

Міністерство освіти і науки України
НТУУ «КПІ ім. Ігоря Сікорського»
Навчально-науковий інститут атомної та теплової енергетики
Кафедра цифрових технологій в енергетиці

Лабораторна робота №3
з дисципліни
«Проектування інформаційних систем»
Тема «Розробка структури бази даних і контрольного
прикладу для АРМ ІС»
Варіант №18

Студента 4-го курсу НН ІАТЕ гр. ТР-12
Ковальова Олександра
Перевірів: ст. в., Гурін Артем Леонідович

Мета роботи. Закріпити навички створення структури бази даних.

Завдання: Розробити звіт, що містить структуру бази даних та контрольний приклад для АРМ ІС.

Повинні бути визначені:

- склад таблиць: з кожної таблиці – поля, розмірність полів, тип полів;
- взаємозв'язок таблиць: ключові атрибути;
- структура: намалювати структуру бази даних.

Контрольний приклад повинен забезпечити перевірку функціонування АРМ ІС, у тому числі дій, що виконуються користувачами в процесі експлуатації, та реакції АРМ на дії користувачів. Повинен бути у вигляді таблиці:

№ п/п	Вхідні дані	Реакція системи (вихідні дані)	Опис функціональності, що перевіряється системи – що, власне, перевіряється (Пункт вимог ТЗ)
1.	

Індивідуальне завдання: ІС обліку історичних пам'яток м. Житомир.

Зразковий зміст бази даних:

- код об'єкта, найменування, фотографії;
- історичні характеристики об'єкта (стародавність та ін.), текстова характеристика об'єкта;
- поточний стан об'єкта, для чого використовується в даний час, вартість об'єкта, якщо об'єкт в експлуатації – ким (інформація про експлуатуючу організацію або приватну особу);
- інше.

Хід роботи.

Структура таблиць: поля, тип, розмірність полів, обмеження (constraints):

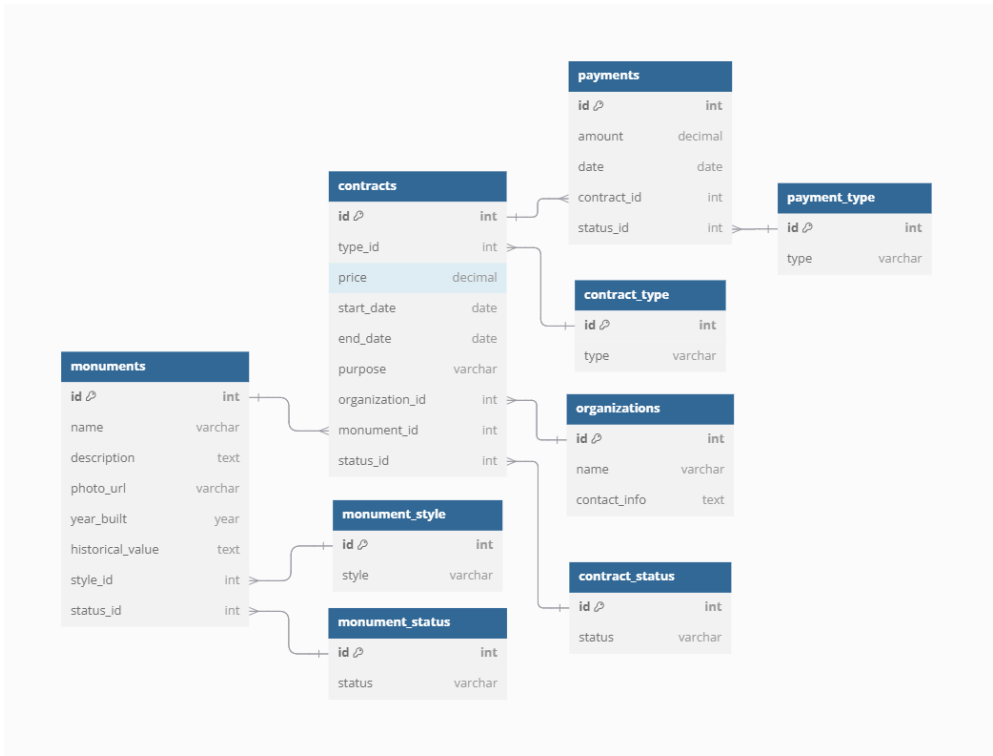
Таблиця	Поле	Тип поля	Обмеження
monuments	id	int	PK, Auto Increment
	name	varchar(255)	Not Null
	description	text	
	photo_url	varchar(255)	
	year_built	year	Not Null
	historical_value	text	
	style_id	int	FK: monument_style.id
	status_id	int	FK: monument_status.id
monument_style	id	int	PK, Auto Increment
	style	varchar(100)	Not Null
monument_status	id	int	PK, Auto Increment
	status	varchar(50)	Not Null

<i>organizations</i>	id	int	PK, Auto Increment
	name	varchar(255)	Not Null
	contact_info	text	
<i>payments</i>	id	int	PK, Auto Increment
	amount	decimal	Not Null
	date	date	Not Null
	contract_id	int	FK: contracts.id
	status_id	int	FK: payment_type.id
<i>payment_type</i>	id	int	PK, Auto Increment
	type	varchar(50)	Not Null
<i>contracts</i>	id	int	PK, Auto Increment
	type_id	int	FK: contract_type.id
	price	decimal	Not Null
	start_date	date	Not Null
	end_date	date	Not Null
	purpose	varchar(255)	Not Null
	organization_id	int	FK: organizations.id
	monument_id	int	FK: monuments.id
	status_id	int	FK: contract_status.id
<i>contract_type</i>	id	int	PK, Auto Increment
	type	varchar(50)	Not Null
<i>contract_status</i>	id	int	PK, Auto Increment
	status	varchar(50)	Not Null

Зв'язки між таблицями:

- «1 до ∞» між таблицями «Пам'ятки» та «Контракти», так як в контракті може бути зазначена лише одна пам'ятка, але контрактів по ній може багато.
- «1 до ∞» між таблицями «Стилі пам'яток» та «Пам'ятки», так як перша таблиця є словником стилів.
- «1 до ∞» між таблицями «Статус пам'ятки» та «Пам'ятки», так як перша таблиця є словником статусів пам'ятки.
- «1 до ∞» між таблицями «Тип контракту» та «Контракти», так як перша таблиця є словником типів контрактів.
- «1 до ∞» між таблицями «Організації» та «Контракти», так як на одну організацію може бути укладено багато контрактів, але в контракті може бути вказана лише одна організація.
- «1 до ∞» між таблицями «Контракти» та «Платежі», так як в одному платежі може бути вказаний один контракт, а платежів багато.
- «1 до ∞» між таблицями «Тип платежу» та «Платежі», так як перша таблиця є словником типів платіжок.
- «∞ до ∞» між таблицями «Пам'ятки» та «Організації» через таблицю «Контракти», так як контракт може бути укладений як і на багато пам'яток однією організацією, так і багатьма організаціями на одну пам'ятку.

Схема:



Декілька контрольних прикладів:

№	Вхідні дії	Реакція системи	Функціональність, яка перевіряється
1	Додавання нового пам'ятника із заповненням назви, опису, року побудови, історичної цінності та стилю.	Система додає новий запис до таблиці monuments, генерує унікальний ID. Автоматично призначається статус «Активний», бо не призначений інший.	Перевірка коректного збереження даних пам'ятки.
2	Оновлення статусу пам'ятника (наприклад, зміна зі статусу "Активний" на "В реставрації").	Система змінює відповідний status_id в обраної пам'ятки.	Перевірка можливості зміни статусу пам'ятника.
3	Створення нового договору на експлуатацію пам'ятника із зазначенням дати початку, кінцевої дати, організації, яка експлуатує об'єкт.	Система додає запис до таблиці contracts, генерує унікальний ID договору.	Перевірка створення договорів та їх прив'язки до організацій і пам'яток.

4	Введення платежу за певний договір, із зазначенням суми та дати платежу.	Система додає новий запис до таблиці payments з прив'язкою до відповідного договору.	Перевірка функціональності обліку платежів по договорам.
5	Перевірка історичної інформації про пам'ятник, включаючи текстовий опис і фото.	Система отримує та відображає повну інформацію з таблиці monuments, включаючи фотографії та опис.	Перевірка відображення історичної інформації та фото об'єктів.

Висновок: У ході виконання лабораторної роботи була досягнута поставлена мета – закріплено навички створення структури бази даних. Було розроблено структуру бази даних для інформаційної системи обліку історичних пам'яток м. Житомир, яка включає таблиці з детальним описом полів, типів даних, їх розмірності, а також визначено взаємозв'язки між ними за допомогою ключових атрибутів.

Було успішно реалізовано індивідуальне завдання, яке включало в себе проектування таблиць для обліку пам'яток, їх історичних характеристик, статусів, фотографій, а також пов'язаних з ними організацій, договорів та платежів. Кожна таблиця має відповідні атрибути та первинні ключі, а також встановлені зв'язки між ними, що забезпечує ефективну обробку та збереження даних.

Окрім створення структури бази даних, було розроблено контрольний приклад для перевірки функціонування автоматизованого робочого місця (АРМ). Було описано кілька тестових сценаріїв із вхідними діями користувачів та реакцією системи, що дозволило оцінити працездатність системи та її реакцію на реальні дії користувачів.

У підсумку всі завдання лабораторної роботи виконані повністю, розроблена структура бази даних та контрольні приклади для АРМ відповідають вимогам завдання, забезпечуючи належну функціональність інформаційної системи.

Додаток А: DBML код схеми бази даних.

```

Table monuments {
  id int [pk, increment]
  name varchar
  description text
  photo_url varchar
  year_built year
  historical_value text
  style_id int [ref: > monument_style.id]
  status_id int [ref: > monument_status.id]
}

Table monument_style {
  id int [pk, increment]
  style varchar
}

Table monument_status {
  id int [pk, increment]
  status varchar
}

```

```

Table organizations {
  id int [pk, increment]
  name varchar
  contact_info text
}

Table payments {
  id int [pk, increment]
  amount decimal
  date date
  contract_id int [ref: > contracts.id]
  status_id int [ref: > payment_type.id]
}

Table payment_type {
  id int [pk, increment]
  type varchar
}

Table contracts {
  id int [pk, increment]
  type_id int [ref: > contract_type.id]
  price decimal
  start_date date
  end_date date
  purpose varchar
  organization_id int [ref: > organizations.id]
  monument_id int [ref: > monuments.id]
  status_id int [ref: > contract_status.id]
}

Table contract_type {
  id int [pk, increment]
  type varchar
}

Table contract_status {
  id int [pk, increment]
  status varchar
}

```