

ВСТУП

В сучасному світі важко уявити будь-яку компанію, або підприємство без оптимізованої та ефективної системи контролю даних. Завдяки розвитку технологій, а також зростанню та розповсюдженню таких мереж як Internet збільшується потреба в використанні баз даних, суттєво помітно збільшення обсягів інформації, кількість одночасно під'єднаних користувачів, модернізація самої архітектури бази даних, а також розширення функціоналу. З цих факторів змінюється міркування з приводу того як створювати, будувати, розробляти а також користуватися базою даних. Бази даних мають у своєму складі логічний опис та бізнес-правила у формі сценарійних мов. Ці елементи грають суттєву роль у створенні діагностичних та активних веб-сторінок.

Для того щоб виконувати всі наші потреби існує такий інструмент СКБД (Система керування базою даних) він виконує такі задачі як: аналіз, зберігання, обробка інформації. СКБД це дуже могутнє та злагоджене інтеграційне оточення, яке для своєї роботи потребує дуже значної кількості ресурсів комп'ютера, або навіть цілої мережі комп'ютерів, а також дуже кваліфікованих спеціалістів. До моменту написання курсової роботи у світі існує безліч СКБД які задовільняють потреби корпорацій. Незважаючи на їх кількість, вони майже всі виконують одноманітні вимоги, такі як: мультиплатформеність; можливість швидкого відновлення втрачених даних, безперебійний доступ та обробку даних, надійну цілісність даних та систему безпеки, підтримку різних мов програмування, тощо.

Зі швидким темпом розвитку бізнесу та інформації СКБД постійно потребує вдосконалення та змін. СКБД MySQL має велику відомість серед розробників, а також має більш значні переваги ніж конкуренти. Це найкращий вибір для керування, масштабування та підтримки різних інформаційних середовищ.

					КП.ІПЗ-321.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

1 ЕТАПИ ПРОЕКТУВАННЯ БД. МОДЕЛЬ БД. ER-ДІАГРАМИ

1.1 РБД (Реляційна база даних)

База даних (БД) - це систематизоване зібрання інформації про об'єкти, події або явища, що відносяться до певної предметної області, організоване за встановленими правилами і збережене в пам'яті комп'ютера. Мета бази даних - задовольнити інформаційні потреби користувачів та ефективно зберігати ці дані.

Реляційна база даних складається з взаємопов'язаних таблиць, кожна з яких містить інформацію про об'єкти певного типу. Кожен рядок таблиці містить дані про об'єкт, а стовпці таблиці містять різні характеристики цих об'єктів, відомі як атрибути.

Для ідентифікації записів використовується первинний ключ - унікальний набір полів у таблиці, який однозначно ідентифікує кожен запис. Системи керування базами даних (СКБД) визначають структуру бази даних, обробляють дані та забезпечують управління даними.

Структура бази даних включає в себе визначення таблиць і зв'язків між ними. Розробка такої структури є фундаментальним завданням при проектуванні бази даних. Будь-яка СУБД дозволяє виконувати операції над даними, такі як додавання, видалення та оновлення записів, а також пошук за певними умовами, використовуючи мову структурованих запитів (SQL).

Управління даними включає захист від несанкціонованого доступу, підтримку декількох користувачів і забезпечення цілісності та узгодженості даних.

					КП.ІПЗ-321.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

1.2 Проектування РБД

Етапи проектування реляційної бази даних визначають процес розробки ефективної структури для зберігання та управління даними. Цей процес дуже складний, оскільки об'єкти реального світу та їхні взаємозв'язки не завжди вписуються в реляційну модель даних. Розробник повинен перетворити складні реальні структури в абстрактні концепції таблиць, полів, атрибутів і записів.

На першому етапі, аналізі вимог, розробник повинен визначити, який тип даних буде зберігатися і як вони будуть використовуватися. Це включає в себе структуру даних та їх взаємозв'язки, а також питання про те, як користувачі будуть взаємодіяти з даними.

На другому етапі створюється логічна структура бази даних. На цьому етапі визначається, як дані повинні бути логічно згруповані, використовуючи терміни об'єкти програми та їхні зв'язки. Дизайнер візуалізує, як об'єкти взаємодіють один з одним на рівні даних.

На заключному етапі логічна структура трансформується у фізичну структуру з урахуванням аспектів продуктивності. Дизайнер визначає, як концепції та взаємозв'язки в таблицях, атрибутах і стовпцях будуть реалізовані в системі управління базами даних (СУБД), обраній для реалізації бази даних. Це також включає оптимізацію для швидкого та ефективного доступу до даних. Таким чином, на етапі проектування визначається оптимальна структура, яка забезпечить ефективне управління та використання даних у реляційній базі даних.

1.2.1 Специфікація вимог

Як правило, вимоги до додатків баз даних визначаються в результаті досліджень і обговорень з кінцевими користувачами. Це ітеративний процес, в якому розробники поступово визначають структуру діалогу, документують критерії пошуку та визначають можливі реакції користувачів.

					КП.ІПЗ-321.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

Одним з найважливіших кроків у визначенні та документуванні вимог до бази даних є створення словника даних. Словник даних перераховує і визначає окремі елементи даних, які повинні зберігатися в базі даних. Створення словника даних - це ефективний спосіб розпочати визначення вимог до бази даних.

Однак одного словника даних майже у всіх випадках, завжди не достатньо для визначення структури бази даних, оскільки він не визначає взаємозв'язки між елементами, способи створення, оновлення та відбору даних, а також способи використання баз даних. Тому необхідна функціональна специфікація, яка включає інформацію про кількість одночасних користувачів, частоту введення та оновлення записів, а також спосіб відбору інформації з бази даних.

Функціональні специфікації та словники даних зазвичай розробляються паралельно. Це пов'язано з тим, що ці документи доповнюють один одного і дозволяють отримати більш повне уявлення про систему. Важливою частиною аналізу вимог є передбачення потреб користувачів. Це пов'язано з тим, що користувачі не завжди мають повних і чітко визначених вимог до проекту. Функціональні описи призначені для того, щоб представити систему якнайповніше і найфункціональніше, враховуючи всі аспекти роботи системи.

1.2.2 Моделювання. ER-діаграма

Моделювання баз даних та ER-діаграми є важливою частиною процесу розробки баз даних. Ця методологія, відома як моделювання "сутність-зв'язок" (ER), дозволяє описати структуру бази даних в термінах сутностей, їхніх зв'язків та атрибутів.

ER-діаграма допомагає зрозуміти структуру бази даних, визначає взаємозв'язки між різними частинами системи і служить основою для створення схеми бази даних в технічному сенсі. Вона полегшує комунікацію між розробниками та зацікавленими сторонами і визначає основні елементи, які слід враховувати при створенні та управлінні базою даних.

					КП.ІПЗ-321.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

1.2.3 Елементи БД: Об'єкти, атрибути та ідентифікатори

У цьому розділі було більш детально розглянуто структуру бази даних і зосереджуся на об'єктах, атрибутах і ключах, які є основними елементами реляційної моделі.

Об'єкти та атрибути:

Починаючи з визначення об'єктів бази даних, розглядаємо кожен об'єкт як сутність, яка має певні властивості. Атрибути, в свою чергу, представляють характеристики або елементи даних, які визначають об'єкт. Аналізуючи словник даних, визначаємо об'єкти та їхні атрибути і розширюємо словник за потреби.

Реляційна модель та зв'язки:

Реляційна модель бази даних використовує відношення і ключі для встановлення зв'язків між об'єктами. Терміни "відношення" і "зв'язок" мають важливу відмінність в контексті реляційних баз даних. Відношення - це неупорядковані двовимірні таблиці, а схеми даних формуються між відношеннями за допомогою спільних атрибутів, які виступають в ролі ключів.

Типи ключів:

У реляційній моделі існує кілька різних типів ключів. Первинний ключ однозначно ідентифікує кожен рядок у таблиці, і кожна таблиця може мати лише один ключ. Якщо для ідентифікації рядка потрібно використовувати більше одного атрибута, створюється складений ключ.

Зовнішні ключі існують лише у зв'язках між двома таблицями. Це атрибут, який діє як первинний ключ в іншій таблиці і створює зв'язок між ними.

У прикладі з поштою унікальний ідентифікатор пошти (код пошти) може бути використаний як первинний ключ для відношення. Крім того, зовнішні ключі можна використовувати для зв'язку з іншими таблицями, що полегшує організацію зв'язків у базі даних.

Завдяки такому підходу можна створити добре структуровану базу даних з чітко визначеними об'єктами, атрибутами і ключами, що полегшує ефективне

					КП.ІПЗ-321.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

управління даними і виконання запитів.

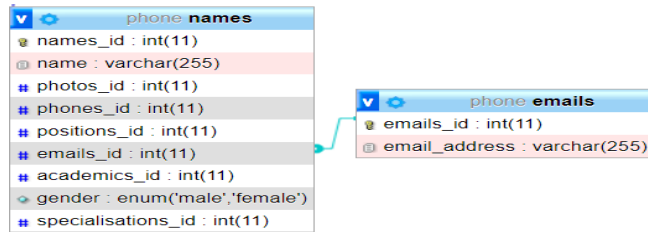


Рисунок 1.1 - Приклад зовнішнього ключа

1.2.4 Нормалізація таблиць: Розкриття функціональних залежностей

Нормалізація таблиць означає виявлення функціональних залежностей і впорядкування бази даних відповідно до певної нормальної форми (НФ).

Функціональна залежність:

Функціональна залежність виникає тоді, коли значення одного атрибута можна визначити за значенням іншого атрибута. У такому випадку атрибут, який можна визначити, називається функціонально залежним атрибутом, а атрибут, який залежить від нього, називається специфікатором. Неключовий атрибут відношення є функціонально залежним від первинного ключа, оскільки первинний ключ однозначно ідентифікує кожен рядок.

Типи залежностей:

- Багатозначна залежність: коли один атрибут не ідентифікує однозначно інший атрибут, а обмежується набором можливих значень.
- Часткова залежність: Вказує на те, що атрибут зв'язку функціонально залежить лише від частини ключа.
- Транзитивна залежність: Виникає, коли атрибут неключового зв'язку функціонально залежить від іншого атрибута неключового зв'язку у відношенні.

Нормальна форма (НФ):

- Перша нормальна форма (1НФ): Всі таблиці повинні мати атомарні значення для кожного поля, тобто значення не повинні містити ніякої внутрішньої структури.

- Друга нормальна форма (2НФ): Кожен непервинний атрибут повністю залежить від первинного ключа.
- Третя нормальна форма (3НФ): Кожен непервинний атрибут тимчасово не залежить від первинного ключа.
- Нормальна форма Бойса-Кодда (BCNF): Всі домени є потенційними ключами, тобто не існує транзитивної залежності первинного атрибуту від непервинного атрибуту.
- Четверта нормальна форма (4НФ): Відсутня залежність з деякими нефункціональними значеннями.
- П'ята нормальна форма (5НФ): Визначає умови, за яких початковий зв'язок може бути відновлений без втрати інформації за допомогою можливих ключів.

Процес нормалізації дозволяє створювати базу даних відповідно до визначеної нормальної форми для забезпечення ефективного зберігання та обробки даних.

1.2.5 Зв'язки в БД

Зв'язки виникають через взаємодію між сутностями в базі даних. Це можуть бути зв'язки типу "є" (наприклад, між "Номер телефону" і "посадою") або інші логічні зв'язки.

Типи зв'язків:

- Один до одного (1:1): кожен елемент у відношенні пов'язаний лише з одним елементом в іншому відношенні, і навпаки.
- Один до багатьох (1:N): відношення може мати багато елементів, пов'язаних з одним елементом іншого відношення, але кожен елемент другого відношення пов'язаний лише з одним елементом першого.
- Багато до багатьох (M:N): кожен елемент зв'язку може мати багато елементів, пов'язаних з іншими зв'язками, і навпаки.

1.3 Тригери: Забезпечення Цілісності та Безпеки Даних

Тригер можна розглядати як автоматичну реакцію бази даних на певні події або дії. Вони використовуються для забезпечення цілісності (коректності) та безпеки даних.

Типи тригерів:

- DML-тригери: реагують на події обробки даних (INSERT, UPDATE, DELETE). Вони можуть виконуватися до або після визначених дій.
- DDL-тригери: спрацьовують при виконанні команд, пов'язаних зі зміною структури бази даних (CREATE, ALTER, DROP, GRANT, DENY, REVOKE, UPDATE STATISTICS).
- LOGON: Активуються при встановленні з'єднання з сервером бази даних (MS SQL Server).

Тригери використовуються для визначення умов, які повинні бути виконані до або після додавання, оновлення, видалення даних, щоб забезпечити їх коректність, а також для обмеження доступу до бази даних і контролю змін, щоб гарантувати, що незаконні дії не допускаються.

Таким чином, тригери є потужним інструментом для автоматизації та моніторингу подій і дій у базі даних для забезпечення її цілісності та безпеки.

					КП.ІПЗ-321.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

2 СХЕМА ТА ОПИС БАЗИ ДАНИХ

2.1 Призначення

Телефонний довідник це один з найважливіших інструментів при адмініструванні системи вищого навчального закладу. Цей діловий набір паперу містить дуже інформативну та корисну інформацію про інститути, факультети, спеціальності, викладачів, кабінети, пошти, посади та інше.

Телефонний довідник переважно використовують для швидкого зв'язку з викладачем для вирішення різних типів питань. Ціллю даної розробки є створення автоматизованої системи оснований на базі даних довідника для керування списками викладачів, прискорення пошуку контактів, пошуку точної локації кабінета з цим викладачем, а також для більш детальнішого ознайомлення з проектами, роботами, життям працівників закладу.

Система повинна мати такі функції:

1. зберігати персональні дані;
2. вести облік спеціальностей, кабінетів, факультетів, інститутів;
3. надавати доступ до інформації про кожного окремого викладача.

2.2 Логічна модель

Модель розробленої бази даних будуємо наступним чином:

1. Таблиця “institutions” містить такі поля: ідентифікатор, назва інституту.



institutions	
institutions_name	string
institutions_id	int

Рисунок 2.1 - таблиця інститути

2. Таблиця “faculties” містить такі поля: ідентифікатор(інституту), ідентифікатор(факультету), ідентифікатор, назву факультету, спеціальність.

faculties	
institutions_id	int
faculties_id	int
faculties_name	string
specialisations_id	int
specialisations_name	string

Рисунок 2.2 - таблиця факультетів

3. Таблиця “hull” містить такі поля: ідентифікатор (спеціальності), вебсайт, кабінет.

hull	
adress	varchar(32)
website_and_social_networks	varchar(40)
specialisations_id	int

Рисунок 2.3 - таблиця корпусів

4. Таблиця “Name” (найважливіша) містить такі поля: ідентифікатори (ім’я, фото, телефону, пошти, посада, фото, проектів, спеціальності), гендер, ім’я.

Name	
Name_id	int
Name	string
Photo_id	int
Phone_id	int
position_id	int
email_id	int
Academic_degrees_id	int
projects_and_publications_id	int
gender	ger
specialisations_id	int

Рисунок 2.4 - таблиця ім’я

5. Таблиці “phone”; “job_position”; email; “academic_data”; “photo”; мають схожу архітектуру: ідентифікатори (фото, пошти, телефону, публікації, посади) назва посади, пошта, телефон, назва публікації, фото.



Рисунок 2.5 – фінальні таблиці

2.3 Зміст полів

1. Таблиця institutions – інститути

institutions_name – назва інституту

institutions_id [pk] – ідентифікатор інституту

2. Таблиця faculties – факультети

institutions_id [pk] – ідентифікатор інституту

faculties_id [pk] – ідентифікатор факультету

faculties_name – назва факультету

specialisations_id [pk] – ідентифікатор спеціальності

specialisations_name – назва спеціальності

3. Таблиця hull – корпуси

adress - кабінет

website_and_social_networks - вебсайт

specialisations_id [pk] - ідентифікатор спеціальності

4. Таблиця Name – ім'я

Name_id [pk] – ідентифікатор імені

Name – ім'я

Photo_id – ідентифікатор фото

Phone_id – ідентифікатор телефону

position_id – ідентифікатор посади

email_id – ідентифікатор пошти

projects_and_publications_id – ідентифікатор проектів

gender – гендер (чоловік/жінка)

specialisations_id [pk] - ідентифікатор спеціальності

5. Таблиця Photo – фото

Photo_id [pk] - ідентифікатор фото

Photo img – фото

6. Таблиця Phone – телефон

Phone_id [pk] – ідентифікатор телефону

Phone – телефон

7. Таблиця job_position – посада

position_id [pk] – ідентифікатор посади

position – назва посади

8. Таблиця email – пошти

email_id int [pk] – ідентифікатор пошти

email – пошта

9. Таблиця Academic_data – проекти

projects_and_publications_id [pk] - ідентифікатор проектів

projects_and_publications string – назва проекту

Створена модель системи телефонний довідник має таблиці для зберігання даних про всі необхідні об'єкти та їхні характеристики з метою знаходження контактної інформації.

					КП.ІПЗ-321.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

3 РЕАЛІЗАЦІЯ БАЗИ ДАНИХ

3.1 Створення Баз Даних

Перед початком роботи завантажуюємо додаток МАРМ для того щоб запустити свій власний локальний сервер.

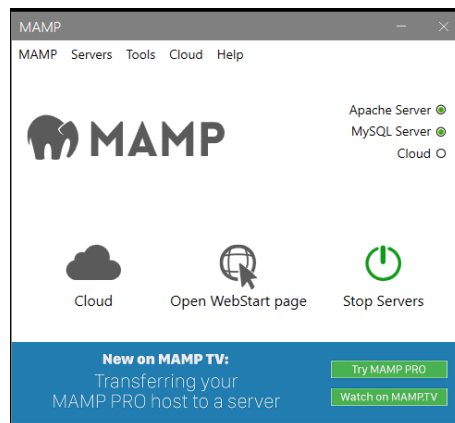


Рисунок 3.1 - Інтерфейс додатку МАРМ

Далі після розгортання серверу переходимо на початкову веб-сторінку phpMyAdmin, налаштовуючи мову, тему та ще деякі конфігурації для власного комфорту і потрапляємо на сторінку створення бази даних та таблиць.

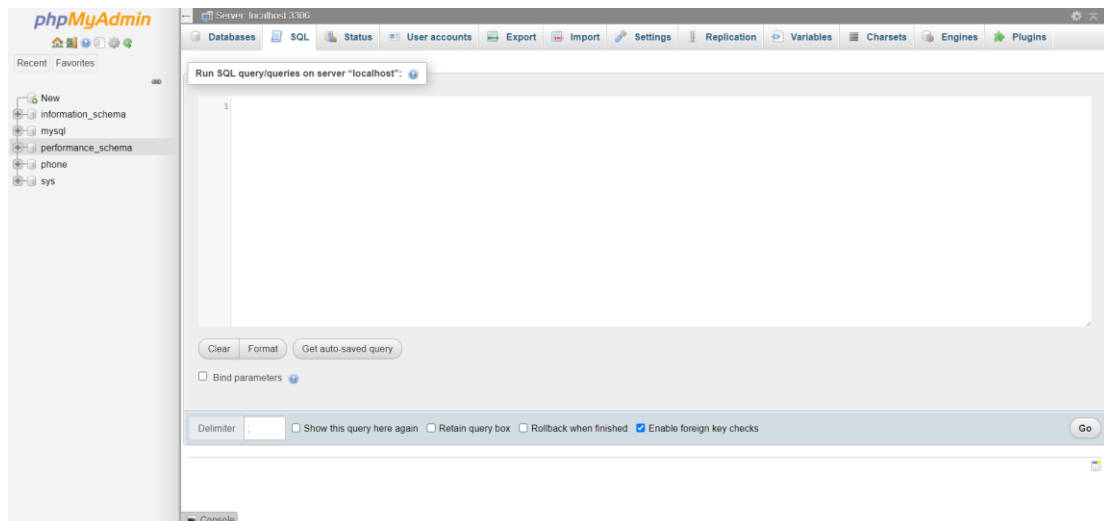


Рисунок 3.2 - Початкова сторінка створення БД

					КП.ІПЗ-321.ПЗ	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

В наше розпорядження потрапляє весь функціонал веб-додатку. Створення таблиці через SQL та через графічний інтерфейс, налаштування та конфігурації, експорт та імпорт бази даних, бібліотека плагінів, консоль, тощо. Обираємо пункт SQL та вводимо команду:

```
CREATE DATABASE phone
```

Ми створили пусту базу даних яка не має таблиць.

3.2 Створення таблиць

Увесь код буде демонструватися одразу зі всіма модифікаціями.

```
CREATE TABLE `institutions` (
  `institutions_id` int(11) NOT NULL,
  `institutions_name` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE `institutions`
  ADD PRIMARY KEY (`institutions_id`);
ALTER TABLE `institutions`
  MODIFY `institutions_id` int(11) NOT NULL AUTO_INCREMENT;
COMMIT;

CREATE TABLE `faculties` (
  `faculties_id` int(11) NOT NULL,
  `faculties_name` varchar(128) NOT NULL,
  `institutions_id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE `faculties`
  ADD PRIMARY KEY (`faculties_id`),
  ADD KEY `institutions_id` (`institutions_id`);
ALTER TABLE `faculties`
  ADD CONSTRAINT `faculties_ibfk_2` FOREIGN KEY (`institutions_id`)
REFERENCES `institutions` (`institutions_id`);
COMMIT;

CREATE TABLE `specialisations` (
  `specialisations_id` int(11) NOT NULL,
  `specialisations_name` varchar(255) NOT NULL,
  `faculties_id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE `specialisations`
  ADD PRIMARY KEY (`specialisations_id`),
  ADD KEY `p_department_id` (`faculties_id`);
ALTER TABLE `specialisations`
  ADD CONSTRAINT `specialisations_ibfk_1` FOREIGN KEY
(`faculties_id`) REFERENCES `faculties` (`faculties_id`);
COMMIT;
```

```

CREATE TABLE `hulls` (
  `specialisations_id` int(11) DEFAULT NULL,
  `address` varchar(255) NOT NULL,
  `website` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE `hulls`
ADD UNIQUE KEY `department_id` (`specialisations_id`);
ALTER TABLE `hulls`
  ADD CONSTRAINT `hulls_ibfk_1` FOREIGN KEY (`specialisations_id`)
REFERENCES `specialisations` (`specialisations_id`);
COMMIT;

CREATE TABLE `names` (
  `names_id` int(11) NOT NULL,
  `name` varchar(255) NOT NULL,
  `photos_id` int(11) DEFAULT NULL,
  `phones_id` int(11) DEFAULT NULL,
  `positions_id` int(11) DEFAULT NULL,
  `emails_id` int(11) DEFAULT NULL,
  `academics_id` int(11) DEFAULT NULL,
  `gender` enum('male','female') NOT NULL,
  `specialisations_id` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE `names`
  ADD PRIMARY KEY (`names_id`),
  ADD KEY `department_id` (`specialisations_id`),
  ADD KEY `phones_id` (`phones_id`),
  ADD KEY `photos_id` (`photos_id`),
  ADD KEY `positions_id` (`positions_id`),
  ADD KEY `emails_id` (`emails_id`),
  ADD KEY `academics_id` (`academics_id`);
ALTER TABLE `names`
  MODIFY `names_id` int(11) NOT NULL AUTO_INCREMENT;
ALTER TABLE `names`
  ADD CONSTRAINT `names_ibfk_1` FOREIGN KEY (`specialisations_id`)
REFERENCES `specialisations` (`specialisations_id`),
  ADD CONSTRAINT `names_ibfk_2` FOREIGN KEY (`phones_id`) REFERENCES
`phones` (`phones_id`),
  ADD CONSTRAINT `names_ibfk_3` FOREIGN KEY (`photos_id`) REFERENCES
`photos` (`photos_id`),
  ADD CONSTRAINT `names_ibfk_5` FOREIGN KEY (`emails_id`) REFERENCES
`emails` (`emails_id`),
  ADD CONSTRAINT `names_ibfk_6` FOREIGN KEY (`academics_id`)
REFERENCES `academics` (`academics_id`);

```



```

CREATE TABLE `academics` (
  `academics_id` int(11) NOT NULL,
  `projects_and_publications` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
CREATE TABLE `emails` (
  `emails_id` int(11) NOT NULL,
  `email_address` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
CREATE TABLE `phones` (
  `phones_id` int(11) NOT NULL,
  `phone_number` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
CREATE TABLE `photos` (
  `photos_id` int(11) NOT NULL,
  `photo_url` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
CREATE TABLE `positions` (
  `positions_id` int(11) NOT NULL,
  `position_name` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE `academics`
  ADD PRIMARY KEY (`academics_id`);
ALTER TABLE `emails`
  ADD PRIMARY KEY (`emails_id`);
ALTER TABLE `phones`
  ADD PRIMARY KEY (`phones_id`);
ALTER TABLE `photos`
  ADD PRIMARY KEY (`photos_id`);
ALTER TABLE `positions`
  ADD PRIMARY KEY (`positions_id`);
ALTER TABLE `academics`
  MODIFY `academics_id` int(11) NOT NULL AUTO_INCREMENT;
ALTER TABLE `emails`
  MODIFY `emails_id` int(11) NOT NULL AUTO_INCREMENT;
ALTER TABLE `phones`
  MODIFY `phones_id` int(11) NOT NULL AUTO_INCREMENT;
ALTER TABLE `photos`
  MODIFY `photos_id` int(11) NOT NULL AUTO_INCREMENT;
ALTER TABLE `positions`
  MODIFY `positions_id` int(11) NOT NULL AUTO_INCREMENT;
ALTER TABLE `positions`
  ADD CONSTRAINT `positions_ibfk_1` FOREIGN KEY (`positions_id`)
REFERENCES `names` (`positions_id`);
COMMIT;

```

Також до файлу курсового проекту буде прикріплений файл з базою даних *Base.sql*

					КП.ІПЗ-321.ІЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

3.3 Наповнення таблиць. Імпорт даних

Будь-яка база даних без даних немає змісту, саме тому нам необхідно наповнити її для того щоб провести базове тестування нашої системи та виконати запити. Для конкретного прикладу були використані абсолютно випадкові дані, і будь-яке співпадіння з реальними факультетами, інститутами чи особами є випадковим. Об'єм даних під час написання практичної частини значно збільшився в обсягах, саме тому файл буде прикріплений окремо під назвою *Data.sql*.

3.4 Порівняння вимог з готовим проектом

Отримана ER-діаграма, успішно побудована у phpMyAdmin, є втіленням наших зусиль протягом усього семестру. Ця діаграма показує структуру та взаємозв'язки між таблицями бази даних. В процесі побудови було враховано важливі аспекти проекту, внесено необхідні зміни та оптимізації для досягнення оптимального вигляду та функціональності.

Кожен елемент діаграми представляє певну таблицю в базі даних, а лінії та зв'язки між ними ілюструють взаємозв'язки між даними. Цей візуальний інструмент допомагає нам легше зрозуміти структуру нашої бази даних і взаємозв'язки між різними елементами, що сприяє більш ефективному розумінню та управлінню нашим проектом.



Рисунок 3.3 - ER-діаграма (прототип бази даних)

Отримана ER-діаграма, побудована в phpMyAdmin після написання коду для бази даних, є визначальним етапом нашої роботи під час виконання курсової роботи. Ця діаграма показує не тільки структуру бази даних, але й зв'язки між різними сутностями та таблицями:

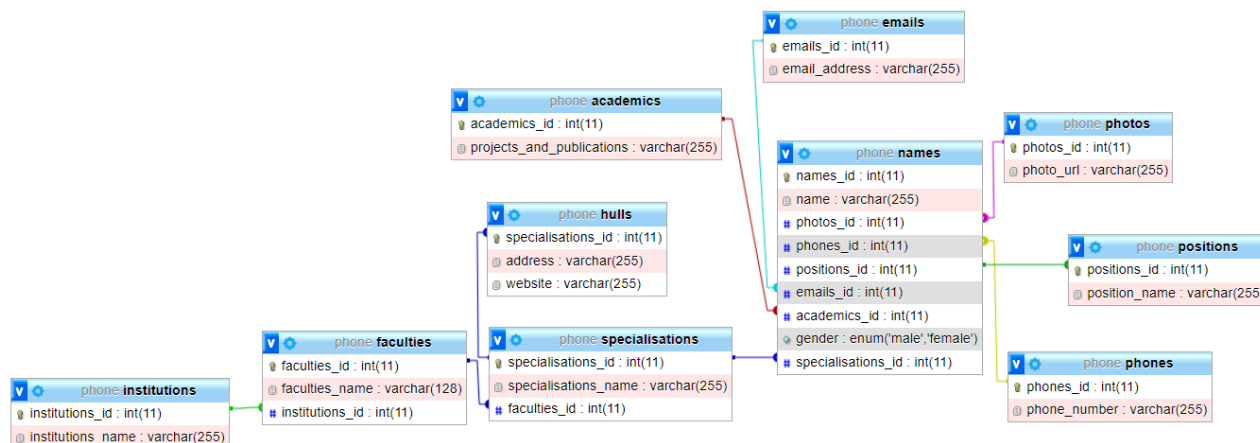


Рисунок 3.4 - ER-діаграма (готова база даних в phpMyAdmin)

В процесі розробки бази даних автор намагався максимально наблизитися до концепції прототипу. Незважаючи на деякі відмінності, які виникли через мої індивідуальні підходи та досвід у проектуванні, вважаю, що мені успішно вдалося передати основну структуру та архітектуру бази даних.

Під час розробки було враховано власний досвід, та на накопичені знання в галузі проектування баз даних. Навіть незважаючи на труднощі відтворення ідеального прототипу, було визнано, що мені вдалося успішно передати ключові елементи та концепції, які забезпечують ефективну роботу бази даних в рамках визначених цілей курсової роботи.

Важливою також є ідея готовності адаптувати архітектурні рішення до конкретних вимог і контексту проекту. Це визнає необхідність досягнення оптимального балансу між ідеалом і практикою, і можна сказати впевнено, що отримана база даних відповідає основним вимогам і сприятиме успішному розвитку як майбутнього спеціаліста.

3.5 Запити

Спираючись на вже розроблену та готову базу даних, було придумано 10 запитів, кожен з яких спрямований на вирішення конкретного завдання та оптимізацію взаємодії з даними. Давайте розглянемо кожен запит та їхнє призначення:

Запит 1: Вивести назви факультетів які належать інститутам під номерами 1,3,5.

```
SELECT faculties.*
FROM faculties
JOIN institutions ON faculties.institutions_id = institutions.institutions_id
WHERE institutions.institutions_id IN (1, 3, 5);
```

faculties_id	faculties_name	institutions_id
1	Інформаційні технології	1
2	Математика та комп'ютерні науки	1
3	Біологія та екологія	1
4	Мовознавство та література	1
5	Філософія та соціальні науки	1
11	Економіка та бізнес	3
12	Право та юриспруденція	3
13	Медицина та біомедичні науки	3
14	Соціальна робота та соціальна політика	3
15	Інженерія та технології	3
21	Соціологія та політологія	5
22	Інформаційна безпека та кібербезпека	5
23	Фізика та астрономія	5
24	Літературознавство та культурологія	5
25	Психологія мистецтва	5

Рисунок 3.5 - Результат першого запиту

Запит 2: Вивести тільки жінок які належать інститутам під номерами (2,4).

```
SELECT names.name, names.gender, specialisations.specialisations_name,
faculties.faculties_name, institutions.institutions_name
FROM names
JOIN academics ON names.academics_id = academics.academics_id
JOIN positions ON names.positions_id = positions.positions_id
JOIN emails ON names.emails_id = emails.emails_id
JOIN phones ON names.phones_id = phones.phones_id
JOIN photos ON names.photos_id = photos.photos_id
JOIN specialisations ON names.specialisations_id =
specialisations.specialisations_id
JOIN faculties ON specialisations.faculties_id = faculties.faculties_id
JOIN institutions ON faculties.institutions_id = institutions.institutions_id
WHERE institutions.institutions_id IN (2, 4)
AND names.gender = 'female';
```

name	gender	specialisations_name	faculties_name	institutions_name
Нестеренко Лідія Вікторівна	female	Споживчі технології	Фізична культура	Фізичного виховання
Павленко Наталія Олександрівна	female	Економіка туризму	Фізична культура	Фізичного виховання
Федорова Катерина Миколаївна	female	Культурологічні дослідження	Журналістика та масові комунікації	Фізичного виховання
Цветкова Ірина Віталіївна	female	Психотерапія	Мистецтво та дизайн	Фізичного виховання
Шевченко Олена Василівна	female	Спеціальна педагогіка	Мистецтво та дизайн	Фізичного виховання
Юрченко Оксана Ігорівна	female	Фінансовий менеджмент	Психологія та педагогіка	Фізичного виховання
Кузнєцова Катерина Станіславівна	female	Електроенергетика	Архітектура та дизайн	Інженерно-технічний інститут
Білоусова Лілія Михайлівна	female	Лінгвістична антропологія	Хімія та матеріалознавство	Інженерно-технічний інститут
Соколова Анастасія Василівна	female	Теорія перекладу	Хімія та матеріалознавство	Інженерно-технічний інститут
Литвинова Вікторія Віталіївна	female	Екологічний моніторинг	Енергетика та відновлювані джерела енергії	Інженерно-технічний інститут
Бондаренко Єлизавета Олегівна	female	Політична соціологія	Мовознавство та міжкультурна комунікація	Інженерно-технічний інститут
Соловйова Аліна Василівна	female	Інформаційна аналітика	Географія та екологія	Інженерно-технічний інститут

Рисунок 3.6 - Результат другого запиту

Запит 3: Вивести номери телефонів які закінчуються на цифру 1.

```
SELECT phone_number
FROM phones
WHERE RIGHT(phone_number, 1) = '1';
```

	phone_number
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	0666535891
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	0977654321
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	0987654321
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	0951212121
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	0987654321
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	0967654321
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	0910000111
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	0931234321
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	0921212121
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	0944444111
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	0990000111
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	0935555111
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	0941234321
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	0993210321
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	0984444111

Рисунок 3.7 - Результат третього запиту

Запит 4: Підрахувати загальну суму всіх id в таблиці names.

```
SELECT SUM(names_id) AS total_names_id
FROM names;
```

total_names_id
2850

Рисунок 3.8 - Результат четвертого запиту

Запит 5: Вивести інформацію про викладачів які займають посади (5,6,7).

```
SELECT
    names.name,
    names.gender,
    specialisations.specialisations_name,
    faculties.faculties_name,
    institutions.institutions_name
FROM names
JOIN academics ON names.academics_id = academics.academics_id
JOIN positions ON names.positions_id = positions.positions_id
JOIN emails ON names.emails_id = emails.emails_id
JOIN phones ON names.phones_id = phones.phones_id
JOIN photos ON names.photos_id = photos.photos_id
JOIN specialisations ON names.specialisations_id =
specialisations.specialisations_id
JOIN faculties ON specialisations.faculties_id = faculties.faculties_id
```

```
JOIN institutions ON faculties.institutions_id = institutions.institutions_id
WHERE positions.positions_id IN (5, 6, 7);
```

name	gender	specialisations_name	faculties_name	institutions_name
Маркін Дмитро Станіславович	male	Інформаційні системи	Соціальна робота та соціальна політика	Інститут мистецтв
Захарова Ксенія Миколаївна	female	Електротехніка	Соціальна робота та соціальна політика	Інститут мистецтв
Гусева Софія Юріївна	female	Органічна хімія	Інженерія та технології	Інститут мистецтв
Козирева Анастасія Володимирівна	female	Фізична хімія	Інженерія та технології	Інститут мистецтв
Ігнат'єва Дарина Олександрівна	female	Аналітична хімія	Інженерія та технології	Інститут мистецтв
Тарасенко Іван Сергійович	male	Відновлювана енергія	Архітектура та дизайн	Інженерно-технічний інститут
Калашніков Василь Ігорович	male	Енергозбереження	Архітектура та дизайн	Інженерно-технічний інститут
Кузнецова Катерина Станіславівна	female	Електроенергетика	Архітектура та дизайн	Інженерно-технічний інститут
Білоусова Лілія Михайлівна	female	Лінгвістична антропологія	Хімія та матеріалознавство	Інженерно-технічний інститут
Мельничук Валентин Сергійович	male	Мовознавство та глобалізація	Хімія та матеріалознавство	Інженерно-технічний інститут
Соколова Анастасія Василівна	female	Теорія перекладу	Хімія та матеріалознавство	Інженерно-технічний інститут
Коваль Євген Віталійович	male	Географія та геоінформаційні технології	Енергетика та відновлювані джерела енергії	Інженерно-технічний інститут
Олійник Олег Олександрович	male	Охорона природи	Енергетика та відновлювані джерела енергії	Інженерно-технічний інститут
Литвинова Вікторія Віталіївна	female	Екологічний моніторинг	Енергетика та відновлювані джерела енергії	Інженерно-технічний інститут
Іванов Станіслав Віталійович	male	Соціологічні дослідження	Мовознавство та міжкультурна комунікація	Інженерно-технічний інститут
Климов Олексій Сергійович	male	Гендерна соціологія	Мовознавство та міжкультурна комунікація	Інженерно-технічний інститут
Бондаренко Єлизавета Олегівна	female	Політична соціологія	Мовознавство та міжкультурна комунікація	Інженерно-технічний інститут
Солопова Аліна Василівна	female	Інформаційна аналітика	Географія та екологія	Інженерно-технічний інститут
Кузьмін Дмитро Володимирович	male	Комп'ютерна безпека	Географія та екологія	Інженерно-технічний інститут
Поляков Артем Володимирович	male	Кібератаки та захист від них	Географія та екологія	Інженерно-технічний інститут
Мартиненко Сергій Олександрович	male	Теоретична фізика	Соціологія та політологія	Інститут гуманітарних наук
Шевчук Юрій Станіславович	male	Експериментальна фізика	Соціологія та політологія	Інститут гуманітарних наук
Григоренко Олексій Ігорович	male	Астрофізика	Соціологія та політологія	Інститут гуманітарних наук

Рисунок 3.9 - Результат п'ятого запиту

Запит 6: Вивести інформацію про викладача за номером 0666535891 таку як: ім'я, пошта, посилання на фото, посада.

```
SELECT n.names_id, n.name, p.phone_number, e.email_address, po.position_name,
ph.photo_url
FROM names n
JOIN phones p ON n.names_id = p.phones_id
JOIN emails e ON n.names_id = e.emails_id
JOIN positions po ON n.names_id = po.positions_id
JOIN photos ph ON n.names_id = ph.photos_id
WHERE p.phone_number = '0666535891';
```

names_id	name	phone_number	email_address	position_name	photo_url
1	Рижкін Олександр Костянтинів	0666535891	john.doe@gmail.com	Викладач інформаційних технологій	https://pnu.edu.ua/

Рисунок 3.10 - Результат шостого запиту

Запит 7: Вивести кабінети та посилання на вебсайт спеціальностей починаючи з 5-тої з кроком 5

```
SELECT h.address, h.website
FROM hulls h
INNER JOIN specialisations s ON h.specialisations_id = s.specialisations_id
WHERE s.specialisations_id IN (5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75);
```

address	website
Кабінет 005	https://pnu.edu.ua/
Кабінет 010	https://pnu.edu.ua/
Кабінет 015	https://pnu.edu.ua/
Кабінет 020	https://pnu.edu.ua/
Кабінет 025	https://pnu.edu.ua/
Кабінет 030	https://pnu.edu.ua/
Кабінет 035	https://pnu.edu.ua/
Кабінет 040	https://pnu.edu.ua/
Кабінет 045	https://pnu.edu.ua/
Кабінет 050	https://pnu.edu.ua/
Кабінет 055	https://pnu.edu.ua/
Кабінет 060	https://pnu.edu.ua/
Кабінет 065	https://pnu.edu.ua/
Кабінет 070	https://pnu.edu.ua/
Кабінет 075	https://pnu.edu.ua/

Рисунок 3.11 - Результат сьомого запиту

Запит 8: Отримати імена та контактну інформацію для всіх академіків, які працюють на певному факультеті: 9, 10, 11

```
SELECT n.name, p.phone_number, e.email_address
FROM names n
INNER JOIN academics a ON n.academics_id = a.academics_id
INNER JOIN phones p ON n.phones_id = p.phones_id
INNER JOIN emails e ON n.emails_id = e.emails_id
INNER JOIN specialisations s ON n.specialisations_id = s.specialisations_id
INNER JOIN faculties f ON s.faculties_id = f.faculties_id
WHERE f.faculties_id IN (9, 10, 11);
```

name	phone_number	email_address
Цветкова Ірина Віталіївна	0948888888	george.martin@gmail.com
Чернов Денис Сергійович	0969999999	betty.williams@gmail.com
Шевченко Олена Василівна	0901111222	matthew.moore@gmail.com
Щербань Євген Ігорович	0922222333	judy.brown@gmail.com
Юрченко Оксана Ігорівна	0998765432	joseph.miller@gmail.com
Яковлев Вадим Михайлович	0912345678	dorothy.thomas@gmail.com
Даниленко Ірина Станіславівна	0935555666	christopher.wilson@gmail.com
Максимов Олександр Вікторович	0974444555	sandra.harris@gmail.com
Сидельникова Лариса Ігорівна	0982222333	andrew.smith@gmail.com

Рисунок 3.12 - Результат восьмого запиту

Запит 9: Вивести випадкову роботу та ім'я викладача

```
SELECT a.projects_and_publications AS academic_project,
n.name AS instructor_name
FROM names n
INNER JOIN academics a ON n.academics_id = a.academics_id
ORDER BY RAND() LIMIT 1;
```

projects_and_publications	name
Архітектура та дизайн: трансформації простору та е...	Нестеренко Лідія Вікторівна

Рисунок 3.13 - Результат дев'ятого запиту

Запит 10: Вивести інститут факультет спеціальність ім'я кабінет телефон та фото викладача під номером 21

```
SELECT i.institutions_name, f.faculties_name ,
s.specialisations_name , n.name , h.address , p.phone_number,
ph.photo_url
FROM names n
INNER JOIN academics a ON n.academics_id = a.academics_id
INNER JOIN phones p ON n.phones_id = p.phones_id
INNER JOIN photos ph ON n.photos_id = ph.photos_id
INNER JOIN specialisations s ON n.specialisations_id = s.specialisations_id
INNER JOIN faculties f ON s.faculties_id = f.faculties_id
INNER JOIN institutions i ON f.institutions_id = i.institutions_id
INNER JOIN hulls h ON s.specialisations_id = h.specialisations_id
WHERE n.names_id = 21;
```

institutions_name	faculties_name	specialisations_name	name	address	phone_number	photo_url
Фізичного виховання	Туризм та готельно-ресторанна справа	Радіожурналістика	Ткаченко Людмила Олександрівна	Кабінет 021	0977654321	https://bing.com/

Рисунок 3.14 - Результат десятого запиту

ВИСНОВКИ

У даному курсовому проекту було проведено дослідження теми «Створення телефонного довідника», створивши базу даних яка в повній мірі відображає роботу такого проекту в умовах університету. Цю базу можна використати для дійсно існуючого вищого навчального закладу, або як шаблон для будь-якої некомерційної компанії. Під час написання цієї курсової роботи були здобуті навички проектування, моделювання а також створення бази даних. Навчившись класифікувати, а також угруповувати дані. Коли дані організовані, вони стають більш гнучкими, тобто їх легше змінювати та модифікувати в майбутньому. Спробувавши на практичній частині основні можливості середовища phpMyAdmin та мови SQL.

					КП.ІПЗ-321.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		30

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кузь М. Навчально-методичний посібник до курсового проектування. Івано-Франківськ: ПНУ ім. Василя Стефаника, 2018. 38 с.

2. Документація по MySQL [Електронний ресурс]

URL: <https://www.mysql.com/> (дата звернення: 31.12.2023)

3. Загальна інформація з Вікіпедії для побудови теоретичної частини [Електронний ресурс]

URL: <https://uk.wikipedia.org/wiki/SQL> (дата звернення: 31.12.2023)

4. Гайдаржи В. І., Ізварін І. В. Бази даних в інформаційних системах. Тернопіль: Університет "Україна", 2018. 418 с.

5. W3Schools SQL Tutorial [Електронний ресурс]

URL: https://www.w3schools.com/sql/sql_foreignkey.asp

(дата звернення: 01.01.2024)

					КП.ІПЗ-321.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

