

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

МОДЕЛЮВАННЯ ТА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ОРГАНІЗАЦІЇ РОЗКЛАДУ ТЕЛЕПЕРЕДАЧ НА РІЗНИХ ТЕЛЕКАНАЛАХ (тема)

Виконав:
студент 4 курсу, групи ІТІНФ-18-1

Тарасенко Д.О.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Творошенко І.С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

Кобилін О.А.
(прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)Кафедра Інформатики
(повна назва)Рівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« ____ » _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Тарасенку Денису Олександровичу
(прізвище, ім'я, по батькові)1. Тема роботи Моделювання та реалізація інформаційної системи для організації розкладу телепередач на різних телеканалахзатверджена наказом університету від 16 травня 2022 року № 541Ст2. Термін подання студентом роботи до екзаменаційної комісії 23 травня 2022 р.3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, CASE-засіб візуального проектування даних ERwin, середовище розробки програмних систем Microsoft Visual Studio, технологія доступу до даних ADO.NET, СУБД Microsoft SQL Server, інтегроване середовище SQL Server Management Studio.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Аналіз існуючих принципів моделювання та механізмів реалізації баз даних.

2. Моделювання структури бази даних інформаційної системи вибраної предметної області.

3. Моделювання та реалізація інформаційної системи для організації розкладу телепередач на різних телеканалах.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) схема моделей даних за синтаксисом Чена, модель даних, згенерована в SQL Server, діаграма реалізації можливостей користувачів застосунку, скріншоти наповнених таблиць бази даних, скріншоти форм застосунку.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Консультант з дотримання діючих стандартів та норм	Доцент Белова Н.В.		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	18.04.2022	
2	Аналіз завдання, підбір літератури	18.04.22-21.04.22	
3	Аналіз літератури з досліджуваної проблеми	22.04.22-25.04.22	
4	Аналіз технічних засобів	26.04.22-30.04.22	
5	Розробка інформаційної системи вибраної предметної області	01.05.22-14.05.22	
6	Програмна реалізація	15.05.22-23.05.22	
7	Оформлення пояснювальної записки	24.05.22-26.05.22	
8	Перевірка на плагіат	27.05.22	
9	Рецензування	28.05.22	
10	Підготовка презентації та доповіді	29.05.22-30.05.22	
11	Занесення роботи в електронний архів	31.05.22	
12	Попередній захист кваліфікаційної роботи	31.05.22	

Дата видачі завдання 18 квітня 2022 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Творошенко І.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 53 с., 12 табл., 22 рис., 2 дод., 36 джерел.

ІНФОРМАЦІЙНА СИСТЕМА, СУБД, РЕЛЯЦІЙНА БАЗА ДАНИХ, НОРМАЛІЗАЦІЯ ДАНИХ, ER-МОДЕЛЬ, МОДЕЛЬ «СУТНІСТЬ – ЗВ’ЯЗОК», МОВА ЗАПИТІВ SQL, CASE-ЗАСІБ ERWIN, MS ACCESS, MS SQL SERVER, MS SQL SERVER MANAGEMENT STUDIO.

Об’єктом роботи є розклад телепередач на різних телеканалах.

Метою роботи є моделювання та реалізація інформаційної системи для організації розкладу телепередач на різних телеканалах.

Робота присвячена розробці інформаційної системи «Телепрограма». Дана інформаційна система зберігає інформацію про розклад телепередач на різних телеканалах. Має 3 рівня доступу: адміністратор (може змінювати дані усіх таблиць), укладач розкладу (змінює розклад телепередач) та користувач (може продивлятися розклад за побажаннями).

Під час розробки інформаційної системи були використані: теорія проєктування реляційних баз даних на базі будування ER-моделі; основні принципи нормалізації реляційних баз даних; мова запитів SQL; CASE-засіб візуального проєктування даних ERwin, середовище розробки програмних систем MS Visual Studio, мова програмування C#.

INFORMATION SYSTEM, DBMS, RELATIVE DATABASE, NORMALIZATION OF DATA, ER-MODEL, MODEL “ENTITY – CONNECTION”, SQL QUERY LANGUAGE, ERWIN CASE TOOL, MS ACCESS, MS SQL SERVER, MS SQL SERVER MANAGEMENT STUDIO.

The object of work is the schedule of TV programs on various TV channels.

The goal of the work is to model and implement an information system for organizing TV schedules on different TV channels.

The work is devoted to the development of the information system “TV program”. This information system stores information about the schedule of TV programs on various TV channels. It has 3 levels of access: administrator (can change the data of all tables), scheduler (can change the schedule of TV programs), and user (can view the schedule at will).

During the development of the information system were used: the theory of relational database design based on the construction of the ER-model; basic principles of normalization of relational databases; SQL query language; CASE-tool for visual data design ERwin, software development environment MS Visual Studio, C # programming language.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	7
Вступ.....	8
1 Аналіз існуючих принципів моделювання та механізмів реалізації баз даних.....	9
1.1 Аналіз основних етапів життєвого циклу	9
1.2 Особливості сучасних підходів до управління ІТ-проектами.....	11
1.3 Існуючі властивості реляційних баз даних	11
1.4 Вивчення механізмів мови SQL для реалізації баз даних	13
1.5 Принципи моделювання структури баз даних.....	14
1.6 Постановка задачі	16
2 Моделювання структури бази даних інформаційної системи вибраної предметної області	18
2.1 Основні характеристики та особливості предметної області «Організація розкладу телепередач на різних телеканалах».....	18
2.2 Особливості структури бази даних інформаційної системи вибраної предметної області.....	19
2.2.1 Специфікація вимог до інформаційної системи	19
2.2.2 Розробка бізнес-правил бази даних.....	19
2.2.3 Розробка концептуальної моделі інформаційної системи	20
2.2.4 Побудова логічної моделі даних інформаційної системи	23
2.3 Моделювання структури та наповнення бази даних інформаційної системи вибраної предметної області.....	30
2.4 Фізичне проектування інформаційної системи вибраної предметної області	31
3 Моделювання та реалізація інформаційної системи для організації розкладу телепередач на різних телеканалах.....	36
3.1 Вибір інструментальних засобів для реалізації	36

3.2	Етапи реалізації інформаційної системи для організації розкладу телепередач на різних телеканалах	37
3.3	Тестування розробленої інформаційної системи та аналіз результатів	40
3.4	Перспективи подальшої роботи	41
	Висновки	42
	Перелік джерел посилання	43
	Додаток А Відношення-екземпляри розробленої бази даних	47
	Додаток Б Ілюстрація роботи розробленої інформаційної системи	50

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ADO.NET – набір класів, які надають послуги доступу до даних для програмістів .NET Framework

API – Application Programming Interface (програмний інтерфейс програми)

CASE – Computer-Aided Software Engineering (комп'ютерна інженерія програмного забезпечення)

DQL – Data Query Language (мова запитів до даних)

DML – Data Manipulation Language (мова маніпулювання даними)

DDL – Data Definition Language (мова визначення даних)

ER – Entity-Relationship («сутність – зв'язок»)

GUI – Graphical User Interface (графічний інтерфейс користувача)

IBM – назва американської електронної корпорації

IDE – Integrated Development Environment (інтегроване середовище розробки)

IT – Information Technologies (інформаційні технології)

RDBMS – Relational Database Management System (система управління реляційними базами даних)

SQL – Structured Query Language (структурована мова запитів)

TV – television (телебачення)

UML – Unified Modeling Language (уніфікована мова моделювання)

ЖЦ – життєвий цикл

ІС – інформаційна система

ПО – предметна область

СУБД – система управління базами даних

ТБ – телебачення

ВСТУП

Кожен день люди споживають багато інформації і роблять це різними способами. Хтось шукає цікаве в соціальних мережах, хтось – на сайтах новин, хтось дивиться кіно та серіали на спеціальних сервісах, а хтось дивиться телебачення, або, так зване, інтернет-ТБ, яке поєднує в собі все зазначене раніше. Програми показуються у телеефірі строго за розкладом, а тому необхідно, щоб цей розклад був чітко спланований.

Данна робота присвячена моделюванню та реалізації інформаційної системи для організації розкладу телепередач на різних телеканалах.

Для цього було використано бази даних. Серед переваг їх використання можна виділити те, що вони дозволяють зменшити надмірність даних, зменшити кількість помилок оновлення та підвищити узгодженість, підвищити цілісність даних і незалежність від прикладних програм, покращити доступ до даних для користувачів завдяки використанню мов хосту та запитів, покращити безпеку даних, зменш витрат на введення, зберігання та пошук даних, сприяти розробці нових прикладних програм.

Для роботи з базою даних використовувалася СУБД Microsoft SQL Server. Застосунок було написано мовою С# з використанням Windows Forms для розробки інтерфейсу та технології доступу до даних ADO.NET. В якості середовища розробки було обрано Microsoft Visual Studio.

Актуальність роботи полягає у тому, що згідно з опитуванням, проведеним минулого року, 92% з 1200 респондентів дивляться телеканали, а 67% дивляться їх щоденно, 50% дивляться онлайн в інтернеті. Оскільки телебачення (в тому числі й інтернет-ТБ) є популярним, відповідно є й потреба у складанні розкладу телепередач для телеканалів.

1 АНАЛІЗ ІСНУЮЧИХ ПРИНЦИПІВ МОДЕЛЮВАННЯ ТА МЕХАНІЗМІВ РЕАЛІЗАЦІЇ БАЗ ДАНИХ

1.1 Аналіз основних етапів життєвого циклу

Життєвий цикл управління проектами – це уніфікована модель прогресу проекту, від його початку до завершення [1-6]. Кожен процес проекту складається з п'яти ключових стадій: початкова стадія, планування, виконання, моніторинг та контроль, завершення. Ці етапи детально описують усі аспекти проекту – від найменших елементів, таких як залежності та індивідуальні обов'язки, до основних складових проекту, таких як графік, бюджет та основні етапи.

Поглиблене знання життєвого циклу управління проектами є особливо важливим для керівників проектів, оскільки дозволяє їм ефективно планувати та виконувати проект. Структурування проекту на етапи надає їм погляд з пташиного польоту протягом усього процесу, що дозволяє повністю контролювати кожен крок.

Початкова стадія закладає основу для всіх майбутніх кроків до завершення проекту. На цьому етапі команда визначає ключові проблеми, які новий проект буде прагнути вирішити, вирішує наскільки цей проект здійснений і чи має насправді сенс продовжувати його виконання. До кінця цього етапу всі учасники повинні бути дуже пильними щодо обґрунтування проекту.

На цьому етапі починається розподіл ресурсів та пошук корисних джерел, збирається команда, визначаються бюджети тощо.

Менеджер розглядає низку важливих питань: з якими проблемами може зіткнутися реалізація проекту? Як довго триватиме проект? Які результати та цілі будуть досягнені наприкінці роботи.

Якщо відповіді на всі запитання будуть знайдені та доречно інтегруються в узгодження замовника та виконавця, команда може переходити на наступний етап.

Як тільки етап ініціації закінчиться, проєкт переходить на етап планування. Перш ніж приступити до виконання проєкту, потрібно створити детальний план дій. План проєкту систематизує роботу, не даючи проєкту стати дуже хаотичним процесом. На етапі планування слід чітко розуміти наступні моменти: бачення проєкту, графік та терміни, склад команди, групові та індивідуальні обов'язки, доступні ресурси, включаючи бюджети, і як вони будуть використані, робочі процеси проєкту.

Плануючи, команда складає «дорожню карту», яка визначає основні етапи проєкту з кроками, які будуть втілені для їх досягнення. Крім того, команда буде мати більш чітке розуміння строків проєкту, та термінів яких повинен дотримуватися кожен член команди. Найголовніше – збираються та розподіляються ресурси туди, де вони потрібні.

Наступний етап – виконання проєкту. На цьому етапі проєкт нарешті «оживає». Це найдовший і найважливіший етап проєкту, в якому крок за кроком виконується всі дії для реалізації кінцевого результату. Наприкінці цього етапу мозковий штурм перетворюється у фізичний прояв.

Основна увага приділяється результатам, як внутрішнім, так і зовнішнім: від звітів, які діляться між командами, до проміжних продуктів, які роздаються замовникам. Менеджеру проєктів потрібно буде стежити за тим, як прогресує проєкт, і уважно дивитися за кожним етапом процесу.

Наступний етап контроль та моніторинг. Основні обов'язки керівника проєкту на цьому етапі полягають у забезпеченні прогресу проєкту згідно плану та графіку, затвердженого на другому етапі. Це передбачає управління бюджетами та всіма іншими ресурсами та переконання, що кожен крок проєкту відповідає строкам. Під час цього етапу менеджер проєкту також несе відповідальність за передбачення та виявлення будь-яких проблем та

швидку діяльність щодо їх усунення. На цьому етапі відбувається оцінка та аналіз, включаючи оцінку індивідуальної та командної ефективності.

Етап завершення проєкту стосується аналізу та оцінки. На цьому етапі керівник та команда підбивають підсумки та міркують над тим, що можна було б покращити.

1.2 Особливості сучасних підходів до управління ІТ-проєктами

Традиційна модель управління має свої обмеження і може не відповідати потребам глобального розвитку.

Сучасні підходи є більш динамічні та гнучкі. Вони використовують декілька бізнес-стратегій, компанії прагнуть покращувати робочий процес, оновлювати свої конкурентні переваги, а працівники мають покращувати свої знання та вміння, щоб залишатися на ринку.

Сучасний підхід характеризується відсутністю строгої ієрархії [7-11], натомість використовується так звана плоска ієрархія, яка орієнтована на командну роботу та співробітництво. Це сприяє мозковому штурму, підвищенню ефективності роботи [12-16]. Сучасне управління проєктами надає працівникам більше свободи та гнучкості для виконання своєї роботи.

1.3 Існуючі властивості реляційних баз даних

Реляційна база даних – це набір формально описаних таблиць, з яких можна отримати доступ або повторно зібрати дані різними способами без необхідності реорганізації таблиць бази даних. Стандартний інтерфейс програмування користувачів та прикладних програм (API) реляційної бази даних – це структурована мова запитів (SQL). Оператори SQL

використовуються як для інтерактивних запитів щодо інформації з реляційної бази даних, так і для збору даних для звітів.

Реляційну базу даних винайшов у 1970 році Е. Ф. Кодд, тодішній молодий програміст в ІВМ. У своїй роботі «Реляційна модель даних для великих спільних банків даних» Кодд запропонував перейти від зберігання даних в ієрархічних або навігаційних структурах до організації даних у таблицях, що містять рядки та стовпці.

Кожна таблиця, яку іноді називають відношенням, у реляційній базі даних містить одну або більше категорій даних у стовпцях або атрибутах. Кожен рядок, який також називається записом або кортежем, містить унікальний примірник даних або ключ для категорій, визначених стовпцями. Кожна таблиця має унікальний первинний ключ, який ідентифікує інформацію в таблиці. Відносини між таблицями можуть бути встановлені за допомогою використання зовнішніх ключів – поля в таблиці, яке посилається на первинний ключ іншої таблиці.

Створюючи реляційну базу даних, можна визначити область можливих значень у стовпці даних та інші обмеження, які можуть застосовуватися до цього значення даних. Наприклад, домен можливих клієнтів може дозволити до 10 можливих імен клієнтів, але бути обмеженим в одній таблиці, щоб дозволити визначити лише три з цих імен клієнтів.

Два обмеження стосуються цілісності даних та первинного та зовнішнього ключів:

- цілісність об'єкта гарантує, що первинний ключ у таблиці унікальний, і що значення не встановлено на нуль;
- цілісність за посиланням вимагає, щоб кожне значення стовпця з зовнішнім ключем знаходилося у первинному ключі таблиці, з якої воно походить.

Основна перевага реляційних баз даних полягає в тому, що вони дозволяють користувачам легко класифікувати та зберігати дані, які згодом можуть бути запитані та відфільтровані для отримання конкретної інформації

для звітів. Реляційні бази даних також легко розширюються і не залежать від фізичної організації. Після створення оригінальної бази даних можна додати нову категорію даних без зміни всіх існуючих застосунків.

Інші переваги реляційних баз даних включають:

- точність: дані зберігаються лише один раз, виключаючи дедуплікацію даних;
- гнучкість: складні запити для користувачів легко виконувати;
- співпраця: кілька користувачів можуть отримати доступ до однієї бази даних;
- довіра: реляційні моделі баз даних зрілі та добре зрозумілі;
- безпека: дані в таблицях в RDBMS можуть бути обмежені, щоб дозволити доступ лише певним користувачам.

1.4 Вивчення механізмів мови SQL для реалізації баз даних

Більшість сучасних реляційних СУБД використовують саме мову SQL. Американський інститут національних стандартів (American National Standards Institute – ANSI) та Міжнародна організація стандартів (International Standards Organization – ISO) займаються описом і підтримкою стандартів цієї мови. Усі сучасні СУБД підтримують певний стандарт, проте є й відхилення, які в кожному конкретному випадку специфікуються в документації програмного продукту. Окрім того, у багатьох системах розроблено розширення SQL, що дають змогу використовувати мову запитів у середовищі програмування.

SQL надає такі можливості:

- створювати й видаляти таблиці бази даних, а також змінювати заголовки таблиць;
- вставляти, змінювати й видаляти рядки в таблицях;

- виконувати пошук даних у багатьох таблицях та впорядковувати результати цього пошуку;
- описувати процедури підтримки цілісності;
- визначати та змінювати інформацію про захист даних.

1.5 Принципи моделювання структури баз даних

Правильно розроблена база даних надає доступ до актуальної та точної інформації. Процес моделювання бази даних керується певними принципами. Перший принцип полягає в тому, що дублююча інформація (яку також називають надлишковими даними) є поганою, оскільки витрачає простір і збільшує ймовірність помилок і невідповідностей. Другий принцип – важлива правильність і повнота інформації. Якщо база даних містить неправильну інформацію, будь-які звіти, які витягують інформацію з бази даних, також міститимуть невірну інформацію. В результаті будь-які рішення, що приймаються на основі цих звітів, будуть дезінформовані.

Таким чином, добре змодельована база даних – це та, яка:

- розділяє інформацію на таблиці на основі тематики, щоб зменшити зайві дані;
- забезпечує доступ до інформації, необхідної для об'єднання інформації в таблицях, якщо це необхідно;
- допомагає підтримувати та гарантувати точність і цілісність інформації;
- задовольнить потреби в обробці даних та звітності.

Процес моделювання складається з наступних етапів:

- визначення призначення бази даних – це підготовчий етап до наступних кроків;

- знаходження та впорядкування потрібної інформації. На цьому етапі збираються усі типи інформації, які треба записати в базу даних, наприклад: назва продукту та номер замовлення;

- розділення інформації на таблиці. На цьому етапі інформаційні елементи розділяються на основні сутності або предмети, такі як продукти або замовлення. Потім кожен предмет стає таблицею;

- перетворення елементів інформації на стовпці. На цьому етапі визначається, яку інформацію необхідно зберегти в кожній таблиці. Кожен елемент відображається у вигляді стовпця в таблиці як поле;

- вказання первинних ключів. На цьому етапі обирається первинний ключ для кожної таблиці. Первинний ключ – це стовпець або комбінація стовпців, які однозначно ідентифікують один рядок і відрізняють його від будь-якого іншого рядка;

- налаштування зв'язків між таблицями. На цьому етапі визначається, як дані в одній таблиці пов'язані з даними в інших таблицях. За потреби необхідно додати поля до таблиць або створити нові таблиці для уточнення зв'язків;

- вдосконалення моделі. На цьому етапі необхідно проаналізувати модель на наявність помилок. Треба створити таблиці та додати кілька записів зразкових даних. З'ясувати, чи можна отримати потрібні результати з таблиць. За потреби необхідно внести коригування в модель;

- застосування правил нормалізації. На цьому етапі необхідно застосувати правила нормалізації даних для перевірки правильності структурованості таблиці. За потреби необхідно внести корективи в таблиці.

Чому ці кроки важливі? Тому що це допомагає створювати системи баз даних, які відповідають вимогам користувачів та мають високу продуктивність.

Процес моделювання бази даних у СУБД має вирішальне значення для високопродуктивної системи баз даних.

1.6 Постановка задачі

Об'єктом роботи є розклад телепередач на різних телеканалах.

Метою роботи є моделювання та реалізація інформаційної системи для організації розкладу телепередач на різних телеканалах.

Нижче наведені опис предметної області, вимоги користувача до даних і функціоналу інформаційної системи:

- а) програма повинна показувати розклад показу телепередач на різних телеканалах;
- б) користувач програми повинен мати можливість:
 - 1) самостійно обрати, телепрограму яких каналів він хоче подивитися;
 - 2) обрати жанр телепередачі для показу у телерозкладі;
 - 3) подивитися телепрограму на сьогодні, на 3 дня, на тиждень;
 - 4) подивитися детальну інформацію про телепродукт.

Редагувати розклад має можливість укладач розкладу. Змінювати дані усіх таблиць, редагувати структуру таблиць, змінюючи типи, атрибути тощо може тільки адміністратор.

Для досягнення мети необхідно вирішити наступні задачі:

- вивчити такі теоретичні питання: теорію проектування реляційних БД на основі побудови ER-моделі, основні принципи нормалізації БД, принципи побудови запитів до БД (мови DQL SQL, DML SQL, DDL SQL);
- ознайомитися з наступними інструментами проектування і програмування ІС: CASE-засобом візуального проектування даних ERwin, MS Visual Studio;
- застосувати отримані теоретичні та практичні знання для розробки ІС для ПО «Телепрограма»;
- вивчивши опис ПО і вимоги до функціональності ІС, розробити бізнес-правила і глосарій;

- на основі бізнес-правил розробити ER-модель і відобразити її за допомогою ER-діаграми в синтаксисі Чена;
- розробити модель даних за допомогою CASE-засобу візуального проектування даних ERwin;
- сформувати структуру БД в СУБД;
- розробити дизайн ІС;
- розробити програмний код для забезпечення необхідної функціональності ІС;
- провести тестування ІС;
- відобразити процес і результати розробки ІС в пояснювальній записці.

Для розробки ІС необхідно використовувати такі інструменти:

- ERwin – CASE-засіб проектування БД;
- MS SQL Server – СУБД;
- MS Visual Studio, C# – середа і мова розробки програмного забезпечення.

2 МОДЕЛЮВАННЯ СТРУКТУРИ БАЗИ ДАНИХ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВИБРАНОЇ ПРЕДМЕТНОЇ ОБЛАСТІ

2.1 Основні характеристики та особливості предметної області «Організація розкладу телепередач на різних телеканалах»

Процес організації розкладу телепередач на різних телеканалах виконує укладач розкладу.

Укладач розкладу має таблицю з телеканалами, таблицю з телепередачами, і на основі цих даних має скласти розклад. Також треба брати до уваги жанри телепередач та тематику телеканалів, щоб, наприклад, інформаційно-новинний канал не транслював розважальні шоу.

Укладач розкладу повинен враховувати, що в один час телеканал не може транслювати декілька передач. Кожна телепередача має свою тривалість, але на телеканалі є рекламні блоки, тож треба це також мати на увазі, і додати певний час в залежності від частоти та тривалості рекламного блоку. Також у розкладі не має бути «вікон», тобто після закінчення однієї передачі має починатися друга.

В телеканалі можуть бути години, в які транслюються тільки певні телепередачі. Наприклад, новини завжди йдуть в один і той самий час. Тому якщо перша передача ще не закінчилася, а друга вже має початися, то перша може продовжитися після закінчення другої, і відповідний запис повинен бути в розкладі.

Розклад складається мінімум на тиждень вперед.

Кожна телепередача має своїх акторів, режисерів, кореспондентів тощо. Окрім цього в кожного телепродукту є своя компанія-виробник. Ці дані також мають бути доступні до користувача, якщо він захоче подивитися детальну інформацію про передачу.

2.2 Особливості структури бази даних інформаційної системи вибраної предметної області

2.2.1 Специфікація вимог до інформаційної системи

Для розробки інформаційної системи має бути використана мова програмування C# та технологія доступу до даних ADO.NET, тому що ця технологія дозволяє розробляти застосунки з графічним інтерфейсом та включає до свого складу засоби для роботи з базами даних різних форматів. З їх допомогою можна виконувати будь-які SQL-запити та оброблювати їх результати, отримувати доступ до окремих таблиць бази даних, працювати з транзакціями а також використовувати особливі моделі для виводу вмісту таблиць або запитів у будь-якому з компонентів-представлень.

Аби застосунок працював повноцінно, користувач повинен мати персональний комп'ютер, на якому він буде запускати систему, монітор, мишу та клавіатуру для вводу даних та взаємодії з програмою. В якості операційної системи рекомендовано використовувати Windows 7 або новіша.

2.2.2 Розробка бізнес-правил бази даних

Для предметної області «Телепрограма» виділено такі бізнес-правила:

- на телеканалі у певний час може йти тільки один продукт;
- в продукту може бути декілька компаній-виробників, компанія виробляє безліч продуктів;
- компанія-виробник зареєстрована в одній країні;
- на продукті в певну дату людина може працювати тільки в одному статусі, але вона може працювати у тому ж статусі через деякий час;
- в продукту може бути декілька жанрів, до одного жанру належить безліч продуктів;
- людина може деякий час не працювати на жодному телепродукті;

– продукт можуть деякий час не показувати на жодному телеканалі.

Беручи за основу ці бізнес-правила, можна переходити до побудови функціональної структури обраної предметної області.

2.2.3 Розробка концептуальної моделі інформаційної системи

Проектування бази даних полягає в побудові комплексу взаємозв'язаних моделей даних. Моделювання даних починається з концептуального моделювання даних. Ця модель даних визначає, що містить система.

Концептуальні моделі даних використовують стандартну систему символів, які утворюють формальну, хоча й нескладну мову, яка передає велику кількість знань про інформацію, що моделюється.

Концептуальне моделювання дозволяє врахувати логічне уявлення структури даних у базі даних. Правильно розроблена модель бази даних має підтримувати усі явлення користувачів. Концептуальне моделювання є основою подальшого проектування бази даних та застосунку для її обробці.

Розробка концептуальної моделі предметної області є найважливішим етапом проектування бази даних є, не орієнтованим на конкретну СУБД.

Концептуальна модель предметної області будується першою та полягає у структуризації наочної області: об'єкти реального миру піддаються класифікації, фіксується сукупність тих, що підлягають відображенню в БД об'єктів [17-19].

Для кожного об'єкту фіксується сукупність властивостей, за допомогою яких описуватимуться конкретні екземпляри об'єкту, і відносини (взаємозв'язки) з іншими об'єктами.

Потім вирішуються питання про те, яка інформація про об'єкти повинна бути представлена в БД і як її представити за допомогою даних.

Ця модель даних в основному визначає, що містить система. Зацікавлені сторони бізнесу та архітектори даних, як правило, створюють концептуальні моделі даних з наміром організувати та визначити різні бізнес-концепції та правила та встановити їх параметри чи сферу дії.

Таким чином, на цьому етапі проектування треба:

- визначити перелік типів сутностей, інформація про які зберігатиметься у базі даних;
- на підставі опису предметної області визначити зв'язки між сутностями створеної бази даних, навести їх опис;
- визначити тип зв'язків та обмеження участі їх членів;
- визначити попередній перелік атрибутів та зв'язати їх з конкретними типами сутностей;
- визначити первинні та потенційні ключі для кожного об'єкту бази даних;
- побудувати ER-діаграму;
- вилучити зайві зв'язки.

Визначимо перелік сутностей та зробимо їх опис та особливості використання (табл. 2.1).

Таблиця 2.1 – Відомості про типи сутностей

Назва сутності	Опис	Особливості використання
1	2	3
Channel	Назва каналу	Містить в собі перелік телеканалів
TVshow	Телепрограма	Містить розклад телепередач на телеканалі

Продовження таблиці 2.1

1	2	3
Product	Назва телепродукту	Містить інформацію про телепродукт
Company	Назва компанії-виробника	Містить в собі інформацію про компанію-виробник
Country	Назва країни	Містить в собі країни, в яких зареєстровані компанії-виробники
People	ПІБ працівника	Містить інформацію про працівників
Genre	Назва жанру	Містить перелік жанрів

Далі встановимо типи зв'язків між обраними сутностями в моделі даних (табл. 2.2).

Таблиця 2.2 – Відомості про типи зв'язків між сутностями

Назва сутності	Назва сутності	Зв'язок
Channel	TVshow	1:M
Product	TVshow	1:M
Country	Company	1:M
People	Product	M:N
Company	Product	M:N
Product	Genre	M:N

На основі цих таблиць побудуємо концептуально модель даних у синтаксисі Чена (рис. 2.1).

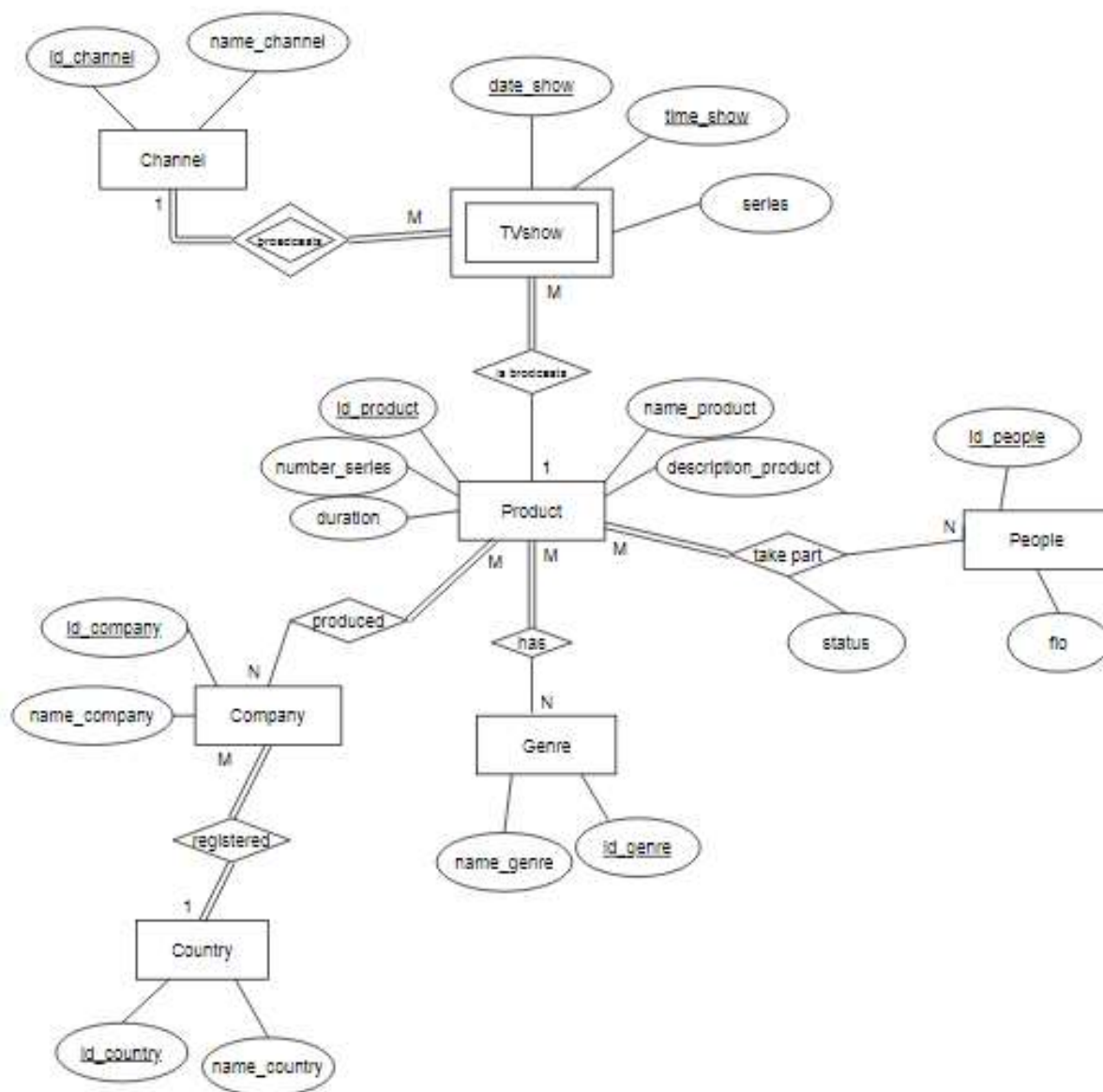


Рисунок 2.1 – Концептуальна модель даних

2.2.4 Побудова логічної моделі даних інформаційної системи

Логічна модель даних визначає, як система має бути реалізована незалежно від СУБД. Вона складається з трьох основних компонентів:

- суб'єкти: кожна сутність представляє набір речей, осіб або концепцій, що мають відношення до бізнесу;
- відносини: будь-які відносини представляють асоціацію між двома з перерахованих вище сутностей;

– атрибути: кожен атрибут є описовою частиною, характеристикою або будь-якою іншою інформацією, яка корисна для подальшого опису сутності.

Кожному з цих компонентів логічної моделі даних дається ім'я та текстове визначення. Вони служать для постійного документування бізнес-правил та визначення вимог до інформації. Однак вищезазначені компоненти обмежуються лише описом вимог бізнесу. Їх не хвилює те, як обробляються, реалізуються або зберігаються зазначені бізнес-вимоги.

З огляду на те, що дані втілюють найважливіший аспект будь-якого застосунку чи системи, якісні системи обробки та зберігання даних повинні будуватися на міцній і точній структурі даних. Надійна структура даних дає розробникам застосунків свободу розробити найкращий користувальницький інтерфейс, систему обробки або статистичний аналіз і налаштування звітності [20-22].

Ось найважливіші характеристики логічної моделі даних:

– логічна модель даних може описати потреби в даних для кожного окремого проєкту. Вона розроблена для легкої інтеграції з іншими логічними моделями даних, якщо цього вимагає проєкт;

– логічна модель даних може бути розроблена незалежно від системи управління базою даних. Тип системи управління базами даних не дуже впливає на це;

– атрибути даних містять типи даних з точною довжиною і точністю;

– у логічній моделі даних визначено первинний або вторинний ключ.

На цьому рівні моделювання даних потрібно перевірити й налаштувати деталі з'єднувача, які були встановлені до визначення зв'язків;

– логічна модель даних схожа на графічне представлення інформаційних вимог бізнес-сфери. Це не база даних чи сама система керування базами даних;

– логічна модель даних не залежить від будь-якого фізичного пристрою зберігання даних, наприклад файлової системи;

– логічна модель даних повинна бути розроблена так, щоб вона була незалежною від технології, щоб не вплинути на швидкі зміни в технології.

Логічна модель даних служить для визначення того, як система має бути реалізована незалежно від використовуваної системи управління базою даних. Архітектори даних і бізнес-аналітики зазвичай є творцями логічної моделі даних. Метою створення логічної моделі даних є розробка високотехнічної карти основних правил і структур даних.

Логічне моделювання даних належить до моделі зв'язку сутності, побудованої за допомогою діаграми зв'язків об'єктів (відомої як ERD), стандартної техніки моделювання, яка використовується як інструмент зв'язку розробниками моделювання даних у всьому світі. У ньому міститься повний набір бізнес-вимог, але не технічні компоненти [23, 24].

Переваги логічної моделі даних:

– оскільки дані залишаються стабільними з часом, логічна модель даних також є стабільною і дуже сприяє повторному використанню даних і обміну фізичними даними, що в кінцевому підсумку призводить до зменшення обсягу зберігання зайвих даних;

– компоненти логічної моделі даних можуть бути перероблені, повторно використані та адаптовані, оскільки більше команд буде відповідати їхнім (часто змінним) потребам;

– витрати, пов'язані зі створенням та підтримкою логічної моделі даних, у довгостроковій перспективі компенсуються перевагами, які вона дає, не в останню чергу шляхом визначення та інтеграції всіх бізнес-потреб і правил з самого початку;

– компоненти процесу побудови, а саме проектування, кодування, тестування та розгортання, відбуваються швидше, як прямий результат інтеграції та уточнення бізнес-правил;

– наявність логічної моделі даних спрощує, а отже, і економічно вигідніше вносити зміни, виправляти помилки або вводити відсутні дані протягом самого життєвого циклу розробки до впровадження;

- логічні моделі даних можна використовувати для аналізу впливу, оскільки кожен бізнес-процес плюс правило пов'язано з ним;
- оскільки об'єкти в логічній моделі даних мають текстові визначення діловою мовою, це полегшує підтримку та доступ до системної документації.

Побудуємо логічну модель у синтаксисі Чена (рис. 2.2).

У ER-діаграмі наявні 10 таблиць: «Channel», «TVshow», «Product», «Production», «Company», «Country», «Participation», «People», «Genre_product», «Genre». У кожній таблиці є свій список атрибутів (табл. 2.3 – табл. 2.12).

Зробимо розкриття для цих таблиць.

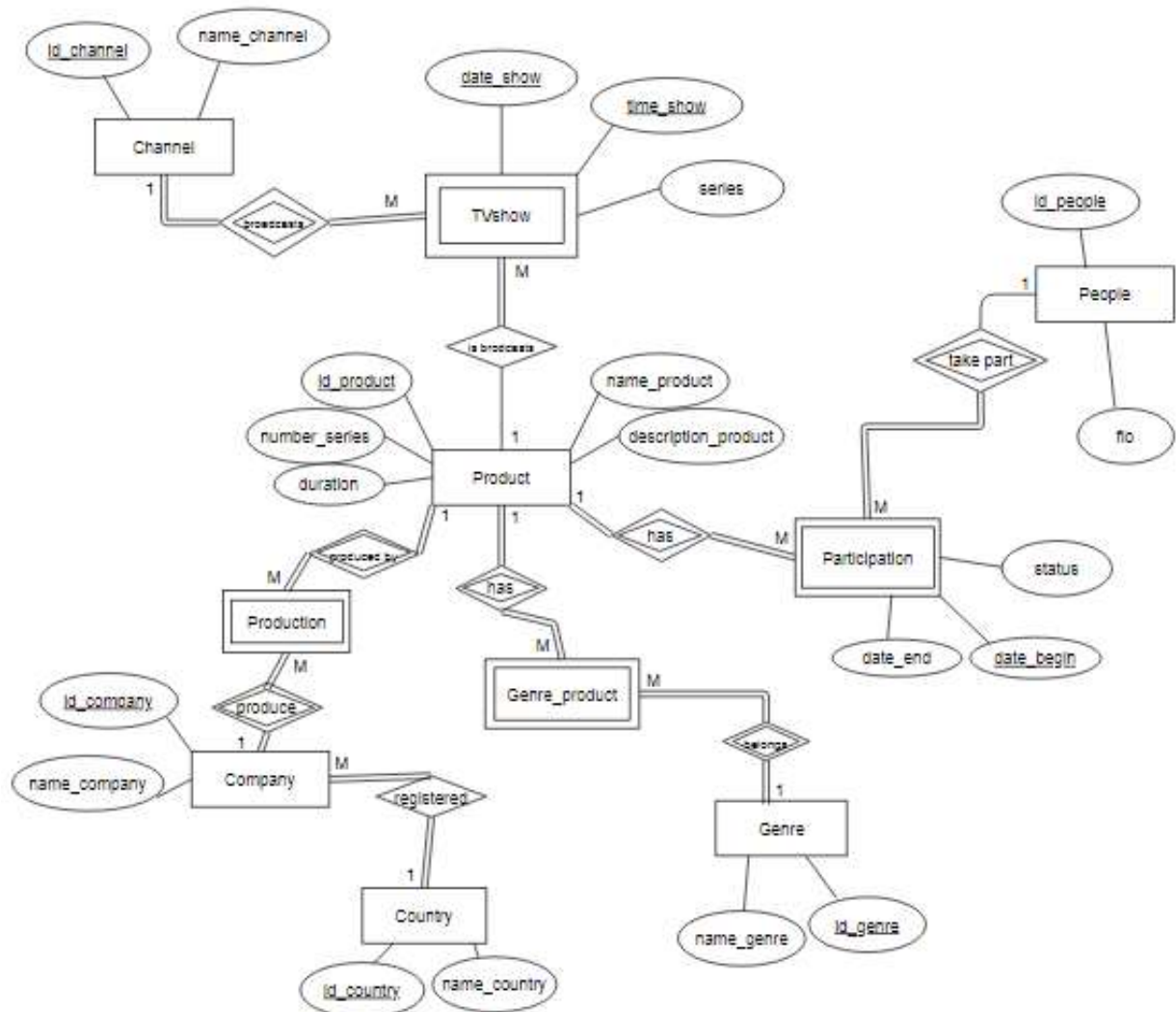


Рисунок 2.2 – Логічна модель даних

Таблиця 2.3 – Опис структури таблиці «Channel» («Канал»)

Ключ	Ім'я поля	Тип даних	Розмір поля	Опис
PK	id_channel	int	4	Унікальний код телеканалу
	name_channel	varchar	20	Назва телеканалу

Таблиця 2.4 – Опис структури таблиці «TVshow» («Телепрограма»)

Ключ	Ім'я поля	Тип даних	Розмір поля	Опис
PK, FK	id_channel	int	4	Код телеканалу
PK	date_show	date	3	Дата показу телепередачі
PK	time_show	time(7)	5	Час показу телепередачі
FK	id_product	int	4	Код продукту
	series	int	4	Номер випуску/серії

Таблиця 2.5 – Опис структури таблиці «Product» («Продукт»)

Ключ	Ім'я поля	Тип даних	Розмір поля	Опис
1	2	3	4	5
PK	id_product	int	4	Унікальний код продукту
	name_product	varchar	20	Назва телепродукту
	description_product	varchar	50	Опис продукту, може бути не заповнене

Продовження таблиці 2.5

1	2	3	4	5
	number_series	int	4	Кількість серій/випусків
	duration	time(7)	5	Тривалість

Таблиця 2.6 – Опис структури таблиці «Production» («Виробництво»)

Ключ	Ім'я поля	Тип даних	Розмір поля	Опис
PK, FK	id_product	int	4	Код продукту
PK, FK	id_company	int	4	Код компанії

Таблиця 2.7 – Опис структури таблиці «Company» («Компанія»)

Ключ	Ім'я поля	Тип даних	Розмір поля	Опис
PK	id_company	int	4	Унікальний код компанії
	name_company	varchar	20	Назва компанії
FK	id_country	int	4	Код країни, де зареєстрована компанія

Таблиця 2.8 – Опис структури таблиці «Country» («Країна»)

Ключ	Ім'я поля	Тип даних	Розмір поля	Опис
PK	id_country	int	4	Унікальний код країни
	name_country	varchar	20	Назва країни

Таблиця 2.9 – Опис структури таблиці «Participation» («Участь у продукті»)

Ключ	Ім'я поля	Тип даних	Розмір поля	Опис
PK, FK	id_product	int	4	Код продукту
PK, FK	id_people	int	4	Код людини
PK	date_begin	date	3	Дата початку роботи
	date_end	date	3	Дата закінчення роботи, може бути не заповнене
	status	varchar	20	Статус, у якому людина працює не продукті

Таблиця 2.10 – Опис структури таблиці «People» («Працівники»)

Ключ	Ім'я поля	Тип даних	Розмір поля	Опис
PK	id_people	int	4	Унікальний код людини
	fio	varchar	30	ПІБ працівника

Таблиця 2.11 – Опис структури таблиці «Genre_product» («Жанр продукту»)

Ключ	Ім'я поля	Тип даних	Розмір поля	Опис
PK, FK	id_product	int	4	Код продукту
PK, FK	id_genre	int	4	Код жанру

Таблиця 2.12 – Опис структури таблиці «Genre» («Жанр»)

Ключ	Ім'я поля	Тип даних	Розмір поля	Опис
РК	id_genre	int	4	Унікальний код жанру
	name_genre	varchar	40	Назва жанру

Розкриття таблиць:

- а) Channel (id_channel, name_channel);
- б) TVshow (id_channel, date_show, time_show, id_product, series);
- в) Product (id_product, name_product, description_product, number_series, duration);
- г) Production (id_product, id_company);
- д) Company (id_company, name_company, id_country);
- е) Country (id_country, name_country);
- ж) Participation (id_product, id_people, date_begin, date_end, status);
- к) People (id_people, fio);
- л) Genre_product (id_product, id_genre);
- м) Genre (id_genre, name_genre).

2.3 Моделювання структури та наповнення бази даних інформаційної системи вибраної предметної області

Беручи за основу концептуальну та логічну моделі, використаємо інструменти програми візуального проектування ERwin для побудови моделі даних (рис. 2.3).

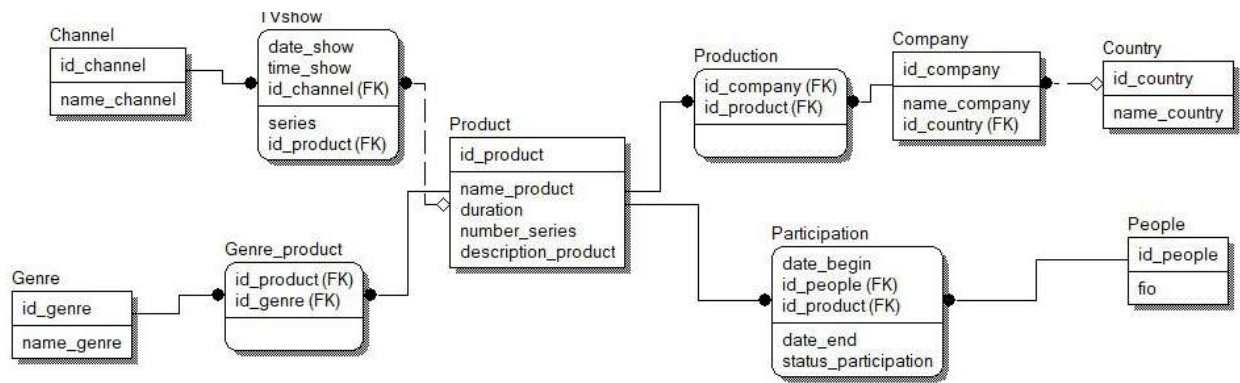


Рисунок 2.3 – Логічна модель в ERwin

2.4 Фізичне проєктування інформаційної системи вибраної предметної області

Фізичне проєктування – створення схеми бази даних для конкретної СУБД. Специфіка конкретної СУБД може включати в себе обмеження на іменування об’єктів бази даних, обмеження на підтримувані типи даних та інші [25-27].

Характеристики фізичної моделі даних:

- фізична модель даних описує потребу в даних для окремого проєкту або програми, хоча вона може бути інтегрована з іншими моделями фізичних даних на основі обсягу проєкту;
- модель даних містить зв’язки між таблицями, які вирішують потужність зв’язків і можливість їх нульового значення;
- розроблено для конкретної версії СУБД, розташування, сховища даних або технології, яка буде використовуватися в проєкті;
- стовпці повинні мати точні типи даних, призначені довжини та значення за замовчуванням;
- визначені первинні та зовнішні ключі, представлення даних, індекси, профілі доступу, авторизації тощо.

Після переносу моделі даних «Телепрограма» з ERwin у SQL Management Studio отримаємо результат у вигляді діаграми, зображеної на рисунку 2.4.

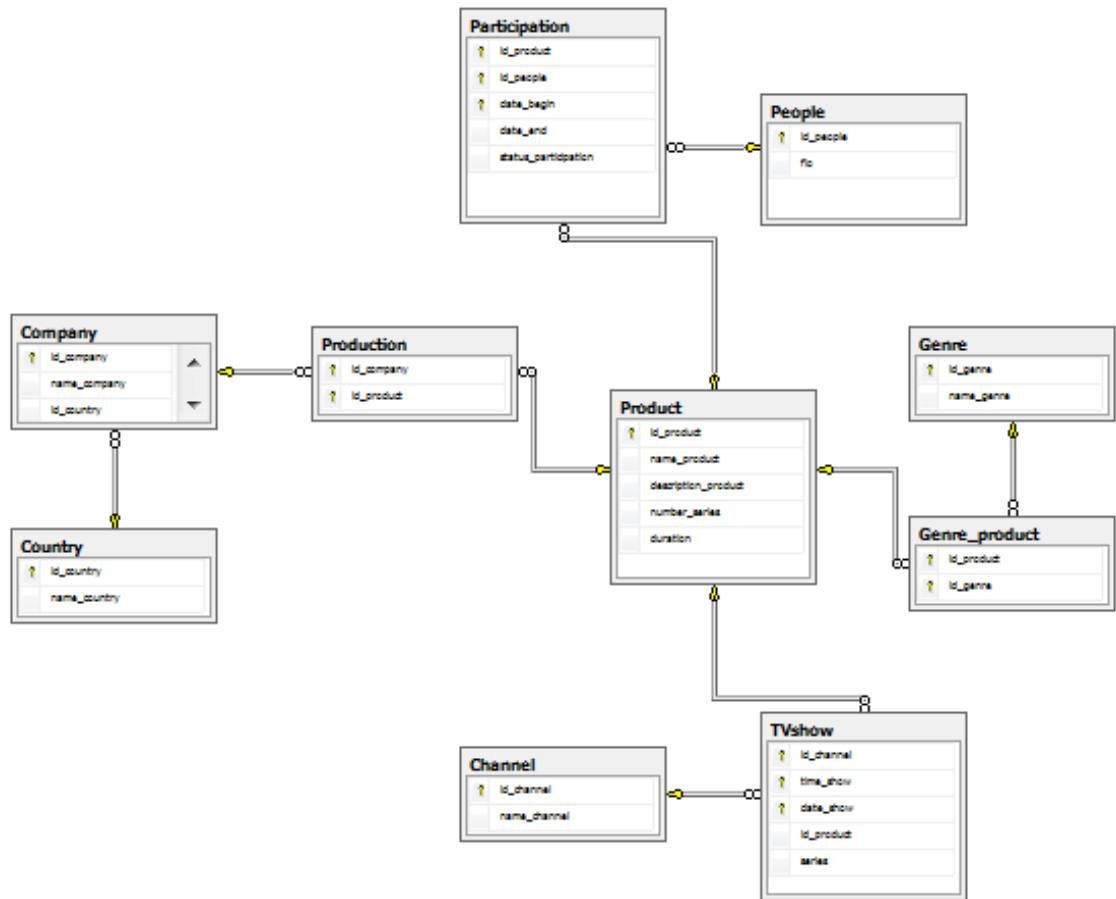


Рисунок 2.4 – Згенерована модель даних в SQL Server

Фізична структура бази даних для інформаційної системи «Телепрограма» наведена нижче за допомогою відповідних команд CREATE TABLE.

Лістинг 2.1 Створення таблиці «Channel»:

```
create table Channel
(id_channel int identity(1,1) constraint PK_Channel primary key,
name_channel varchar(30) not null unique)
```

Лістинг 2.2 Створення таблиці «Country»:

```
create table Country
(id_country int identity(1,1) constraint PK_Country primary key,
name_country varchar(30) not null unique)
```

Лістинг 2.3 Створення таблиці «Company»:

```
create table Company
(id_company int identity(1,1) constraint PK_Company primary key,
name_company varchar(30) not null unique,
id_country int not null
constraint FK_Company references Country(id_country)
on update cascade
on delete cascade)
```

Лістинг 2.4 Створення таблиці «Genre»:

```
create table Genre
(id_genre int identity(1,1) constraint PK_Genre primary key,
name_genre varchar(40) not null unique)
```

Лістинг 2.5 Створення таблиці «People»:

```
create table People
(id_people int constraint PK_People primary key,
fio varchar(40) not null constraint Check_People check (fio like '[А-Я]%'
[А-Я].[А-Я].'))
```

Лістинг 2.6 Створення таблиці «Product»:

```
create table Product
(id_product int identity(1,1) constraint PK_Product primary key,
name_product varchar(30) not null,
description_product varchar(50)not null,
number_series int not null,
duration time not null)
```

Лістинг 2.7 Створення таблиці «Production»:

```
create table Production
(id_company int constraint FK_Production1 references
Company(id_company) on update cascade on delete cascade,
id_product int constraint FK_Production2 references Product(id_product)
on update cascade on delete cascade,
constraint PK_Production primary key(id_company, id_product))
```

Лістинг 2.8 Створення таблиці «Participation»:

```
create table Participation
(id_product int constraint FK_Participation1 references
Product(id_product) on update cascade on delete cascade,
id_people int constraint FK_Participation2 references People(id_people) on
update cascade on delete cascade,
date_begin date,
date_end date,
status_participation varchar(30) not null,
constraint PK_Participation primary key(id_product, id_people,
date_begin))
```

Лістинг 2.9 Створення таблиці «TVshow»:

```
create table TVshow  
(id_channel int constraint FK_TVshow1 references Channel(id_channel) on  
update cascade on delete cascade,  
time_show time not null,  
date_show date not null,  
id_product int not null constraint FK_TVshow2 references  
Product(id_product) on update cascade on delete cascade,  
series int not null,  
constraint PK_TVshow primary key (id_channel, time_show, date_show));
```

Лістинг 2.10 Створення таблиці «Genre_product»:

```
create table Genre_product  
(id_product int constraint FK_Genre_product1 references  
Product(id_product) on update cascade on delete cascade,  
id_genre int constraint FK_Genre_product2 references Genre(id_genre) on  
update cascade on delete cascade,  
constraint PK_Genre_product primary key(id_product, id_genre));
```

Заповнимо створенні таблиці даними (рис. А.1 – рис. А.10).

3 МОДЕЛЮВАННЯ ТА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ОРГАНІЗАЦІЇ РОЗКЛАДУ ТЕЛЕПЕРЕДАЧ НА РІЗНИХ ТЕЛЕКАНАЛАХ

3.1 Вибір інструментальних засобів для реалізації

Інтегроване середовище розробки (IDE) – це програмне забезпечення, яке об'єднує основні інструменти, необхідні для тестування та написання програмного забезпечення.

Без IDE розробнику довелося б вибирати й керувати всіма цими інструментами окремо, але IDE об'єднує всі ці інструменти як єдиний фреймворк або сервіс [28-31].

IDE включає в себе такі основні функції:

- текстовий редактор – IDE – це перш за все текстовий редактор, який включає в себе кілька інших специфічних функцій;
- компілятор або інтерпретатор – інтегрована програма, яка перекладає код на машину мову для читання. Це дозволяє коду виконуватися, що називається виконанням сценарію. Інтерпретатор – це програма, яка входить в комплект. Його мета полягає у виконанні коду без необхідності його попередньої компіляції [32, 33];
- встроювання або створення інтеграції – це включені способи автоматизації необхідних процесів;
- налагоджувач (debugger) – включена програма, яка перевіряє код на наявність помилок для їх усунення;
- підсвічування синтаксису – частини коду відображаються кольорами, які відповідають різним елементам, таким як властивості, теги, атрибути та все інше. Це допомагає візуально полегшити сканування коду та побачити, де допущені невеликі помилки, наприклад, не закритий тег;

- графічний користувальницький інтерфейс (GUI) – це екран, який легко читається людиною і за яким легко переміщатися, а також виконувати завдання. Він може містити кнопки, меню та інші подібні елементи, а не лише текст і можливість ввести текстову команду [34, 35];

- інші можливі функції – багато IDE розробляють і включають нові функції, яких інші IDE можуть не пропонувати.

Існує безліч IDE для різних цілей та різних мов програмування. Серед них зокрема можна виділити лінійку продуктів компанії Microsoft – Microsoft Visual Studio, продукти компанії JetBrains, такі як: IntelliJ IDEA, PhpStorm, PyCharm, WebStorm тощо, Oracle Application Express, Microsoft SQL Server та pgAdmin для моделювання та створення баз даних.

Для реалізації інформаційної системи для організації розкладу телепередач на різних телеканалах було обрано такі програмні засоби:

- Microsoft Visual Studio – для написання застосунку застосунку з графічним інтерфейсом на мові C#;

- Microsoft SQL Server та SQL Server Management Studio – для реалізації бази даних;

- ERwin – для проєктування бази даних, створення моделей даних та діаграм Чена.

3.2 Етапи реалізації інформаційної системи для організації розкладу телепередач на різних телеканалах

Під час запуску програми відкривається головна форма, зображена на (рис. Б.1), яка представляє собою вікно для авторизації. В залежності від логіну і паролю, який введе користувач, відкриється конкретна дій, які дозволені його статусу. Усього їх 3 – адміністратор, укладач розкладу та простий користувач. Користувач може зайти без авторизації за допомогою кнопки «Увійти як гість».

Після авторизації відкривається вікно «Телепрограма», де користувач може знайти телепрограму за будь-яким смаком (рис. Б.2).

Якщо зайти як укладач розкладу, то відкриються нові кнопки, за допомогою яких можна додати, змінити та видалити інформацію з розкладу (рис. Б.3 – рис. Б.5).

Якщо увійти як адміністратор, то додаються кнопки, завдяки яким можна додавати та видаляти дані з усіх таблиць (рис. Б.6, рис. Б.7).

Для кращого розуміння можливостей різних користувачів побудуємо UML діаграму, яка наглядно ілюструє загальні та універсальні можливості роботи з інформацією (рис. 3.1).



Рисунок 3.1 – Діаграма реалізації можливостей

Для зображення функціональності ІС сформуємо основні SQL запити, які використовуються в програмі.

Лістинг 3.1 Запит для додавання розкладу:

```
insert into TVshow (id_channel, date_show, time_show, id_product, series)
values(" + num1 + ", " + dateTimePicker1.Value + ", " +
numericUpDown1.Value + ":" + numericUpDown2.Value + ":00', " + num2 + ",
" + Convert.ToInt32(textBox1.Text) + ")
```

Лістинг 3.2 Запит для видалення розкладу:

```
delete from TVshow where id_channel=" + num1 + " and date_show=" +
dataGridView1.CurrentRow.Cells["Дата"].Value.ToString() + " and
time_show=" + dataGridView1.CurrentRow.Cells["Время"].Value.ToString() + "
```

Лістинг 3.3 Запит для оновлення розкладу:

```
update TVshow set id_product = " + num2 + ", series = " +
Convert.ToInt32(textBox4.Text) + " where id_channel = " + num1 + " and
date_show = " + dataGridView1.CurrentRow.Cells["Дата"].Value.ToString() +
" and time_show=" +
dataGridView1.CurrentRow.Cells["Время"].Value.ToString() + "
```

Лістинг 3.4 Запит для перегляду детальної інформації про телепродукт:

```
select name_product, description_product, number_series, duration from
Product where name_product in (" +
dataGridView1.CurrentRow.Cells["Продукт"].Value.ToString()+")
```

```
select name_genre from (Genre inner join Genre_product on
Genre.id_genre=Genre_product.id_genre) inner join Product on
Product.id_product=Genre_product.id_product where name_product in (" +
dataGridView1.CurrentRow.Cells["Продукт"].Value.ToString() + ")")
```

Лістинг 3.5 Запит для перегляду розкладу телепередач на свій смак:

```
select distinct Channel.name_channel as [Канал], date_show as [Дата],
time_show as [Время], Product.name_product as [Продукт], series as
[Выпуск/серия] from (((Channel inner join TVshow on
Channel.id_channel=TVshow.id_channel) inner join Product on
TVshow.id_product=Product.id_product) inner join Genre_product on
Product.id_product=Genre_product.id_product) inner join Genre on
Genre.id_genre=Genre_product.id_genre where Channel.name_channel in (" +
str1 + ") and name_genre in (" + str2 + ") and date_show between
convert(date,GETDATE()) and convert(date,(GETDATE()+ " + num + ")) order by
date_show, Channel.name_channel, time_show
```

3.3 Тестування розробленої інформаційної системи та аналіз результатів

Програмний застосунок був протестований чотирма одногрупниками. Були знайдені декілька багів, які були виправлені. Наприклад, виправлена можливість додавання в розклад підряд одного і того ж телепродукту, тобто раніше можна було ставити підряд одну і ту ж серію серіалу. Тепер це зробити неможливо завдяки додатковим перевіркам. Також виправлено зауваження стосовно редагування в розкладі програми, яка вже вийшла в ефір. Тепер такі програми захищені від редагування.

Також були надані пропозиції щодо покращення розробленої інформаційної системи.

Застосунок має зрозумілий та простий інтерфейс для користування, тому питань з цього приводу не було.

Реєстрація та авторизація користувачів пройшли успішно. Користувачі з роллю адміністратора можуть налаштовувати таблиці, не зламуючи логіку системи.

3.4 Перспективи подальшої роботи

Завдяки тестуванню та відгукам одногрупників було визначено перелік питань для покращення інформаційної системи та застосунку. В подальшому можна:

- додати можливість онлайн-оновлення розкладу без натискання на кнопку «Переглянути телепрограму»;
- додати підсвічування програми, яка йде зараз в ефірі;
- оновити дизайн, додати декілька тем для різних категорій користувачів;
- додати у застосунок безпосередньо трансляцію для користувача;
- розширити та масштабувати систему завдяки додаванням реєстрації медіахолдінгу. Кожен телеканал є частиною якогось конкретного медіахолдінгу, і складати розклад для нього та змінювати дані можуть лише укладачі розкладу та адміністратори цього медіахолдінгу відповідно.

ВИСНОВКИ

У рамках кваліфікаційної роботи було змодельовано та реалізовано інформаційну систему для організації розкладу телепередач на різних телеканалах.

Дана інформаційна система дозволяє додавати інформацію до розкладу та переглядати її, використовуючи пошук за каналами, жанрами, та обираючи час, на який необхідно переглянути розклад. Має 3 рівня доступу: адміністратор, укладач розкладу та користувач.

Система дозволяє виконати вхід від адміністратора для налаштування усіх таблиць ІС «Телепрограма», від укладача розкладу – для складання розкладу телепередач, від звичайного користувача – для перегляду розкладу.

Систему було протестовано, за результатами чого було визначено перелік питань для подальшого покращення системи.

Результати роботи апробовано у вигляді тез доповідей під час XIX Міжнародної науково-практичної конференції «Modern problems in science» [36].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ahmad M.A., Tvoroshenko I., Baker J.H., Kochura L., Lyashenko V. (2020) Interactive Geoinformation Three-Dimensional Model of a Landscape Park Using Geoinformatics Tools, *International Journal on Advanced Science, Engineering and Information Technology*, 10(5), pp. 2005-2013.
2. Творошенко І.С. (2021) Технології прийняття рішень в інформаційних системах: навч. посібник. Харків: ХНУРЕ, 120 с.
3. Tvoroshenko, I. (2019). Development of models of spatial analysis of status of interactive processes of complex systems.
4. Кучеренко Е.И., Творошенко И.С. (2003). Процессы принятия решений в сложных системах на основе нечетких интервальных представлений. *Вісник Національного технічного університету «ХПІ»*. Тематичний випуск: Системний аналіз, управління та інформаційні технології. Х.: НТУ «ХПІ», 1(7), С. 79-86.
5. Гороховатський В.О., Творошенко І.С. (2021) Методи інтелектуального аналізу та оброблення даних: навч. посібник. Харків: ХНУРЕ, 92 с.
6. Кучеренко Є.І., Творошенко І.С., Анопрієнко Т.В. (2016) Моделювання та оцінювання станів складних об'єктів із застосуванням формальної логіки, *Системи обробки інформації*, 2, С. 76-82.
7. Tvoroshenko I., and Babochkin O. (2021). Object identification method based on image keypoint descriptors.
8. Творошенко І. С., Табашник В. А. (2018) Розробка просторової моделі геоінформаційної підтримки людей з обмеженими можливостями, що пересуваються на інвалідних колясках, у місті Харків. *Збірник наукових праць Харківського національного університету Повітряних Сил*, 1, 122-128.
9. Творошенко И.С. (2010) Анализ процессов принятия решений в интеллектуальных системах, *Системи обробки інформації*, 2, С. 248-253.

10. Tvoroshenko I., Ahmad M.A., Mustafa S.K., Lyashenko V., and Alharbi A.R. (2020) Modification of Models Intensive Development Ontologies by Fuzzy Logic, *International Journal of Emerging Trends in Engineering Research*, 8(3), pp. 939-944.
11. Lyashenko V., Mustafa S.K., Tvoroshenko I., and Ahmad M.A. (2020) Methods of Using Fuzzy Interval Logic During Processing of Space States of Complex Biophysical Objects, *International Journal of Emerging Trends in Engineering Research*, 8(2), pp. 372-377.
12. Gorokhovatskyi, V., Rusakova, N., and Tvoroshenko, I. (2020) The application of image analysis methods and predicate logic in applied problems of magnetic monitoring, *Telecommunications and Radio Engineering*, 79(20), pp. 1801-1811.
13. Ahmad M.A., Gorokhovatskyi V., Tvoroshenko I., Vlasenko N., Mustafa S.K. (2021) The Research of Image Classification Methods Based on the Introducing Cluster Representation Parameters for the Structural Description, *International Journal of Engineering Trends and Technology*, 69(10), pp. 186-192.
14. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Al-Dhaifallah M. (2022) Classification of Images Based on a System of Hierarchical Features, *Computers, Materials & Continua*, 72(1), pp. 1785-1797.
15. Tvoroshenko I., and Gorokhovatskyi V. (2022) The Application of Hybrid Intelligence Systems for Dynamic Data Analysis, *International Journal of Engineering and Information Systems*, 6(2), pp. 40-48.
16. Tvoroshenko I., and Tkachenko D. (2020) Mechanisms of image classification based on descriptors of local features, *Abstracts of IV International Scientific and Practical Conference «Integration of scientific bases into practice»* (October 12-16, 2020). Stockholm, Sweden, pp. 443-448.
17. Garcia-Molina, H. (2008). *Database systems: the complete book*. Pearson Education India.
18. Date, C. J. (1975). *An introduction to database systems*. Pearson Education India.

19. Connolly, T. M., & Begg, C. E. (2005). *Database systems: a practical approach to design, implementation, and management*. Pearson Education.
20. Ramez, E., & Shamkant B, N. (2022). Fundamentals of database systems.
21. Гороховатський В.О., Творошенко І.С. (2022) Аналіз багатовимірних даних за описом у формі множини компонент: монографія. Харків: ХНУРЕ. 124 с.
22. Bogusławski, A., Veselý, P., Husenicová, L., & Čupka, O. (2022). Creating Database Models in Rational Data Architect. In *Developments in Information & Knowledge Management for Business Applications* (pp. 731-766). Springer, Cham.
23. Gupta, S., Pal, S., & Chakraborty, M. (2019, August). A Study on Various Database Models: Relational, Graph, and Hybrid Databases. In *International Ethical Hacking Conference* (pp. 141-149). Springer, Singapore.
24. Bjeladinovic, S., Marjanovic, Z., & Babarogic, S. (2020). A proposal of architecture for integration and uniform use of hybrid SQL/NoSQL database components. *Journal of Systems and Software*, 168, 110633.
25. Bjeladinovic, S. (2018). A fresh approach for hybrid SQL/NoSQL database design based on data structuredness. *Enterprise Information Systems*, 12(8-9), 1202-1220.
26. Gimadiev, T., Nugmanov, R., Batyrshin, D., Madzhidov, T., Maeda, S., Sidorov, P., & Varnek, A. (2021). Combined graph/relational database management system for calculated chemical reaction pathway data. *Journal of Chemical Information and Modeling*, 61(2), 554-559.
27. Зінов'єва, І. С., & Артемчук, В. О. (2019). Сучасні підходи до подальшої еволюції концепції баз даних. In *III International scientific and practical conference «Dynamics of the development of world science»*, Canada (pp. 34-44).

28. Yousef Ibrahim Daradkeh, and Iryna Tvoroshenko (2020) Application of an Improved Formal Model of the Hybrid Development of Ontologies in Complex Information Systems, *Applied Sciences*, 10(19). p. 6777.

29. Kobylin O., Gorokhovatskyi V., Tvoroshenko I., and Peredrii O. (2020) The application of non-parametric statistics methods in image classifiers based on structural description components, *Telecommunications and Radio Engineering*, 79(10), pp. 855-863.

30. Daradkeh, Y.I., Tvoroshenko, I., Gorokhovatskyi, V., Latiff, L.A., and Ahmad, N. (2021) Development of Effective Methods for Structural Image Recognition Using the Principles of Data Granulation and Apparatus of Fuzzy Logic, *IEEE Access*, 9, pp. 13417-13428.

31. Tvoroshenko I., and Gorokhovatskyi V. (2022) The Application of Hybrid Intelligence Systems for Dynamic Data Analysis, *International Journal of Engineering and Information Systems*, 6(2), pp. 40-48.

32. Tvoroshenko, I., & Almakaieva, A. (2020). Application of procedural generation of game content using software algorithms.

33. Tvoroshenko, I., & Andrieieva, A. (2021). Development of web applications for remote learning of English.

34. Tvoroshenko, I., & Temchur, K. (2021). Features of software application development for food recognition using deep machine learning methods.

35. Tvoroshenko, I., & Kukharchuk, V. (2021). Current state of development of applications for recognition of faces in the image and frames of video captures.

36. Tarasenko, D. (2022). Analysis of the main stages of the project management life cycle and project management approaches.