

МИКОЛАЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ В.О.СУХОМЛИНСЬКОГО  
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

**КУРСОВА РОБОТА**

з «Програмна інженерія (ч.1 Інженерія програмного забезпечення)»  
(назва дисципліни)  
на тему: «Проектування архітектури інформаційної системи  
«Рекламне агентство»

Студента 1 курсу 182 групи  
спеціальності

123 Комп'ютерна інженерія

Іваненко О.І.

(прізвище та ініціали)

Керівник д.т.н., професор

Борисенко В.Д.

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

Національна шкала                     

Кількість балів:                      Оцінка: ECTS                     

Члени комісії

Кузьма К.Т.

(підпис)

(прізвище та ініціали)

Мельник В.А.

(підпис)

(прізвище та ініціали)

## Зміст

<b>Вступ .....</b>	<b>3</b>
<b>1 Аналіз предметної області та постановка завдання .....</b>	<b>4</b>
1.1 Опис предметної області.....	4
1.2 Постановка завдання .....	5
<b>2 Проектування та реалізація БД .....</b>	<b>6</b>
2.1 Діаграма «СУТНІСТЬ-ЗВ'ЯЗОК» .....	6
2.2 Логічна та фізична моделі даних .....	7
2.3 Створення таблиць мовою MySQL .....	10
2.4 Запити на вибірку даних.....	13
<b>3 Програмна реалізація інформаційної системи</b>	
«Рекламне агентство» .....	15
3.1 Інтерфейси форми.....	15
3.2 Програмування зв'язку інтерфейсу та БД .....	18
<b>Висновки .....</b>	<b>20</b>
<b>Список використаних джерел.....</b>	<b>21</b>
<b>Додаток А Запити на створення таблиць.....</b>	<b>22</b>
<b>Додаток Б Інтерфейси форм.....</b>	<b>24</b>
<b>Додаток В Лістинг коду програми.....</b>	<b>27</b>

					123.182.08.KP						
Змн.	Лист	№ докум.	Підпис	Дата							
Розроб.		Іваненко О.І.			«Проектування архітектури інформаційної системи «Рекламне агентство»			Літ.	Арк.	Аркушів	
Перевір.		Кузьма К.Т.								2	36
								МНУ ім. В.О. Сухомлинського			

## Вступ

У наш час реклама виконує важливу роль у житті сучасного суспільства. Рекламу можна зустріти у будь-якому місці, куди б ви не пішли. До її створення залучаються різноманітні рекламні агентства. А для успішного функціонування таких фірм, організацій і підприємств просто не можливо без розвинутої інформаційної системи, яка дозволяє автоматизувати збір і обробку даних.

Під час функціонування інформаційних систем велика увага приділяється саме процесу створення та подальшій практиці використання бази даних.

Нині існує безліч різних баз даних, але у деяких із них можна виокремити спільні риси, які відносяться як до процесу розробки, так і використання. Такий аналіз дає змогу розібратися з базами даних і відповідним системним та прикладним програмним забезпеченням на реальних прикладах. У якості прикладу, була вибрана база даних MySQL Server (СУБД MySQL), а також її загальні питання, які стосуються проектування, розробки та використання.

Тому, за мету курсової роботи взято проектування архітектури інформаційної системи рекламного агентства.

Для того, щоб задовольнити мету, було поставлено такі завдання:

- 1) виконати опис предметної області;
- 2) розробити технічне завдання;
- 3) виконати ескізне проектування;
- 4) розробити ескіз діалогових вікон;
- 5) реалізувати взаємодію інтерфейсу із БД засобами C#.

					123.182.08.КР	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

# 1 Аналіз предметної області та постановка завдання

## 1.1 Опис предметної області

Предметною областю виступає процес автоматизації розміщення та замовлення реклами. Для розв'язання цієї проблеми потрібно розробити базу даних, яка буде зберігати усю важливу інформацію про замовлення та розміщення реклами. Готова версія бази даних має розв'язувати питання з додаванням, видаленням або оновленням інформації, а також бути зручною та надійною в користуванні.

База даних була створена на MySQL. Інтерфейс, який дає доступ до БД, реалізувався мовою C#. На сьогодні MySQL є однією з найбільш надійних систем для керування базами даних. Під час створення БД до уваги бралися найбільш важливі та розповсюдженні розробки, які можуть бути використанні як для збереження всієї інформації, так і для забезпечення комфортного використання системи користувачем.

Розроблена база даних буде зберігати та відтворювати інформацію про всі найважливіші елементи, які мають зв'язок з замовленням та розміщенням реклами.

На рис.1 можна побачити діаграму використання системи замовлення реклами клієнтом.

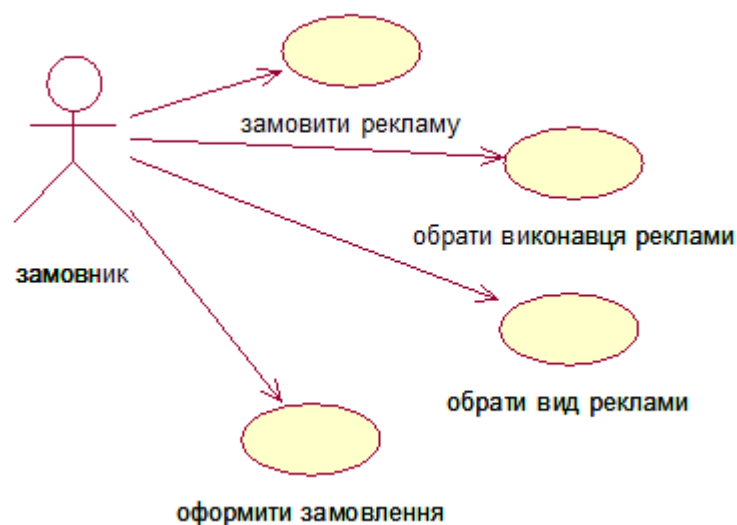


Рис. 1 – Діаграма використання системи замовлення реклами

## 1.2 Постановка завдання

В курсовому проекті за мету взято проектування архітектури інформаційної системи подачі безкоштовних оголошень.

Для того, щоб задовольнити мету, було поставлено такі завдання:

- 1) виконати опис предметної області;
- 2) розробити технічне завдання;
- 3) виконати ескізне проектування;
- 4) розробити ескіз діалогових вікон;
- 5) реалізувати взаємодію інтерфейсу із БД засобами С#.

					123.182.08.KP	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2 Проектування та реалізація БД

### 2.1 Діаграма «СУТНІСТЬ-ЗВ'ЯЗОК»

До початку реалізації бази даних в СУБД, тобто вживу, необхідно зобразити її структуру у вигляді ER-діаграми. Модель "сутність-зв'язок" (ER-модель) – модель даних, яка дозволяє описувати концептуальні схеми за допомогою узагальнених конструкцій блоків.

Модель "сутність-зв'язок" ґрунтується на важливій семантичній інформації про реальний світ і призначена для логічного представлення даних. Вона визначає значення даних в контексті їх взаємозв'язку з іншими даними.

ER-діаграма допомагає заздалегідь продумати схему бази даних, тобто це схоже на блок-схеми алгоритмів. В ER-діаграмі таблиці зображуються прямокутниками, зв'язки між таблицями (сутностями) – ромбами, між якими проводять лінії з відповідними стрілочками чи без них.

У моїй ER-діаграмі, зображеній на рисунку 2, я використовую зв'язки між сутностями типу "один до багатьох". Вони відображаються на діаграмі стрілочками лише в одному напрямку. Також є зв'язки "один до одного" та "багато до багатьох".

"Один до одного" може позначатися на діаграмі в формі ліній, які допомагають з'єднувати сутності з ромбиком, в середині якого і знаходиться вся інформація.

"Багато до багатьох" зображуються на ER-діаграмі у вигляді стрілочок, які направлені в обидві сторони.

Моя ER-діаграма налічує 4 сутності та зв'язки між ними.

					123.182.08.КР	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

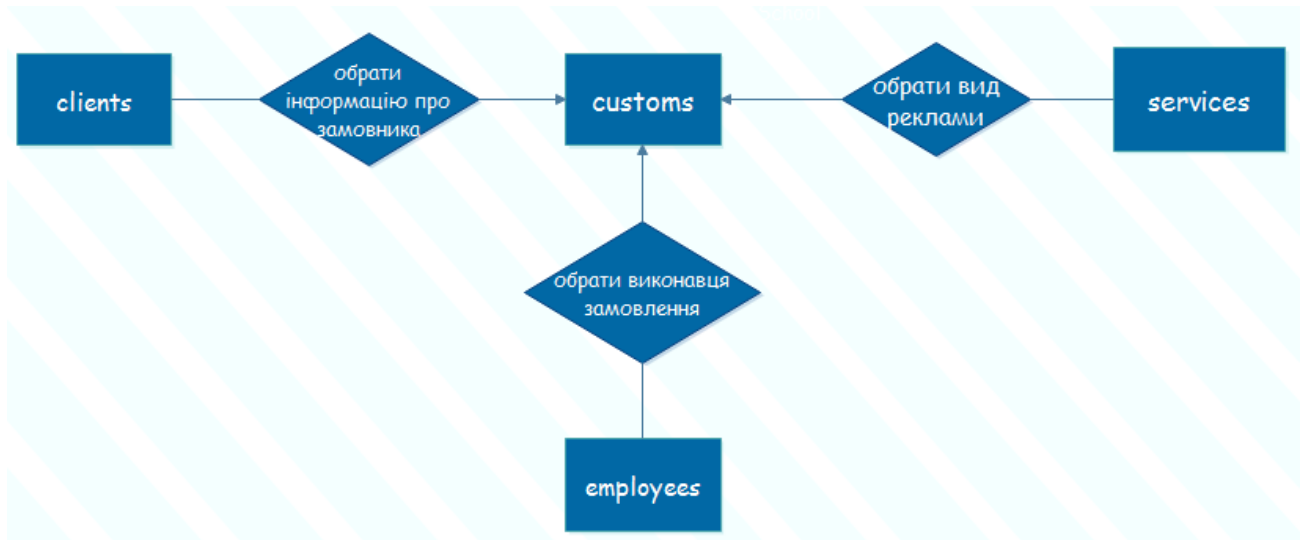


Рис. 2 – Діаграма «СУТНІСТЬ-ЗВ'ЯЗОК»

## 2.2 Логічна та фізична моделі даних

Структура бази даних може складатися як з логічної, так і фізичної моделі даних.

Взаємини між таблицями, список користувачів, правила й інші об'єкти БД визначає логічна модель.

Логічна модель - це абстрактний погляд на дані. На ньому дані представляються так, як виглядають у реальному світі. Об'єкти і моделі, що представляються на логічному рівні, називаються сутностями і атрибутами. Логічна модель даних є універсальною і ніяк не пов'язана з конкретною реалізацією СУБД.

Моя база даних складається з чотирьох таблиць і до неї було зроблено відповідну логічну модель, зображену на рисунку 3. У цій логічній моделі використовується тип зв'язку 1:Б.

					123.182.08.KP	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

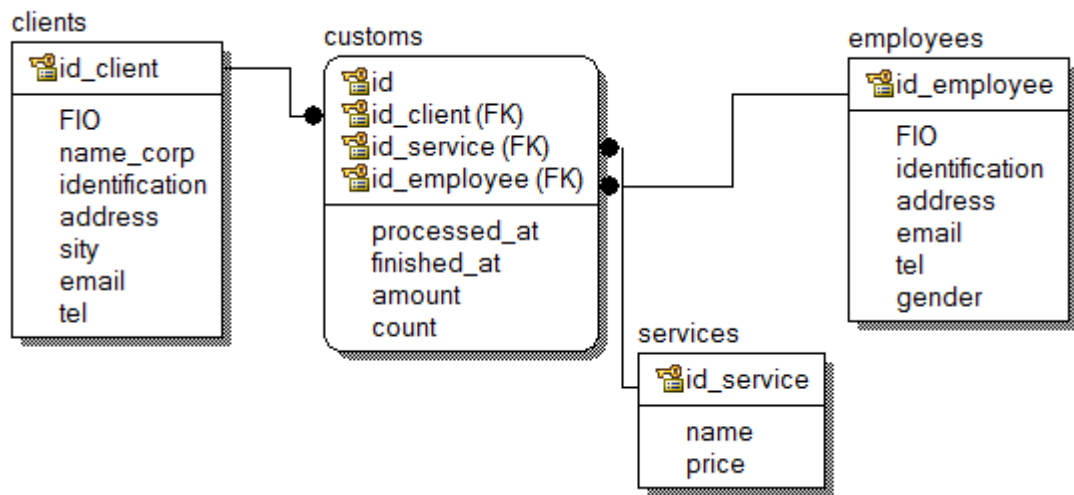


Рис. 3 – Логічна модель даних

Фізична модель даних, на відміну від логічної, є фактичним відображенням системного каталогу та залежить від певної СУБД. У даній моделі міститься важлива інформація про всі об'єкти, які містяться в БД. Виходячи з того, що стандартів на об'єкти БД існувати не може, то фізична модель має залежність від конкретної реалізації бази даних. Тобто, одній і тій самій логічній моделі відповідає лише частина фізичних моделей. При цьому, якщо логічна модель не містить в собі ніякого значення про конкретний тип даних, то важливим етапом є опис всієї важливої інформації у фізичній моделі.

На рис. 4 наведено фізичну модель розробленої БД. У таблицях 1-4 наведено детальний опис кожної сутності.

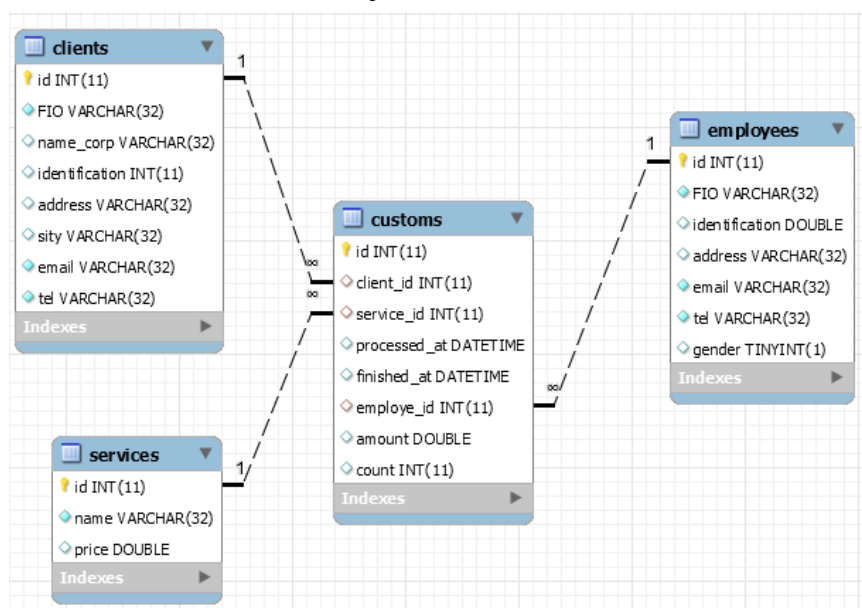


Рис. 4 – Фізична модель даних



Таблиця 1 – Структура таблиці «Services»

Назва поля	Тип даних	Опис
id	Int	Ключове поле, індексація
name	Varchar(32)	Назва виду реклами
price	double	Ціна реклами

Таблиця 2 – Структура таблиці «Clients»

Назва поля	Тип даних	Опис
Id	Int	Ключове поле, індексація
FIO	Varchar(32)	ППП замовника
Name_corp	Varchar(32)	Компанія замовника
Identification	Int	Ідентифікаційний код замовника
address	Varchar(32)	Адреса замовника
Sity	Varchar(32)	Місто проживання замовника
Email	Varchar(32)	Імейл замовника
tel	Varchar(32)	Номер телефону замовника

Таблиця 3 – Структура таблиці «Customs»

Назва поля	Тип даних	Опис
Id	Int	Ключове поле, індексація
Client_id	Int	ППП замовника, індекс
Service_id	Int	Вид реклами, індекс
Processed_at	Datetime	Дата підпису замовлення
Finished_at	Datetime	Дата отримання замовлення
Employee_id	Int	ППП виконавця замовлення, індекс
amount	double	Сума замовлення
count	int	Кількість замовлень

Таблиця 4 – Структура таблиці «Employees»

Назва поля	Тип даних	Опис
Id	Int	Ключове поле, індексація
FIO	Varchar(32)	ППП виконавця замовлення
Identification	Double	Ідентифікаційний код виконавця
Address	Varchar(32)	Адреса виконавця
Email	Varchar(32)	Імейл виконавця
Tel	Varchar(32)	Номер телефону виконавця
gender	tinyint	Стать виконавця

### 2.3 Створення таблиць мовою MySQL

У мові MySQL кожна із таблиць створюється окремим оператором CREATE TABLE. Створення таблиці включає в себе присвоєння імені таблиці і визначення її стовпців і типу даних для кожного стовпця. Основний синтаксис оператора CREATE TABLE виглядає наступним чином:

```
CREATE TABLE <Ім'я таблиці>
(<Ім'я стовпця1> <Тип стовпця1> [<Властивості стовпця1>],
<Ім'я стовпця2> <Тип стовпця2> [<Властивості стовпця2>],
...
[<Інформація про ключові стовпці та індекси>])
[<Опціональні властивості таблиці>];
```

CREATE TABLE — це ключове слово, яке надає системі баз даних, що ви хочете зробити. В даному випадку ми хочемо створити нову таблицю. Після оператора CREATE TABLE вказується її унікальне ім'я, або ідентифікатор таблиці.

Потім в дужках вказується список, визначає кожен стовпець таблиці тип даних для нього.

Моя база даних складається із 4 таблиць.

Перша таблиця це – «Services». В цій таблиці зберігаються дані про види реклам, які можна замовити та їх ціну. Структура таблиці зображена на рисунку 5, який наведено нижче.

					123.182.08.KP	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
⚡ id	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
💠 name	VARCHAR(32)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
💠 price	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Рис. 5 – Структура таблиці «Services»

Наступна таблиця – «Clients». У ній зберігається необхідна інформація про замовника реклами: ім'я, компанія, ідентифікаційний код, адреса, місто проживання замовника, імейл та номер телефону. Ознайомитися із структурою таблиці можна за наведеним нижче рисунком 6.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
⚡ id	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
💠 FIO	VARCHAR(32)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
💠 name_corp	VARCHAR(32)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
💠 identification	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
💠 address	VARCHAR(32)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
💠 sity	VARCHAR(32)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
💠 email	VARCHAR(32)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
💠 tel	VARCHAR(32)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Рис. 6 – Структура таблиці «Clients»

Тепер перейдемо до наступної таблиці «Employees», яка зберігає у собі інформація про людину, яка виконує замовлення – виконавець. Вона містить у собі ті ж поля, які присутні у таблиці «Clients». Структуру таблиці наведено на рисунку 7.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
⚡ id	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
💠 FIO	VARCHAR(32)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
💠 identification	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
💠 address	VARCHAR(32)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
💠 email	VARCHAR(32)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
💠 tel	VARCHAR(32)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
💠 gender	TINYINT(1)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'

Рис. 7 – Структура таблиці «Employees»

Остання таблиця в базі даних – «Customs». Таблиця є головною та містить у собі інформацію про зроблені замовлення, про клієнтів та виконавців, дату підписання угоди та час виконання замовлення. На рисунку 8 наведено структуру цієї таблиці.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
client_id	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
service_id	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
processed_at	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
finished_at	DATETIME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
employee_id	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
amount	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
count	INT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Index Name	Type	Foreign Key Name	Referenced Table
PRIMARY	PRIMARY		
client_id__clients_idx	INDEX	client_id__clients	`promotion`.`clients`
service_id__order_ser...	INDEX	employee_id__employees	`promotion`.`employees`
employee_id__employ...	INDEX	service_id__order_services	`promotion`.`services`

Рис. 8 – Структура таблиці «Customs»

Запити на створення усіх таблиць наведено в Додатку А.

## 2.4 Запити на вибірку даних

Для роботи з базою даних використовуються запити. SQL-запити - це запити, які складаються з послідовності SQL-інструкцій. Ці інструкції задають, що треба зробити з вхідним набором даних для генерації вихідного набору.

Існує кілька типів запитів: на вибірку, на оновлення, на додавання, на видалення, перехресний запит, створення таблиць. Найбільш поширеним є запит на вибірку. Запити на вибірку використовуються для відбору потрібної користувачу інформації, що міститься в таблицях. Вони створюються тільки для пов'язаних таблиць.

Всі запити у MySQL поділені на прості та складні.

Прості запити – це запити, у яких бере участь одна таблиця бази даних.

Складні запити – це запити, у яких можуть брати участь дві і більше таблиць БД.

MySQL дозволяє включати один запит до складу іншого. Вкладені запити називають також підлеглими. Вкладений запит у складі запиту SELECT поміщається в круглі дужки. Спочатку виконується вкладений запит, а потім отримані результати використовуються при обробці головного запиту

Загальний формат оператора SELECT наступний:

SELECT [DISTINCT] { \* | <Список полів> }

FROM <Список таблиць >

[WHERE <Умова добору записів >]

[GROUP BY <Список полів для групування>]

[HAVING <Умови добору для груп>]

[ORDER BY <Список полів для сортування>]

Саме цей оператор і використовувався при створенні курсової роботи.

Наведені нижче запити, відносяться до простих та містять у собі інструкції FROM, WHERE та LIKE:

*SELECT \* FROM `clients`* - обирає всі записи із таблиці «Clients».

*SELECT \* FROM `clients` WHERE FIO LIKE '%{0}%'* – обирає із всіх записів таблиці «clients», де «FIO» відповідає введеному значенню.

					123.182.08.KP	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

Наступний запит – складний. Він об’єднує у собі чотири таблиці: clients, customs, services, employees.

```
SELECT cus.id, cl.FIO as 'Замовник', em.FIO as 'Виконувач',
sr.name as 'Вид реклами', cus.count as 'Кіл-ть', concat(DATEDIFF(cus.finished_at,
cus.processed_at), ' days') as 'Кіл-ть днів на виконання', cus.amount as 'Вартість
замовлення', cus.processed_at as 'Дата замовлення', cus.finished_at as 'Дата
здачі замовлення' FROM customs cus
JOIN clients cl ON cus.client_id = cl.id
JOIN services sr ON cus.service_id = sr.id
JOIN employees em ON cus.employe_id = em.id;
```

					123.182.08.KP	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3 Програмна реалізація інформаційної системи «Рекламне агентство»

#### 3.1 Інтерфейси форми

Для зручного користування інформаційною системою було розроблено наступний інтерфейс, зображений на рисунку 9-10.

Рис. 9 – Інтерфейс головної форми Form1.cs

Рис. 10 – Вкладка «Редагувати»

Для користування розробленим додатком, користувачу необхідно скопіювати файл Coursework\_DB.exe на комп'ютер та встановити наступне ПЗ: MySQL Workbench, MySQL Server, MySQL Connector NET, MySQL for Visual Studio, Microsoft Visual Studio 2015.

Після запуску файлу Coursework\_DB.exe на екрані з'явиться головне вікно програми .

Для того, щоб виконати дію, потрібно обрати будь-який запис із DataGridView та натиснути Редагувати або Видалити (рис.11).

Рис. 11 – Меню «Видалити/Редагувати»



Щоб додати замовлення, потрібно по черзі (на вкладці «Оформити замовлення») вибрати вид реклами, виконувача та замовника із списку, який буде доступний при натиску на кнопку поруч з відповідним полем. Також обираємо дату отримання та кількість замовлень.

Ці ж самі дії, що й для вибору замовлення, потрібно виконати для корекції запису на вкладці «Редагувати».

Також є можливість додати нового замовника. Для цього потрібно поставити відмітку у відповідному місці (радіо-кнопка «Новий») після чого відкриється нове вікно, де можна внести нові дані про клієнта (рис. 12).

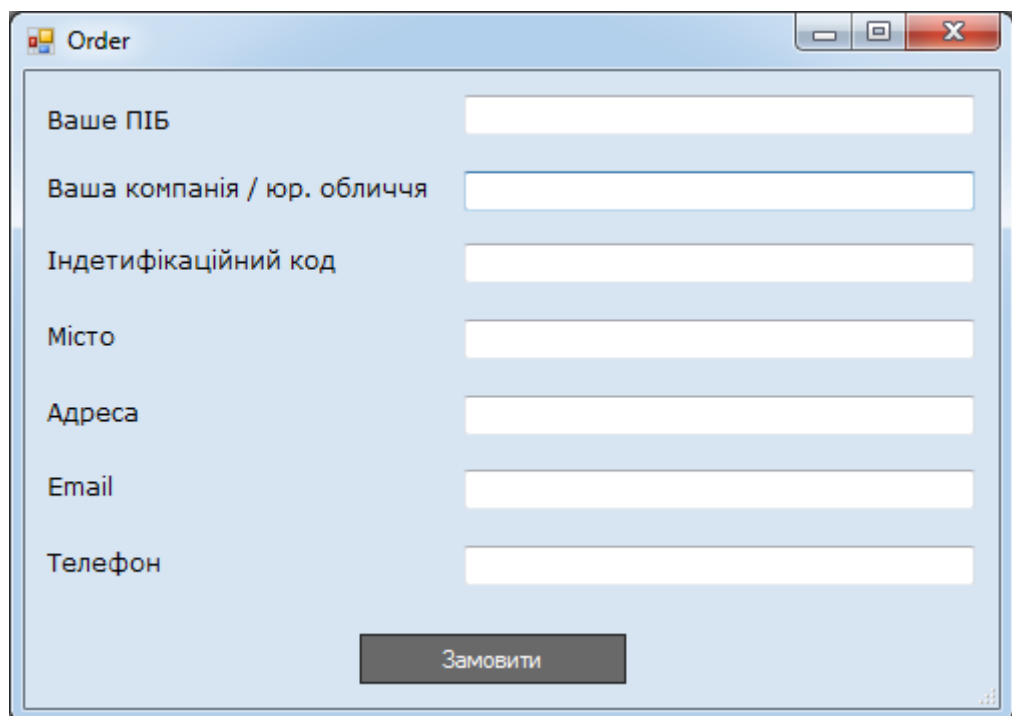
The image shows a web browser window with a title bar that says "Order". Inside the window is a form with several text input fields. The labels for these fields are in Ukrainian: "Ваше ПІБ" (Your PIB), "Ваша компанія / юр. обличчя" (Your company / legal entity), "Ідентифікаційний код" (Identification code), "Місто" (City), "Адреса" (Address), "Email", and "Телефон" (Phone). Each label is followed by an empty text input field. At the bottom of the form, there is a button labeled "Замовити" (Order).

Рис. 12 – Форма додавання нового замовника

Також в додатку працює система пошуку. Приклад використання зображено на рисунку 13.

id	Заказчик	Исполнитель	Вид рекламы	Кол-ть	Кол-ть дней на выполнение	Стоимость заказа
4	Самсонов Саму...	Титов Августин ...	Широкоформат...	1	27 days	150
5	Самсонов Саму...	Самойлова Зин...	Алюминиевые ...	3	45 days	420

Поиск:

Оформить заказ Редактировать

Вид рекламы:

Исполнитель:

Количество:

Заказчик:

Дата получения:

☒ Новый ☐ Обрати

Рис. 13 – Робота функції пошуку

### 3.2 Програмування зв'язку інтерфейсу та БД

Для того, щоб налаштувати зв'язок між БД та VisualStudio, потрібно виконати наступні дії:

1. Створити новий проект і додати посилання на MySQL.Data;
2. Потрібно додати до проекту DataGridView (знаходиться на панелі інструментів);
3. Далі потрібно в простір імен додати «using MySql.Data.MySqlClient;»
4. Після чого можна зв'язуватись із MySql та виконувати запити, відображати результати і т.д.
5. Далі потрібно розширити клас form1. Код підключення до БД має наступний вигляд:

```
public partial class Form1 : Form
{
    static string cs = new MySqlConnectionStringBuilder() //Створюємо рядок
    підключення
    {
        UserID = "root",// записуємо реквізити доступу, тепер якщо ви запускаєте
        додаток
        Password = "*****",//на комп'ютері де назва БД або паролі не
        співпадають,
        Database = "cs_test",// просто редагуєте ці рядки і все
        Server = "127.0.0.1"
    }.ConnectionString;

    public MySqlConnection con=new MySqlConnection(cs); //Оголошуємо клас
    підключення

    public Form1()
    {
        InitializeComponent();//цей рядок залишаємо
        con.Open(); //Підключаємося
        tabControl1.SelectedIndex=0; //Показуємо вкладку "Додати"
        fill_grid("SELECT * FROM `users`;", dataGridView1); //При відображенні
        форми показуємо на сітці таблицю користувачів
    }
}
```

					123.182.08.КР	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public void fill_grid(string q,DataGridView d){ //Виконуємо запит і
відображаємо результат на сітці де q –рядок запиту, d – сітка яка приймає дані
для відображення
    DataTable dt = new DataTable();//створюємо проміжну таблицю
    if (con.State.ToString()!="Open") con.Open();//якщо з'єднання закрите
відкриваємо
    MySqlCommand com = new MySqlCommand(q, con); //підготуємо запит
    MySqlDataReader dr = com.ExecuteReader(); //Виконання із поверненням
результатів запиту
    dt.Load(dr); //витягуємо рядки
    d.DataSource = dt; //поміщаємо на сітку
    dr.Close();//закриваємо допоміжну компоненту
}

public void run_query(string q){ //виконання запиту без повернення
результатів q –рядок запиту
    if (con.State.ToString()!="Open") con.Open();//якщо з'єднання закрите
відкриваємо
    MySqlCommand com = new MySqlCommand(q, con); //підготуємо запит
    com.ExecuteNonQuery(); //Виконання без повернення результатів запиту
}

```

					123.182.08.КР	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

## Висновки

Під час виконання курсової роботи були вивчені основи проектування концептуальної та логічної БД та розроблено інформаційну систему «Рекламне агентство».

Під час проектування системи спочатку була розроблена ER-діаграма бази даних з детальним описом усіх сутностей у вигляді атрибутів та складена схема даних. Виходячи з поставлених завдань, були розроблені таблиці, запити, форми та реалізовано взаємодію та роботу інтерфейсу із БД при використанні засобів C#. Для покращення умов використання бази даних, користувач має діалогове вікно, яке допомагає взаємодіяти з усією інформацією, додавати нові записи або змінювати уже існуючі.

Крім цього, я вивчив основні принципи складання запитів на MySQL. Користуючись зрозумілими за зручними командами мови MySQL, я зробив п'ять запитів, які наочно показують ефективну працездатність всіх функцій системи.

В результаті виконання даної курсової роботи всі поставлені цілі і завдання були виконані. Завдяки створенню цієї бази даних, я дізнався більше нового про роботу в цій області та отримав безліч корисної інформації. Використання мови MySQL виявилось цікавим та дуже незвичним завданням, адже при написанні програм потрібно бути не лише обізнаним в цій мові, а й проявляти винахідливість.

					123.182.08.КР	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

Список використаних джерел

					123.182.08.KP	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

## Додаток А Запити на створення таблиць

### Таблиця «Clients»

```
CREATE TABLE `clients` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `FIO` varchar(32) NOT NULL,  
  `name_corp` varchar(32) DEFAULT NULL,  
  `identification` int(11) DEFAULT NULL,  
  `address` varchar(32) DEFAULT NULL,  
  `sity` varchar(32) DEFAULT NULL,  
  `email` varchar(32) NOT NULL,  
  `tel` varchar(32) NOT NULL,  
  PRIMARY KEY (`id`)  
)
```

### Таблиця «Employees»

```
CREATE TABLE `employees` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `FIO` varchar(32) NOT NULL,  
  `identification` double DEFAULT NULL,  
  `address` varchar(32) DEFAULT NULL,  
  `email` varchar(32) NOT NULL,  
  `tel` varchar(32) NOT NULL,  
  `gender` tinyint(1) DEFAULT '0',  
  PRIMARY KEY (`id`)  
)
```

					123.182.08.KP	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

### Таблиця «Services»

```
CREATE TABLE `services` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(32) NOT NULL,  
  `price` double DEFAULT NULL,  
  PRIMARY KEY (`id`)  
)
```

### Таблиця «Customs»

```
CREATE TABLE `customs` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `client_id` int(11) DEFAULT NULL,  
  `service_id` int(11) DEFAULT NULL,  
  `processed_at` datetime DEFAULT NULL,  
  `finished_at` datetime DEFAULT NULL,  
  `employe_id` int(11) DEFAULT NULL,  
  `amount` double DEFAULT NULL,  
  `count` int(11) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `client_id__clients_idx` (`client_id`),  
  KEY `service_id__order_services_idx` (`service_id`),  
  KEY `employe_id__employees_idx` (`employe_id`),  
  CONSTRAINT `client_id__clients` FOREIGN KEY (`client_id`)  
REFERENCES `clients` (`id`),  
  CONSTRAINT `employe_id__employees` FOREIGN KEY  
(`employe_id`) REFERENCES `employees` (`id`),  
  CONSTRAINT `service_id__order_services` FOREIGN KEY  
(`service_id`) REFERENCES `services` (`id`)  
)
```

					123.182.08.KP	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		



## Додаток Б Інтерфейси форм

Form1

Інформація о заказах Таблица связей

	id	Замовник	Виконувач	Вид реклами	Кіл-ть	Кіл-ть днів на виконання	Вартість замовле
▶	4	Самсонов Саму...	Титов Августин ...	Широкоформат...	1	27 days	150
	5	Самсонов Саму...	Самойлова Зин...	Алюминиевые ...	3	45 days	420
	3	Мартынов Адам...	Носова Божена ...	Наружная рекл...	4	33 days	480
	6	Мартынов Адам...	Мухина Зарина ...	Интерьерная пе...	3	27 days	390
	1	Антонов Богдан...	Бобров Исак Пр...	Наружная рекл...	1	19 days	120
	2	Антонов Богдан...	Бобров Исак Пр...	Наружная рекл...	1	46 days	120
*							

Поиск

Оформить замовлення Редагувати

Обрати вид реклами

Виконувач

Кількість 1

Замовник

Дата отримання 26 мая 2019 г.

Новий Обрати

Далі

Рис. Б.1 – Головне вікно додатку

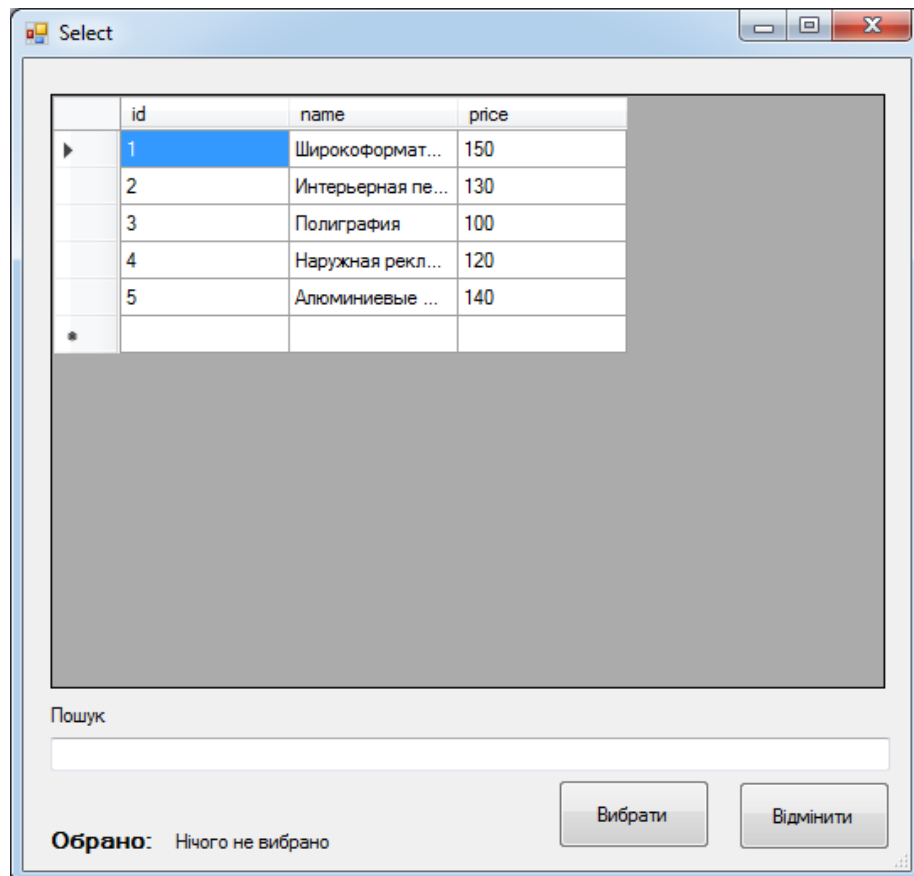


Рис. Б.2 – Вікно вибору виду реклами

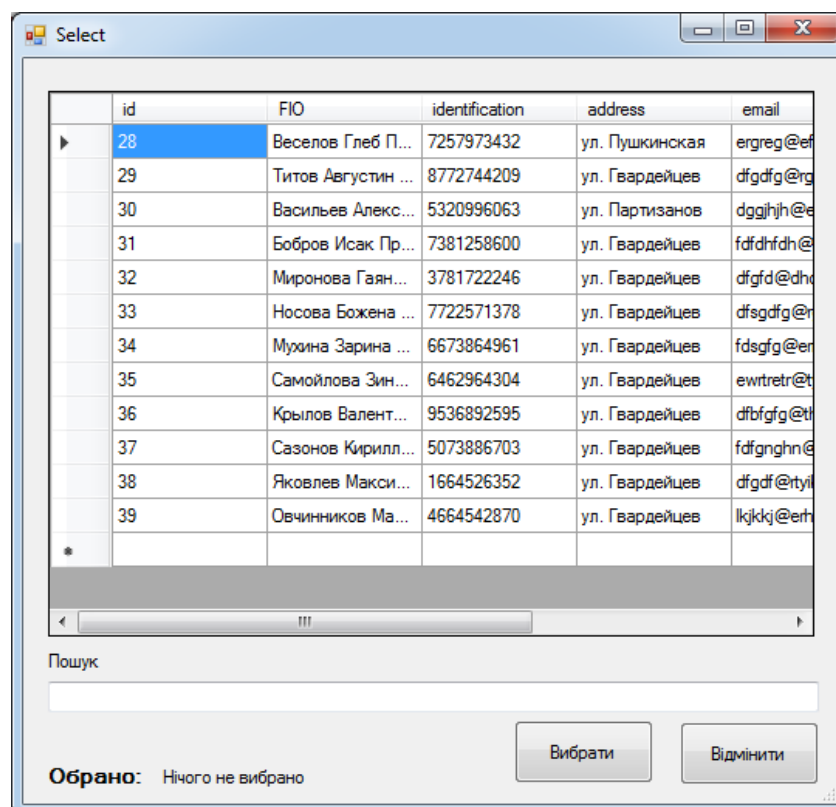


Рис. Б.3 – Вікно вибору виконавця

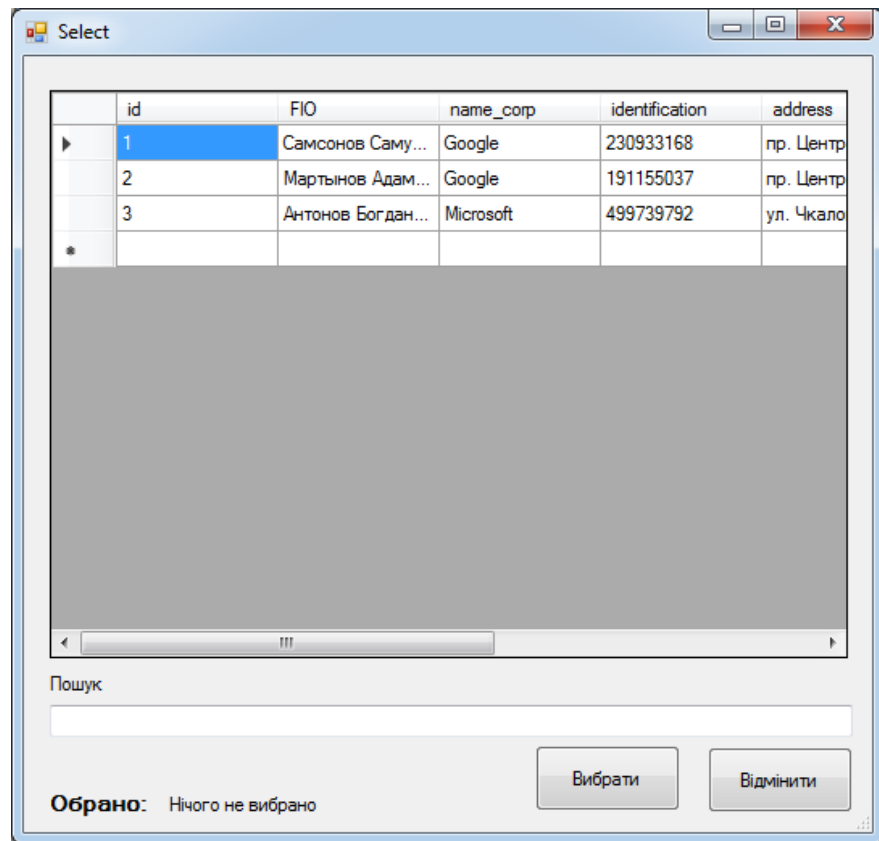


Рис. Б.4 – Вікно вибору замовника

Order

Ваше ПІБ

Ваша компанія / юр. обличчя

Ідентифікаційний код

Місто

Адреса

Email

Телефон

Замовити

Рис. Б.5 – Вікно додавання нового замовника

## Додаток В Лістинг коду програми

### Головна форма – Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

/**
 * КУРСОВАЯ РАБОТА
 */

namespace Coursework_DB
{
    public partial class Form1 : Form
    {
        public string sql_all = "SELECT cus.id, cl.FIO as 'Замовник', em.FIO as 'Виконувач',
        "+
            "sr.name as 'Вид реклами', cus.count as 'Кіл-ть', " +
            "concat(DATEDIFF(cus.finished_at, cus.processed_at), ' days') as
        'Кіл-ть днів на виконання', " +
            "cus.amount as 'Вартість ззамовлення', " +
            "cus.processed_at as 'Дата замовлення', " +
            "cus.finished_at as 'Дата здачі замовлення' " +
            "FROM customs " +
            "JOIN clients cl ON cus.client_id = cl.id " +
            "JOIN services sr ON cus.service_id = sr.id " +
            "JOIN employees em ON cus.employe_id = em.id";

        static string cs = new MySqlConnectionStringBuilder() //Створюємо рядок підключення
        {
            UserID = "promotion",
            Password = "GDEjuV4Up0kmvKbj",
            Database = "promotion",
            Server = "127.0.0.1",
            Port = 3306
        }.ConnectionString;

        public static MySqlConnection con = new MySqlConnection(cs); //Оголошуємо
        підключення

        private void fill_grid(string q, DataGridView d)
        { //Виконуємо запит і відображаємо результат на сітці
            DataTable dt = new DataTable();
            if (con.State.ToString() != "Open") con.Open();
            MySqlCommand com = new MySqlCommand(q, con);
            MySqlDataReader dr = com.ExecuteReader(); //Виконання із поверненням кузультатів
        запити
            dt.Load(dr);
            d.DataSource = dt;
            dr.Close();
        }

        private void run_query(string q)
        {
            if (con.State.ToString() != "Open") con.Open();
        }
    }
}
```

					123.182.08.KP	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        MySqlCommand com = new MySqlCommand(q, con);
        com.ExecuteNonQuery(); //Виконання без повернення результатів запиту
    }
    private int run_withquery(string q, int coll)
    {
        if (con.State.ToString() != "Open") con.Open();
        MySqlCommand com = new MySqlCommand(q, con);
        MySqlDataReader data = com.ExecuteReader(); //Виконання з поверненням
        результатів запиту
        data.Read();
        int cell = Convert.ToInt32(data[coll]);
        data.Close();
        return cell;
    }

    private Color enableButton(bool mode)
    {
        Color activ = new Color();
        if (mode)
            activ = Color.MediumAquaMarine;
        else
            activ = Color.Pink;
        return activ;
    }

    public Form1() //створення (конструктор) форми
    {
        InitializeComponent();
        fill_grid(sql_all + "; ", dataGridView1);
        fill_grid("SELECT * FROM `customs` ", dataGridView2);
        // в переліку полів оператор as переіменовує поля для того, щоб у нас не було
        // полів типу name_1
        tabControl1.SelectedIndex = 0; // перемикаємося на вкладку додати
    }

    private int getLastAddUser()
    {
        int user = run_withquery(" SELECT * FROM `clients` ORDER BY id DESC LIMIT 1; ",
0);
        return Convert.ToInt32(user);
    }
    private int getAmountCustom(int id_ser, int num)
    {
        int amount = run_withquery(" SELECT * FROM `services` WHERE id=" + id_ser + ";
", 2) * num;
        return amount;
    }

    private void видалитиToolStripMenuItem_Click(object sender, EventArgs e)
    {
        string id = dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells[0].Value +
""; // беремо id з сітки
        if (id != "") // якщо вдалося взяти значення
        {
            run_query("DELETE FROM `customs` WHERE `id`=" + id + "; "); //видаляємо
            fill_grid(sql_all + "; ", dataGridView1);
            //оновлюємо сітку
            fill_grid("SELECT * FROM `customs` ", dataGridView2);
            tabControl1.SelectedIndex = 0; //показуємо вкладку додати
        }
    }

```

					123.182.08.KP	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
}
private void button1_Click(object sender, EventArgs e) //вибір книжки
{
    string select_query = "SELECT * FROM `services`;";//запит який буде
відображатися на сітці форми 2
    string search_query = "SELECT * FROM `services` WHERE name LIKE '%{0}%';";
//запит для пошуку
    // замість {0} буде підставлений текст поля вводу для пошуку
    Select slct = new Select(con, select_query, search_query, "id", "name");//
створюємо другу форму
    //"id" - поле ідентифікатору яке повертається "name" - поле яке буде відображено
в полі вводу
    if (slct.ShowDialog() == DialogResult.OK) //якщо користувач здійснив вибір
    {
        textBox1.Text = slct.name_field; //показуємо що вибрав
        textBox1.Tag = slct.id_field;//запам'ятовуємо ідентифікатор обраного запису
    }
}

private void button2_Click(object sender, EventArgs e)
{
    string select_query = "SELECT * FROM `employees`;";//запит який буде
відображатися на сітці форми 2
    string search_query = "SELECT * FROM `employees` WHERE FIO LIKE '%{0}%';";
//запит для пошуку
    // замість {0} буде підставлений текст поля вводу для пошуку
    Select slct = new Select(con, select_query, search_query, "id", "FIO");//
створюємо другу форму
    //"id" - поле ідентифікатору яке повертається "name" - поле яке буде відображено
в полі вводу
    if (slct.ShowDialog() == DialogResult.OK) //якщо користувач здійснив вибір
    {
        textBox2.Text = slct.name_field; //показуємо що вибрав
        textBox2.Tag = slct.id_field;//запам'ятовуємо ідентифікатор обраного запису
    }
}

private void button3_Click(object sender, EventArgs e)
{
    string select_query = " SELECT * FROM `clients`;";//запит який буде
відображатися на сітці форми 2
    string search_query = " SELECT * FROM `clients` WHERE FIO LIKE '%{0}%';";
//запит для пошуку
    // замість {0} буде підставлений текст поля вводу для пошуку
    Select slct = new Select(con, select_query, search_query, "id", "FIO");//
створюємо другу форму
    //"id" - поле ідентифікатору яке повертається "name" - поле яке буде відображено
в полі вводу
    if (slct.ShowDialog() == DialogResult.OK) //якщо користувач здійснив вибір
    {
        textBox3.Text = slct.name_field; //показуємо що вибрав
        textBox3.Tag = slct.id_field;//запам'ятовуємо ідентифікатор обраного запису
    }
}

/*
 * Метод нужен для проверки введенной даты.
 * Не указал ли пользователь срок сдачи заказа на сегодня или на неделю назад?
 */
private double limit_Check_Date(DateTime dateTime){
    DateTime now = DateTime.Now;
    DateTime user = dateTime;

```

					123.182.08.KP	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        TimeSpan limit = user - now;
        double NrOfDays = limit.TotalDays;

        if (NrOfDays > 0)
            return NrOfDays;
        else
            return 0;
    }

    private void button4_Click(object sender, EventArgs e)
    {
        if (textBox1.Text.Length > 0 && textBox2.Text.Length > 0)    // Выбран вид
        рекламы && Выбран исполнитель
        {
            if (limit_Check_Date(dateTimePicker1.Value) > 0)
            { // Выбран НЕ сегодняшний и НЕ вчерашний день
                string finish = dateTimePicker1.Value.ToString("yyyy-MM-dd");
                string now = DateTime.Now.ToString("yyyy-MM-dd");
                int num = Convert.ToInt32(numericUpDown1.Value);
                int amount = getAmountCustom(Convert.ToInt32(textBox1.Tag), num);

                if (textBox3.Text.Length > 0 && radioButton2.Checked == true)
                {

                    //MessageBox.Show(limit+" "+amount);

                    run_query(String.Format(" INSERT INTO `customs`
VALUES(NULL,{0},{1},{2},{3},{4},{5},{6}); ",
                    textBox3.Tag, textBox1.Tag, "" + now + "", "" + finish + "",
                    textBox2.Tag, amount, num));
                    //оновлюємо сітку
                    fill_grid(sql_all + "; ", dataGridView1);
                    fill_grid("SELECT * FROM `customs`; ", dataGridView2);

                }
            }
            else
            {
                Order user = new Order(con);
                if (user.ShowDialog() == DialogResult.OK)
                {
                    run_query(String.Format(" INSERT INTO `customs`
VALUES(NULL,{0},{1},{2},{3},{4},{5},{6}); ",
                    getLastAddUser(), textBox1.Tag, "" + now + "", "" +
                    finish + "", textBox2.Tag, amount, num));
                    //оновлюємо сітку
                    fill_grid(sql_all+"; ", dataGridView1);
                    fill_grid("SELECT * FROM `customs`; ", dataGridView2);

                }
            }
            //очищаємо введені данні
            textBox1.Text = ""; textBox2.Text = "";
            textBox3.Text = ""; dateTimePicker1.Value = DateTime.Now;
            radioButton1.Checked = true;
            numericUpDown1.Value = 1;
        }
        else MessageBox.Show("Оберіть коректну дату!");
        else MessageBox.Show("Введіть всі значення!");
    }

    private void textBox4_TextChanged(object sender, EventArgs e)
    {
        fill_grid(sql_all +
        // використовується функція для об'єднання значень полів, а потім

```

					123.182.08.KP	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        //шукається співпадання вже з об'єднаним полем
        " WHERE CONCAT(`cl`.`FIO`,`em`.`FIO`,`sr`.`name`,`cus`.`processed_at`) LIKE '%" +
textBox4.Text + "%';", dataGridView1);
    }
    private void button5_Click(object sender, EventArgs e)
    {
        textBox4.Text = "";
    }
    private void редагуватиToolStripMenuItem_Click(object sender, EventArgs e)
    {
        tabControl1.SelectedIndex = 1; //показуємо вкладку редагувати
        //переносимо дані рядка сітки в поля редагування
        textBox7.Text =
dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells["Замовник"].Value + "";
        textBox7.Tag =
dataGridView2.Rows[dataGridView1.CurrentRow.Index].Cells["client_id"].Value + "";
        textBox6.Text =
dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells["Виконувач"].Value + "";
        textBox6.Tag =
dataGridView2.Rows[dataGridView1.CurrentRow.Index].Cells["employee_id"].Value + "";
        textBox5.Text = dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells["Вид
реклами"].Value + "";
        textBox5.Tag =
dataGridView2.Rows[dataGridView1.CurrentRow.Index].Cells["service_id"].Value + "";
        numericUpDown2.Value =
(int)dataGridView2.Rows[dataGridView1.CurrentRow.Index].Cells["count"].Value;
        dateTimePicker2.Value =
DateTime.Parse(dataGridView2.Rows[dataGridView1.CurrentRow.Index].Cells["finished_at"].Value
+ "");

        radioButton3.Checked = true;

    }
    private void button9_Click(object sender, EventArgs e)
    {
        //процедура аналогічна іншим викликам другої форми
        //див. вище
        string select_query = "SELECT * FROM `services`"; //запит який буде
відображатися на сітці форми 2
        string search_query = "SELECT * FROM `services` WHERE name LIKE '%{0}%';";
        //запит для пошуку
        // замість {0} буде підставлений текст поля вводу для пошуку
        Select slct = new Select(con, select_query, search_query, "id", "name"); //
створюємо другу форму
        //"id" - поле ідентифікатору яке повертається "name" - поле яке буде відображено
в полі вводу
        if (slct.ShowDialog() == DialogResult.OK) //якщо користувач здійснив вибір
        {
            textBox5.Text = slct.name_field; //показуємо що вибрав
            textBox5.Tag = slct.id_field; //запам'ятовуємо ідентифікатор обраного запису
        }
    }
    private void button7_Click(object sender, EventArgs e) // вибір бібліотекаря
    {
        //процедура аналогічна іншим викликам другої форми
        //див. вище
        string select_query = "SELECT * FROM `employees`";
        string search_query = "SELECT * FROM `employees` WHERE `FIO` LIKE '%{0}%';";
        Select f2 = new Select(con, select_query, search_query, "id", "FIO");
        if (f2.ShowDialog() == DialogResult.OK)
        {
            textBox6.Text = f2.name_field;
            textBox6.Tag = f2.id_field;
        }
    }
}

```

					123.182.08.KP	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32



```

private void button6_Click(object sender, EventArgs e)
{
    string select_query = "SELECT * FROM `clients`";
    string search_query = "SELECT * FROM `clients` WHERE `FIO` LIKE '%{0}%'";
    Select f2 = new Select(con, select_query, search_query, "id", "FIO");
    if (f2.ShowDialog() == DialogResult.OK)
    {
        textBox7.Text = f2.name_field;
        textBox7.Tag = f2.id_field;
    }
}
//Відмінити
private void button10_Click(object sender, EventArgs e)
{
    textBox5.Text = ""; textBox6.Text = "";
    radioButton4.Checked = true;
    numericUpDown2.Value = 1;
    dateTimePicker2.Value = DateTime.Now;

    tabControl1.SelectedIndex = 0;
}
//Змінити
private void button8_Click(object sender, EventArgs e)
{
    string id = dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells[0].Value +
    "";
    if (id != "")//якщо вдалося визначити ідентифікатор запису
    {
        if (textBox5.Text.Length > 0 && textBox6.Text.Length > 0) // Выбран вид
        рекламы && Выбран исполнитель
        {
            if (limit_Check_Date(dateTimePicker2.Value) > 0)
            { // Выбран НЕ сегодняшний и НЕ вчерашний день
                string finish = dateTimePicker2.Value.ToString("yyyy-MM-dd");
                string now = DateTime.Now.ToString("yyyy-MM-dd");
                int num = Convert.ToInt32(numericUpDown2.Value);
                int amount = getAmountCustom(Convert.ToInt32(textBox5.Tag), num);

                if (textBox7.Text.Length > 0 && radioButton3.Checked == true)
                {

                    //MessageBox.Show(limit+" "+amount);

                    run_query(String.Format(" UPDATE `customs` SET client_id={0},
                    service_id={1}, " +
                    "processed_at={2}, finished_at={3}, employe_id={4},
                    amount={5}, count={6} " +
                    "WHERE id={7}; ",
                    textBox7.Tag, textBox5.Tag, "" + now + "", "" + finish + "",
                    textBox6.Tag, amount, num, id));
                    //Оновлюємо сітку
                    fill_grid(sql_all+"", dataGridView1);
                    fill_grid("SELECT * FROM `customs`", dataGridView2);

                }
            }
            else
            {
                Order user = new Order(con);
                if (user.ShowDialog() == DialogResult.OK)
                {
                    run_query(String.Format(" UPDATE `customs` SET
                    client_id={0}, service_id={1}, " +
                    "processed_at={2}, finished_at={3}, employe_id={4},
                    amount={5}, count={6} " +

```

					123.182.08.KP	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

```

        "WHERE id={7}"; ",
        getLastAddUser(), textBox5.Tag, "" + now + "", "" +
finish + "", textBox6.Tag, amount, num, id));
//оновлюємо сітку
fill_grid(sql_all+"; ", dataGridView1);
fill_grid("SELECT * FROM `customs`; ", dataGridView2);
    }
    //очищаємо введені данні
textBox5.Text = ""; textBox6.Text = "";
textBox7.Text = ""; dateTimePicker2.Value = DateTime.Now;
radioButton4.Checked = true;
numericUpDown2.Value = 1;
}
else MessageBox.Show("Оберіть коректну дату!");
else MessageBox.Show("Введіть всі значення!");

tabControl1.SelectedIndex = 0;//перемикаємося на вкладку додати
}
}

private void radioButton2_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton2.Checked == true)
    {
        button3.Enabled = true;
        button3.BackColor = enableButton(true);
    }
    else
    {
        textBox3.Text = "";
        button3.Enabled = false;
        button3.BackColor = enableButton(false);
    }
}

private void radioButton4_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton3.Checked == true)
    {
        button6.Enabled = true;
        button6.BackColor = enableButton(true);
    }
    else
    {
        textBox7.Text = "";
        button6.Enabled = false;
        button6.BackColor = enableButton(false);
    }
}
}
}

```

					123.182.08.KP	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

## Форма 2 – «Orders»

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace Coursework_DB
{
    public partial class Order : Form
    {
        private static MySqlConnection con;
        public Order(MySqlConnection c)
        {
            InitializeComponent();
            con = c;
        }
        private void fill_grid(string q, DataGridView d)
        { //Виконуємо запит і відображаємо результат на сітці
            DataTable dt = new DataTable();
            if (con.State.ToString() != "Open") con.Open();
            MySqlCommand comm = new MySqlCommand(q, con);
            MySqlDataReader dr = comm.ExecuteReader(); //Виконання із поверненням
            кузультатів запити
            dt.Load(dr);
            d.DataSource = dt;
            dr.Close();
        }

        private void run_query(string q)
        {
            if (con.State.ToString() != "Open") con.Open();
            MySqlCommand comm = new MySqlCommand(q, con);
            comm.ExecuteNonQuery(); //Виконання без повернення результатів запити
        }

        private void button1_Click(object sender, EventArgs e)
        {
            bool flag = true;
            int[] allLength = new int[7] {
                textBox1.Text.Length, textBox2.Text.Length,
                textBox3.Text.Length, textBox4.Text.Length,
                textBox5.Text.Length, textBox6.Text.Length, textBox7.Text.Length};
            for (int i = 0; i < allLength.Length; i++)
            {
                if (allLength[i] > 5){flag = true;}
                else{flag = false;}
            }
            if (flag)
            {
                run_query(String.Format(" INSERT INTO `clients` VALUES(NULL, '{0}', '{1}',
                '{2}', '{3}', '{4}', '{5}', '{6}'))",
                textBox7.Text, textBox1.Text, textBox2.Text, textBox4.Text,
                textBox3.Text, textBox5.Text, textBox6.Text));
                this.DialogResult = DialogResult.OK;
            }
            else { MessageBox.Show("Введите верные значения!"); }
        }
    }
}
```

					123.182.08.KP	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

### Форма 3 – «Select»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

/**
 * КУРСОВАЯ РАБОТА
 */

namespace Coursework_DB
{
    public partial class Select : Form
    {
        private static MySqlConnection con;

        private void fill_grid(string q, DataGridView d)
        { //Виконуємо запит і відображаємо результат на сітці
            DataTable dt = new DataTable();
            if (con.State.ToString() != "Open") con.Open();
            MySqlCommand com = new MySqlCommand(q, con);
            MySqlDataReader dr = com.ExecuteReader(); //Виконання із поверненням кузультатів
запиту
            dt.Load(dr);
            d.DataSource = dt;
            dr.Close();
        }

        private void run_query(string q)
        {
            if (con.State.ToString() != "Open") con.Open();
            MySqlCommand com = new MySqlCommand(q, con);
            com.ExecuteNonQuery(); //Виконання без повернення результатів запиту
        }

        public string id_field { get; set; } //зберігаємо ід. запису для повернення
        public string name_field { get; set; } //зберігаємо назву запису для повернення
        private string sel_q; //зберігаємо запит який потрібно відобразити на сітці
        private string srch_q; //зберігаємо запит пошуку по таблиці
        private string id_name; //назва поля фдентифікатору запису
        private string name_name; //назва поле фмені яке повертається для відображення

        public Select(MySqlConnection c, string sel_q, string sq, string id, string
name) //конструктор форми
        { //с з'єднання, sel_q - запит для відображення, sq - запит для пошуку, id - поле
ідентифікатору, name - поле назви
            InitializeComponent();
            con = c; //зберігаємо підключення
            run_select(sel_q, sq, id, name); //процедура ініціалізації даних форми
        }

        public void run_select(string select_query, string search_query, string id_, string
name_)
        {
            sel_q = select_query; //зберігаємо запит на відображення
            srch_q = search_query; //зберігаємо запит для пошуку
            id_name = id_; // зберігаємо поле ідентифікатору
        }
    }
}

```

					123.182.08.KP	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        name_name = name_;//зберігаємо поле імені
        fill_grid(sel_q, dataGridView1);//заповнюємо сітку форми
        label2.Text = "Нічого не вибрано";//мітка для відображення вибору
    }

    private void textBox1_TextChanged(object sender, EventArgs e)
    {
        fill_grid(String.Format(srch_q, textBox1.Text), dataGridView1);//форматуємо
        запит для пошуку - підставляємо значення поля в потрібне місце
    }

    private void button2_Click(object sender, EventArgs e)
    {
        this.DialogResult = DialogResult.Cancel;//форма закривається, статус відмінено
    }

    private void button1_Click(object sender, EventArgs e)
    {
        id_field =
        dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells[id_name].Value + "";//беремо
        значення ідентифікатору
        name_field =
        dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells[name_name].Value + "";//беремо
        значення поля імені
        if ((id_field != "") && (name_field != ""))//якщо значення не порожні
        { this.DialogResult = DialogResult.OK; }//форма закривається, статус - ОК
        else { MessageBox.Show("Потрібно щось обрати!"); }//повідомлення про помилку
    }

    private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
    {
        label2.Text = "(" +//пишемо на мітці поточний вибір
        dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells[id_name].Value + ")" +
        dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells[name_name].Value + "";
    }
}
}

```

					123.182.08.КР	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

					123.182.08.KP	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

					123.182.08.KP	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

					123.182.08.KP	
Змн.	Арк.	№ докум.	Підпис	Дата	Арк. 40	