

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка автоматизованої системи управління для санаторію-профілакторію «Барвінок» Зборівського району

Виконав: студент IV курсу, групи СНз-41

спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

(підпис)

Боднар М.О.

(прізвище та ініціали)

Керівник

(підпис)

Щербак Л.М.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Шимчук Г.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Лупенко С.А.

(прізвище та ініціали)

Тернопіль
2021

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)

«__» _____ 2021 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Боднару Миколі Орестовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка автоматизованої системи управління для санаторію-профілакторію «Барвінок» Зборівського району

Керівник роботи Щербак Леонід Миколайович, д.т.н., професор кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «02» 03 2021 року № 4/7-170

2. Термін подання студентом завершеної роботи 14.06 2021р.

3. Вихідні дані до роботи Наукові літературні джерела

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ, Розділ 1. Аналіз предметної області, 1.1 Теоретичне обґрунтування, 1.2 Важливість даних, 1.5 Основи проектування БД, 1.6 Висновок до першого розділу, Розділ 2. Практична реалізація, 2.1 Огляд систем керування базами даних, 2.2 Огляд медичних інформаційних систем, 2.3 Опис об'єкту автоматизації, 2.4 Структурна схема розроблюваної системи, 2.5 Реалізація системи, 2.6 Демонстрація та результати роботи системи, 2.7 Висновок до другого розділу, Розділ 3. Безпека життєдіяльності, основи хорони праці, висновки, перелік літературних джерел

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1 Тема, мета, завдання. 2. Схема зв'язків бази даних «реєстратура», 3. Структурні схеми модулів програми. 4. Блок-схема алгоритму роботи програми. 5. Перелік таблиць бази даних реєстратура. 6. Схема зв'язків між таблицями. 7. Головне вікно програми «реєстратура». 8. Форма для введення даних про пацієнта. 9. Форма для пошуку даних. 10. Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці	Гурик О.Я., доцент кафедри МТ		

7. Дата видачі завдання 17 травня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	17.05.2021	<i>Виконано</i>
2.	Аналіз літературних джерел	18.05.2021- 19.05.2021	<i>Виконано</i>
3.	Виконання дослідження щодо розробки автоматизованої системи управління	20.05.2021- 21.05.2021	<i>Виконано</i>
4.	Оформлення розділу «Аналіз предметної області»	22.05.2021- 28.05.2021	<i>Виконано</i>
5.	Оформлення розділу «Практична реалізація»	29.06.2021- 01.06.2021	<i>Виконано</i>
6.	Виконання завдання до підрозділу «Безпека життєдіяльності»	2.06.2021	<i>Виконано</i>
7.	Виконання завдання до підрозділу «Основи хорони праці»	2.06.2021	<i>Виконано</i>
8.	Оформлення кваліфікаційної роботи	6.06.2021	<i>Виконано</i>
9.	Нормоконтроль	7.06.2021	<i>Виконано</i>
10.	Перевірка на плагіат	7.06.2021	<i>Виконано</i>
11.	Попередній захист кваліфікаційної роботи	11.06.2021	<i>Виконано</i>
12.	Захист кваліфікаційної роботи	14.06.2021	

Студент

_____ (підпис)

Боднар М.О.

_____ (прізвище та ініціали)

Керівник роботи

Щербак Л.М..

АНОТАЦІЯ

Розробка автоматизованої системи управління для санаторію-профілакторію «Барвінок» Зборівського району // Кваліфікаційна робота освітнього рівня «Бакалавр» // Боднар Микола Орестович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНз-41 // Тернопіль, 2021 // С. – рисунок – , таблиці – кресл. – , додат. – , бібліогр. – .

Ключові слова: Delphi, Acces, АІС, БД, СКБД, ПЕОМ, ODBC, ІМ

Об'єкт аналізу – медичні інформаційні системи. Мета роботи – проектування автоматизованої системи санаторію-профілакторію, розробка програмного забезпечення до задачі створення АІС, пошуку інформації в БД.

Основні результати – АІС, сучасні комп'ютерні технології, БД. Розроблена теоретична модель проекту автоматизованої системи ведення ведення обліку санаторію-профілакторію. Розроблена структурна схема модулів системи, схема бази даних. Проведений аналіз існуючих медичних інформаційних систем. Розроблені завдання виконано засобами Delphi та Access. Розроблено програмне забезпечення до задачі збереження та редагування даних, пошуку інформації.

ANNOTATION

Automatic control system development for the health care center “Barvinok” Zboriv district // Qualification work of educational level «Bachelor» // Bodnar Mykola Orestovych // Ternopil’ Ivan Pul’uj National Technical University, Faculty of Computer Information System and Software Engineering, Department of Computer Science, group SNz-41 // Ternopil', 2021 // P. , Fig. – , Tables – , References – Annexes. – .

Keywords: Delphi, Acces, AIS, DB, DBMS, PC, ODBC, IM

The object of analysis is medical information systems. The purpose of work - designing of the automated system of sanatorium-dispensary, development of the software to a task of creation of AIS, search of the information in a DB.

The main results - AIS, modern computer technology, database. The theoretical model of the project of the automated system of conducting the account of sanatorium-dispensary is developed. The structural scheme of modules of system, the scheme of a database is developed. The analysis of existing medical information systems is carried out. The developed tasks were performed using Delphi and Access. Developed software for the task of saving and editing data, information retrieval.

ЗМІСТ

ВСТУП.....	10
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1 Теоретичне обґрунтування.....	11
1.2 Важливість дизайну бази даних.....	12
1.3 Життєвий цикл розробки баз даних	13
1.4 Методи проектування баз даних.....	14
1.5 Основи проектування БД	17
1.6 Висновок до першого розділу.....	25
РОЗДІЛ 2. ПРАКТИЧНА РЕАЛІЗАЦІЯ	26
2.1 Огляд систем керування базами даних	26
2.2 Огляд медичних інформаційних систем	27
2.3 Опис об'єкту автоматизації.....	29
2.3.1 Характеристика предметної області	29
2.4 Структурна схема розроблюваної системи	32
2.4.1 Постановка задачі	32
2.4.2 Опис функціональної структури автоматизованої підсистеми ...	33
2.5 Реалізація системи	39
2.5.1 Модель бази даних в Delphi	39
2.5.2 Створення таблиці	39
2.5.3 Програма управління базою даних	42
2.5.4 Механізм з'єднання з сховищем даних ADO	44
2.5.5 Приклад зв'язку з Access через ADO	45
2.6 Демонстрація та результати роботи системи.....	46
2.7 Висновок до другого розділу	49
РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ ...	50
3.1 Гігієнічні вимоги до організації обладнання робочих місць з ВДТ ..	50
3.2 Дія електричного струму на організм людини, види електротравм ..	51
3.3 Безпека життєдіяльності в екстремальних умовах	54

	9
3.4 Наслідки забруднення навколишнього середовища.....	57
ВИСНОВКИ	61
ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	62

ВСТУП

Актуальність теми. Задача обробки великих обсягів інформації вже давно є однією з найболючіших проблем організації системи охорони здоров'я в Україні. У структурі майже кожного лікувального закладу існує ряд вузлів, куди стікаються потоки інформації, яку слід зберігати та обробляти.

Мета і задачі дослідження. Провести розробку автоматизованої системи управління і вирішити такі завдання:

1. розглянути існуючі інформаційні системи в галузі медицини.
2. проаналізувати об'єкт дослідження.
3. розглянути існуючі інформаційні системи ведення обліку відділу реєстратури.
4. враховувати недоліки попередньої системи.
5. розробити автоматизована система ведення обліку пацієнтів для відділу реєстратури лікувально-профілактичного закладу.

Практичне значення одержаних результатів. В даній кваліфікаційній роботі освітнього рівня «Бакалавр» розглянуто програмне рішення що до організації баз даних (БД) для відділу реєстратури у медичних закладах. Для роботи була використана БД – Microsoft Access, а додаток управління нею розроблений на мові програмування Delphi, використовуючи механізм з'єднання з сховищем даних ADO. За допомогою даного програмного продукту користувач (працівник відділу реєстратури) має можливість вносити у базу нові дані, змінювати їх, сортувати та здійснювати пошук по заданих критеріях. Для зручного представлення даних та їх використання розроблені відповідні засоби інтерфейсу.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Теоретичне обґрунтування

База даних включає об'ємну інформацію, депоновану в рамках, що полегшує пошук та вивчення відповідної інформації. Добре розроблена база даних містить точну та сучасну інформацію, що дозволяє легко отримувати дані за потреби. Легко зрозуміти важливість бази даних для компанії, яка регулярно займається масами даних. Однак важливо зазначити, що для цього потрібен дизайн даних, який може зробити аналіз швидшим та надійнішим.

Дизайн бази даних визначається як сукупність кроків, які допомагають розробляти, створювати, впроваджувати та підтримувати системи управління даними бізнесу.

Хороший процес проектування баз даних регулюється певними правилами. Перше правило вимагає уникати зайвих даних; оскільки це марно витрачає простір і збільшує ймовірність несправностей та розбіжностей у базі даних. Наступне правило полягає в тому, що точність та вичерпність інформації вкрай необхідні. Якщо база даних містить помилкову інформацію, будь-які документи, які отримують дані з такої бази даних, також включатимуть недостовірну інформацію. Отже, будь-які рішення, засновані на цих документах, будуть вводити в оману, тим самим збільшуючи важливість дизайну бази даних, який враховує всі вищезазначені правила.

Отже, як ви можете переконатися, що ваш дизайн бази даних хороший? Добре розроблена база даних - це та, яка:

- розподіляє ваші дані у таблиці на основі конкретних предметних областей, щоб зменшити надмірність даних;
- надає базі даних інформацію, необхідну для зв'язку даних у таблицях;
- забезпечує підтримку та гарантує точність та надійність даних;
- задовольняє ваші вимоги щодо обробки інформації та звітності;

– максимально функціонує в інтерактивному режимі з операторами баз даних.

1.2 Важливість дизайну бази даних

Дизайн бази даних визначає структуру бази даних, яка використовується для планування, зберігання та управління інформацією. Точність даних може бути досягнута лише в тому випадку, якщо база даних призначена для зберігання лише цінної та необхідної інформації.

Добре розроблена база даних має обов'язкове значення для забезпечення послідовності інформації, усунення зайвих даних, ефективного виконання запитів та покращення продуктивності бази даних. Скрупульозне проектування бази даних позбавляє вас від втрати часу та розчарувань на етапі розробки бази даних. Хороший дизайн бази даних також дозволяє легко отримувати доступ та отримувати дані, коли це потрібно.

Надійність даних залежить від структури таблиці; тоді як створення первинних та унікальних ключів гарантує однаковість збереженої інформації. Реплікації даних можна уникнути шляхом формування таблиці ймовірних значень та використання ключа для позначення значення. Отже, коли значення змінюється, зміна відбувається лише один раз в головній таблиці.

Оскільки загальна ефективність бази даних залежить від її конструкції, хороший дизайн бази даних використовує прості запити та швидшу реалізацію. Його легко підтримувати та оновлювати; тоді як виправлення тривіальних перебоїв у поганому дизайні бази даних може зашкодити збереженим подіям, представленням та утилітам.

1.3 Життєвий цикл розробки баз даних

У розробці баз даних існують різні етапи. Однак не потрібно послідовно виконувати кожен із етапів. Життєвий цикл можна розділити на три етапи: аналіз потреб, проектування баз даних та впровадження.

1. Аналіз вимог

Аналіз вимог вимагає двох кроків:

- Планування. На цьому етапі визначається план всього життєвого циклу розробки баз даних. Це також вимагає аналізу стратегії інформаційних систем організації.
- Визначення системи. Цей етап пояснює та викладає сферу застосування запропонованої системи баз даних.

2. Проектування бази даних

Фактичне проектування бази даних враховує дві ключові моделі:

- логічна модель, вона стосується використання заданих вимог для створення моделі бази даних. Повна конструкція викладена на папері на цьому етапі, без урахування будь-яких конкретних вимог СУБД або їх фізичної реалізації.
- Фізична модель, цей етап настає після логічної моделі і, отже, передбачає фізичну реалізацію логічної моделі. Вона враховує СУБД та інші фактори фізичної реалізації.

3. Впровадження

Етап реалізації життєвого циклу розробки баз даних стосується:

- Перетворення та завантаження даних: воно включає імпорт даних та перетворення даних, що надходять зі старої системи в нову базу даних.
- Тестування: нарешті, на цьому етапі виявляються помилки в новій системі та переконуються, що всі вимоги до бази даних виконуються.

1.4 Методи проектування баз даних

Дві найпоширеніші техніки, використовувані для проектування бази даних, включають:

- нормалізація: таблиці організовані таким чином, що це зменшує надмірність даних та залежність. Більші таблиці поділяються на менші таблиці і зв'язуються між собою за допомогою взаємозв'язків.
- моделювання взаємозв'язків між сутностями (ER): це графічний підхід до проектування баз даних, що моделює сутності, їх атрибути та визначає взаємозв'язки між цими сутностями для позначення об'єктів реального життя. Сутність - це будь-який предмет із реального світу, який відрізняється або унікальний від оточення.

Як створити базу даних: кроки проектування бази даних

Проектування бази даних зазвичай починається з визначення мети вашої бази даних. Потім відповідні дані збираються та упорядковуються у таблиці. Далі ви вказуєте первинні ключі та аналізуєте зв'язки між різними таблицями для ефективного проектування даних. Після уточнення таблиць останнім кроком є застосування правил нормування для стандартизації таблиць.

Давайте розглянемо ці кроки проектування бази даних:

- Визначте мету вашої бази даних.

Першим кроком є визначення мети вашої бази даних. Наприклад, якщо ви малий домашній бізнес, ви можете розробляти базу даних клієнтів, яка веде список споживчої інформації для створення електронних листів та звітів. Отже, розуміння важливості бази даних є життєво важливим.

Наприкінці цього кроку у вас буде сильна заява про місію, на яку ви можете посилатись протягом усього процесу проектування бази даних. Це допоможе вам зосередитися на своїх цілях під час прийняття важливих рішень.

- Знайдіть і консолідуйте необхідні дані.

Наступним кроком є збір всілякої інформації, яку ви можете захотіти зберегти в базі даних. Почніть із поточної інформації. Подумайте над питаннями, на які ви хочете відповісти у своїй базі даних, і це допоможе вам вирішити, які дані потрібно записати.

- Розподіліть дані по таблицях.

Після того, як ви накопичите всі необхідні елементи даних, наступним кроком буде розподіл їх на основні сутності або предметні області. Наприклад, якщо ви роздрібний торговець, деякими основними організаціями можуть бути товари, клієнти, постачальники та замовлення. Потім кожна сутність стане окремою таблицею.

- Змінити елементи даних у стовпці.

Дані розділено на таблиці, так що кожен елемент даних стає полем і відображається у вигляді стовпця. Наприклад, таблиця Клієнта може містити такі поля, як ім'я, адреса, адреса електронної пошти та місто.

Визначивши попередній набір стовпців для кожної таблиці, ви можете їх уточнити. Наприклад, ім'я замовника може бути записано як два окремі стовпці: ім'я та прізвище. Так само ви можете зберігати адресу у п'яти окремих стовпцях на основі адреси, міста, штату, поштового індексу та регіону. Це дозволить вам зручно фільтрувати інформацію.

- Визначте первинні ключі.

Наступним кроком для вдосконалення дизайну бази даних є вибір первинного ключа для кожної таблиці. Цей первинний ключ - це стовець або набір стовпців, який використовується для чіткого визначення кожного рядка. Наприклад, у вашій таблиці клієнтів первинним ключем може бути ідентифікатор клієнта. Це дозволить вам однозначно ідентифікувати кожен рядок на основі ідентифікатора клієнта.

Також може існувати більше одного первинного ключа, який називається складеним ключем, включаючи кілька стовпців. Наприклад, у вашій таблиці Деталі замовлення первинними ключами можуть бути

ідентифікатор замовлення та ідентифікатор товару. Складений ключ можна зробити за допомогою полів із подібними або різними типами даних.

Подібним чином, якщо ви хочете отримати уявлення про продажі вашої продукції, ви можете ідентифікувати ідентифікатор товару в таблиці Товари та номер замовлення або ідентифікатор у таблиці Замовлення.

- Визначте, як пов'язані таблиці.

Поділивши дані на таблиці, інформацію слід зібрати значущим чином. Отже, вивчіть кожну таблицю та визначте, як дані однієї таблиці пов'язані з даними іншої таблиці. За потреби можна додати поля або сформувати нові таблиці, щоб спростити взаємозв'язок на основі типів інформації.

На цьому кроці ви створите взаємозв'язок "один-до-одного", «один до багатьох» та/або «багато-до-багатьох» між різними записами таблиці.

Коли лише один елемент із таблиці пов'язаний з елементом з іншої таблиці, це називається співвідношенням один до одного (1:1). У взаємозв'язку «один до багатьох» (1:M) елемент в одній таблиці пов'язаний з багатьма елементами в іншій таблиці, наприклад один клієнт, який робить кілька замовлень. Зв'язок багато-до-багатьох (M:N) виникає, якщо багато елементів з однієї таблиці пов'язані з багатьма елементами в інших таблицях.

- Покращення дизайну бази даних.

Тепер, коли у вас є всі необхідні таблиці, поля та зв'язки, наступним кроком є вдосконалення дизайну бази даних шляхом створення та заповнення таблиць інформацією про макет. Експериментуйте із зразками даних, створюючи запити або додаючи нові елементи. Це допоможе вам проаналізувати вашу конструкцію на наявність несправностей, і ви зможете виділити можливі помилки. За потреби скоригуйте дизайн, щоб пом'якшити ці проблеми.

- Впровадити норми нормування.

Останнім кроком є впровадження правил нормалізації для проектування вашої бази даних. Це систематичний підхід, який усуває

надмірність та небажані характеристики, такі як неправильність вставки, оновлення та видалення.

Цей багатоступеневий процес зберігає дані у табличній формі, виключаючи надлишкові дані з таблиць відношень.

1.5 Основи проектування БД

Щоб розробити схеми баз даних, згадаймо 7 нормальних форм та самі концепції нормалізації та денормалізації. Вони лежать в основі всіх правил дизайну.

Зробимо опис 7 нормальних форм:

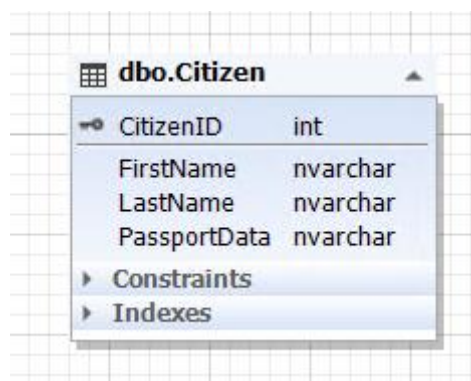
1. Персональні стосунки.

Обов'язкові відносини

Прикладом може бути громадянин з паспортом (кожен громадянин повинен мати паспорт, а паспорт є одним для кожного громадянина).

Цей взаємозв'язок реалізується двома шляхами:

В одному об'єкті (таблиця).



dbo.Citizen	
CitizenID	int
FirstName	nvarchar
LastName	nvarchar
PassportData	nvarchar
▶ Constraints	
▶ Indexes	

Рисунок 1.1 – Суб'єкт «громадянин»

Тут таблиця Citizen представляє сутність громадянина, а атрибут (поле) PassportData містить усі паспортні дані громадянина і не може бути порожнім.

У двох різних об'єктах (таблицях):

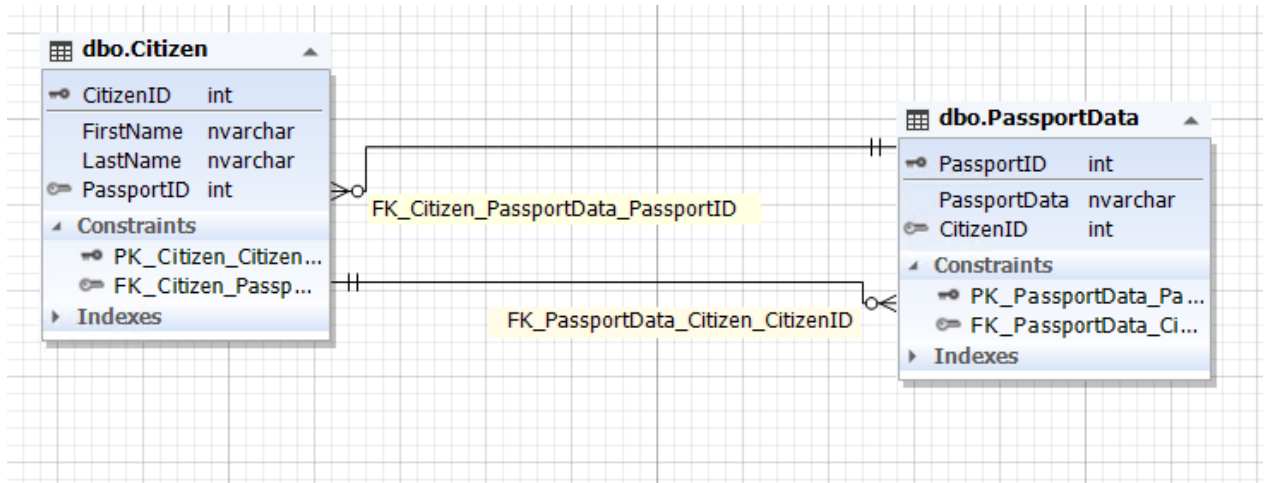


Рисунок 1.2 – Взаємовідносини між особами Citizen та PassportData

Таблиця Citizen представляє сутність громадянина, а таблиця PassportData – сутність паспортних даних громадянина. Сутність громадянина містить атрибут PassportID (поле), який посилається на первинний ключ таблиці PassportData. Тоді як суб'єкт паспортних даних має атрибут CitizenID (поле), який посилається на первинний ключ CitizenID таблиці Citizen.

Також важливо гарантувати цілісність поля CitizenID і таблиці PassportData, щоб забезпечити взаємозв'язок «один на один». Тобто поле PassportID у таблиці Citizen та поле CitizenID у таблиці PassportData повинні посилатися на той самий запис, як якщо б це була одна сутність (таблиця), яка була проілюстрована на рисунку 1.2.

2. Необов'язкові відносини.

Прикладом тут може бути особа, яка може мати паспортні дані і не мати вказаної країни. Отже, в першому випадку він є громадянином даної країни, а в другому – ні.

Цей взаємозв'язок реалізується двома шляхами. В одній сутності (таблиця).

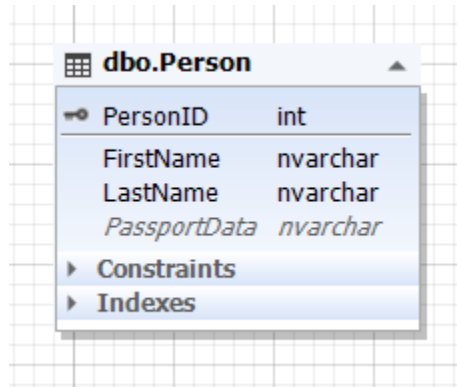


Рисунок 1.3 – Сутність особи

Тут таблиця Person представляє сутність особи, а атрибут (поле) PassportData містить усі паспортні дані особи і може бути порожнім (NULL). У двох об'єктах (таблиці).

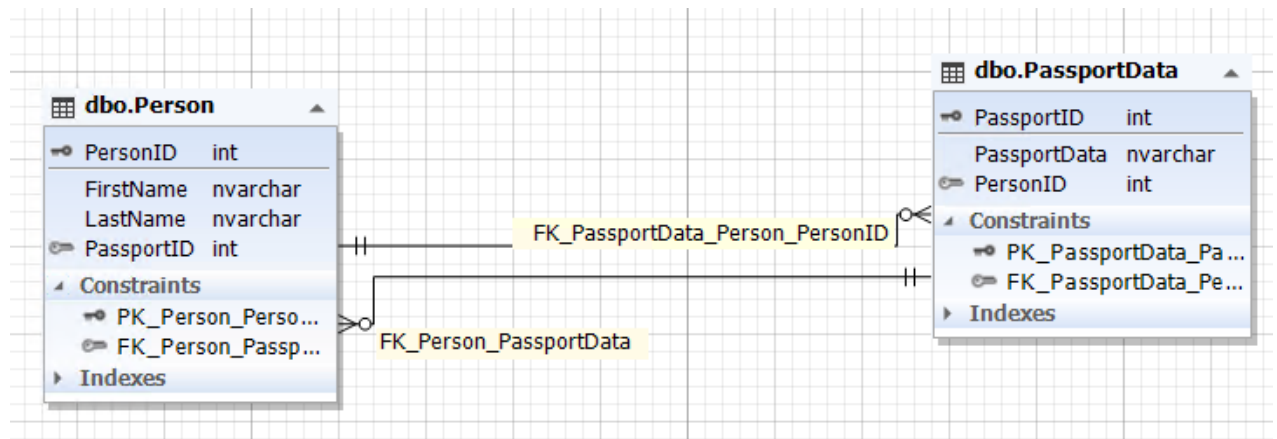


Рисунок 1.4 – Взаємовідносини між Person і PassportData

Тут таблиця Person представляє сутність особи, а таблиця PassportData – сутність паспортних даних людини (тобто сам паспорт). Сутність особи містить атрибут (поле) PassportID, який посилається на первинний ключ таблиці PassportData. Тоді як об'єкт паспортних даних має атрибут PersonID (поле) в таблиці Person. PassportID поле Person таблиці може бути порожнім (NULL).

Також важливо гарантувати цілісність поля PersonID і таблиці PassportData, щоб забезпечити взаємозв'язок "один на один". Тобто поле PassportID таблиці Person та поле PersonID таблиці PassportData повинні

посилатися на ті самі записи, як якщо б це була одна сутність (таблиця), або ці поля повинні бути невказаними, тобто містять NULL.

3. Відносини один до багатьох.

Обов'язкові відносини. Ілюстрацією цього можуть бути батьки та їх діти. У кожного з батьків є принаймні одна дитина.

Ви можете реалізувати ці відносини двома способами:

- В одній сутності (таблиця).

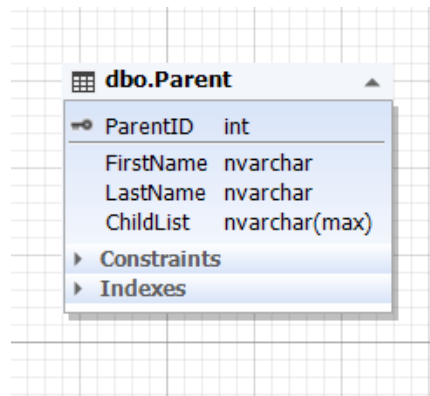


Рисунок 1.5 – Суб'єкт батьків

Тут батьківська таблиця представляє батьківську сутність, а атрибут (поле) ChildList містить інформацію про дітей, тобто самих дітей. Це поле не може бути порожнім (NOT NULL). ChildList поле типу, як правило, частково структурованих даних (NoSQL), такі як XML, JSON, і так далі.

- У двох об'єктах (таблиці).

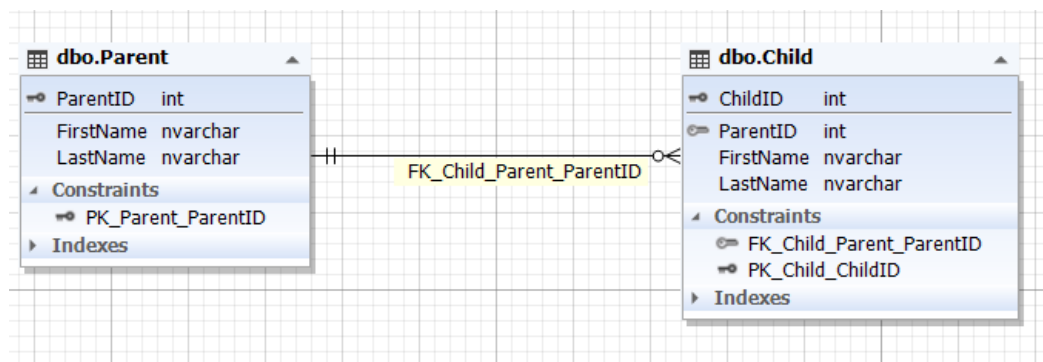


Рисунок 1.6 – Взаємозв'язок між батьками та дитиною

Тут батьківська таблиця представляє батьківську сутність, а дочірня таблиця – дочірню сутність. Дитячий столик має ParentID поле, яке відноситься до первинного ParentID ключу Батько таблиці. ParentID поле дитини таблиці не може бути порожнім (NOT NULL).

Необов'язкові відносини. Прикладом може бути людина, яка може мати дітей або не мати дітей.

Цей взаємозв'язок реалізується двома шляхами:

- В одному об'єкті (таблиця):

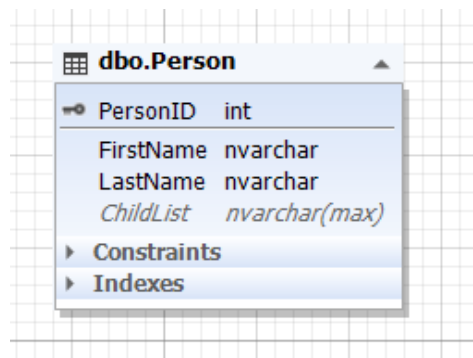


Рисунок 1.7 – Сутність особи

Тут батьківська таблиця представляє батьківську сутність, а атрибут (поле) ChildList містить інформацію про дітей, тобто самих дітей. Це поле може бути порожнім (NULL). Звичайний тип поля ChildList – це напівструктуровані дані (NoSQL), такі як XML, JSON та інші.

- У двох об'єктах (таблиці).

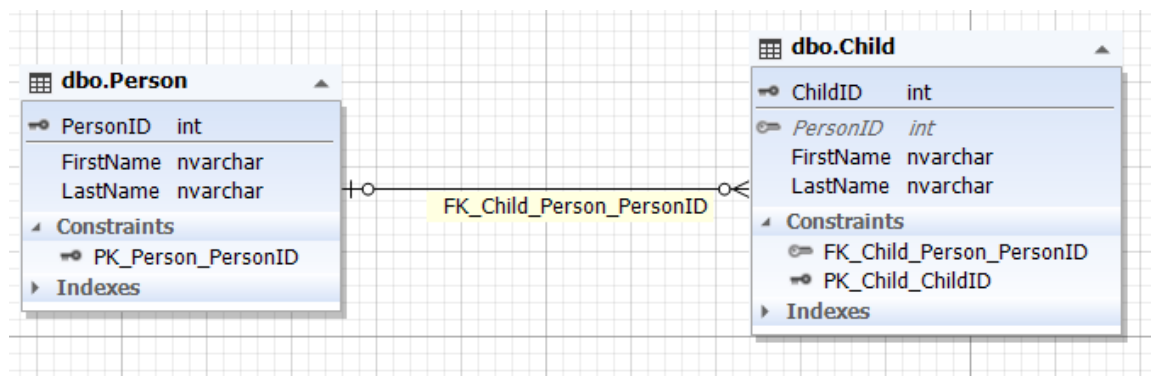


Рисунок 1.8 – Взаємозв'язок між особами "Особа" та "Дитина"

Тут батьківська таблиця представляє батьківську сутність, а дочірня таблиця – дочірню сутність. Дитячий столик має ParentID поле, яке відноситься до первинного ParentID ключу Батько таблиці. ParentID поле дитини таблиці може бути порожнім (NULL).

Також існує третій спосіб реалізації однієї сутності, який посилається на себе, за умови, що дочірній та батьківській сутності (таблиці) мають однаковий набір атрибутів (полів) без посилання на батьківський.

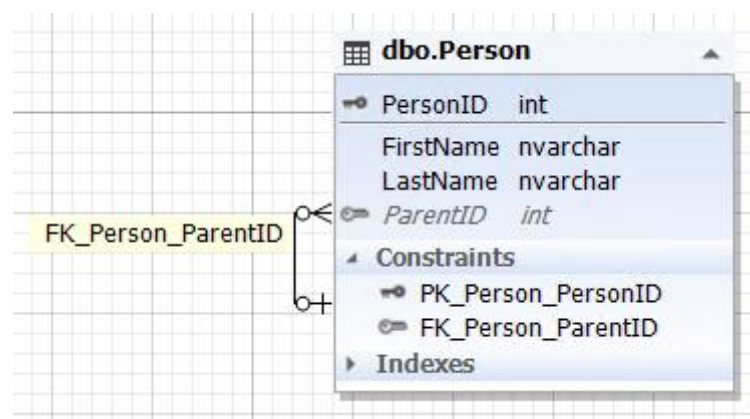


Рисунок 1.9 – Суб'єкт Особи

Тут сутність Person (таблиця) містить атрибут (поле) ParentID, який посилається на первинний ключ PersonID тієї ж таблиці Person і може мати порожнє значення (NULL).

Це реалізація багатосторонніх відносин з необов'язковим характером.

3. Відносини багато до одного.

Ці відносини відображають ілюстровані вище відносини «один до багатьох». Це відносини між дочірньою сутністю та батьківською сутністю, де обов'язкові відносини можливі, якщо у дитини є хоча б один із батьків, і якщо ми беремо всіх дітей, включаючи дітей, що перебувають у будинках-сиротах, тоді такі відносини мають необов'язковий характер.

Взаємозв'язки «один-до-багатьох» і «багато-до-одного» також можуть бути реалізовані через більш ніж 2 об'єкти, додавши необхідні атрибути, що посилаються на первинні ключі відповідних об'єктів.

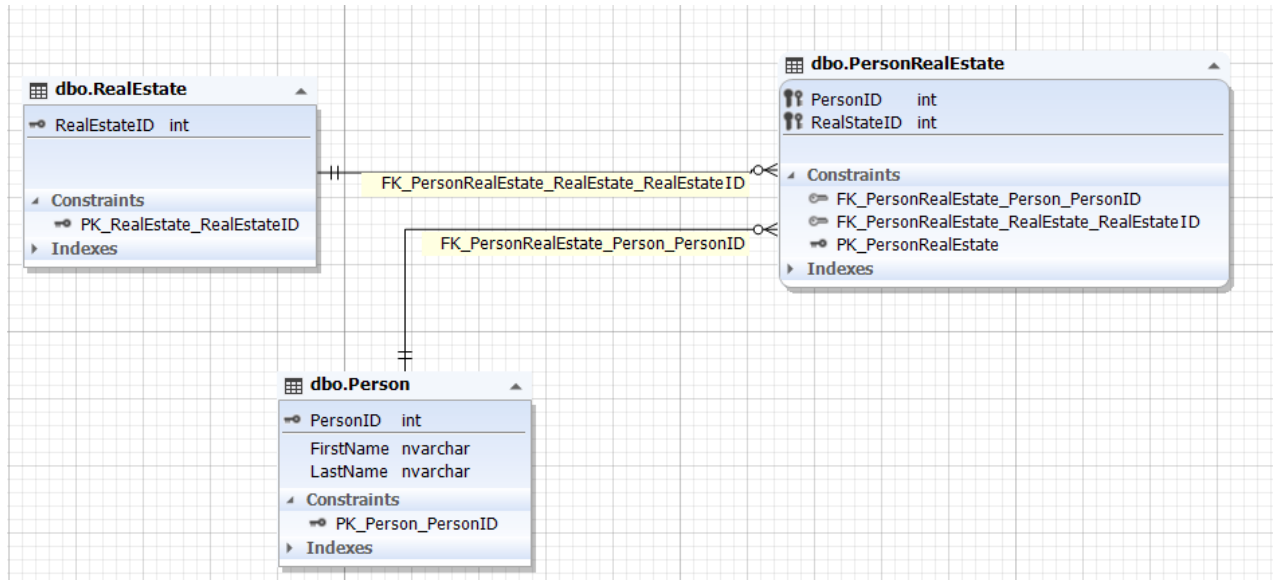


Рисунок 1.10 – Взаємовідносини між особами та особами RealEstate

4. Відносини багато-до-багатьох.

У цьому випадку прикладом може бути нерухомість, якою може володіти одна особа або кілька людей. Одночасно людина може володіти кількома будинками або мати частку власності на багато будинків.

Ви можете реалізувати ці відносини з NoSQL способами, описаними вище для попередніх відносин. Однак у рамках реляційної моделі цей зв'язок зазвичай реалізується через 3 сутності (таблиці).

Тут таблиці `Person` та `RealEstate` представляють сутності особи та нерухомості відповідно. Ці об'єкти (таблиці) пов'язані з допомогою `PersonRealEstate` об'єкта (таблиці) через `PersonId` і `RealEstateID` атрибути (поля), які посилаються на первинні ключі `PersonId` з `Person` таблиці і `RealEstateID` з `RealEstate` таблиці відповідно. Зверніть увагу, що пара (`PersonID` ; `RealEstateID`) завжди є унікальною для таблиці `PersonRealEstate`,

тому вона може бути первинним ключем для самої PersonRealEstate суттєва сутність (таблиця).

Цей зв'язок може бути реалізований через більш ніж 3 сутності шляхом додавання необхідних атрибутів, що посилаються на первинні ключі відповідних сутностей.

Отже, де 7 нормальних форм, ви можете задатися питанням?

Просто ці 7 нормальних форм згруповані у 4 функціональних блоки у тексті вище.

Нормалізація усуває надмірність даних, а отже, зменшує ризики аномалій даних. Однак нормалізація при декомпозиції сутностей (таблиць) призводить до складнішої побудови запитів для маніпулювання даними (вставка, оновлення, виділення та видалення).

Протилежний процес – денормалізація. Це спрощує обробку запитів для доступу до даних за допомогою додавання надлишкових даних.

Якось у джерелах не зазначено, що ці відносини між сутностями не просто вигадані, а виявлені. Тобто з самого початку вони насправді існували в реальному світі між суб'єктами та об'єктами.

На додаток до цього, ці відносини можуть змінюватися, переходячи від «одного до одного», до «одного до багатьох», або «до багатьох до одного», або до «багатьох до багатьох», змінюючи свій обов'язковий характер або зберігаючи його.

Отримавши уявлення про 7 звичайних форм, ви можете легко розробити модель бази даних будь-якої складності для будь-якої інформаційної системи.

Добре розроблена модель бази даних, з урахуванням змін відносин в реальності та в предметній області, не вимагає жодних змін протягом тривалого часу. Це особливо важливо для зберігання даних, коли зміни передбачають збереження великих обсягів даних, від декількох гігабайт до багатьох терабайт.

1.6 Висновок до першого розділу

Давайте підведемо підсумок, що таке дизайн бази даних. Процес проектування бази даних допомагає спростити проектування, розробку, виконання та обслуговування вашої корпоративної системи управління даними.

Хороший дизайн бази даних може допомогти заощадити місце на диску, зменшивши надмірність даних. Поряд із підтримкою точності та надійності даних, це дозволяє отримувати доступ до даних різними способами. Більше того, добре розроблена база даних простіша у використанні та обслуговуванні, що робить інтеграцію простою.

Як видно, проектування систем – це просто перетворення об'єктів і предметів з реальності на сутності баз даних, де взаємозв'язок між цими об'єктами фіксується в певний момент часу, беручи до уваги майбутні зміни. Що саме ми беремо з реальності та реалізуємо як сутність схеми, і які відносини будуюмо в моделі, залежить від того, що ми хочемо від інформаційної системи загалом, зараз і в майбутньому. Тобто, які дані ми хочемо отримати на даний момент та деякий час у майбутньому.

РОЗДІЛ 2. ПРАКТИЧНА РЕАЛІЗАЦІЯ

2.1 Огляд систем керування базами даних

База даних – це сховище різноманітної інформації із певною структурою. Сама база даних зберігається і обробляється при допомозі відповідного програмного забезпечення, яке зазвичай називається сервером баз даних. Сервер баз даних – інформаційна система, яка виконує роботу з даними, регламентує доступ до них і призначена забезпечити їх збереження при допомозі резервування. З сервером бази даних взаємодіють програми, написані сторонніми розробниками. Подібні програми називаються системами управління базою даних (СКБД). СКБД взаємодіючи з сервером отримує можливість оперувати даними: додавати, видаляти змінювати і отримувати їх за запитом користувача. СКБД має інтерфейс регламентуючий в тій чи іншій мірі дії користувача, виконує обробку даних і створює на їх основі різноманітні звіти.

В залежності від розташування СКБД розрізняють локальні та розподілені системи. Всі компоненти локальної СКБД, тобто сам сервер і таблиці з даними, розміщені на ПК кінцевого користувача. У випадку розподіленої бази даних на ПК кінцевого користувача розміщена лише клієнтська програма, яка взаємодіє з сервером баз даних через мережу. Бази даних можуть мати багаторівневу архітектуру. Частіше за все зустрічаються двох рівневі і трьох рівневі СКБД.

При використанні двох рівневої системи відбувається безпосередня взаємодія клієнтського додатку з сервером БД. При використанні триланкової СУБД клієнтський додаток взаємодіє з проміжною програмою – сервером додатку. Сервер додатку здійснює обмін даними з сервером БД, отримує від нього дані обробляє їх і передає клієнтові. Таким чином, всі обчислювальні навантаження лягають на сервер додатку, а клієнт стає дуже «легким» оскільки отримує тільки запитані дані. Прикладом легкого клієнта може

служити інтернет-браузер. При роботі з веб-додатками користувач вибирає посилання в браузері і отримує пов'язаний з ним документ. Розподілені СУБД в загальному випадку можуть бути як двох ланковими, так і три ланковими.[5]

2.2 Огляд медичних інформаційних систем

Основним недоліком більшості МІС, які були представлені на ринку в 2000-2002 рр., була їх вузька спеціалізація. Найбільш популярними були системи для відділень стоматології, офтальмології, рентгенології, анестезіології, реаніматології та інших областей. На сьогодні є декілька програм, призначених для автоматизації тільки реєстратури, доцільність застосування яких сумнівна у відриві від автоматизації клінічних відділень, відділу статистики і других основних служб ЛПЗ. Сучасна МІС, на погляд деяких аналітиків, повинна бути універсальною і підтримувати різні види ЛПЗ, а не спеціалізуватися, наприклад, тільки на автоматизацію окремої поліклініки.

З початку 90-х років в лікувальних закладах України з'явилася перша хвиля персональних комп'ютерів. Одне з перших завдань, над яким вони працювали – це оптимізація роботи реєстратури. Програмне забезпечення, що з успіхом використовувалося, – це система управління базами даних (СУБД) FoxPro 2.6, що саме тоді набула великої популярності.

На даний час на ринку в результаті аналізу було знайдено не велику кількість програмного забезпечення яке створене для медичних закладів. Зокрема, серед знайдених програм є як інформаційні системи так і вузько спеціалізовані програмні продукти призначені лише для роботи у відповідних відділеннях лікувального закладу це наприклад ПЗ для стоматологічних кабінетів, відділу реєстратури ЛПЗ, хірургічного відділення. Також у останні роки стало поширеним компютерне тестування організму. Відповідно для цих систем також створюється допоміжне ПЗ призначена наприклад для реєстрації пацієнтів, видачі різноманітної звітної інформації і т д. Серед вітчизняних

продуктів можна виділити декілька, які автоматизують роботу відділу Реєстратури медичних закладів:

- „TherDep 5”;
- „ТАЛАН”;

Система „TherDep” (Therapeutical Department) усі версії – являє собою комплекс програм, за допомогою яких система Therdep5 може автоматизувати роботу:

- Реєстратури поліклініки (АРМ реєстратури).
- Консультативних кабінетів (АРМ лікаря-консультанта).
- Діагностичних кабінетів УЗІ, КТ, ЕКГ і так далі (АРМ лікаря-діагноста).
- Лабораторій будь-якого профілю (АРМ лікаря-лаборанта).
- Приймального і решти всіх відділень стаціонару (АРМ лікаря стаціонару).
- Відділу статистики (АРМ статистика).
- Всіх наукових підрозділів (АРМ для аналізу даних).

Для кожного з цих напрямів роботи передбачені можливості зручного перегляду і введення даних, роздруку облікових і бланкових форм, обліку фінансів, додавання призначених для користувача параметрів, виконання аналізу даних.

Система Therdep5 вдає із себе комплект програм, що працюють в середовищі WINDOWS 7. Вимоги до комп'ютера для роботи Therdep5, мінімальні Робота в локальній мережі заснована на клієнт-серверній технології, що означає високу надійність одночасної роботи в багатокористувацькому середовищі. В даному програмному комплексі для автоматизації ведення обліку призначена спеціальна утиліта <Polyclin>.

Інформаційна система «ТАЛАН» служить для автоматизованої роботи з медичними документами, зокрема є такі поля як:

- профілактичні огляди;

- звернення пацієнта при захворюванні;
- диспансерний нагляд і лікування в стаціонарі.

В даному підрозділі було розглянуто поняття систем управління базами даних, особливості їх побудови і використання в інформаційних системах. Описано основні можливості і прийоми роботи в сучасних СУБД. Проаналізовано програмне забезпечення у сфері медичного документообігу.

2.3 Опис об'єкту автоматизації

2.3.1 Характеристика предметної області

Блок задач по обліку пацієнтів є основною роботою відділу реєстратури. Даний облік повинен забезпечити в лікувальному закладі правильний розподіл прийому та проходження медичних оглядів пацієнтами.

Об'єктом автоматизації даної дипломної роботи є інформаційні процеси, пов'язані з обліком та реєстрацією пацієнтів у відділі реєстратури на прикладі Зборівської поліклініки.

Поліклініка №2 у м. Зборів створена у 1975 році. Схема організаційної структури відділення реєстратури даної поліклініки зображена на рисунку 2.1.

Основною задачею створення поліклініки було надання медичних послуг для місцевого населення. У 1991 році деякі відділи поліклініки було оснащено компютерами. Компютерна система обліку пацієнтів у відділенні реєстратури була введена у 1998 році, до цього вся інформація зберігалась на паперових носіях. завдяки комп'ютеризації повністю змінилася технологія обробки інформації, суттєво зменшився час обробки даних, полегшилась та автоматизувалась робота лікарів-реєстраторів. Від початку введення компютерної системи, програмне забезпечення яке використовувалось було повністю змінено один раз і до лише частково вдосконалювалось у відповідності до потреб відділу.

Даний відділ виконує наступні функції:

- створення амбулаторних карток для нових пацієнтів;
- реєстрація хворих на прийом до лікарів;
- видача талонів на прийом та обстеження;
- надання довідкової інформації для пацієнтів щодо прийому лікарів;
- реєстрація викликів лікарів «на дому».

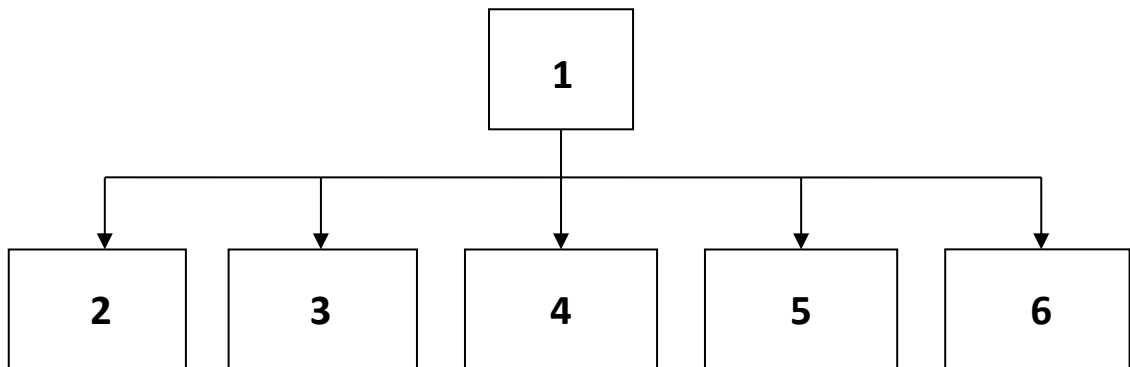


Рисунок 2.1 – Схема організаційної структури відділу реєстратури

1– головний реєстратор,
2,3,4,5,6 – лікарі реєстратори.

Основними документами на основі яких здійснюється реєстрація та зміна наявних даних є паспорт, довідка про народження, довідка про смерть.

У відділі реєстратури (ВЄ), при прийнятті нового пацієнта, який є ще не зареєстрованим, складається нова амбулаторна картка(АК), яка має унікальний код що відповідає лише даному пацієнту. Також створюється новий запис у БД. Даний запис містить наступні дані:

- ПІП пацієнта;
- дату народження;
- стать;
- місце проживання;
- номер амбулаторної карти та ін.

Видача талонів на прийом до певного лікаря проходить наступним чином: пацієнт, який хоче записатись на прийом до лікаря перш за все повинен звернутись у реєстратуру щоб взяти талон і власну амбулаторну карту. Для створення обстеження талону обстеження пацієнт говорить реєстратору свої дані ПІП та адресу або, якщо знає номер АК та стан здоров'я щоб реєстратор міг призначити відповідного лікаря. У комп'ютері дані пацієнта заносяться у форму яка і є талоном, також сюди заносяться дані про дату та час видачі талону. Після цього талон друкується і видається разом із відповідною АК.

Після відвідування лікаря пацієнт повертає амбулаторну картку назад у реєстратуру, а талон залишається у лікаря на ньому записуються додаткові дані про обстеження такі як:

- дата та час прийому;
- особистий номер лікаря (ідентифікатор);
- шифр діагнозу.

По закінченню робочої зміни талони здаються у інформаційно-аналітичний відділ. У цьому відділі дані із талона заносяться у статистичну БД. За допомогою цієї бази даних при потребі можна вивести статистичні звіти по окремому пацієнту, по кількості хворих з певним діагнозом та ін.

Також відділ реєстратури надає довідкову інформацію по телефоні, для пацієнтів які телефонують дізнатись про розпорядок прийому на обстеження до вузькоспеціалізованих лікарів. Для цього у відділені на робочому місці реєстратора, встановлено звичайні таблиці (листи), в яких містяться дані про розпорядок роботи кожного із лікарів. Це є незручним при роботі, тому що при запитанні пацієнтом розпорядку певного лікаря реєстратор переглядає весь список лікарів. При цьому втрачається час на відповідь. Також реєстратор здійснює записи у журнал про виклики лікаря до хворого на дім. Реєстрація здійснюється вручну на паперових носіях. Відповідно, при пошуку певного запису втрачається більше часу у порівнянні з винанням даної операції на компютері. Тож з впевненістю можна сказати що при обладнанні даного

робочого місця компютером з відповідним ПЗ продуктивність роботи реєстратора зросла б у кілька разів.

Отже, основними недоліками даної інформаційної системи є те, що всі операції ведуться устарівшою компютерною системою або вручну, введення даних відбувається на декількох ком'ютерах, а потім дані переносяться на звичайних магнітних носіях для створення єдиної бази. Також немає єдиної бази даних у якій би зберігались всі дані про пацієнта, про відвідування ним лікарів і результати обстеження, дані про виклики на дім та ін.

2.4 Структурна схема розроблюваної системи

2.4.1 Постановка задачі

В кваліфікаційній роботі проектується інформаційна система для розв'язку задачі по обліку пацієнтів в організації.

Пропонується автоматизувати робоче місце працівника відділу реєстрації. Основною метою створення АРМу працівника відділу реєстрації є удосконалення обліку пацієнтів на основі автоматизації виконуваних функцій, реалізація елементів безпаперової технології при його веденні, оперативне формування у базі даних інформації про реєстрацію, прийом, виклики та дані обстежень пацієнтів.

Розробка даного АРМу ведеться з можливістю об'єднання його, при необхідності, в мережу разом з АРМ працівників Інформаційно-аналітичного відділу, де ведуться статистичні дані на основі даних отриманих з відділу реєстрації та інших підрозділів поліклініки.

2.4.2 Опис функціональної структури автоматизованої підсистеми

На першому етапі реалізації задачі було проведено аналіз вимог і згруповано дані вимоги за відповідними класами задач. Тому було прийнято рішення, щоб розв'язок певного класу задач здійснювати в окремому структурному модулі (див. рисунок 2.1, додаток Б).

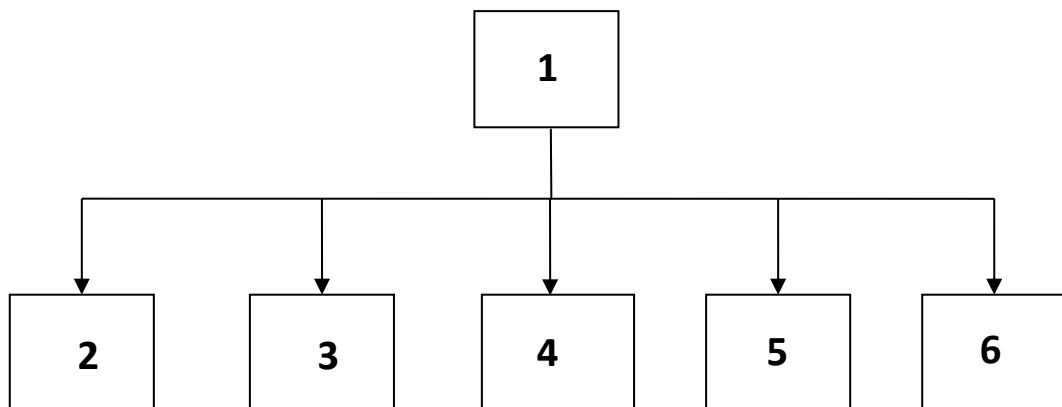


Рисунок 2.1 – Структурна схема модулів АРМ

1. АРМ – атоматизоване робоче місце;
2. стіл довідок;
3. журнал реєстрації викликів;
4. база даних пацієнтів;
5. журнал реєстрації обстежень;
6. довідкова система захворювань.

В процесі деталізації основної задачі було проведено декомпозицію блоків системи. Нижче приведемо структурні схеми блоків, над якими було проведено декомпозицію. Зокрема, на рисунок 2.2 зображено деталізовану структурну схему модуля „Стіл Довідок”.

Модуль «Стіл довідок» відповідає за виконання наступних задач:

- відображення бази даних лікарів;
- перегляд графіків роботи підповідного лікаря;
- вибірка (фільтрація) необхідних даних.

Реалізація вищезгаданих задач здійснюється наступним чином:

- Перегляд списку лікарів в даній програмі здійснюється за допомогою відображення таблиці з базою даних штатних лікарів.
- Графіки роботи відповідного лікаря відображається в додатковій таблиці в залежності від того який запис вибраний в БД «Лікарі».
- Вибірка або фільтрація даних відбувається в залежності від умови яку задає користувач. Фільтрація може здійснюватись по коду, посаді та ППІ лікаря.

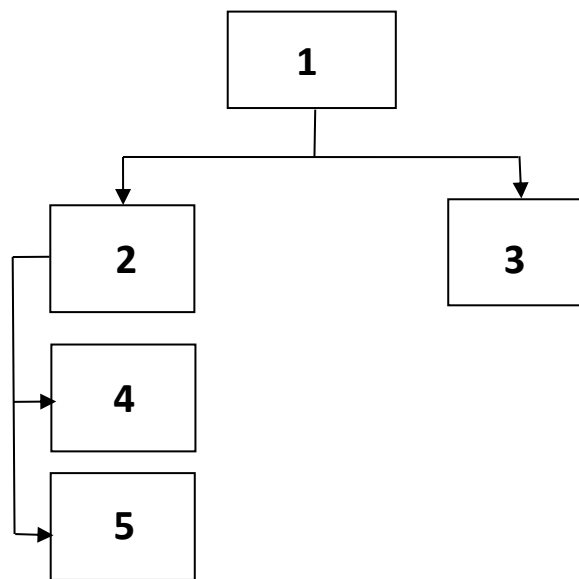


Рисунок 2.2 – Структурна схема модуля «Стіл Довідок»

1. «Стіл довідок».
2. БД лікарів.
3. графіки роботи лікарів.
4. Редагування БД лікарів.
5. Пошук.

Модуль «Стіл довідок» відповідає за виконання наступних задач:

- відображення бази даних лікарів;
- перегляд графіків роботи підповідного лікаря;
- вибірка (фільтрація) необхідних даних.

Реалізація вищезгаданих задач здійснюється наступним чином:

- Перегляд списку лікарів в даній програмі здійснюється за допомогою відображення таблиці з базою даних штатних лікарів.
- Графіки роботи відповідного лікаря відображається в додатковій таблиці в залежності від того який запис вибраний в БД «Лікарі».
- Вибірка або фільтрація даних відбувається в залежності від умови яку задає користувач. Фільтрація може здійснюватись по коду, посаді та ППП лікаря.

Опишемо модуль «Журнал реєстрації викликів» який відповідає за виконання наступних задач:

- реєстрація даних пацієнта, що здійснив виклик: ППП, адреса, номер амбулаторної карти, якщо пацієнт зареєстрований та причина виклику;
- реєстрація даних виклику, таких як номер п/п, дата та час;
- реєстрація додаткових записів лікаря що здійснював виїзд до пацієнта.

Запис даних, що використовуються в даному модулі зручно реалізувати за допомогою однієї таблиці (бази) «Журнал викликів» та звязати її відповідними полями з іншими базами в програмі такими як «Лікарі» та «Пацієнти». Реалізація модуля здійснюється наступним чином.

Для даного модуля створюється окрема форма на якій буде відображено базу всіх викликів, для зручності додано відповідні засоби для пошуку відповідних записів та введення нових.

Опишемо модуль „База пацієнтів”, структурна схема якого зображена на рисунку 2.3.

Даний модуль реалізує наступні задачі:

- Створення запису про нового пацієнта.
- Перегляд та редагування записів.
- Вилучення та збереження даних.
- Пошук та фільтрація даних про пацієнтів.
- Сортування даних за певними параметрами.

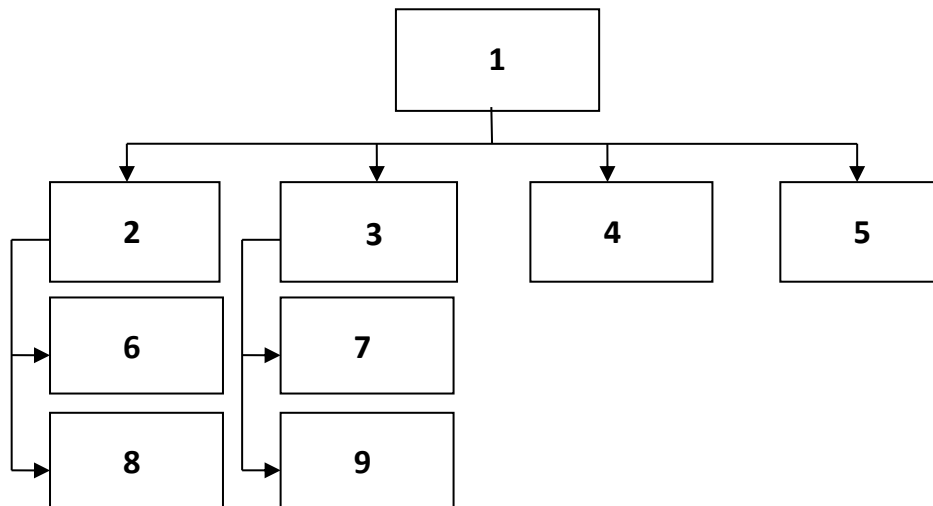


Рисунок 2.3 – Структурна схема модуля „База пацієнтів”

1. модуль „База пацієнтів”;
2. робота з даними;
4. сортування даних;
6. створення та редагування записів;
7. пошук;
8. знищення та збереження записів;
9. фільтрація.

Для реалізації даних задач підсистеми в модулі „Пацієнти ” реалізовано наступні процедури:

- Перегляд – процедура реалізовує перегляд всієї інформації про зареєстрованих пацієнтів.
- Редагування – процедура реалізовує операції щодо зміни раніше введених даних.
- Пошук. Дана процедура реалізовує пошук інформації по ключових словах, значеннях номерах карток, та інших даних пацієнта.
- Фільтрація забезпечує перегляд даних лише певного пацієнта.
- Створення нового запису. Дана процедура дозволяє створити зареєструвати нового пацієнта.

Модуль «Журнал обстежень» (див. рисунок 2.4) відповідає за виконання наступних функцій:

- Реєстрація даних нового обстеження.
- Збереження, зміна та пошук даних обстеження.

Для реєстрації всіх даних про обстеження створена одна таблиця, яка має зв'язок із БД «Лікарі» та «Пацієнти».

Для реєстрації обстежень створена окрема форма на якій є відповідні поля для занесення даних про:

- ППІ хворого.
- Дата та час проведення обстеження.
- Дані лікаря яким виконано обстеження.
- Результати обстеження у вигляді коду захворювання та додаткових даних.

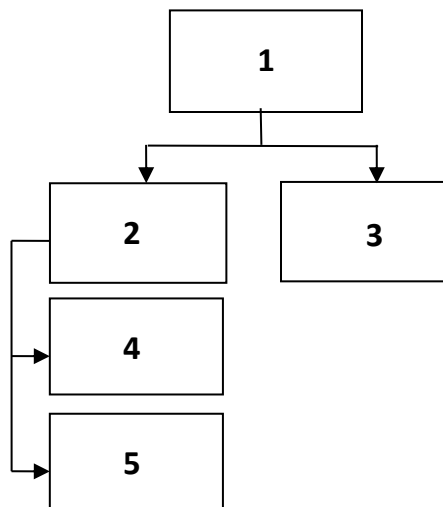


Рисунок 2.4 – Структурна схема модуля „Журнал реєстрації обстежень”

1. модуль „Журнал реєстрації обстежень”;
2. робота з даними;
3. пошук даних;
4. створення та знищення запису;
5. редагування даних.

Модуль «Довідкова система захворювань» (див. рисунок 2.5) служить для надання довідкових даних про захворювання. В ньому реалізовані наступні функції:

- Присвоєння коду захворювання.
- Опис захворювання (відповідно до коду).
- Пошук даних.
- Відображення додаткових даних.

Для реалізації даного модуля створена окрема БД захворювань. Модуль реалізований на окремій формі, на якій є таблиця для відображення БД та функціональні засоби для роботи з базою.

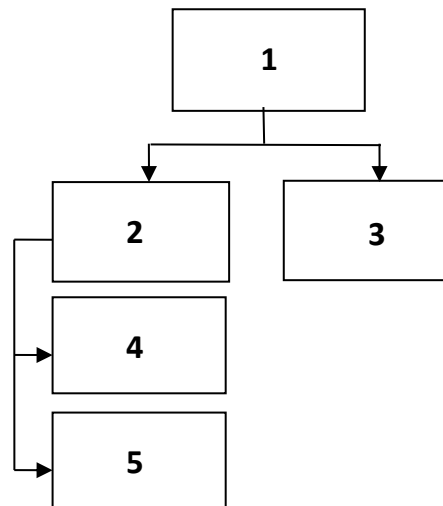


Рисунок 2.5 – Структурна схема модуля „Довідкова система захворювань”

1. модуль „Довідкова система захворювань”;
2. база даних захворювань;
3. пошук;
4. редагування даних;
5. додавання нового запису.

2.5 Реалізація системи

2.5.1 Модель бази даних в Delphi

Таблиці зберігаються фізично в окремих файлах.

Таблиці було створено за допомогою утиліти «Borland Database Desktop».

Створення бази даних можна представити як послідовність кроків:

- Створюємо каталог.
- Створюємо таблиці.
- Створюємо СУБД.

2.5.2 Створення таблиці

Важливим моментом при створенні бази даних є розподіл інформації між полями запису. Очевидно, що інформація може бути розподілена між полями різним чином. Наприклад, відомості про пацієнта можуть бути організовані у вигляді записів, що складаються з полів "Номер п/п", "Прізвище" і "ім'я", "Дата народження", „Адреса” і т.д. У виконуватиметься вибірка інформації по

деякому критерію, то інформацію, що забезпечує можливість цієї вибірки, слід помістити в окреме поле.

Після того, як визначені поля запису, необхідно виконати розподіл полів по таблицях. У простій базі даних всі поля можна розмістити в одній таблиці. У складній базі даних поля розподіляють по декількох таблицях, і введенням деякої додаткової інформації, що однозначно ідентифікує кожен запис, забезпечується зв'язок між таблицями.

Після того, як визначена структура записів бази даних, можна приступити безпосередньо до створення таблиці наприклад у Microsoft Access.

Наприклад при створенні бази даних для реєстратури дані що потрібно зберігати я розподілив таблицях:

- пацієнти;

- лікарі;
- графіки роботи;
- обстеження;
- виклики;
- захворювання.

Цим самим звівши їх до 2НФ.

У головну таблицю заносяться лише дані про пацієнта тобто: прізвище, імя, по-батькові, дата народження, адреса та ін.

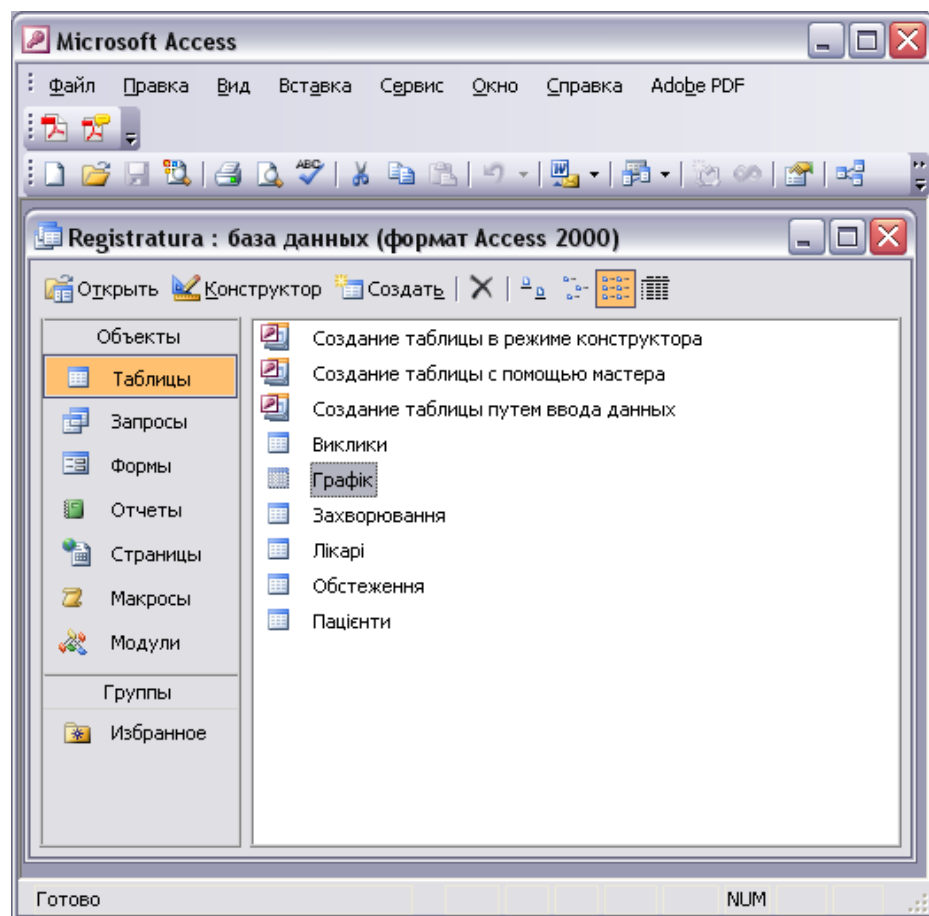


Рисунок 2.6 – Перелік таблиць бази даних Реєстратура

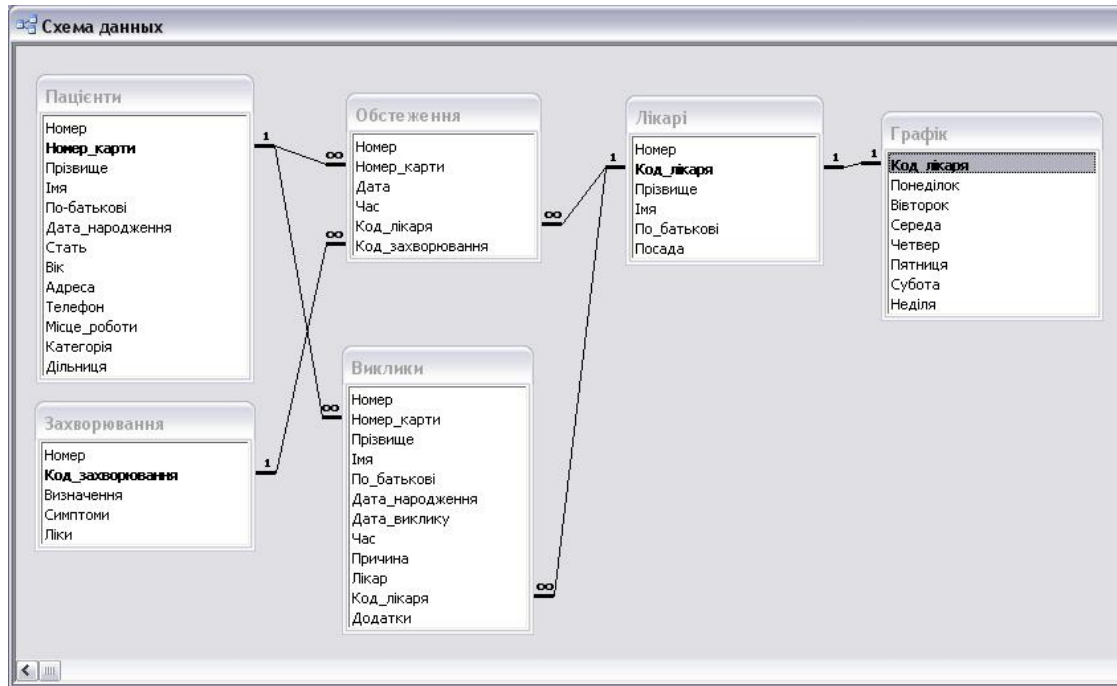


Рисунок 2.7 – Схема зв'язків між таблицями

Ця таблиця є головною інші зв'язуються з нею за допомогою ключового поля «Номер_карти», яке є унікальним тобто не повторюється. Тип зв'язку який використовується «один до багатьох». В таблицю «Лікарі» будуть заноситись дані лікарів, а саме Прізвище, імя, по-батькові, посада та персональний код. В таблицю «Обстеження» будуть заноситись дані про дату і тип обстежень а також результати або висновки лікарів. У таблиці «Графік» міститься інформація про погодинний графік роботи кожного лікаря. Вищезгадані таблиці створюємо в у Microsoft Access за допомогою конструктора. (див. рисунок 2.6).

Після створення таблиць зв'язуємо їх між собою відповідними типами зв'язків у вікні «Схема даних» (див. рисунок 2.7).

2.5.3 Програма управління базою даних

Тепер можна приступити до розробки додатку. Методика розробки додатку роботи з базою даних нічим не відрізняється від методики створення звичної програми: до форми додаються необхідні компоненти, встановлюються значення властивостей компонентів, розробляються необхідні процедури обробки подій.

Додаток роботи з базою даних повинен містити компоненти, що забезпечують доступ до даних, можливість перегляду і редагування вмісту полів. Компоненти доступу до даних знаходяться на вкладці Data Access палітри компонентів, а компоненти відображення даних – на вкладці Data Controls.

Доступ до бази даних (таблиці)

На рисунку 2.8 наведено компоненти для вкладок.

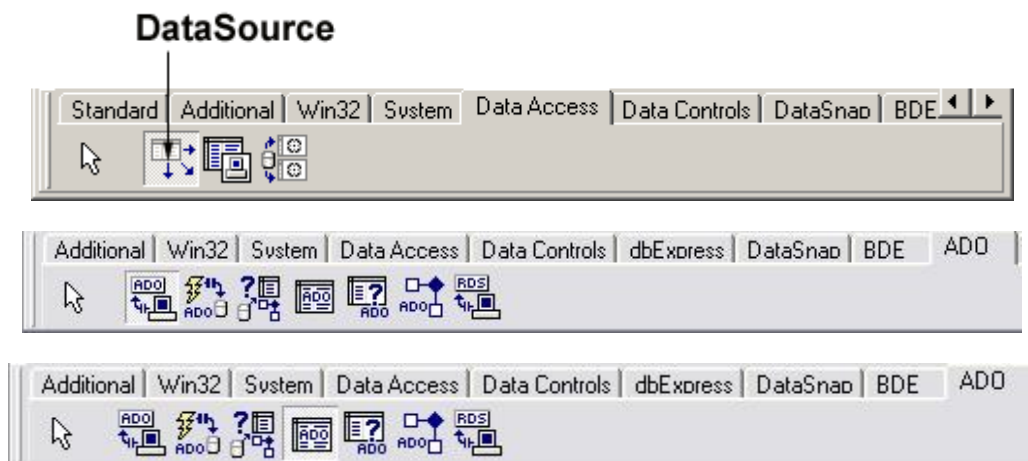


Рисунок 2.8 – Компоненти для вкладок

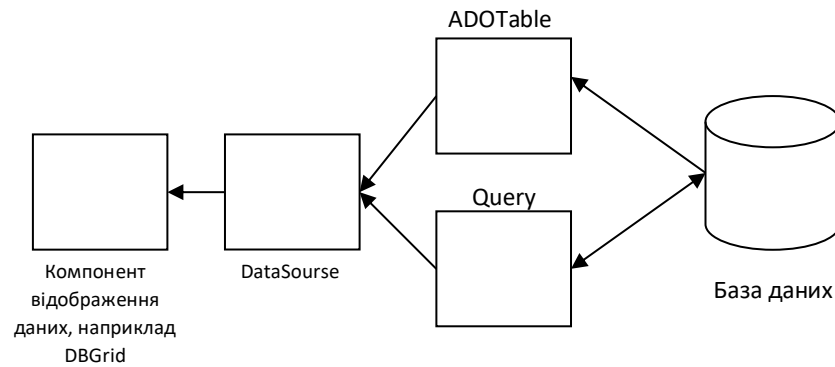


Рисунок 2.9 – Взаємодія компонентів

Взаємодія компонентів показано на рисунку 2.9.

Рисунок 2.10 ілюструє взаємодію компонентів для перегляду і редагування.

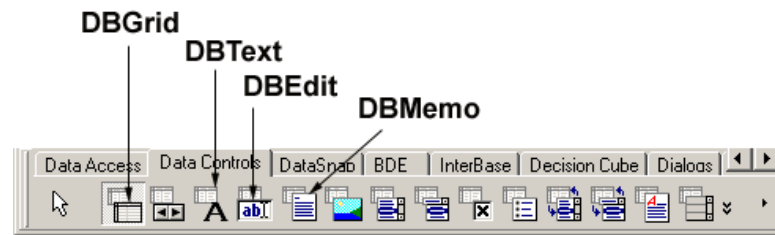


Рисунок 2.10 – Взаємодія компонентів для перегляду і редагування

Компонент Tadoconnection призначений для з'єднання з сховищами даних. З одним компонентом Tadoconnection може бути зв'язаний декілька компонентів Tadotable tadoquery. З'єднання з сховищем даних відкривається і закривається при допомозі властивості Connected або методу Open. Методу Open можна передати параметри USER і Password, в яких зберігаються логін і пароль. Закрити з'єднання можна, викликавши метод Close.



Рисунок 2.11 – Компонент TadoConnection

Властивість `Connectoptions` визначає тип з'єднання. Можна створювати синхронне і асинхронне з'єднання. За умовчанням з'єднання визначається як синхронне. У тому випадку, коли сервер працює досить поволі, вибирається асинхронне з'єднання.

Властивість `Keepconnection` визначає, чи може даний додаток підтримувати зв'язок з базою даних, якщо немає відкритих наборів даних. Коли властивість має значення `True`, з'єднання утримуватиметься у відкритому стані. Для з'єднань з видаленими СУБД або для додатків, які часто відкривають і закривають набори даних, установка значення властивості в `True` значно зменшує мережевий трафік і збільшує швидкість роботи додатку оскільки не потрібно кожного разу проходити аутентифікацію на сервері

Властивість `State` дозволяє дізнатися, в якому стані знаходиться набір даних.

А властивість `ConnectionString` містить рядок, в якому вказується інформація, необхідна для установки з'єднання з джерелом даних.

2.5.4 Механізм з'єднання з сховищем даних ADO

Перед створенням з'єднання необхідно визначити його параметри. Для цього як вже мовилося, призначена властивість `ConnectionString`.

Набір параметрів змінюється залежно від типу використовуваного провайдера і може настраюватися як вручну, так і за допомогою редактора. Для того щоб викликати редактор з'єднань, необхідно двічі клацнути на компоненті `Tadocconnection`. В результаті буде активовано діалогове вікно показане на рисунок 2.12. У цьому вікні можна набудувати з'єднання, використовуючи поле `Use Connection String` або завантажити параметри з'єднання з файлу в розділі `Use Data Link File`. Параметри з'єднання зберігаються у файлах `UDL`, що є звичайні текстові файли, що містять параметри з'єднання.

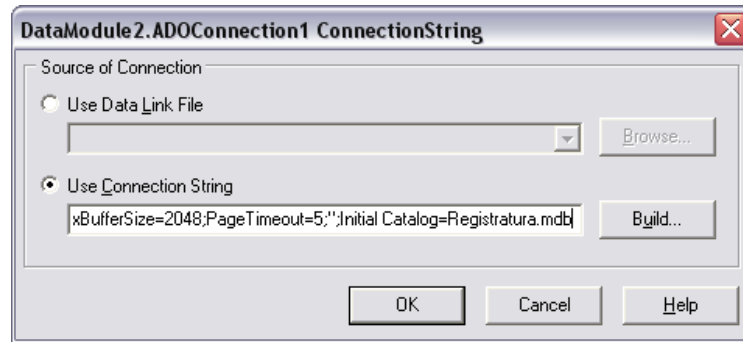


Рисунок 2.12 – Вікно редактора рядка з'єднання

2.5.5 Приклад зв'язку з Access через ADO

У новому проекті необхідно створити окремий модуль даних. У нім треба розташувати компонент Tadoconnection і набудувати з'єднання з базою даних. Властивості Loginprompt слід привласнити значення False і активувати з'єднання. Також буде потрібно зв'язати один з одним компоненти Tadotable і Tdatasource.

Компонент Tadotable має бути пов'язаний з компонентом Tadoconnection при допомозі властивості Connection. У властивості Tablename слід вибрати таблицю, а потім активувати компонент. На головній формі треба розмістити компонент Tdbgrid і пов'язати його з компонентом Tdatasource. Також буде потрібно один компонент Tmainmenu. У цьому меню будуть реалізовані пункти для створення нового запису, ухваленні змін збереженні змін, видалення запису, записи даних у файл і завантаження їх з файлу, а також для установки з'єднання з базою даних і відключення від неї. Код прикладу приведений в лістингу. Методи Savetofile і Loadfromfile дозволяють зберегти набір даних у файл, а пізніше рахувати його звідти. Використовуючи дані методи, можна завантажити набір даних за допомогою компоненту Tadoconnection, зберегти його і потім працювати з локальною базою даних. Ця технологія досить проста:

- Завантажити дані за допомогою Tadoconnection в Tadotable.
- Методом Savetofile компоненту Tadotable зберегти набір даних у вибраному файлі.

- Видалити компонент Tadoconnection.
- Методами Loadfromfile і Savetofile можна завантажувати набір даних з файлу і зберігати його у файлі. [5]

У даному розділі розглянуто створення системи управління базою даних на мові Delphi. Описано способи створення програмного додатку за допомогою відповідних компонентів. Описано основні процедури які використовуються компонентами при виконанні певних функцій. Поетапно розглянуто спосіб з'єднання з БД та створення додатку управління базою даних.

2.6 Демонстрація та результати роботи системи

Комп'ютерна програма Реєстратура є автоматизованою підсистемою для ведення обліку відділу реєстратури ЛПЗ. Практична реалізація поставлених задач була здійснена за допомогою візуального середовища програмування Delphi 7 і СУБД Access.

Розглянемо дії користувача при роботі з підсистемою. Алгоритм по якому працює розроблена програма зображений у додатку В. Користувач підсистеми запускає програму на виконання (виконувальний файл), або за допомогою ярлика, який викликає цей файл. На екрані з'являється головне вікно програми (див. рисунок 2.13)

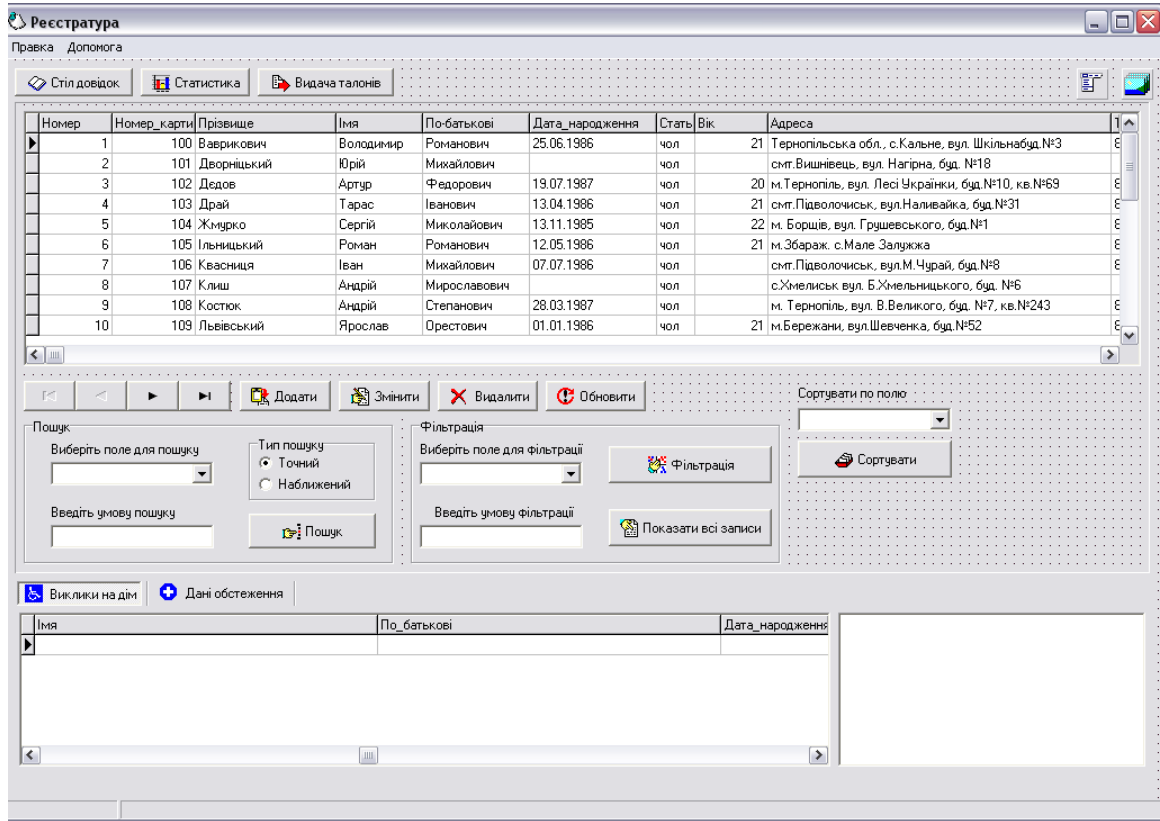


Рисунок 2.13 – Головна вікно програми «Реєстратура»

Головне меню складається із таких пунктів:

- Редагування, в ньому містяться команди для редагування бази даних;
- Про програму.

В нижній частині головної форми є декілька закладок призначених для відображення детальнішої інформації про пацієнта, а саме дані конкретних обстежень, обстежень медичних оглядів, та інформація щодо викликів на дім.

При виборі в меню „додати запис” з’являється форма для введення даних (див. рисунок 2.14).

Рисунок 2.14 – Форма для введення даних про пацієнта

За допомогою даної форми користувач має можливість вносити нові дані та редагувати наявні. Після занесення або зміни даних потрібно натиснути кнопку «Зберегти зміни» для того щоб дані були збереглись у базу.

Для перегляду даних про пацієнта користувач використовує головну форму програми, при цьому у даній формі одразу ж можна здійснювати пошук та вибірку даних за заданими критеріями. Також для пошуку є окрема форма (див. рисунок 2.15) яка з'являється при натисканні кнопки головного меню Редагувати>Пошук.

Номер	Номер_карти	Прізвище	Імя	По-батькові	Дата_народження
1	100	Ваврикович	Володимир	Романович	25.06.1986
2	101	Дворніцький	Юрій	Михайлович	
3	102	Дедов	Артур	Федорович	19.07.1987
4	103	Драй	Тарас	Іванович	13.04.1986
5	104	Жмурко	Сергій	Миколайович	13.11.1985

Виберіть поле для пошуку: Введіть умову:

Рисунок 2.15 – Форма для пошуку даних

Для надання довідкової інформації, щодо розпорядку роботи лікарів у дані програмі створена окрема форма, яка викликається за допомогою клавіші на головній формі «Стіл довідок». На даній формі відображається таблиця із переліком вузькоспеціалізованих лікарів та таблиця із робочим графіком. При переході на запис іншого лікаря у додатковій таблиці відображається відповідний йому графік роботи. Також на даній формі для зручності розміщено засоби пошуку та сортування бази даних лікарів.

2.7 Висновок до другого розділу

В даному розділі проведено огляд існуючих автоматизованих систем, та наведено практичну реалізацію розробленої системи і наведено практичні результати роботи розробленої системи.

3 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ

3.1 Гігієнічні вимоги до організації обладнання робочих місць з ВДТ

Обладнання і організація робочого місця з ВДТ мають забезпечувати відповідність конструкції всіх елементів робочого місця та їх взаємного розташування ергономічним вимогам з урахуванням характеру і особливостей трудової діяльності (ГОСТ 12.2.032-78, ГОСТ 22.269-76, ГОСТ 21.889-76).

Конструкція робочого місця користувача ВДТ має забезпечити підтримання оптимальної робочої пози.

Робочі місця з ВДТ слід так розташовувати відносно світлових прорізів, щоб природне світло падало збоку, переважно зліва. При розміщенні робочих столів з ВДТ слід дотримуватись таких відстаней: між бічними поверхнями ВДТ – 1,2 м; від тильної поверхні одного ВДТ до екрана іншого – 2,5 м.

Екран ВДТ має розташовуватися на оптимальній відстані від очей користувача, що становить 600...700 мм, але не ближче ніж за 600 мм з урахуванням розміру літерно-цифрових знаків і символів. Розташування екрана ВДТ має забезпечувати зручність зорового спостереження у вертикальній площині під кутом $+30^\circ$ до нормальної лінії погляду працюючого.

Клавіатуру слід розташовувати на поверхні столу на відстані 100...300 мм від краю, звернутого до працюючого. У конструкції клавіатури має передбачатися опорний пристрій (виготовлений із матеріалу з високим коефіцієнтом тертя, що перешкоджає мимовільному її зсуву), який дає змогу змінювати кут нахилу поверхні клавіатури у межах $5...15^\circ$.

Для забезпечення захисту і досягнення нормованих рівнів комп'ютерних випромінювань необхідно застосовувати приєкранні фільтри, локальні світлофільтри (засоби індивідуального захисту очей) та інші засоби

захисту, що пройшли випробування в акредитованих лабораторіях і мають щорічний гігієнічний сертифікат.

При оснащенні робочого місця з ВДТ лазерним принтером параметри лазерного випромінювання повинні відповідати вимогам ДСанПіН 3.3.2.007-98.

3.2 Дія електричного струму на організм людини, види електротравм

Найважливішими питаннями охорони праці є електробезпека, яка являє собою систему організаційних, технічних заходів та засобів, що забезпечують захист людей від шкідливого та небезпечного впливу електричного струму, електричної дуги, електромагнітного поля і статичної електрики.

До небезпечних і шкідливих виробничих факторів відносять підвищене значення напруги в електричному колі, замкнення якого може відбутися через тіло людини, підвищений рівень статичної електрики, електромагнітних випромінювань, підвищену напруженість електричного та магнітного полів.

Електричне обладнання становить велику потенційну небезпеку для людини, особливо у зв'язку з тим, що органи почуттів не відчують на відстані електричну напругу на відміну від теплоти, світла, елементів, що рухаються, запаху та інших шкідливих і небезпечних виробничих факторів. Тому, коли струм впливає на людину, її закисна реакція проявляється тільки після безпосереднього контакту з частинами обладнання, що є під напругою.

Дія електричного струму на живу тканину, на відміну від інших фізичних факторів, носить своєрідний і різнобічний характер.

Механізм ураження людини електричним струмом надзвичайно складний і супроводжується термітним, електролітичним та біологічним впливами. При цьому можливі незворотні порушення функціональної діяльності життєво важливих органів людини.

Термічний вплив характеризується нагріванням тканин тіла, кров'яних судин, нервів, серця та інших органів, які знаходяться на шляху струму.

Електролітичний вплив розкладає кров, лімфу та плазму, порушує їх фізико-хімічний склад. Біологічний вплив виявляється у порушенні біологічних процесів, які відбуваються в організмі, що супроводжуються подразненням або руйнуванням нервових та інших тканин та опіками, аж до повного припинення діяльності органів дихання та кровообігу. За наслідками електротравми поділяються на місцеві, що супроводжуються явно визначеними місцевими ушкодженнями організму, та загальні, або електричні удари, які призводять до ураження всього організму через порушення функцій життєдіяльності найважливіших органів та систем. Більшість електротравм (-55%) це сукупність місцевих електротравм та електричних ударів.

Небезпека місцевих електротравм і складність їх лікування залежить від характеру і ушкодження тканини, реакції організму на це ушкодження. Як правило, місцеві електротравми виліковуються і працездатність потерпілого відновлюється повністю або частково. Інколи (частіше при тяжких опіках) людина гине. У цьому разі безпосередньою причиною смерті є не електричний струм (або дуга), а місцеве ушкодження організму, викликане струмом (або електричною дугою). Характерні види місцевих електротравм: електричні опіки, електричні знаки, металізація шкіри, електрофтальмія і механічні пошкодження.

Електричний опік – найбільш поширена електротравма, що зустрічається у 60...65% потерпілих від електричного струму, більшість з яких складає оперативний персонал, що обслуговує діючі електроустановки.

Електричні опіки виникають в місцях контакту поверхні тіла людини з електродом (контактний або струмовий опік) або під впливом електричної дуги (дуговий опік).

При виділенні великої кількості теплоти опіки можуть уражати не тільки шкіру, але й підшкірний жировий прошарок, м'язи, нерви і кістки. Такі опіки називаються глибинними і заживають досить довго.

Електричні знаки, або електричні позначки, виникають на шкірі людини в місцях її щільного контакту із струмопровідними частинами. Це затверділі плями сірого або блідо-жовтого кольору, як правило, округлої або овальної форми. На відміну від опіків, знаки не викликають больових відчуттів і лікування їх закінчується добре.

Металізація шкіри – проникнення в поверхневі шари шкіри найдрібніших частинок металу, що розплавляється та розпорошується під дією електричної дуги. Це може відбутися під час коротких замикань, відключення роз'єднувачів та рубильників під напругою і та ін. Цей вид електротравми зустрічається у -10% потерпілих. Металізації шкіри можна уникнути, застосовуючи спецодяг Г захисні окуляри,

Електрофтальмія – запалення роговиці зовнішніх оболонок очей, що виникають внаслідок впливу потужного потоку ультрафіолетових променів, які негайно поглинаються клітинами організму і викликають в них хімічні зміни.-Таке опромінення можливе при наявності електричної дуги (що виникає, наприклад, при короткому замиканні), яка є джерелом інтенсивного випромінювання не тільки видимого електричного світла, але й ультрафіолетових та інфрачервоних променів. Застосування захисних окулярів з безбарвним склом, яке майже не пропускає ультрафіолетове випромінювання, сприяє попередженню захворювання очей.

Механічні пошкодження виникають внаслідок різких мимовільних судомних скорочень м'язів під дією струму, що проходить через людину. В результаті можуть відбутися розриви шкіри, кров'яних судин і нервової тканини, а також вивихи суглобів, навіть переломи кісток. Механічні пошкодження, як правило, є суттєвими травмами, що потребують довгочасного лікування, вони бувають дуже рідко. Механічні пошкодження, викликані, наприклад тим, що людина падає з висоти внаслідок дії струму, до електротравм не відносяться.

Електричний удар – найбільш небезпечний вид електротравми, що супроводжується ураженням організму, при якому спостерігається параліч

м'язів опорно-рушійного апарата, м'язів грудної клітини (дихальних), м'язів шлуночків серця. В першому випадку судомні скорочення м'язів не дозволяють людині самотійно звільнитися від контакту з електроустановкою. При паралічі дихання припиняється газообмін та постачання організму кисню, внаслідок чого виникає задуха. При паралічі м'язів серця його робота або припиняється повністю, або деякий час супроводжується тремтінням (фібриляцією).

Фібриляція – це хаотично-швидкі та різночасні скорочення волокнин серцевого м'язу (фібрил), при яких серце перестає працювати як насос, тобто воно неспроможне забезпечувати рух крові по судинах. Внаслідок цього припиняється кровообіг, зупиняється постачання кисню до тканин та органів, що й викликає загибель організму.

Медичною практикою встановлено, що після припинення роботи серця та дихання в результаті кисневого голоду через 5...6 хвилин гинуть клітини центральної нервової системи, від чого настає втрата свідомості та припинення управління функціями усіх органів тіла. Цей стан носить назву "клінічної (уявної) смерті", оскільки клітини інших органів тіла ще живі. Але при тривалій відсутності дихання та кровообігу відбувається припинення життєдіяльності решти клітин та органів і настає незворотна біологічна смерть. Тому необхідно зразу після вивільнення людини від дії електричного струму, не пізніше перших 5...6 хвилин, надати долікарську допомогу шляхом штучного дихання та непрямого масажу серця, що дасть можливість запобігти смертельному випадку.

3.3 Безпека життєдіяльності в екстремальних умовах

Екстремальна ситуація (ЕС) – це сукупність обставин, що виникають в природі або в процесі діяльності людини, при яких психофізичні параметри можуть перевищити межі компенсації організму, що призводить до порушення безпеки життєдіяльності людини.

Наприклад, високі і низькі температури, фізичне навантаження, вражаючі токсичні дози отруйних речовин, високі дози опромінення тощо.

Поняття «надзвичайний» трактується як «винятковий, дуже великий, що перевершує усі» (С.І. Ожегов). Словосполучення «надзвичайна ситуація» відноситься до сукупності небезпечних подій або явищ, які призводять до порушення безпеки життєдіяльності.

Надзвичайна ситуація (НС) – це несподівані обставини, що раптово виникла на певній території або об'єкті економіки в результаті аварії, катастрофи, небезпечного природного явища або стихійного лиха, що можуть призвести до людських жертв, матеріальних втрат і порушення умов життєдіяльності людей, завдати шкоди здоров'ю людей або навколишньому середовищу.

НС класифікуються:

- за причиною виникнення: навмісні та ненавмісні;
- за природою виникнення: техногенні, природні, екологічні, біологічні, антропогенні, соціальні та комбіновані;
- за масштабами поширення наслідків: локальні, об'єктові, місцеві національні, регіональні, глобальні;
- за можливістю запобігання: неминуючі (наприклад, природні) та такі, яким можна запобігти (наприклад, техногенні, соціальні).

До техногенних відносяться НС, походження яких пов'язане з технічними об'єктами: вибух, пожежі, аварії на хімічно небезпечних об'єктах, викиди радіоактивних речовин на радіаційних небезпечних об'єктах, аварії з викидом екологічно небезпечних речовин, завалення будівель, аварії на системах життєзабезпечення тощо.

До природних відносяться НС, пов'язані із проявами стихійних сил природи: землетруси, цунамі, повені, виверження вулканів, зсуви, селі, урагани, смерчі, бурі, природні пожежі тощо.

До екологічного лиха (НС) відносяться аномальні зміни стану природного середовища: забруднення біосфери, руйнування озонового шару, опустинювання, кислотні дощі тощо.

До біологічних НС відносяться події, що відбуваються в суспільстві, міжнаціональні конфлікти із застосуванням сили, тероризм, грабунки, насильства, протиріччя між державами (війни).

Антропогенні НС є наслідком помилкових дій людей.

Надзвичайні ситуації характеризуються якісними та кількісними критеріями. До якісних критеріїв відносяться (часовий (раптовість та швидкість розвитку подій); соціально-екологічний (людські жертви, виведення із обігу великих площ); соціально-психологічний (масові стреси); економічний. Наприклад, ситуації відносяться до надзвичайних, якщо загинуло більше 10 людей, ГДК перевищена більше ніж в 100 разів, викид нафтопродуктів перевищив 10 т та ін.

Основні причини виникнення НС:

- внутрішні: складність технологій, недостатня кваліфікація персоналу, проектно-конструкторські недоробки, фізичний і моральний знос обладнання, низька дисципліна праці та низька технологічна дисципліна;
- зовнішні: стихійні лиха, неочікуване припинення подачі електроенергії, газу, води, технологічних продуктів, тероризм, війни.

Характер розвитку НС. Виникнення НС обумовлене наявністю залишкового ризику. Відповідно до концепції надлишкового ризику абсолютну безпеку забезпечити неможливо. Тому приймається така безпека, яку приймає і може забезпечити суспільство в даний період часу.

Умови виникнення НС: наявність джерела ризику (тиску, вибухових, отруйних, радіоактивних речовин), дія факторів ризику (викид газу, вибух, загоряння); перебування у вогнищі ураження людей, сільськогосподарських тварин та угідь.

Аналіз причин та ходу розвитку НЗ різного характеру показує їх загальну рису – стадійність. Виділяють п'ять стадій (періодів) розвитку НС:

- накопичення негативних ефектів, що призводять до аварії;
- період розвитку катастрофи;
- екстремальний період, за якого виділяється основна частка енергії;
- період затухання;
- період ліквідації наслідків.

Вогнища ураження техногенного характеру. Існують такі вогнища ураження (ВУ): під час вибухів конденсованих вибухових речовин, газоповітряних, пароповітряних, пилоповітряних сумішей; під час хімічних, радіаційних та біологічних гідродинамічних аварій, пожеж, аварій поїздів, падінні літаків.

Для прикладу розглянемо вогнища ураження під час аварій на хімічно та радіаційних небезпечних об'єктах.

3.4 Наслідки забруднення навколишнього середовища

У наслідок господарської діяльності людини у природному середовищі нагромаджуються не властиві йому речовини. Серед них тверді відходи (сміття) та хімічні сполуки, які призводять до забруднення довкілля. Забрудненою може бути невелика територія, зазвичай навколо промислового підприємства або населеного пункту. Якщо ж забруднення охоплює всю планету і виявляється у будь-якій точці Землі навіть на значній відстані від джерела забруднення, то говорять про глобальне забруднення.

Основними джерелами забруднення повітря є промислові підприємства (заводи, фабрики, теплові електростанції) і транспорт. Спалюючи паливо або виробляючи продукцію, вони викидають в атмосферу пил, сажу, різні хімічні сполуки. Забруднене повітря стає загрозою для всього живого. Воно подразнює очі, ніс та горло людини, викликає отруєння, вбиває рослини. Забруднене повітря охоплює великі райони і безперешкодно надходить у різні країни, залежно від напрямку вітрів. Наслідками забруднення атмосферного

повітря стали такі глобальні проблеми, як потепління клімату, випадання кислотних дощів і утворення озонових дір.

Про глобальне потепління клімату нині часто йдеться на сторінках газет і в теленовинах. Викликане воно збільшенням кількості вуглекислого газу і пилу в атмосфері. Забруднене повітря перешкоджає випромінюванню тепла від Землі назад у космічний простір. Тепло нагромаджується і викликає порушення звичних кліматичних умов. Підвищення температури повітря на Землі навіть на 1°C , призведе до танення криги в Арктиці і Антарктиці. Невдовзі підвищиться рівень Світового океану. Внаслідок підняття рівня води в ньому навіть на 1 м будуть затоплені густонаселені прибережні низовини материків, а мільйони людей – позбавлені своїх місць проживання. Через потепління клімату почастишали стихійні лиха: урагани, смерчі, зливи, повені.

У забрудненому повітрі з'явився новий вид атмосферних опадів – кислотні дощі. Вони виникають внаслідок сполучення з атмосферою вологою хімічних речовин, що викидаються з труб заводів і ТЕС. Кислотні дощі спричиняють захворювання людей, висихання лісів, отруєння води озер і ставків, унаслідок чого в них гине риба. З цієї причини, наприклад, за останні 50 років зникла форель з численних озер Норвегії і Швеції.

Гази, що потрапляють у повітря, руйнують озоновий шар стратосфери. В результаті виникають озонові діри. Найбільша з них виявлена над Антарктидою. Зменшення концентрації озону призводить до збільшення кількості згубного для всього живого ультрафіолетового випромінювання, яке надходить на Землю.

Щоб запобігти забрудненню повітря і екологічним проблемам, що з цим пов'язані, необхідно встановлювати очисні споруди на промислових підприємствах, які б зменшили викиди хімічних речовин.

Основними джерелами забруднення води є промислові підприємства, які скидають у річки і водойми неочищені стоки. Вода забруднюється також змитими з полів мінеральними добривами та отрутохімікатами. До них додаються побутові стоки. Забруднена вода стає непридатною для пиття,

купання та поливу рослин. Небезпечно забрудненими є багато річок та озер на різних материках. “Стічними канавами” Європи називають річки Рейн і Дунай. До них можна віднести й наш Дніпро.

Забруднену воду річки несуть у моря й океани. Там додається ще й забруднення нафтою внаслідок аварій трубопроводів і танкерів, що її транспортують. Нафта утворює на поверхні води нафтову плівку, яка перешкоджає надходженню кисню, а тому небезпечна для всього живого. Нині значна частина поверхні Світового океану вкрита нафтовими плямами і перетворена на безжиттєву пустелю. Сильно забруднені Північне і Балтійське моря, Мексиканська і Перська затоки.

Основними заходами боротьби із забрудненням води є очищення стоків, що надходять у водойми. Велике значення має запровадження сучасних безстічних технологій у промисловому виробництві.

Практично всі забруднюючі речовини, що спочатку потрапили в повітря, згодом опиняються у ґрунтах. Крім того, ґрунти забруднюються і при надмірному внесенні в них отрутохімікатів та мінеральних добрив. Самоочищення землі відбувається дуже повільно, тому отруйні хімічні речовини нагромаджуються там. Їх поглинають рослини, вживання яких викликає захворювання людей і тварин. Щоб запобігти забрудненню ґрунтів, мінеральні добрива потрібно вносити дуже обережно, у науково обґрунтованій кількості.

Забруднення місцевості радіоактивними речовинами може виникнути внаслідок аварії на атомній електростанції. Радіоактивні речовини спричиняють зовнішнє і внутрішнє опромінення людини, що викликає надзвичайно небезпечні для життя захворювання. Унаслідок аварії на Чорнобильській АЕС в Україні у 1986 р. радіоактивно забрудненими виявилися території багатьох областей.

Райони з катастрофічним забрудненням природного середовища називають районами екологічного лиха. Такими, наприклад, є південні райони Африки, надмірно забруднені викидами численних ТЕС і промислових

підприємств. В Євразії районами екологічним лиха є Аральське море і Перська затока, в Україні – 30-кіломерова зона навколо Чорнобильської АЕС, частини Донецької і Луганської областей.

ВИСНОВКИ

У даній кваліфікаційній роботі було здійснено наступні кроки для створення інформаційної системи:

1. Розглянуто існуючі інформаційні системи в галузі медицини. Було проаналізовано ринок медичних інформаційних продуктів, розглянуто та порівняно їх переваги та недоліки. Вивчення предметної області інформаційних технологій в медицині дозволило виявити загальні тенденції розвитку програмних продуктів, методи та шаблони програм.

2. Проаналізовано об'єкт дослідження. Розглянуто існуючу інформаційну систему ведення обліку відділу реєстратури. Визначено основні недоліки даної інформаційної системи. Вивчено вхідну та вихідну документацію. Зібрано всю необхідну інформацію, яка буде використовуватись при створенні нової ІС.

3. Враховуючи недоліки попередньої системи, а також поставлені задачі було розроблено проект нової інформаційної системи ведення обліку відділення реєстратури. На першому етапі реалізації задачі було проведено аналіз вимог і згруповано дані вимоги за відповідними класами задач. Тому було прийнято рішення, щоб розв'язок певного класу задач здійснювати в окремому структурному модулі програми.

4. В загальному розроблена автоматизована система ведення обліку пацієнтів для відділу реєстратури лікувально-профілактичного закладу, що є завершеним програмним продуктом, який реалізує автоматизацію процесу занесення даних про пацієнтів, пошуку, редагування даних в базі, детальний пошук по даних. Дане програмне забезпечення вирізняється простою і зручністю у використанні.

При вирішенні даної задачі вибрана БД Access, а додаток управління базою розроблений на мові програмування Delphi. Програма має високі швидкісні характеристики, набір команд і функції відповідають сучасним

вимогам до представлення і обробки інформації, реалізується багато віконний інтерфейс.

Також було розглянуто питання з розділу охорона праці та безпека життєдіяльності.

ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. С. Боровський. Delphi 5. Учебный курс.– Питер.– Санкт-Петербург.2001
2. Джефф Когсвелл. Программирование баз данных в Delphi 5.0.- Минск, 1997.– 446с.
3. В.П.Марценюк. Медична інформатика. Проектування та використання баз даних. Тернопіль 1999.
4. Фаронов В.В., Шумаков П.В. Delfi 4. „Руководство разработчика баз данных” – М.: „Нолидж”, 1999.
5. Сорокин А. В. Delphi. Разработка баз данных. — СПб.: Питер, 2005. — 477 с : ил.
6. Кириллов В.В. CitForum „Основы проектирования реляционных баз данных”
7. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений. – КОРОНА принт, 2003. – 672с.
8. Стоцкий Ю.А. «Office XP» . – СПб.:Питер, 2002.-576с.
9. Демина Л.М. Использование СУБД при решении экономических задач: Учебное пособие. М.: МГИУ, 1999.
10. Як проводиться атестація робочих місць за умовами праці [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <https://hrliga.com/index.php?module=news&op=view&id=20009>
11. ДБН В.2.5-28-2006 [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <http://kbu.org.ua/assets/app/documents/dbn2/95.1.%20ДБН%20В.2.5-28-2006.%20Природне%20i%20штучне%20освітлення.pdf>
12. Установлюємо доплату за роботу з важкими та шкідливими умовами праці [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://interbuh.com.ua/ua/documents/ib/8902/123191>

13. Вимоги до освітлення робочого місця [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: https://pidruchniki.com/10560412/bzhd/osvitlennya_primischen_robochih_mists