

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА  
ФРАНКА  
ФАКУЛЬТЕТ УПРАВЛІННЯ ФІНАНСАМИ ТА БІЗНЕСУ**

**Кафедра цифрової економіки та бізнес-аналітики**

**КУРСОВА РОБОТА**

**з навчальної дисципліни „Проектування та адміністрування БД і СД”**

на тему:

**„Інформаційна система для фірми що займається ремонтом  
електропобутової техніки”**

**Галузь знань:** 05 «Соціальні та поведінкові науки»

**Спеціальність:** 051 «Економіка»

**Спеціалізація:** «Інформаційні технології в бізнесі»

**Освітній ступінь:** бакалавр

**Науковий керівник:**

к.ф.-м.н., доц. Депутат Б.Я.  
(науковий ступінь, посада, прізвище,  
ініціали)

\_\_\_\_\_ “\_\_\_” травня  
2021 р. (підпис)

**Виконавець:**

Студент групи УФЕ-31с  
Книш О.Г.

\_\_\_\_\_ “\_\_\_” травня  
2021 р. (підпис)

**Загальна кількість балів** \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_  
(підпис, ПП членів комісії)

**ЛЬВІВ 2021**

## ЗМІСТ

<b>РОЗДІЛ 1. Аналіз вимог .....</b>	<b>5</b>
<b>1.1 Постановка завдання.....</b>	<b>Ошибка! Закладка не определена.</b>
<b>1.2 Розробка моделі варіантів використання бази даних.....</b>	<b>5</b>
<b>РОЗДІЛ 2. Розробка бази даних.....</b>	<b>6</b>
<b>2.1 Опис моделі даних .....</b>	<b>6</b>
<b>2.2 Нормалізація реляційних відношень .....</b>	<b>10</b>
<b>2.3 Визначення типів даних.....</b>	<b>11</b>
<b>2.4 Обмеження цілісності даних .....</b>	<b>13</b>
<b>2.5 Реалізація SQL-скрипту .....</b>	<b>14</b>
<b>Висновки .....</b>	<b>18</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>	<b>19</b>

## ВСТУП

В даній курсовій роботі розглянуто основні відомості про розробку бази даних. Спроектовано базу даних для фірми яка займається ремонтом побутової техніки.

**Актуальність** даної теми полягає у тому що, менеджери фірми повинні швидко реагувати на запити клієнтів і так само швидко відповідати на них. Також варто зазначити, що у фірми є багато постачальників запчастин які потрібні для ремонту, тому облік запчастин, затрати на їх купівлю потребують обліку а це важко робити без зручної інформаційної системи.

Внесення всіх цих даних в єдину базу дозволить систематизувати їх, а також спростити співробітникам доступ до необхідної для роботи інформації, що в наслідок тягне за собою заощадження коштів і часу, потрібних для вилучення необхідних відомостей, що повинно позитивно позначитися на ефективності роботи фірми. Використання баз даних для спрощення роботи фірм набуло широкого поширення в сучасному світі. Кінцевою метою розробки бази даних фірми та автоматизованої системи для роботи з нею є підвищення якості обслуговування клієнтів і мінімалізації витрат на це.

Сучасною формою інформаційних систем є банки даних, до складу яких входять обчислювальна система, одна чи декілька баз даних (БД), система управління базами даних та набір прикладних програм (ПП). В БД використовуються класичні моделі даних: ієрархічна, мережева, реляційна та об'єктно-орієнтовна. Ядром будь-якої бази даних є модель даних. Модель даних являє собою безліч структур даних, обмежень цілісності і маніпулювання даними. За допомогою моделі даних можуть бути представлені об'єкти предметної області та взаємозв'язки між ними.

**Мета дослідження:** Метою курсової роботи є дослідження та аналіз актуальності інформаційної системи для фірми, що займається ремонтом

електропобутової техніки. Для досягнення мети в роботі поставлено й вирішено такі теоретичні та практичні завдання: – визначити особливості створення бази даних для фірми, що займається ремонтом електропобутової техніки;

– перевірити програмний продукт на наявність дефектів та знайти шляхи покращення.

**Завдання курсової роботи:**

1. Проаналізувати проблему створення бази даних для фірм, що займаються ремонтом електропобутової техніки.
2. Створити базу даних для фірми, що займається ремонтом електропобутової техніки.

**Об’єктом дослідження є:** особливість розробки інформаційної системи для фірми, що займається ремонтом електропобутової техніки за допомогою системи управління базами даних MySQL.

**Предметом дослідження є:** огляд можливостей використання бази даних для фірми, Створити базу даних для фірми, що займається ремонтом електропобутової техніки.

Дана курсова робота складається з двох розділів. У першому розділі проаналізовано предметну область інформаційної системи. У другосу розділі описано розробку навчальної бази даних.

У кінці роботи зроблено висновки.

## **РОЗДІЛ 1. Аналіз вимог**

### **1.1 Постановка завдання**

Головні ідеї сучасних інформаційних технологій базуються на концепції баз даних. Відповідно до цієї концепції, основою інформаційних технологій є дані, які повинні бути організовані в бази даних в цілях адекватного відображення мінливого реального світу і задоволення інформаційних потреб користувачів. Одним з найважливіших і основоположних понять в теорії баз даних є поняття інформації. Під інформацією розуміються будь-які відомості про будь-яку подію, процеси, об'єкти. У свою чергу, дані - це інформація, представлена в певному вигляді, що дозволяє автоматизувати її збір, зберігання і подальшу обробку людиною або інформаційним засобом. Для комп'ютерних технологій дані - це інформація в дискретному, фіксованому вигляді, зручна для зберігання, обробки, а також для передачі по каналах зв'язку.

### **1.2 Розробка моделі варіантів використання бази даних**

Система бази даних фірми складається з п'яти елементів :

- Робітники
- Постачальники
- Деталі
- Послуги
- Клієнти

Головний користувач БД – це адміністратор. Він має змогу наповнювати БД системи новою інформацією, тобто даними про деталі які надходять , добавляти послуги , описувати замовлення, ціну тощо.

Для розробки моделі бази даних обрана реляційна модель даних. Вона найкраще підходить для вирішення цієї задачі, адже вона має ряд наступних переваг:

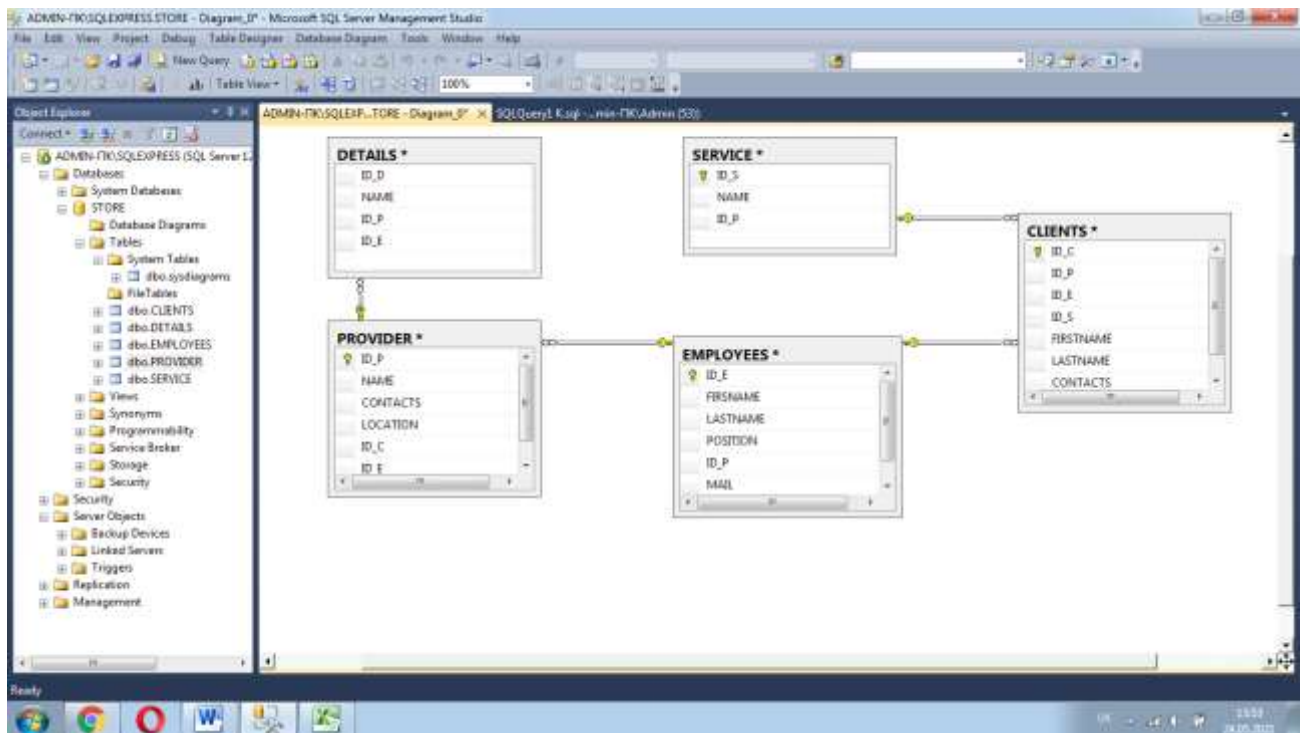
- незалежність програм від даних. Ідея використання баз даних та систем управління базами даних передбачає використання додаткового рівня між прикладними програмами та власне даними, завдяки чому прикладні програмісти можуть абстрагуватися від реалізації самої бази даних, а зосередити свою увагу на логіці обробки даних;

- простота розробки та моделювання інформаційного ресурсу як плата за деякі обмеження та уніфікацію на рівні реалізації операцій над даними;

- наявність умов керування даними за допомогою операцій над множинами.

В процесі проектування структури бази даних потрібно створити діаграму концептуальної моделі даних. На основі визначених елементів і зв'язків створити ER – діаграму.

Діаграма концептуальної моделі даних представлена на рисунку 1.1



## **РОЗДІЛ 2. Розробка бази даних**

Проектована база даних повинна володіти певними властивостями.

- Цілісність бази даних досягається внаслідок введення обмежень, пов'язані з нормалізацією БД. У кожен момент часу існування БД відомості, що містяться в ній, повинні бути несуперечливі.
- Відновлюваність - можливість відновлення БД після збою системи або окремих видів поломок системи.
- Безпека - припускає захист даних від навмисного і ненавмисного доступу, модифікації або руйнування.
- Ефективність. Властивість ефективності зазвичай оцінюється двома параметрами: мінімальний час реакції на запит користувача і мінімальні потреби в пам'яті, а також поєднанням цих параметрів.

У свою чергу, система управління базами даних, за допомогою якої користувачі можуть визначати, створювати і підтримувати базу даних, а також здійснювати до неї контрольований доступ, ділиться на два класи - система загального призначення та спеціалізована система. СУБД загального призначення не орієнтовані на якусь конкретну предметну область або на інформаційні потреби конкретної групи користувачів. Кожна система такого роду реалізується як програмний продукт, здатний функціонувати на деякій моделі ЕОМ у певній операційній обстановці. СУБД загального призначення володіє засобами налаштування на роботу з конкретною БД в умовах конкретного застосування.

У деяких ситуаціях СУБД загального призначення не дозволяють добитися необхідних проектних та експлуатаційних характеристик (продуктивність, зайнятий обсяг пам'яті та інше).

## 2.1. Опис моделі даних

База даних (БД) - упорядкований набір логічно взаємопов'язаних даних, що використовується спільно, та призначений для задоволення інформаційних потреб користувачів. У технічному розумінні включно й система управління БД. Система управління базами даних (СУБД) - це комплекс програмних і мовних засобів, необхідних для створення баз даних, підтримання їх в актуальному стані та організації пошуку в них необхідної інформації.

Централізований характер управління даними в базі даних передбачає необхідність існування деякої особи (групи осіб), на яку покладаються функції адміністрування даними, що зберігаються в базі.

Головним завданням БД є гарантоване збереження значних обсягів інформації та надання доступу до неї користувачеві або ж прикладній програмі. Таким чином БД складається з двох частин: збереженої інформації та системи управління нею. З метою забезпечення ефективності доступу записи даних організовують як множину фактів (елемент даних).

Реляційна модель орієнтована на організацію даних у вигляді двовимірних таблиць. Кожна реляційна таблиця являє собою двовимірний масив і має наступні властивості:

- кожен елемент таблиці - один елемент даних;
- всі осередки в стовпчику таблиці однорідні, тобто всі елементи в стовпчику мають однаковий тип (числовий, символьний тощо);
- кожен стовпчик має унікальне ім'я;
- однакові рядки в таблиці відсутні;
- порядок проходження рядків і стовпчиків може бути довільним.

Об'єктна СУБД ідеально підходить для інтерпретації складних даних, на відміну від реляційних СУБД, де додавання нового типу даних досягається ціною втрати продуктивності або за рахунок різкого збільшення термінів і вартості розробки додатків. Об'єктна база, на відміну від реляційної, не вимагає модифікації ядра при додаванні нового типу даних. Новий клас і його



екземпляри просто надходять у зовнішні структури бази даних. Система управління ними залишається без змін.

Кожна таблиця БД представляється як сукупність рядків і стовпців, де рядки (записи) відповідають екземпляру об'єкту, конкретній події або явищу, а стовпці (поля) - атрибутам (ознакам, характеристикам, параметрам) об'єкту, події, явища.

У кожній таблиці БД необхідна наявність первинного ключа - так іменують поле або набір полів, що однозначно ідентифікує кожен екземпляр об'єкту або запис. Значення первинного ключа в таблиці БД повинне бути унікальним, тобто в таблиці не допускається наявність два і більш за записи з однаковими значеннями первинного ключа. Він повинен бути мінімально достатнім, а значить, не містити полів, видалення яких не відіб'ється на його унікальності.

Зв'язки між об'єктами реального миру можуть знаходити своє віддзеркалення в структурі даних, а можуть і матися на увазі, тобто бути присутнім на неформальному рівні.

Між двома або більш таблицями бази даних можуть існувати відносини підлеглості, які визначають, що для кожного запису головної таблиці (батьківській) можлива наявність одного або декількох записів в підлеглій таблиці (дочірній).

Виділяють три різновиди зв'язку між таблицями бази даних:

- «один-до-багатьох»;
- «один-до-одного»;
- «багато-до-багатьох».

#### Зв'язок «один-до-багатьох»

Зв'язок «один-до-багатьох» має місце, коли одному запису батьківської таблиці може відповідати декілька записів дочірньої.

Зв'язок «один-до-багатьох» іноді називають зв'язком «багато-до-одного». І у тому, і в іншому випадку суть зв'язку між таблицями залишається незмінною. Зв'язок «один-до-багатьох» є найпоширенішим для реляційних баз даних. Вона дозволяє моделювати також ієрархічні структури даних.

#### Зв'язок «один-до-одного»

Зв'язок «один-до-одного» має місце, коли одному запису в батьківській таблиці відповідає один запис в дочірній. Це відношення зустрічається набагато рідше, ніж відношення «один-до-багатьох». Його використовують, якщо не хочуть, щоб таблиця БД «розпухала» від другорядної інформації, проте для читання зв'язаної інформації в декількох таблицях доводиться проводити ряд операцій читання замість однієї, коли дані зберігаються в одній таблиці.

Зв'язок «багато-до-багатьох» застосовується в наступних випадках:

- одному запису в батьківській таблиці відповідає більш за один запис в дочірній;
- одному запису в дочірній таблиці відповідає більш за один запис в батьківській.

Будь-який зв'язок «багато-до-багатьох» в реляційній базі даних необхідно замінити на зв'язок «один-до-багатьох» (одну або більш) за допомогою введення додаткових таблиць.

Процес розробки бази даних являє собою складний процес проектування відображення опису предметної області у схему внутрішньої моделі даних. Перебіг цього процесу є послідовністю більш простих процесів проектування менш складних відображень. Ця послідовність у процесі проектування весь час уточнюється, вдосконалюється таким чином, щоб були визначені об'єкти, їх властивості та зв'язки, які будуть потрібні майбутнім користувачам системи. Для реалізації серверної частини було обрано СУБД MySQL. Для управління базами даних використовується програма PHPMyAdmin.

PHPMyAdmin – це сукупність скриптів написаних на PHP, що забезпечують повноцінну, в тому числі віддалену роботу з базами даних

MySQL, через вебінтерфейс. Основним етапом створення таблиць вважається вибір кодування. Адже від нього залежить правильне відображення даних на веб-сторінці.

У процесі розробки проекту у базі даних «Store» створено таблиці:

- PROVIDER – таблиця, яка містить дані про постачальників деталей
- DETAILS – таблиця, яка містить дані про наявні деталі
- EMPLOYEES – таблиця, яка містить дані про працівників фірми
- CLIENTS – таблиця, яка містить дані про клієнтів фірми
- SERVICE – таблиця, яка містить дані про послуги які надає фірма

## **2.2 Нормалізація реляційних відношень**

Нормалізація - це розбивка таблиці на дві або більш, які характеризуються кращими властивостями при доповненні, зміні і вилученні даних. Процес нормалізації є одним з найважливіших, що визначає процедури приведення моделі бази даних до виду, формує ефективну базу даних, що реалізується в коректному поданні і високошвидкісний обробці даних.

Щоб виконати нормалізацію моделі бази даних, розробнику необхідно представити модель у вигляді сутності з набором атрибутів, які се будуть характеризувати, визначаючи інформаційне опис предметної області.

У цій суті будуть виникати поєднання атрибутів, які представляються функціональними залежностями. Під функціональною залежністю розуміють відношення з безліччю атрибутів, де одна сукупність атрибутів кожним своїм значенням однозначно функціонально залежить від єдиного значення сукупності інших атрибутів. У теорії реляційних баз даних прийнято виділяти таку послідовність нормальних форм:

- перша нормальна форма, 1NF (First Normal Form);
- друга нормальна форма, 2NF (Second Normal Form);
- третя нормальна форма, 3NF (Third Normal Form);
- нормальна форма Бойса – Кодда, BCNF (Brice – Codd Normal Form);

- четверта нормальна форма, 4NF (Fourth Normal Form);
- п'ята нормальна форма, 5NF (Fifth Normal Form)

### 2.3 Визначення типів даних

Структура MySQL тривірнева: бази даних – таблиці – записи. Бази даних і таблиці. Ім'я бази даних MySQL унікальна в межах системи, а таблиці - в межах бази даних, поля - в межах таблиці. База даних з точки зору MySQL - це звичайний каталог, що містить файли певного формату - таблиці. Таблиці складаються із записів, а записи, у свою чергу, складаються з полів. Поле має два атрибути - ім'я і тип даних.

Основні типи даних, які використовувались:

- VARCHAR – може зберігати не більше 255 символів. На відміну від CHAR, для зберігання значення даного типу виділяється необхідна кількість пам'яті;
- TEXT – може зберігати не більше 65 535 символів;
- INT – діапазон від -2 147 483 648 до 2 147 483 647;

Створивши нову базу даних «STORE» вона містить таблиці даних фірми.

Таблиця 2.1

Склад таблиці «PROVIDER»

№	Ім'я атрибуту	Тип даних	Розмір поля
1	ID_P	INT	
2	NAME	VARCHAR	50
3	CONTACTS	VARCHAR	30
4	ID_M	VARCHAR	
5	LOCATION	VARCHAR	30

Таблиця 2.2

Склад таблиці «DETALIS»

№	Ім'я атрибуту	Тип даних	Розмір поля
1	ID_D	INT	
2	NAME	VARCHAR	80

Таблиця 2.3

Склад таблиці «EMPLOYEES»

№	Ім'я атрибуту	Тип даних	Розмір поля
1	ID_E	INT	
2	FIRSTNAME	VARCHAR	30
3	LASTNAME	VARCHAR	30
4	POSITION	VARCHAR	30
5	ID_P	INT	
6	MAIL	VARCHAR	30

Таблиця 2.4

Склад таблиці «CLIENTS»

№	Ім'я атрибуту	Тип даних	Розмір поля
1	ID_C	INT	
2	ID_P	INT	
3	ID_E	INT	
4	ID_S	INT	
5	FIRSTNAME	VARCHAR	30
6	LASTNAME	VARCHAR	30
7	CONTACTS	VARCHAR	30

Таблиця 2.5

Склад таблиці «SERVICE»

№	Ім'я атрибуту	Тип даних	Розмір поля
1	ID_S	INT	
2	NAME	VARCHAR	30
3	ID_P	INY	

## 2.4 Обмеження цілісності даних

Правила цілісності бази даних - це правила, які дають змогу уникнути введення некоректних даних у БД, а також забезпечити можливість зв'язування декількох таблиць. Ці правила можуть бути описані при створенні чи модифікації таблиці. В sql виділяють такі типи цілісності даних:

- CHECK - Контроль допустимих значень атрибутів;
- NOT NULL/NULL - Заборона/ дозвіл на використання не заданих або не визначених значень.
- UNIQUE - Контроль унікальності значень атрибутів.
- PRIMARY KEY - Первинний ключ.
- FOREIGN KEY - Зовнішній ключ.

PRIMARY KEY (PK) використовується для ідентифікації рядків таблиці, в нього є певні особливості: - В одній таблиці БД може бути тільки один PK. - Рядки, які мають це правило, не можуть мати не визначені або не задані дані.

FOREIGN KEY (FG) використовується для того, щоб зв'язати дві таблиці, в нього є такі особливості : - Поле у головній таблиці, на яке буде адресуватися FG повинне бути оголошене PK або UNIQUE. - Поле у головній таблиці і поле з якої відбувається адресація повинні мати один й той самий тип даних Також за допомогою FG можна реалізувати каскадне видалення і оновлення даних. При розробці даної бази даних „STORE” використовувалось: primary key.

## 2.5 Реалізація SQL-скрипту

```
//СТВОРЮЄМО БАЗУ ДАНИХ//CREATE DATABASE STORE
```

```
//ДАЛІ ЩОБ ПРАЦЮВАТИ В СТВОРЕНІЙ БАЗІ ДАНИХ ПИШЕМО//USE  
STORE
```

```
//СТВОРЮЄМО ТАБЛИЦЮ PROVIDER// CREATE TABLE PROVIDER
```

```
//далі даємо назву стовпцю і вказуємо дані які він буде містити// (ID_P INT,  
NAME VARCHAR(50), CONTACTS VARCHAR(30), LOCATION  
VARCHAR(30), ID_M INT)
```

```
// створюємо таблицю DETAILS //CREATE TABLE DETAILS
```

```
//даємо назву стовпцям і вказуємо які дані вони будуть містити//(ID_D INT,  
NAME VARCHAR(80))
```

```
CREATE TABLE EMPLOYEES (ID_E INT, FIRSNAME VARCHAR(30),  
LASTNAME VARCHAR(30), POSITION VARCHAR(30), ID_P INT, MAIL  
VARCHAR (30), PRIMARY KEY (ID_E))
```

```
CREATE TABLE CLIENTS(ID_C INT, ID_P INT, ID_E INT, ID_S INT,  
FIRSTNAME VARCHAR(30), LASTNAME VARCHAR(30), CONTACTS  
VARCHAR(30), PRIMARY KEY(ID_C))
```

```
CREATE TABLE SERVICE (ID_S INT, NAME VARCHAR(30), ID_P INT  
PRIMARY KEY (ID_S))
```

```
ALTER TABLE PROVIDER
```

```
ADD ID_E INT
```

```
SELECT*
```

```
FROM CLIENTS
```

```
ALTER TABLE PROVIDER
```

DROP COLUMN ID\_M

//Заповнюю базу даних//

INSERT INTO PROVIDER(ID\_P,NAME, CONTACTS, LOCATION, ID\_C, ID\_E)  
VALUES(1, 'LG', '232-80-85', 'LVIV', 1, 1)

INSERT INTO PROVIDER(ID\_P,NAME,CONTACTS, LOCATION, ID\_C, ID\_E)  
VALUES(2,'BOSCH', '453-60-00', 'FRANCE',2,1)

INSERT INTO PROVIDER(ID\_P,NAME,CONTACTS, LOCATION, ID\_C, ID\_E)  
VALUES(3, 'SAMSUNG', '555-66-00', 'ITALY', 3,1)

INSERT INTO PROVIDER(ID\_P,NAME,CONTACTS, LOCATION, ID\_C, ID\_E)  
VALUES(4, 'PHILIPS', '456-87-99', 'GERMANY', 4, 2)

INSERT INTO PROVIDER(ID\_P,NAME,CONTACTS, LOCATION, ID\_C, ID\_E)  
VALUES(5,'GORENJE', '657-00-34', 'DENMARK',5, 2)

INSERT INTO PROVIDER(ID\_P,NAME,CONTACTS, LOCATION, ID\_C, ID\_E)  
VALUES(6, 'ELECTROLUX', '555-32-11', 'POLAND', 6, 2)

INSERT INTO PROVIDER(ID\_P,NAME,CONTACTS, LOCATION, ID\_C, ID\_E)  
VALUES(7, 'ATLANT', '444-34-50', 'HUNGARY' 7, 3)

INSERT INTO PROVIDER(ID\_P,NAME,CONTACTS, LOCATION, ID\_C, ID\_E)  
VALUES(8, 'BEKO', '888-00-00', 'SPAIN', 8, 3)

INSERT INTO EMPLOYEES(ID\_E, FIRSNAME, LASTNAME, POSITION, ID\_P,  
MAIL)

VALUES( 1, 'OLEG', 'KNYSH', 'MANAGER', 1,'OLEGLVIVUA@GMAIL.COM')

INSERT INTO EMPLOYEES(ID\_E, FIRSNAME, LASTNAME, POSITION, ID\_P,  
MAIL)

VALUES(2, 'YEVHEN', 'OSOBA', 'MANAGER', 2, 'OSOBA10@GMAIL.COM')

INSERT INTO EMPLOYEES(ID\_E, FIRSNAME, LASTNAME, POSITION, ID\_P,  
MAIL)

VALUES(3, 'NAZAR', 'BEREZUK', 'MANAGER', 3, 'NAZAR47@GMAIL.COM')



```
SELECT EMPLOYEES.LASTNAME, PROVIDER.NAME
FROM      EMPLOYEES      JOIN      PROVIDER      ON
EMPLOYEES.ID_E=PROVIDER.ID_P
```

```
INSERT INTO CLIENTS(ID_C, ID_P, ID_E, ID_S, FIRSTNAME, LASTNAME,
CONTACTS)
```

```
VALUES(1,1,1,1, 'OLEG', 'OSOBA', '0673567234')
```

```
INSERT INTO CLIENTS(ID_C, ID_P, ID_E, ID_S, FIRSTNAME, LASTNAME,
CONTACTS)
```

```
VALUES(2,2,1,2, 'ANDRII', 'BORYSENKO', '0674545123')
```

```
INSERT INTO CLIENTS(ID_C, ID_P, ID_E, ID_S, FIRSTNAME, LASTNAME,
CONTACTS)
```

```
VALUES(3,3,2,3, 'OLEKSII', 'KOVALENKO', '0678496957')
```

```
INSERT INTO CLIENTS(ID_C, ID_P, ID_E, ID_S, FIRSTNAME, LASTNAME,
CONTACTS)
```

```
VALUES(4,4,2,4, 'OLEG', 'MARCHENKO', '0673455123')
```

```
INSERT INTO CLIENTS(ID_C, ID_P, ID_E, ID_S, FIRSTNAME, LASTNAME,
CONTACTS)
```

```
VALUES(5,5,3,5, 'LEONID', 'MARYNYCH', '0674358321')
```

```
INSERT INTO CLIENTS(ID_C, ID_P, ID_E, ID_S, FIRSTNAME, LASTNAME,
CONTACTS)
```

```
VALUES(5,5,3,5, 'PETRO', 'FILIPCHUK', '0674568900')
```

```
INSERT INTO DETAILS(ID_D, NAME)
```

```
VALUES(1, 'TO THE TV')
```

```
INSERT INTO DETAILS(ID_D, NAME)
```

```
VALUES(2, 'TO THE WASHING MACHINES')
```

```
INSERT INTO DETAILS(ID_D, NAME)
```

VALUES(3, 'TO THE REFRIGERATORS')

INSERT INTO DETAILS(ID\_D, NAME)

VALUES(4, 'TO THE COFFEE MACHINES')

INSERT INTO DETAILS(ID\_D, NAME)

VALUES(5, 'TO THE SMALL HOUSEHOLD APPLIANCES')

INSERT INTO SERVICE(ID\_S, NAME, ID\_P)

VALUES(1, 'TV REPAIR', 1)

INSERT INTO SERVICE(ID\_S, NAME, ID\_P)

VALUES(2, ' WASHING MACHINES REPAIR', 2)

INSERT INTO SERVICE(ID\_S, NAME, ID\_P)

VALUES(3, 'REFRIGERATORS REPAIR', 3)

INSERT INTO SERVICE(ID\_S, NAME, ID\_P)

VALUES(4, 'COFFEE MACHINES REPAIR', 4)

INSERT INTO SERVICE(ID\_S, NAME, ID\_P)

VALUES(5, 'SMALL HOUSEHOLD APPLIANCES REPAIR', 5)

INSERT INTO SERVICE(ID\_S, NAME, ID\_P)

VALUES(1, 'TV REPAIR', 6)

INSERT INTO SERVICE(ID\_S, NAME, ID\_P)

VALUES(2, ' WASHING MACHINES REPAIR', 7)

INSERT INTO SERVICE(ID\_S, NAME, ID\_P)

VALUES(4, 'COFFEE MACHINES REPAIR', 8)

## ВИСНОВКИ

У процесі виконання дослідження мета була досягнута, а завдання вирішені, а саме успішно виконано теоретичні і практичне завдання, побудова бази даних для фірми, що займається ремонтом електроробутової техніки. Результати проведеного дослідження дають підставу зробити такі висновки:

1) З вище опрацьованих джерел використаної літератури та з власного аналізу інформаційної системи, безумно можна стверджувати про актуальність бази даних для фірми.

2) Особливість будь якої бази даних полягає в його функціоналі. Перш ніж створити базу даних необхідно детально проаналізувати призначення її призначення, зробити макет. У даному випадку, під час аналізу було вирішено створити базу даних яка спростить роботу менеджерам фірми

Отож, враховуючи усе вище сказане, можна стверджувати про важливу роль інформаційних систем, зокрема побудована інформаційна система позитивно впливає на розвиток бізнесу, а саме спрощує роботу працівникам, які надають ті чи інші послуги.

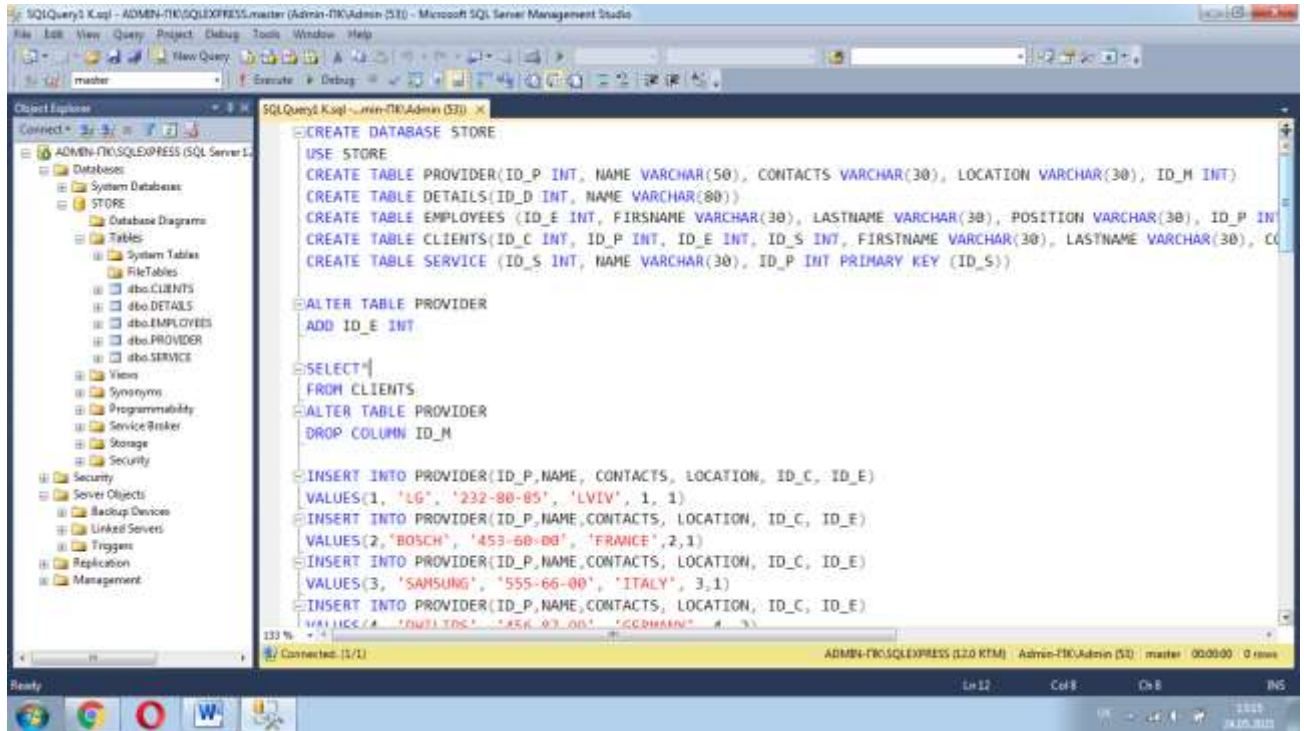
## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мартин Грабер :SQL; Видавництво «Лори», 2016. -643 с.
2. Bootstrap [Електронний ресурс]. – Режим доступу: <https://timeweb.com/ru/community/articles/plyusy-i-minusy-bootstrap-1>
3. Переваги PHP [Електронний ресурс]. – Режим доступу: <https://blog.ithillel.ua/ua/articles/chomu-vybyraiut-7-perevah-movuprohramuvannia-php>.
4. Моделі даних [Електронний ресурс]. – Режим доступу: [https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%B%D1%96\\_%D0%B1%D0%B0%D0%B7\\_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%B%D1%96_%D0%B1%D0%B0%D0%B7_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85).
5. Типи даних MySQL [Електронний ресурс]. – Режим доступу: [https://uk.wikibooks.org/wiki/SQL/%D0%A2%D0%B8%D0%BF%D0%B8\\_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85\\_MySQL](https://uk.wikibooks.org/wiki/SQL/%D0%A2%D0%B8%D0%BF%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85_MySQL).
6. Буч Г. Язык UML: руководство пользователя / Г. Буч, Д. Рамбо, И. Якобсон // [пер. с англ. Н. Мухина]. – Москва. – 2019. – 493 с.
7. Дивак М.П. Системний аналіз та проектування КІС /М.П.Дивак// Навчальний посібник – Т.: Економічна думка. – 2018.

## Додаток

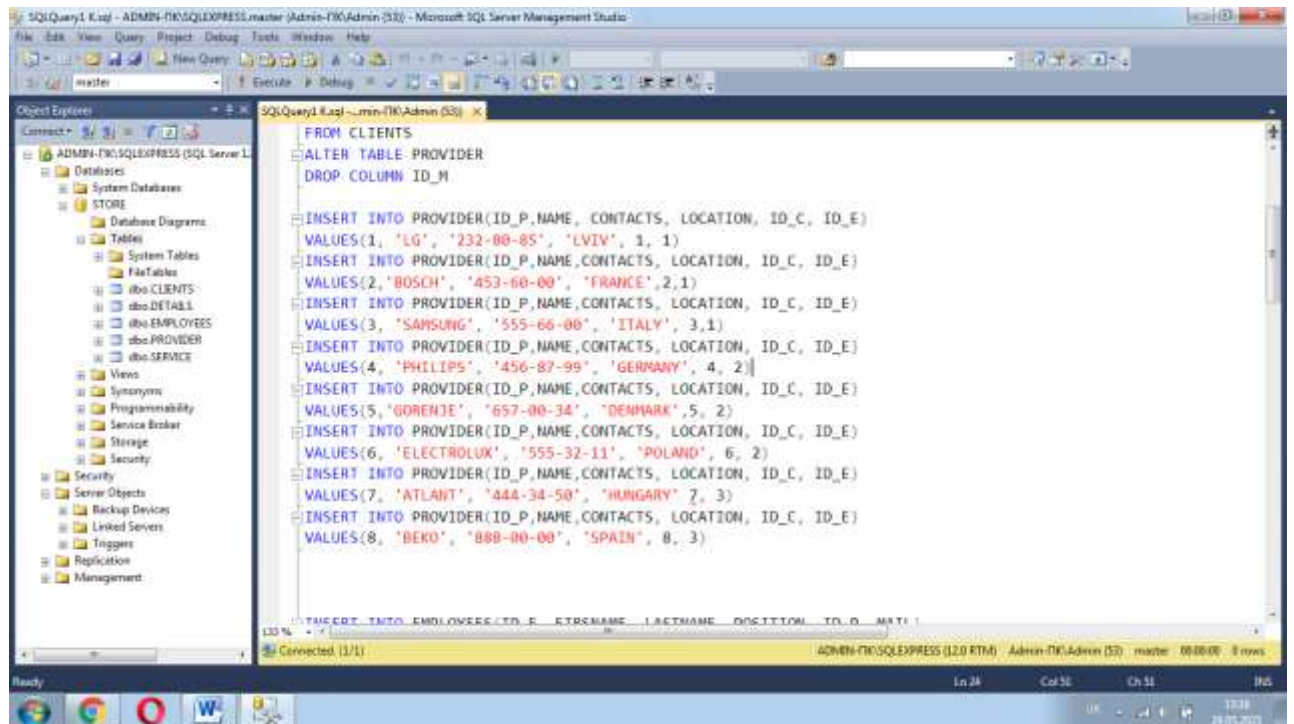
## Додаток А

### DATABASE STORE



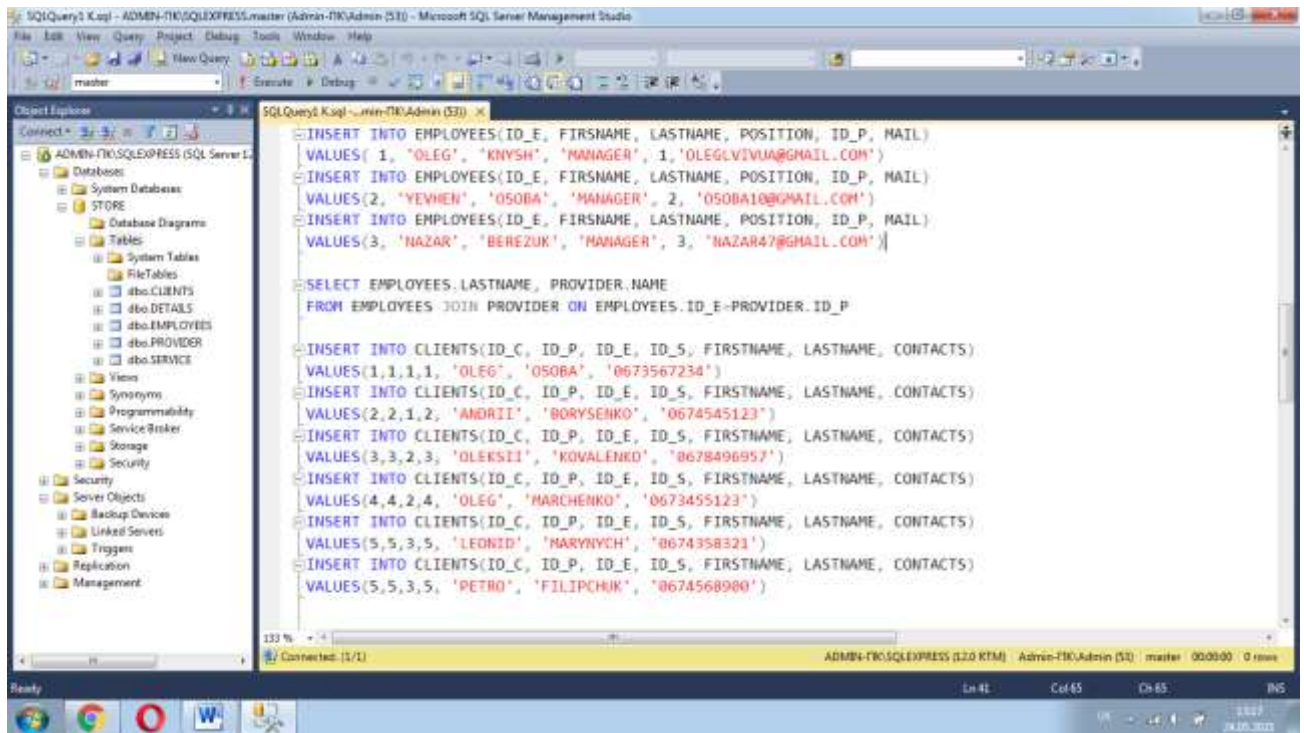
## Додаток Б

## Таблиця PROVIDER



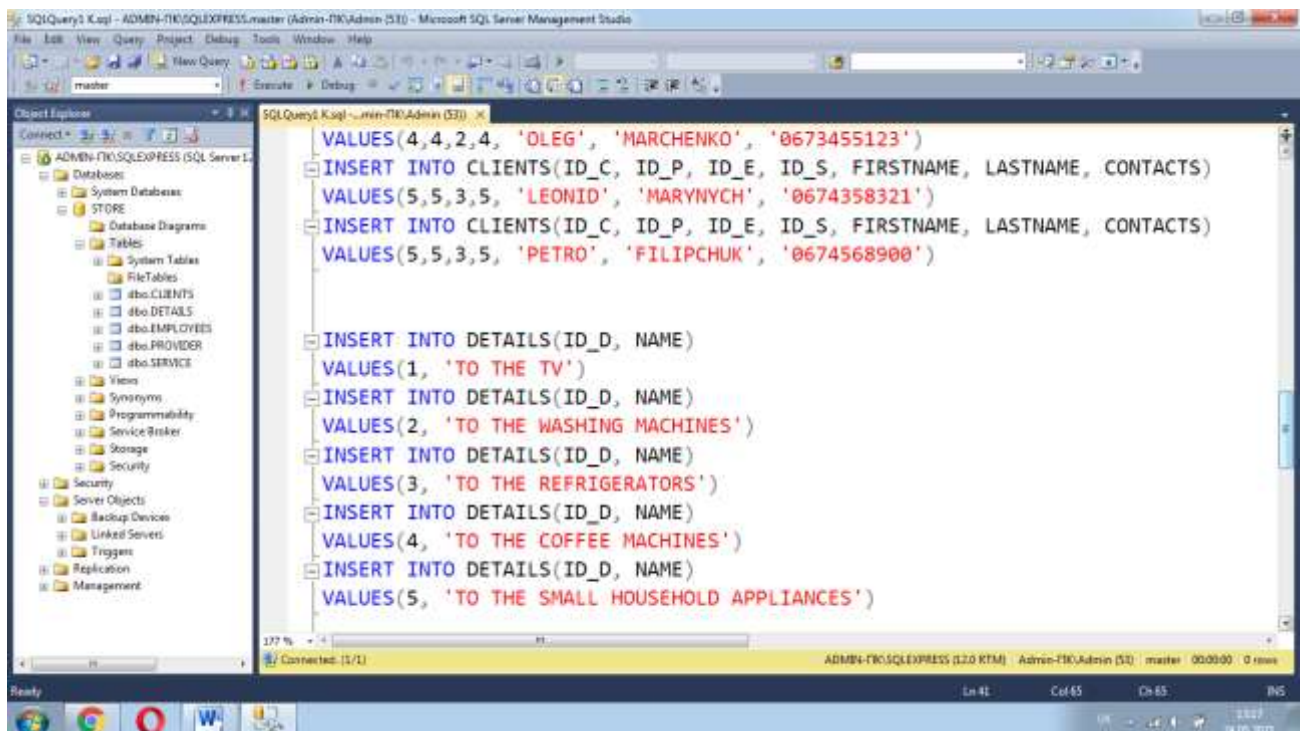
## Додаток В

## Таблиця EMPLOYEES



## Додаток Г

## Таблиця DETAILS



## Додаток Д

## Таблиця SERVICE

