

**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”**

**Тема:
«Інформаційна система для студії веб-дизайну»**

Науковий керівник:

к.е.н., доц. Ярема О.Р.

(прізвище, ім'я, по-батькові)

_____ (підпис)

“ ____ ” _____ 2020 р.

Виконавець:

Левицький Ю.В.

(прізвище, ім'я, по-батькові)

УФЕ-31с група
_____ (підпис)

“ ____ ” _____ 2020 р.

Загальна кількість балів

_____ (підписи, ППП членів комісії)

Львів 2020

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА АНАЛІЗ РОБОТИ СТУДІЇ ВЕБ-ДИЗАЙНУ	5
1.1 Дослідження роботи студії веб-дизайну.	17
1.2 Аналіз роботи студії веб дизайну	5
1.3 Постановка завдання	6
РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ.....	7
2.1 Основні поняття концептуального проектування.	7
2.2 Розробка концептуальної моделі даних (ER-діаграми).....	8
2.2.1 Перелік таблиць бази даних.....	12
2.2.2 Перелік полів таблиць бази даних.....	12
2.3 Нормалізація реляційних відношень	15
2.4 Визначення типів даних.....	17
2.5 Обмеження цілісності даних	17
2.6 Створення бази даних	17
ВИСНОВКИ	20
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	21

ВСТУП.

Інтернет став невід'ємною частиною нашого життя. Подорожуючи по мережі Інтернет, ми зустрічаємо різноманітні веб-сайти. На одних продають речі, на інших запрошують знайти друзів для спілкування. Є широко затребувані інформаційні портали, а є сайти, присвячені окремим людям або компаніям. Тенденція така, що для успіху необхідно заявити про себе в Інтернеті, а значить, власна персональна сторінка повинна бути у будь-якої фірми і навіть в особистості.

Користувачам приємно відвідувати веб-сторінки, які мають стильне оформлення, не обтяжені надмірно графікою і анімацією, швидко завантажуються і правильно відображаються у вікні веб-браузера. Завдання веб-дизайну будь-якого сайту - викликати довіру відвідувача, до якості пропонованих нею послуг, зберігаючи при цьому фірмовий стиль, корпоративну культуру, і позиціонування на ринку.

Дизайн сайту і саме його побудова поєднує в собі різні нюанси, які необхідно враховувати при розробці. Насамперед, це наявність зручної навігації та прозорості структури сайту. При цьому потрібно враховувати художню цінність і витонченість графічного дизайну.

Обрана тема курсової роботи є актуальною, оскільки інформаційна система допомагає забезпечити оптимізацію роботи підприємства.

Метою курсової роботи є дослідження теоретичних основ ІС та її реалізація.

Відповідно до мети було визначено такі завдання:

- проаналізувати предметну область та здійснити проектування БД дизайн студії на основі системи управління базами даних SQL;
- забезпечити редагування даних таблиць БД та взаємозв'язків між різними таблицями;

Об'єктом дослідження курсової роботи є особливості розробки інформаційної системи для дизайн студії з використання системи управління базами даних SQL.

Предметом дослідження є теоритичні основни та практичне застосування технологій SQL, SQL Server Management Studio(SSMS).

Інформаційною базою для написання курсової роботи по даній темі стали, підручники і навчальні посібники, інформаційні матеріали з Інтернету.

Робота складається з двох розділів. У першому розділі описано бізнес-ідею та аналіз вимог. У наступному описано побудову навчальної бази даних.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА АНАЛІЗ РОБОТИ СТУДІЇ ВЕБ-ДИЗАЙНУ.

1.1 Дослідження роботи студії веб-дизайну.

Тема досліджуваної предметної області - «Студія веб-дизайну». Дизайн студія спеціалізується на розробці й оформленні веб - сайтів. Сучасні технології дозволяють вирішувати будь-які поставлені завдання. Вона пропонує інструменти для ведення бізнесу в Інтернет, що впливають на прибуток підприємства, його популярність і попит на його продукцію (послуги).

Головною метою студії є надання якісних послуг, орієнтованих на певного клієнта, виконання роботи (замовлення) в мінімальні терміни і в результаті отримання максимального прибутку.

У сучасний час Інтернет є одним із широко використовуваних засобів інформації, то є як наслідок студія веб - дизайну має великий попит. Головну роль у створенні надаваних послуг грають веб - програміст, веб-дизайнер, і директор. Веб програміст виконує роботу по створенню, функціональності сайту, його модернізації та подальшої підтримки. Веб - дизайнер оформляє дизайн сайту, логотип. Директор відповідає за ведення договорів з клієнтом, ведення бази даних. Замовлення послуг відбувається безпосередньо через директора. Замовник може сам встановити свої вимоги, погодивши їх з веб - розробниками, які в свій час можуть запропонувати свої варіанти роботи.

1.2 Аналіз роботи студії веб дизайну.

На підставі розглянутої предметної області студії веб-дизайну можна скласти ієрархію діаграм. Інформацію представимо в графічному виді (діаграма DF).

Розробимо діаграму в якій буде представлено взаємодія зовнішніх сутностей для рішення задач (Рисунок 1).

Маємо три сутності: головна сутність - "Студія веб-дизайну" і дві зовнішніх - "замовник" і "виконавець".

1. Сутність "замовник" посилає сутності "Студія веб-дизайну" такі потоки даних, як "інформація від замовника". Сутність "Студія веб-дизайну" надсилає відповідь на запит - «інформація для замовника».

2. Сутність "виконавець" посилає сутності "Студія веб-дизайну" такі потоки даних, як "інформація від виконавця". Сутність "Студія веб-дизайну" надсилає відповідь на запит - «інформація для виконавця»

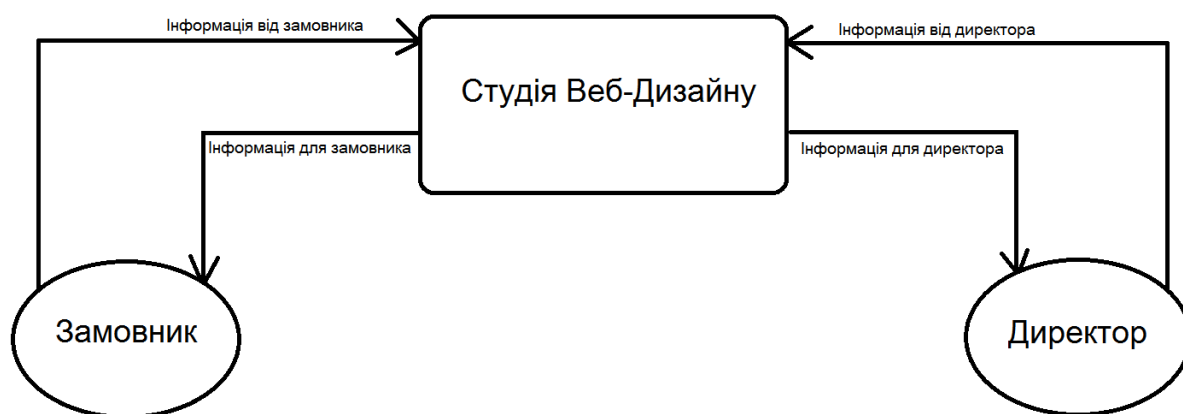


Рисунок 1 – Діаграма потоків даних.

1.3 Постановка завдання.

Досягти поставленої мети можливо шляхом автоматизації роботи, тобто створенням бази даних студії веб-дизайну, яка б зберігала повну необхідну інформацію про послуги, їх виконання.

Створення програмного забезпечення, а саме певної бази мобілізує сам процес і надасть економності часу і простоту у доступі до необхідної інформації.

У базі даних буде міститися інформація:

- 1.Відомості про студію (назва, адреса,сайт).
2. Відомості про послуги (тип послуги, ціна).
3. Відомості про працівників (ПІБ, посада, контактна інформація).

РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ.

2.1 Основні поняття концептуального проектування.

Концептуальну модель даних, що відображає узагальнене представлення про дані, незалежне від типу обраної СУБД. При цьому виділяють інфологічну, чи власне концептуальну модель, що не залежить від типу обраної СУБД узагалі, і логічну, котра відбиває логічне представлення про дані у вигляді, адекватному моделі організації даних у СУБД визначеного типу, але не залежить від конкретної СУБД (тобто це може бути будь-яка СУБД реляційного типу – Access, FoxPro, DB2 тощо).; Проектування баз даних поділяється на декілька етапів:

Ступені зв'язку між двома об'єктами позначаються у вигляді пари:



Рисунок 2.1 Один до одного (1 - 1).



Рисунок 2.2 Один до багатьох (1 – N).

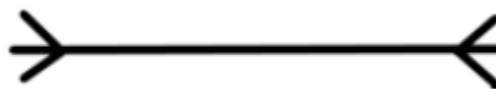


Рисунок 2.3 Багато до багатьох (M – N).

2.2 Розробка концептуальної моделі даних (ER-діаграми).

Концептуальна модель бази даних - модель, яка визначає систему основних понять і правил їх комбінування, які не залежать від засобів розробки бути смисловою структурою предметної області. Для представлення концептуальної моделі бази даних створюється діаграма «сутність-зв'язок» (ERD). Основними конструктивними елементами є сутності, зв'язки між ними та їх властивості (атрибути).

Сутність - будь-який чудовий об'єкт. Сутність володіє одним або декількома атрибутами, які або належать суті, або успадковуються через зв'язок. У даній роботі сутностями є: «Studio», «Services», «Workers», «Contact».

Останнім кроком моделювання є ідентифікація атрибутів. Атрибут - проіменована характеристика сутності. Всі атрибути позначаються через овал. Екземпляр атрибута - це певна характеристика окремого елемента множини. Екземпляр атрибута визначається типом характеристики і її значенням, названим значенням атрибута. В ER-моделі атрибути асоціюються не тільки з конкретними сутностями, але й зв'язками. Таким чином, екземпляр сутності повинен мати єдине певне значення для асоційованого атрибута. Атрибути використовуються для визначення того, яка інформація повинна бути зібрана про сутності.

Складемо діаграму сутностей – ER-діаграму для студії веб-дизайну:

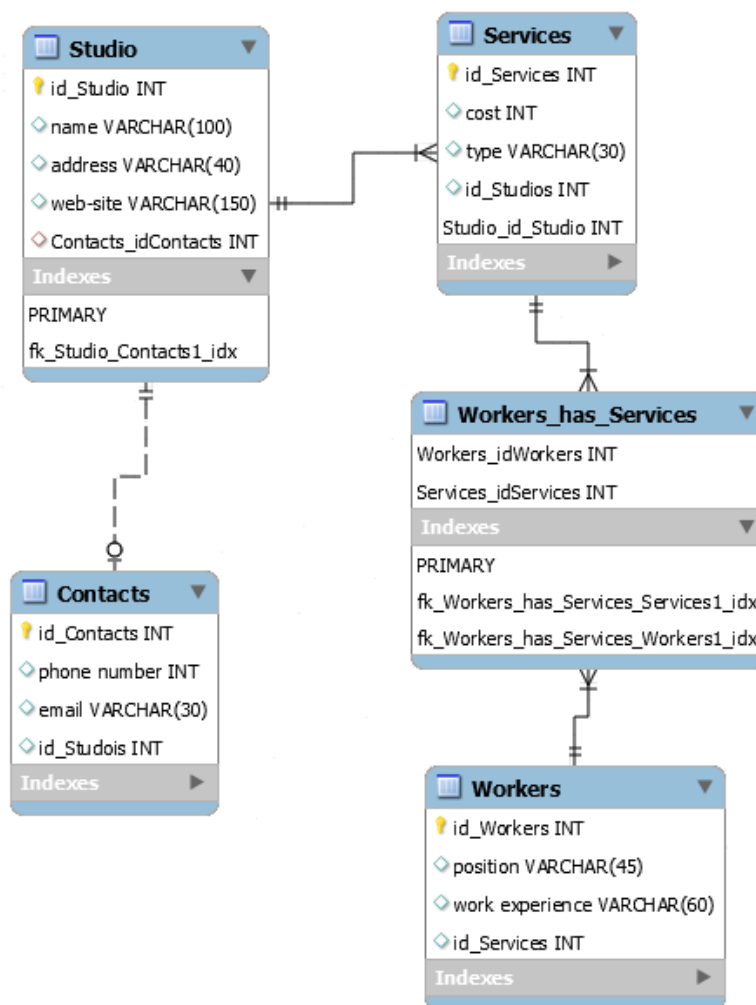


Рисунок 2.4 – ER-діаграма для студії веб-дизайну.

Наприклад, в даній роботі сутність «Studio» пов'язана з сутністю «Services» через зв'язок «1 - N» зображено на рисунку 2.5.

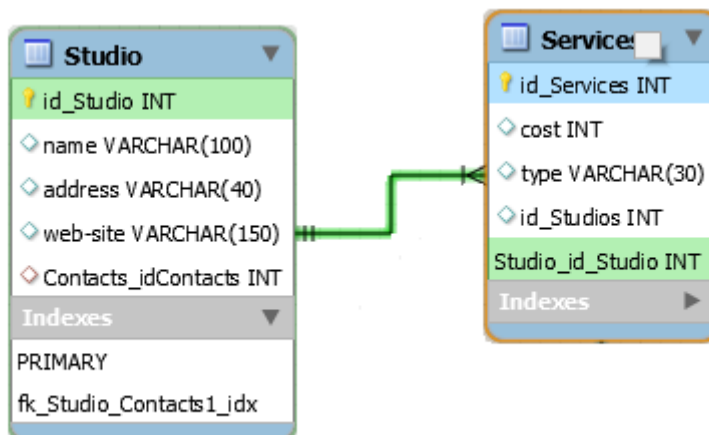


Рисунок 2.5 Сутність «Studio» пов'язана з сутністю «Services» через зв'язок «1 - N».

Сутність «Services» пов'язана з сутністю «Workers» через зв'язок «М - N» зображено на рисунку 2.6.

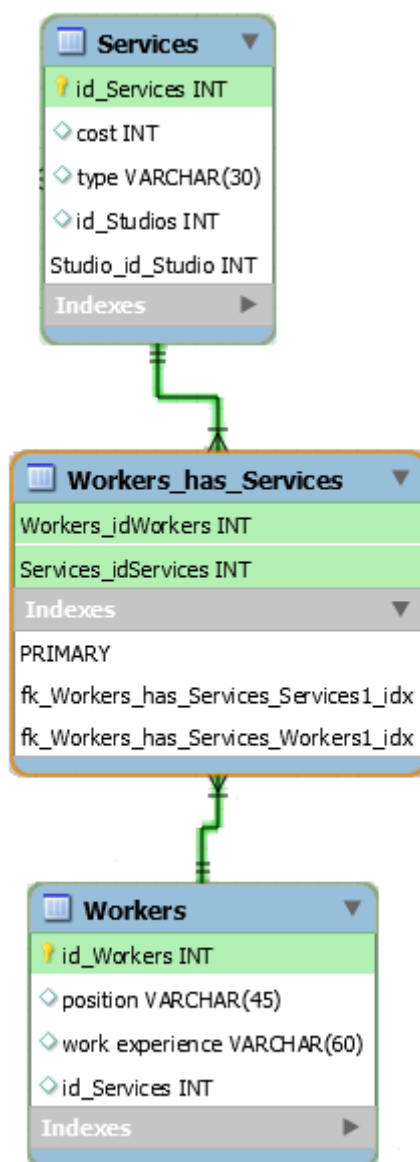


Рисунок 2.6 Сутність «Services» пов'язана з сутністю «Workers» через зв'язок «М - N».

Сутність «Studio» пов'язана з сутністю «Contact» через зв'язок «1 -1 » зображено на рисунку 2.7.

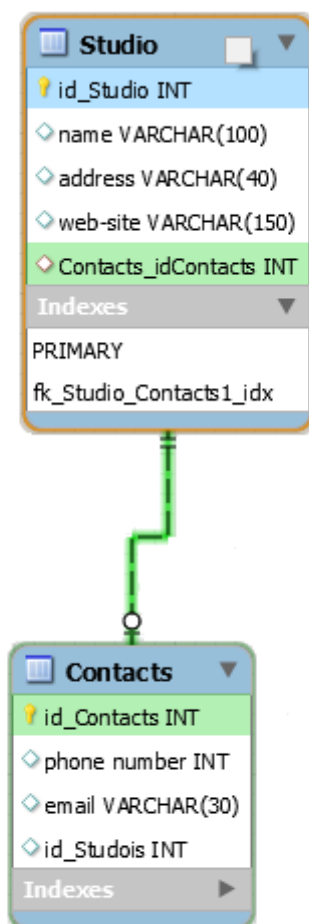


Рисунок 2.7 Сутність «Studio» пов'язана з сутністю «Contact» через зв'язок «1 -1 ».

У кожної сутності й зв'язку є свої атрибути.

2.2.1 Перелік таблиць бази даних.

База даних 'Sdata.sql' містить такі таблиці :

- 1) Studio – таблиця, у якій міститься інформація про студію;
- 2) Service – таблиця, у якій знаходяться доступні послуги для користувача;
- 3) Workers – таблиця, у яку вноситься дані про працівників.
- 4) Contact – таблиця, у якій знаходяться дані про зв'язок із студією.

2.2.2 Перелік полів таблиць бази даних.

Поля, які ідентифікують властивості таблиці «Studio»:

id_Studio - ідентифікаційний код студії;
name – ім'я студії;
adress – адреса студії;
website – веб-сайт студії;

Поля, які ідентифікують властивості таблиці «Service»:

1. id_Services – ідифікаційний код послуги;
2. cost – вартість послуги;
3. type – тип послуги;

Поля, які ідентифікують властивості таблиці «Workers»:

1. id_Workers - індифікаційний код працівників;
2. position – посада;
3. work experience – стаж роботи;

Поля, які ідентифікують властивості таблиці «Contact»:

1. id_Contacts - ідентифікаційний код студії;
2. phone_number – телефон студії;
3. email – емейл студії;

2.3 Нормалізація реляційних відношень.

Нормалізація - це розбивка таблиці на дві або більше частин, які характеризуються кращими властивостями при доповненні, зміні і вилученні даних. Кінцева мета нормалізації зводиться до отримання такого проекту бази даних, у котрому кожний факт з'являється лише в одному місці, тобто виключена надлишковість інформації. Це робиться не стільки з метою економії пам'яті, скільки для виключення можливої суперечливості збережених даних.

Перша нормальна форма (1НФ) - це звичайне відношення, на перетині рядків і стовпів якого розташоване атомарне значення відповідного атрибута, що має унікальне значення ключового атрибута, що не допускає дублювання кортежів. Таке відношення автоматично вже знаходиться в 1НФ.

Відношення знаходиться в другій нормальній формі (2НФ) тоді, коли відношення знаходиться в 1НФ і немає неключових атрибутів, залежних від частини складного ключа. Неключовий атрибут - це атрибут, який не входить до складу жодного потенційного ключа. Варто зазначити, що якщо потенційний ключ відношення є простим, то відношення автоматично перебуває під 2НФ.

Відношення знаходиться в третій нормальній формі (3НФ) тоді, коли відношення знаходиться в 2НФ і всі неключові атрибути взаємно незалежні.

Атрибути називаються взаємно незалежними, якщо жоден з них не є функціонально залежним від іншого.

Поняття функціональної залежності є базовим, так як на його основі формуються визначення всіх інших видів залежностей. Функціональна залежність - це зв'язок між атрибутами.

У відношенні R атрибут Y функціонально залежить від атрибута X (X і Y можуть бути складовими) у тому випадку, якщо кожному значенню X відповідає в точності одне значення Y.

Функціональні залежності між атрибутами в таблиці «Studio», котра містить усю інформацію про студію зображено у Таблиці 2.8. У ній зображено, що ключовим полем є поле ідентифікаційний код користувача, тобто поле «id_Studio», із значенням «Primary». Поля «name», «address», «web-site», залежні від поля «id_Studio».

Функціональні залежності між атрибутами сутності «Studio»

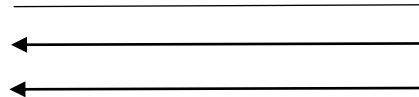
Таблиця 2.8

Найменування атрибутів	Функціональні залежності
id_Studio name address web-site	

Далі розглянемо функціональні залежності між атрибутами в таблиці «Services», у якій міститься інформація про послуги. Графічно вона представлена в Таблиці 2.9. Поля «cost», «type», залежні від поля «id».

Функціональні залежності між атрибутами сутності «Services»

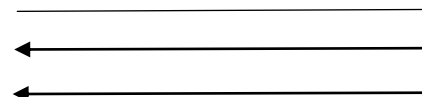
Таблиця 2.9

Найменування атрибутів	Функціональні залежності
id_Services cost type	

Функціональна залежність між атрибутами в таблиці «Workers», у якій міститься інформація про працівників, графічно відображається в Таблиці 2.10. У ній зображується, що ключовим полем є ідентифікаційний код замовлення, тобто поле «id», із значенням «Primary». Усі інші поля підпорядковуються полю.

Функціональні залежності між атрибутами сутності «Workers»

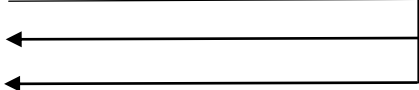
Таблиця 2.10

Найменування атрибутів	Функціональні залежності
id_Workers position work experience	

Функціональні залежності між атрибутами в таблиці «Contact», котра містить усю інформацію про зв'язок із студією зображено у Таблиці 2.11.

Функціональні залежності між атрибутами сутності «Contact»

Таблиця 2.11

Найменування атрибутів	Функціональні залежності
id_Contact phone number email	

2.4 Визначення типів даних.

Структура «SQL» в своїй структурі містить такі елементи:

- база даних;
- таблиці;
- записи таблиць.

Першим етапом є створення бази даних «SQL» з назвою «SData». Наступний етап полягає у наповненні бази даними таблицями: «Studio», «Services», «Workers», «Contact», які використовуються для функціонування сайту. Завершальним етапом є наповнення таблиць даними. Вміст таблиць продемонстрований в Таблицях 2.12-2.15.

Таблиця 2.12

Склад таблиці « Studio»

№	Найменування атрибутів	Тип полів	Розмір полів	Додаткові обмеження цілісності
1	id_Studio	int	10	Первинний ключ
2	name	varchar	100	Обов'язкове
3	address	varchar	30	Обов'язкове
4	web-site	varchar	30	Обов'язкове

Таблиця 2.13

Склад таблиці « Services »

№	Найменування атрибутів	Тип полів	Розмір полів	Додаткові обмеження цілісності
1	id_Services	int	10	Первинний ключ
2	cost	int	10	Обов'язкове
3	type	varchar	30	Обов'язкове

Таблиця 2.14

Склад таблиці « Workers »

№	Найменування атрибутів	Тип полів	Розмір полів	Додаткові обмеження цілісності
1	id_Workers	int	10	Первинний ключ
2	position	varchar	30	Обов'язкове
3	work experience	varchar	30	Обов'язкове

Таблиця 2.15

Склад таблиці « Contact »

№	Найменування атрибутів	Тип полів	Розмір полів	Додаткові обмеження цілісності
1	id_Contact	int	10	Первинний ключ
2	phone number	int	10	Обов'язкове
3	email	varchar	30	Обов'язкове

2.5 Обмеження цілісності даних.

Обмеження цілісності — це правила, які обмежують усі можливі стани бази даних, а також переходи з одного стану в інший. Таким чином, обмеження цілісності визначають множину «допустимих» станів і переходів між ними. База даних перебуває в цілісному стані, якщо вона відповідає всім визначеним для неї вимогам цілісності.

У навчальній базі “SData” застосовано наступні обмеження щодо первинних ключів:

- not null;
- primary key;
- increment;

Not null – обмеження, яке застосовується до всіх полів таблиць бази даних.

2.6. Створення бази даних.

Навчальний сайт реалізує такі запити до бази даних:

//створюємо базу даних

```
CREATE DATABASE system_for_studio;
```

// обираємо базу даних, щоб працювати з нею

```
USE system_for_studio;
```

//створюємо таблицю студії

```
CREATE TABLE Studio (
```

```
    id_Studio INT AUTO_INCREMENT PRIMARY KEY, //авто-  
    ідентифікатор, первинний ключ
```

```
    name VARCHAR(100),
```

```
    adress VARCHAR(30)
```

```
    website VARCHAR(30)); //тип для задавання тексту
```

//заповнюємо таблицю

```
INSERT INTO studio VALUES(NULL, 'Desing Studio', 'www.desing
studio.com', 'Lviv,Shevshenka 88');
```

//створюємо таблицю Services

```
CREATE TABLE Services (
id_Services INT AUTO_INCREMENT PRIMARY KEY,
cost INT,    //задаємо ціні тип INT, щоб можна було записувати
числові дані
type VARCHAR(30),
id_Studios INT,
FOREIGN KEY ( id_Studios ) REFERENCES Studio ( id_Studios ));
```

//зовнішній ключ, необхідний для того, щоб зв'язати дві таблиці

//заповнюємо таблицю

```
INSERT INTO Services VALUES(NULL, 3000, 'design', 1);
INSERT INTO Services VALUES(NULL, 11000, 'technical drawings', 1);
INSERT INTO Services VALUES(NULL, 5000, 'complete set', 1);
INSERT INTO Services VALUES(NULL, 10000, 'author supervision', 1);
INSERT INTO Services VALUES(NULL, 40000, 'construction
management', 1);
```

//створюємо таблицю Workers

```
CREATE TABLE Workers (
id_Workers INT AUTO_INCREMENT PRIMARY KEY,
position VARCHAR(30),
```

```
work experience VARCHAR(60),
id_Services INT,
FOREIGN KEY ( id_Services ) REFERENCES Services ( id_Services ));
```

//заповнюємо таблицю

```
INSERT INTO Workers VALUES(NULL, 'designer', 'years', 1);
INSERT INTO Workers VALUES(NULL, 'architect', 'two years', 2);
INSERT INTO Workers VALUES(NULL, 'architect', 'three years', 3);
INSERT INTO Workers VALUES(NULL, 'designer', 'six years', 4);
INSERT INTO Workers VALUES(NULL, 'architect', 'years', 5);
```

//створюємо таблицю Contacts

```
CREATE TABLE Contacts (
id_Contacts INT AUTO_INCREMENT PRIMARY KEY,
phone_number INT,
email VARCHAR(30),
id_Studios INT,
FOREIGN KEY ( id_Studios ) REFERENCES Studio ( id_Studios ));
```

//заповнюємо таблицю

```
INSERT INTO Contacts VALUES(NULL, 0997658120, 'desing_
studio@gmail.com', 1);
```

ВИСНОВКИ.

Під час розробки проекту бази даних студії веб-дизайну були проведені дослідження і детальний аналіз предметної області, були створені функціональні, концептуальна, логічна і фізична моделі бази даних. Також були детально проаналізовані всі елементи, з яких складається даний програмний продукт і залежності між ними. Для розробки даного програмного продукту було використано середовище розробки бази даних SQL.

Електронна база, розроблена в даній курсовій роботі, набагато підвищує ефективність роботи в процесі обліку надання послуг та роботи студії веб-дизайну, оскільки дозволяє швидко організувати пошуки необхідних послуг за їх назвами або коду, прискорився процес складання звітності. Набагато простіше й швидше Можна редагувати всю інформацію завдяки зручному інтерфейсу.

Дану базу даних можливо використовувати не лише в закладі «Студія веб-дизайну», але й інших закладах, які мають споріднену предметну область – тобто, які надають певні послуги інтернет ресурсів або будь-яких інших.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:

1. Мартин Грабер — Понимание SQL (www.sql.ru/docs/sql/u_sql/)
2. Введение в SQL (М.Грабер) /рус./ (<http://www.mysql.ru/docs/gruber/>)
3. SQL и реляционная теория. Как грамотно писать код на SQL Дейт Крис Дж. 2010.
4. «MySQL. Сборник рецептов» Поля Дюбуа 2012.
5. Create MySQL Tables URL: <https://www.tutorialspoint.com/mysql/mysql-create-tables.htm>.