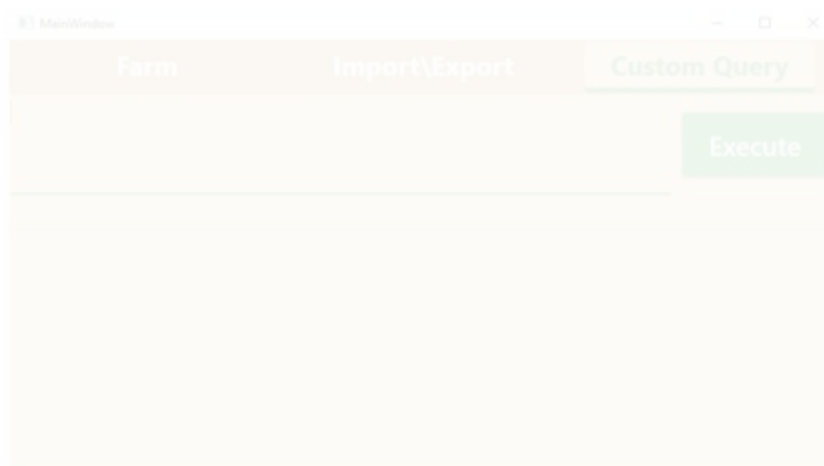


Рис. 3.7. Вигляд вкладки Custom Query

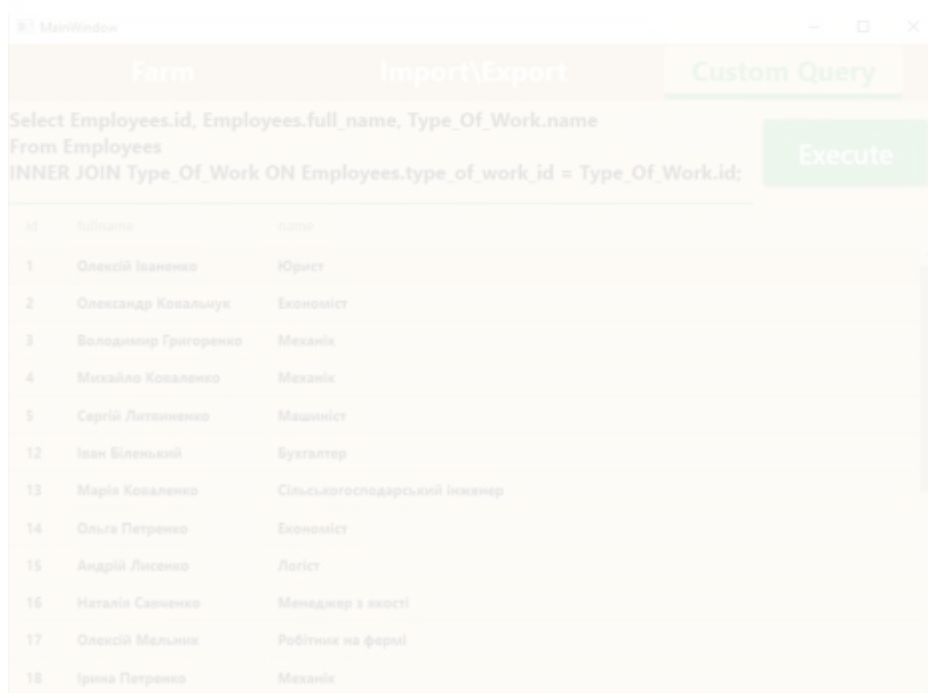


Рис. 3.8. Вигляд вкладки Custom Query з виконаним запитом

Висновок

Роботу виконано, а отже потрібно підбити підсумки. Було поставлено мету, розширити, а в основному проявити навички, які були здобуті, під час проходження курсу «Організація баз даних і знань» на обрану тему курсової роботи «Інформаційна система фермерського господарства».

Наступним пунктом був пошук даних та аналогів, для аналізу цих аналогів та виявлення їх проблем. Ці проблеми можна, також, частково вважати актуальністю теми, оскільки їх потрібно було вирішити.

Після аналізу, виявлення всіх недоліків була постановка задач, а саме, що потрібно фермерському господарству та яке програмне забезпечення використовувати для розробки, також потрібно не забувати, що було описано загальний концепт майбутнього інтерфейсу та бази даних.

Далі був крок до поглибленої роботи з базою даних, були побудовані інфологічна та даталогічна модель. В першій було описано, більш детально які сутності будуть масивним, і які сутності потрібні додатково, для комфортної роботи і розробки, також визналися поля, які потрібні цим сутностям та які приблизно зв'язки будуть між сутностями. В даталогічній моделі, розглянули саме технічну частину, який тип даних для полів потрібен, і який саме зв'язок буде між сутностями в базі даних, і врешті решт було створено та заповнено базу даних. І фінальним кроком роботи з базою даних, це було виконання запитів, для проявлення знань, і виконання мети курсової роботи.

Ну і фінальною частиною всієї курсової роботи, є розробка інтерфейсу і програми, щоб взаємодіяти з базою даних, виконати тему курсової роботи та отримати готовий програмний продукт. Так, програма не ідеальна, дуже багато малих помилок, функціонал і зручність є куди розширяти, але це вже в майбутньому.

Список інформаційних джерел

1. Авраменко А. С., Авраменко В. С., Розломій І. О. Організація баз даних і знань. Навчальний посібник. Черкаси: Черкаський національний університет імені Богдана Хмельницького, 2021. 414 с.

Перевірено: 26.06.2023.

2. Розломій І.О. Методичні вказівки до виконання курсової роботи з дисципліни «Організація баз даних та знань» для студентів спеціальностей 121 – «Інженерія програмного забезпечення», 122 – «Комп'ютерні науки», 123 – «Комп'ютерна інженерія», 124 – «Системний аналіз» денної та заочної форм навчання / Укл.: І.О. Розломій. Черкаси ЧНУ, 2022. 30 с.

Перевірено: 26.06.2023.

3. SQL Tutorial [Електронний документ]. Режим доступу: <https://www.w3schools.com/sql/>;

Перевірено: 26.06.2023.

4. SQL Tutorial [Електронний документ]. Режим доступу: <https://www.sqlitetutorial.net/what-is-sqlite/>;

Перевірено: 26.06.2023.

5. WPF Tutorial [Електронний документ]. Режим доступу: <https://www.tutorialspoint.com/wpf/index.htm>;

Перевірено: 26.06.2023.

6. WPF Tutorial [Електронний документ]. Режим доступу: <https://wpf-tutorial.com/>;

Перевірено: 26.06.2023

Схожість

Джерела з Бібліотеки

201

1	НевмитийОМ_Курсова22-23	ID файлу: 1014690184	Навчальний заклад: Bohdan Khmelnytsky Nation	24 Джерело	4.08%
2	Курсова (ШПАККТ-20)	ID файлу: 1014968159	Навчальний заклад: Bohdan Khmelnytsky National Univ	2 Джерело	3.93%
3	ХворостенкоВА_Курсова23	ID файлу: 1014696174	Навчальний заклад: Bohdan Khmelnytsky Nation	18 Джерело	3.79%
4	ОБДЗ_КС20_ШевченкоДВ_повнаНеМаєДодатків	ID файлу: 1014982033	Навчальний заклад: Bohd	19 Джерело	3.46%
5	ОБДЗ_КС20_ЛипськийСС_повнаНеМаєДодатків	ID файлу: 1014712191	Навчальний заклад: Bohdan Khmeln...		3.37%
6	ГлушковаО_Курсова22-23	ID файлу: 1014705399	Навчальний заклад: Bohdan Khmelnytsky National	51 Джерело	2.47%
7	ОБДіЗ_КН-17_Орлов	ID файлу: 1002637053	Навчальний заклад: Bohdan Khmelnytsky National Univer	6 Джерело	2.43%
8	ОБДіЗ_КН-17_Луцик	ID файлу: 1002685951	Навчальний заклад: Bohdan Khmelnytsky National Univer	3 Джерело	2.33%
9	ОБДЗ_КС20_КачановМД_крімДодатків	ID файлу: 1014712104	Навчальний заклад: Bohdan Khmelnytsky Nat...		2.28%
10	ОБДіЗ_КС-171_Скляр	ID файлу: 1003312688	Навчальний заклад: Bohdan Khmelnytsky National University of C...		2.18%
11	ОБДЗ_КС20_СавенковМР_повнаНеМаєДодатків	ID файлу: 1014982028	Навчальний заклад: Bohdan Khmeln...		2.08%
12	ЛушниковПВ_Курсова23 (2)	ID файлу: 1014696230	Навчальний заклад: Bohdan Khmelnytsky National Univer...		2.03%
13	ОБДіЗ_КН17_Пільгун	ID файлу: 1008195031	Навчальний заклад: Bohdan Khmelnytsky National University of C...		1.91%
14	ГулидаВО_Курсова22-23	ID файлу: 1014705437	Навчальний заклад: Bohdan Khmelnytsky National University...		1.88%
15	КС19_ТрифуюцанЄО	ID файлу: 1010901811	Навчальний заклад: Bohdan Khmelnytsky National University of C...		1.83%
16	ОБДЗ_КС20_СолодКВ_повнаНеМаєДодатків	ID файлу: 1014982030	Навчальний заклад: Bohdan K	13 Джерело	1.78%
17	БештанВ_Курсова23	ID файлу: 1014696172	Навчальний заклад: Bohdan Khmelnytsky National University of C...		1.63%
18	Присяник Є. В._Курсова23	ID файлу: 1014687777	Навчальний заклад: Bohdan Khmelnytsky National Universi...		1.53%
19	ОБДЗ_КС20_ПирігВВ_повнаНеМаєДодатків	ID файлу: 1014711272	Навчальний заклад: Bohdan Khmelnytsk...		1.46%
20	ОБДіЗ_КС-172_Яцун С.С	ID файлу: 1002653511	Навчальний заклад: Bohdan Khmelnytsky National University o...		1.44%

21	КасяновичРА_перевірка	ID файлу: 1014971351	Навчальний заклад: Bohdan Khmelnytsky National U	2 Джерело	1.34%
22	ОБДЗ_КС20_СергієнкоII_повнаНеМаєДодатків	ID файлу: 1014982027	Навчальний заклад: Bohdan Khmelny...		1.29%
23	КС_20_ПетрюкMI	ID файлу: 1014938909	Навчальний заклад: Bohdan Khmelnytsky National University of Cherk...		1.14%
24	ОБДіЗ_ЗапорожецьІВ_КС-171	ID файлу: 1002729254	Навчальний заклад: Bohdan Khmelnytsky National Univ...		1.06%
25	ОБДіЗ_ІльченкоД_КІ-17	ID файлу: 1002653295	Навчальний заклад: Bohdan Khmelnytsky National Un	3 Джерело	0.94%
26	Котляр Д.Я._курсова23	ID файлу: 1014703956	Навчальний заклад: Bohdan Khmelnytsky National University o...		0.87%
27	ЕП-М19-Шевчук СВ	ID файлу: 1000820236	Навчальний заклад: Bohdan Khmelnytsky National Univer	14 Джерело	0.62%
28	12987_ХарченкоВВ.pdf	ID файлу: 1000099250	Навчальний заклад: Bohdan Khmelnytsky National Unі	20 Джерело	0.54%
29	ДонченкоВ.О._КС-181	ID файлу: 1007635503	Навчальний заклад: Bohdan Khmelnytsky National Unіv	8 Джерело	0.54%
30	КН19_ЗагребельнаДО_повнаНеМаєДодатків	ID файлу: 1008229860	Навчальний заклад: Bohdan Khmelnyts...		0.49%
31	КТ_Пономаренко_безДодатків.pdf	ID файлу: 1003481936	Навчальний заклад: Bohdan Khmelnytsky Nationa...		0.49%