

Ім'я користувача:  
приховано налаштуваннями  
конфіденційності

Ковальчук Роман ОК 41  
ID перевірки: 1015398702

Тип перевірки: Doc vs Library

Дата перевірки:  
02.06.2023 16:36:38 EEST

ID користувача: 100011372

Дата звіту:  
02.06.2023 16:44:56 EEST

Назва документа:

Кількість сторінок: 39 Кількість слів: 6146 Кількість символів: 44545 Розмір файлу: 2.06 MB ID файлу: 1015062723

## 10.5% Схожість

Найбільша схожість: 2.6% з джерелом з Бібліотеки (ID файлу: 1003995049)

Пошук збігів з Інтернетом не проводився

Вилучення цитат вимкнене

.....  
...С..т..о..р..і..н..к..а...4..1.....

## 0% Цитат

Вилучення списку бібліографічних посилань вимкнене

## 0% Вилучень

Немає вилучених джерел

## Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723

### 1 ЗАГАЛЬНІ ВІДОМОСТІ ПРО БД

#### 1.1 Визначення БД і СУБД. Основні ідеї і переваги баз даних

База даних – сукупність даних, організованих відповідно до концепції, яка

описує характеристику цих даних і взаємозв'язки між їх елементами. В загальному випадку база даних містить схеми, таблиці, подання, збережені процедури та інші об'єкти. Дані у базі організують відповідно до моделі організації даних. Таким чином, сучасна база даних, крім самих даних, містить їх опис та може містити засоби для їх обробки.

БД забезпечують більш просту та зрозумілу структуру даних порівняно з їх попередниками, такими як традиційні файли чи ручне збереження інформації. У традиційних системах, дані зберігалися у вигляді окремих файлів, що містили різні типи інформації. Це часто призводило до складності в організації та пошуку даних. Кожен файл міг мати свою власну структуру та формат зберігання, що ускладнювало взаємодію з ними.

БД часто ототожнюють із будь-якою сукупністю файлів, що містять набір відомостей. Насправді, сукупність файлів не обов'язково автоматично утворює базу даних. Головні ідеї, що лежать в основі концепції БД:

- Забезпечення цілісності даних.
- Усунення надлишкового дублювання даних.
- Централізація управління даними.

Отже, суть концепції баз даних полягає в інтегрованому збереженні й диференційованому використанні прикладними програмами всієї інформації про об'єкти предметної області, що представляють певний інтерес для організації.

Це забезпечується за рахунок розміщення бази даних у спеціальному програмному середовищі, що виконує функції доступу і перетворення структур даних, і називається системою управління базами даних (СУБД).

СУБД надає інтерфейс для створення структури бази даних, введення, зберігання, оновлення та вибірки даних. Забезпечує цілісність даних, контроль



Схожість Цитати Посилання  Вилучений

текст  Підміна символів Коментарі

Джерела на цій сторінці: **3, 5, 7, 10, 22**

Сторінка 1 з 41

Назва документа: **Ковальчук\_Роман\_ОК\_41** ID файлу: **1015062723**

доступу та забезпечує швидкий доступ до інформації. Вона дозволяє встановлювати зв'язки між таблицями, виконувати складні запити та аналізувати дані за допомогою різноманітних функцій та можливостей.

Що стосується системи охорони здоров'я, то проблема вибору оптимальних способів зберігання і обробки великих обсягів медичної інформації вже давно є однією з актуальних проблем організації системи охорони здоров'я в Україні. У структурі будь-якого лікувального закладу існують центри, куди надходить інформація, яку слід зберігати і обробляти.

За допомогою БД, дані організуються у вигляді таблиць зі зв'язками між ними. Це дозволяє встановлювати структуру даних, що робить дані більш зрозумілими, спрощує їх організацію, пошук та редагування.

Крім того, бази даних надають механізми запитів та фільтрації, що дозволяють вибирати лише потрібну інформацію з бази даних, а також здійснювати різноманітні операції над даними, такі як сортування чи об'єднання, що також спрощує взаємодію з даними та аналіз інформації.

З розвитком інформаційних технологій зростає обсяг та складність інформації, що зберігається. Тому, ще однією причиною використання баз даних є потреба у системах, які можуть ефективно зберігати та опрацьовувати великі обсяги інформації.

Очевидно, що бази даних дозволяють забезпечити більш просту та зрозумілу структуру даних порівняно з її попередниками, що сприяє більш ефективній та зручній роботі з інформацією.

Щодо безпеки даних - у зв'язку зі зростанням кількості та важливості збереженої інформації, захист даних стає пріоритетним завданням. Бази даних надають механізми контролю доступу та резервного копіювання, що забезпечує конфіденційність й цілісність даних.

Враховуючи зазначені переваги, використання баз даних у медичних закладах, стає обґрунтованим та ефективним підходом для збереження, організації та управління медичною інформацією.



Схожість Цитати Посилання  Вилучений

Підміна символів Коментарі  
текст

Джерела на цій сторінці: 3

Сторінка 2 з 41

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723

7

Отже,  
основним

призначенням бази даних є гарантоване систематизоване збереження даних і надання доступу до них користувачеві або комп'ютерній програмі, тобто забезпечення швидкого пошуку даних.

Створюючи базу даних, користувач прагне впорядкувати інформацію про різні ознаки об'єктів і швидко отримати вибірку даних з довільним з'єднанням ознак. Зробити це можливо, тільки якщо дані структуровані.

Структуризація – це введення угод про способи представлення даних. Неструктурованими називають дані, записані, наприклад, в текстовому файлі. Для наочності, далі наведено порівняння структурованих й неструктурованих даних. На рисунку 1.1 наведено приклад неструктурованих даних:

Рисунок 1.1 – Текстовий документ із невпорядкованими даними

Видно, що пошук даних в неструктурованому вигляді дещо ускладнений. Структуровані ж дані - це дані, організовані за певною системою або структурою, що відображає зв'язки між різними елементами. Вони мають чітко визначену форму або шаблон, який дозволяє легко розпізнавати та обробляти дані.

У структурованих даних інформація зберігається у вигляді таблиць, де кожен стовпець представляє певну категорію даних, а кожен рядок відповідає конкретному запису або елементу. Ця організація дозволяє ефективно здійснювати пошук, сортування, фільтрацію та аналіз даних. Приклад структурованих даних демонструється в таблиці 1.1.



[Схожість](#) [Цитати](#) [Посилання](#) [Вилучений](#)

[текст](#) [Підміна символів](#) [Коментарі](#)

Джерела на цій сторінці: **3, 6, 8**

Сторінка 3 з 41

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723

Таблиця 1.1 – Структурована інформація з текстового документа «Кадри».

№ особової справи	Прізвище	Ім'я По батькові	Дата народження
14693	Короленко	Олег Михайлович	08.02.1990
15696	Петрова	Олена Володимирівна	11.03.1990
16685	Кучма	Владислав Борисович	27.04.1999

Щоб автоматизувати пошук і систематизувати ці дані, необхідно виробити певні угоди про способи подання даних, тобто дату народження потрібно записувати однаково кожного разу, вона повинна мати однакову довжину і певне місце серед іншої інформації. Це повинно бути справедливо і для інших даних (номер особової справи, прізвище, ім'я і тому подібне).

Переваги централізованого підходу в управлінні даними, в порівнянні з ситуацією, коли кожна програма незалежно реалізує зберігання своїх даних, полягають в наступному:

- Скорочення надмірності. При відсутності централізованої БД кожен додаток буде окремо зберігати свої дані. Нерідко одні й ті ж дані використовують кілька різних програм. Наприклад, і список співробітників у відділі кадрів, і список співробітників, записаних в бібліотеку підприємства, містить ім'я, адресу, паспортні дані. У централізованій БД такі дані можна об'єднати з повним або частковим усуненням надмірності.

- Можливість усунення суперечливості. Найчастіше протиріччя є наслідком надмірності. Якщо однакові дані про одну людину представлені в двох різних записах і це "роздвоєння" не враховано, то рано чи пізно два записи можуть перестати узгоджуватися: наприклад в один запис внесуть зміни, а в інший - ні. В цьому випадку інформація стане суперечливою. Протиріч можна уникнути, усуваючи надмірність або контролюючи її. В останньому випадку може використовуватися множинне оновлення, коли при внесенні змін в одну із записів автоматично будуть змінені і записи, пов'язані з нею.



■ Можливість спільного доступу до даних. При наявності централізованої БД співробітники різних підрозділів відповідно до їх повноважень можуть спільно використовувати ці дані.

■ Можливість дотримання стандартів. Впровадження єдиних стандартів по обробці даних набагато простіше здійснити в централізованій системі.

## 1.2 Класифікація баз даних. Способи їх організації

Основний поділ БД - за моделлю даних: Між полями і записами існують певні зв'язки. Залежно від характеру цих зв'язків розрізняють три типи організації баз даних:

- ієрархічний;
- мережевий;
- реляційний.

Ієрархічна БД являє собою сукупність елементів, розташованих у порядку їх підпорядкування від загального до приватного і утворюють перевернуте за структурою дерево (граф). До основних понять ієрархічної структури відносяться рівень, зв'язок і вузол.

Вузол — це сукупність атрибутів даних, що описують певний об'єкт. На схемі ієрархічного дерева вузли

представляються вершинами графа. Кожен вузол на більш низькому рівні пов'язаний лише з одним вузлом, що знаходиться на більш високому рівні.

На рисунку 1.2 наведено структуру ієрархічної бази даних. Вгорі знаходиться батько або кореневий елемент, нижче знаходяться дочірні елементи, елементи, що знаходяться на одному рівні називаються братами, або сусідніми елементами.

41

**1015062723**

10



Рисунок 1.2 – Ієрархічна модель БД

У мережевій БД за таких самих основних поняттях (рівень, вузол, зв'язок) кожен елемент може бути пов'язаний з будь-яким іншим елементом. Основна відмінність мережевої моделі від ієрархічної - графічне зображення структури зв'язків сегментів такого типу

моделей є мережею. Сегменти даних в мережевих БД можуть мати множинні зв'язки з сегментами старшого рівня. При цьому напрямок і характер зв'язку в мережевих БД не є очевидними, як в ієрархічних, тому імена і напрямок зв'язків повинні ідентифікуватися при описі БД. На рисунку 1.3 зображена схема мережевої моделі БД.

### Рисунок 1.3 – Мережева модель БД

Реляційні БД. Найбільш зручними для користувача є подання даних у вигляді двовимірної таблиці - більшість сучасних інформаційних систем використовує такі таблиці. Бази даних, що складаються з двовимірних таблиць,



Схожість Цитати Посилання  Вилучений

Підміна символів Коментарі  
текст

Джерела на цій сторінці: 9, 13, 19-20

Сторінка 6 з 41

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723

11

називаються реляційними, (англійською «relation» - відношення). На рисунку 1.4 наведено приклад схеми простої реляційної БД.

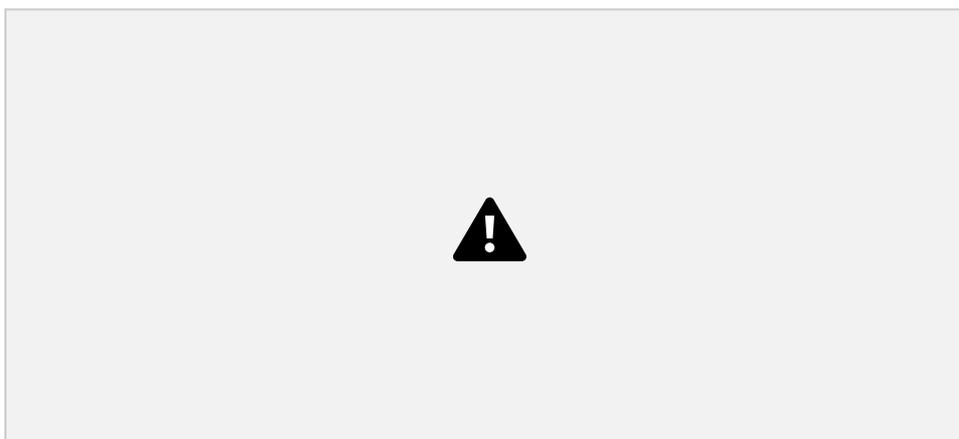


Рисунок 1.4 – Реляційна модель БД Між окремими таблицями реляційної БД проводяться зв'язки (відношення). Пов'язані відношеннями таблиці взаємодіють за принципом головна – підлегла. Головну таблицю називають батьківською, а підпорядковану – дочірньою. Одна і та ж таблиця може бути головною по відношенню до однієї таблиці БД і дочірньою по відношенню до іншої.

Головний принцип реляційної моделі - усунути повторювані поля за допомогою процесу, який називається нормалізацією. Двовимірні нормалізовані таблиці універсальні й прості в розумінні.

Усі реляційні СУБД реалізують той або інший діалект SQL (Structured Query Language – мова структурованих запитів). Універсальна мова, призначена для створення і виконання запитів й обробки даних як у власній базі даних, так і з БД, створених іншими додатками, що підтримують SQL. Ця мова стала фактично стандартною мовою доступу до баз даних.

За обсягом та БД бувають малих або великих розмірів. Бази даних малих розмірів мають невеликі обсяги даних та оброблюються на одному комп'ютері або сервері. Великих ж розмірів містять великі обсяги даних та вимагають розподіленого зберігання та обробки на багатьох серверах.



Схожість Цитати Посилання  Вилючений

 Підміна символів Коментарі  
текст

Джерела на цій сторінці: **16, 21, 24**

Сторінка 7 з 41

Назва документа: **Ковальчук\_Роман\_ОК\_41** ID файлу: **1015062723**

12

За рівнем доступу БД можуть бути однокористувацькими - призначені для використання одним користувачем в один момент часу. Або багатокористувацькими, такими , що Дозволяють одночасний доступ багатьом користувачам до бази даних.

За технологією обробки даних бази даних поділяються на централізовані та розподілені. Централізована база даних зберігається в пам'яті однієї обчислювальної системи. Якщо ця обчислювальна система є компонентом комп'ютерної мережі, є можливим розподілений доступ до такої бази. Такий спосіб використання баз даних часто застосовують у локальних мережах ПК. Розподілена ж база даних складається з декількох, можливо таких, що перетинаються, або навіть дублюють частини, які зберігаються на різних комп'ютерах. Робота з такою базою здійснюється за допомогою системи управління розподіленою базою даних (СУРБД).

За способом доступу до даних бази даних поділяються на бази даних з локальним доступом і бази даних з віддаленим доступом. Загальний поділ баз даних за характеристиками можна представити схемою, зображеною на рис. 1.5.

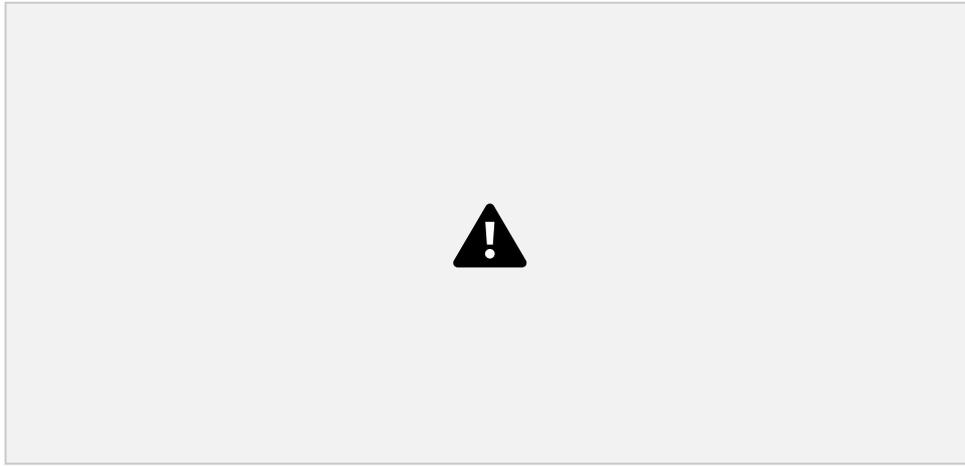


Рисунок 1.5 –

Схема поділу БД за типами організації. модель БД  
Автономні локальні БД (однокористувацькі) є найбільш простими. Вони зберігають свої дані в локальній файлової системі на тому комп'ютері, на якому



Схожість Цитати Посилання  Вилучений

Підміна символів Коментарі  
текст

Джерела на цій сторінці: **2, 4-6, 8-9, 11-12**

Сторінка 8 з 41

Назва документа: **Ковальчук\_Роман\_ОК\_41** ID файлу: **1015062723**

13

встановлені. СУБД, що здійснює до них доступ, знаходяться на тому ж самому комп'ютері. Мережа не використовується. Розробникові автономної БД не доводиться мати справу з проблемою паралельного доступу, коли два клієнти намагаються одночасно змінити один і той самий запис.

Архітектура систем БД з мережним доступом передбачає виділення однієї з машин мережі як центрального сервера файлів. На такій машині зберігається централізована БД, що спільно використовується. Всі інші машини мережі виконують функції робочих станцій, за допомогою яких підтримується доступ призначеної для користувача системи до централізованої бази даних.

Централізовані бази даних з мережним доступом припускають різні архітектури подібних систем: файл-сервер, клієнт-сервер, тривірнева архітектура.



Схожість Цитати Посилання  Вилучений

 Підміна символів Коментарі  
текст

Джерела на цій сторінці: **2, 4, 8-9, 12**

Сторінка 9 з 41

Назва документа: **Ковальчук\_Роман\_ОК\_41** ID файлу: **1015062723**

14

## 2 ВИБІР СЕРЕДОВИЩА РЕАЛІЗАЦІЇ

### 2.1 Аналіз потреб медичного закладу

Медичні заклади, такі як лікарні, поліклініки, лабораторії та інші, мають свою специфіку і особливості, які варто враховувати при розробці бази даних для них.

Загальна специфіка медичного закладу: різні типи медичних закладів мають власні потреби та особливості. Наприклад, лікарня може потребувати додаткових функцій для управління лікарняними відділеннями, операційними

блоками тощо, тоді як поліклініка фокусується на прийомі пацієнтів і веденні журналу прийомів та даних про пацієнтів.

Медичні заклади мають вимоги до точності, надійності та безпеки зберігання медичних даних. Наприклад, конфіденційність даних пацієнтів та доступ до них можуть бути обмежені з метою забезпечення конфіденційності та дотримання вимог законодавства про захист персональних даних.

Облік медичних записів: база даних медичного закладу повинна забезпечувати зберігання та оновлення медичних записів пацієнтів. Це може включати інформацію про особисті дані пацієнтів, медичну історію, результати обстежень та досліджень, призначення лікарів тощо.

Розклади та графіки роботи: база даних медичного закладу має містити інформацію про графіки роботи лікарів, мед-сестер, лаборантів та іншого медичного персоналу. Це допоможе впорядкувати процеси призначення, планування прийому пацієнтів та організацію робочого часу.

Звітність та аналітика: база даних медичного закладу може включати функціональність для створення звітів. Це дозволяє аналізувати дані про прийом пацієнтів, медичні процедури, вести статистику. Звіти можуть бути використані для прийняття управлінських рішень та вдосконалення ефективності медичного закладу.



Схожість Цитати Посилання  Вилучений

Підміна символів Коментарі  
текст 

Сторінка 10 з 41

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723

15

Зважаючи на розглянуту специфіку медичного закладу, функціональність, яку необхідно реалізувати в базі даних, може включати наступні аспекти: Управління пацієнтами:

- Зберігання особистої інформації про пацієнтів, таку як ім'я, адреса проживання, контактні дані тощо.

- Медична історія пацієнта, включаючи результати обстежень, діагнози, призначення та прийоми ліків.

- Реєстрація пацієнтів на прийоми, зберігання графіку прийому.

Управління медичним персоналом:

- Зберігання інформації про лікарів, медсестер, лаборантів та інший медичний персонал.

- Розклад роботи, графіки прийому пацієнтів та планування відпусток медичного персоналу.

- Відстеження кваліфікацій, навичок та ліцензій медичного персоналу. Управління лікарняними відділеннями та операційними блоками (за їх наявності):

- Реєстрація та графіки госпіталізацій та виписок пацієнтів.

- Облік ліжок та розташування пацієнтів у відділеннях.

- Планування та контроль операцій, включаючи резервування операційних залів та ресурсів.

Звітність та аналітика: Функціональність бази даних має включати можливість створення звітів та аналізу медичної статистики, наприклад, статистики за діагнозами, процедурами, лікарями тощо, що може бути корисним для управління та планування роботи медичного закладу.

## 2.2 Порівняння підходящих середовищ реалізації

Огляд найпоширеніших систем управління базами даних (СУБД), які можуть бути використані для медичних закладів:

- MySQL



Схожість Цитати Посилання Вилучений

Підміна символів Коментарі  
текст

Сторінка 11 з 41

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723

16

Переваги: MySQL є однією з найпопулярніших СУБД. Має широкий функціонал, надійність та високу швидкодію. В MySQL зручне управління правами доступу та високий рівень безпеки.

Недоліки: Обмежені можливості управління правами доступу та аудиту активності користувачів. Недостатня масштабованість для великих обсягів даних та високих навантажень.

- PostgreSQL

Переваги: PostgreSQL є відкритою СУБД з потужними можливостями.

Вона підтримує розширені функції, такі як реляційність, JSON-операції, геодани та інші. PostgreSQL забезпечує високу надійність та відновлення в разі відмови системи. Також вона підтримує реплікацію та масштабування для обробки великих навантажень.

Недоліки: Складніше налаштування та адміністрування порівняно з іншими СУБД, що може вимагати додаткових навичок персоналу. Вища ціна комерційної підтримки та ліцензійних витрат порівняно з альтернативними засобами. Порівняно з MySQL може мати трохи меншу швидкодію в певних сценаріях роботи з даними.

#### 📄 Oracle Database

Переваги: Oracle є однією з провідних (а значить і найпоширеніших) комерційних СУБД у світі. Вона має розширений функціонал, включаючи підтримку розподіленого обчислення, високу швидкодію та масштабованість. Oracle має потужну систему управління правами доступу та забезпечує високий рівень безпеки. Надає хороші функціональні засоби для аналітики та звітності.

Недоліки: Високі витрати на ліцензування та підтримку. Складне налаштування та управління системою, що може вимагати спеціалізованого персоналу. Обмежена підтримка вбудованих аналітичних засобів порівняно з деякими іншими СУБД.

Вибір конкретної СУБД для медичного закладу залежить від різних факторів, таких як унікальні потреби медичного закладу, бюджет, наявність кваліфікованого персоналу, вимоги до масштабованості та безпеки даних.

Кожна з



Схожість Цитати Посилання  Вилучений

текст  Підміна символів Коментарі

Сторінка 12 з 41

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723

17

перерахованих систем має свої переваги, недоліки та особливості, і вибір залежатиме від певних потреб та обставин, та нас цікавить Microsoft Access. Це реляційна база даних, яка зазвичай використовується для створення середніх за розміром баз даних. Переваги використання Microsoft Access для медичного закладу наступні:

📄 Простота використання: Microsoft Access має інтуїтивно зрозумілий інтерфейс та інструменти для швидкого створення форм, звітів та запитів, що

полегшує роботу з базою даних.

■ Швидкість розробки: Access дозволяє швидко створювати бази даних без необхідності в програмуванні, що може бути корисним у випадку, коли потрібно швидко запровадити рішення для медичного закладу.

■ Інтеграція з іншими продуктами Microsoft: Microsoft Access добре інтегрується з іншими продуктами Microsoft, такими як Excel, Word і Outlook. Це дозволяє зручно обмінюватись даними та використовувати їх у різних контекстах.

■ Масштабованість: Access може бути використаний для створення середніх за розміром баз даних, але також має можливості розширення для обробки більших обсягів даних та виконання складніших операцій.

■ Гнучкість: Microsoft Access дозволяє гнучко налаштовувати базу даних під конкретні потреби медичного закладу. Дає можливість створювати власні форми, звіти та запити, а також виконувати аналіз даних за допомогою різних інструментів.

Недоліки використання Microsoft Access в медичному закладі такі:

■ Обмежена потужність та обробка даних: Access не є настільки потужною, як деякі інші СУБД. Вона може не підходити для надто великих обсягів даних або складних аналітичних запитів.

■ Обмежена безпека та контроль доступу: Access має обмежені можливості управління правами доступу користувачів, що може бути проблематичним у таких медичних закладах, в яких конфіденційність даних є пріоритетом, та для мого варіанту цілком підійде.



Схожість Цитати Посилання  Вилючений

 Підміна символів Коментарі  
текст

Джерела на цій сторінці: 23, 25

Сторінка 13 з 41

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723

18

### 2.3 Огляд Microsoft Access і його основних функцій

Microsoft Access - це реляційна база даних, що входить до складу пакету Microsoft Office. Вона надає зручні інструменти для створення, управління та аналізу баз даних. Access дозволяє користувачам ефективно створювати

форми, запити, звіти та макроси, що полегшує роботу з даними та автоматизовує багато рутинних завдань.

Інтеграція з іншими продуктами Office: Access добре інтегрується з іншими продуктами Microsoft Office, такими як Excel, Word та Outlook. Це дає змогу обмінюватися даними з цими програмами та використовувати їх функціонал разом з Access.

З кожною наступною версією Microsoft Access, функціональні можливості системи постійно розширювалися. Були додані нові інструменти для створення форм, запитів, звітів та макросів, а також поліпшення в продуктивності та безпеці даних. Microsoft постійно оновлює свої продукти й продовжує випускати нові версії Microsoft Access, вдосконалюючи його функціонал та впроваджуючи нові технології. Остання версія, доступна на час написання цієї доповіді - Microsoft Access 2019. Кожна версія принесла з собою покращення і нові можливості для роботи з базами даних.

Що дуже важливо, Microsoft Access має простий інтерфейс, що робить його доступним користувачам для роботи з базою даних. Це особливо корисно для медичного закладу, де персонал отримує змогу швидко навчитися експлуатувати розроблену базу даних.

Створення та керування базами даних у Microsoft Access є одним з основних функціональних компонентів цієї програми. СУБД Microsoft Access виконує наступні функції:

1. Створення нової бази даних. Бази даних створюються за допомогою вбудованого майстра створення бази даних. Можна вибирати шаблон або почати з порожньої бази даних.



Схожість Цитати Посилання  Вилучений

 Підміна символів Коментарі  
текст

Сторінка 14 з 41

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723

19

2. Створення таблиць. Таблиці призначені для зберігання даних. При створенні таблиць задаються імена полів та їх типи даних, таких як текст, числа, дата/час, зображення, тощо, встановлюються обмеження на дані, визначаються поля первинного ключа.

3. Створення запитів. За допомогою запитів можна, модифікувати й

аналізувати дані в БД. Можна використовувати вбудований майстер для створення простих запитів або використовувати режим SQL для складних запитів.

4. Створення форм. Форми дозволяють створювати користувацький інтерфейс для внесення та відображення даних. Форми створюються для введення записів, їх редагування та відображення даних у зручному форматі.

5. Створення звітів. Звіти представляють дані у зручному форматі для друку або експорту. Звіти можна створювати за допомогою майстра звітів або налаштовувати їх вручну, вибираючи потрібні поля й вигляд звіту.

6. Взаємодія з іншими даними: Microsoft Access підтримує імпорт та експорт даних з інших джерел, таких як Excel, текстові файли, SQL-сервери тощо. Дає можливість збирати дані з різних джерел та об'єднувати їх у вашій базі даних.

7. Управління базою даних: Microsoft Access надає інструменти для управління базою даних, такі як створення резервних копій, розподіл бази даних між користувачами, управління правами доступу та захистом даних.

Ці функції Microsoft Access дозволяють ефективно створювати, управляти та використовувати бази даних для медичного закладу. Короткий опис перспектив розробки таблиць, запитів, форм та звітів для роботи з медичними даними:

Таблиці будуть створені для зберігання медичних даних, таких як інформація про пацієнтів, медичні записи, лікарські препарати, діагнози, лікарняні, тощо.

Запити - можна створювати запити для вибору всіх пацієнтів певного лікаря, пошуку пацієнтів з певними діагнозами або створення звіту про лікарські препарати, що закінчуються.

Форми створюються для введення нових записів про пацієнтів, редагування існуючих записів та перегляду даних у зручному форматі.



Схожість Цитати Посилання  Вилучений

 Підміна символів Коментарі

Сторінка 15 з 41

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723

За допомогою звітів буде створено візуальні представлення медичних даних у зручному форматі для друку або експорту. Є можливість створювати

звіти, які показують статистику пацієнтів, результати медичних досліджень, графіки залежності показників здоров'я тощо.

Перелічені компоненти Microsoft Access дають змогу ефективно керувати й проводити аналіз медичних даних в медичному закладі. Можемо створювати таблиці для зберігання даних, за допомогою запитів отримувати необхідну інформацію, використовувати форми для зручного введення та перегляду даних, а також створювати звіти для аналізу результатів.



Схожість Цитати Посилання  Вилучений

текст  Підміна символів Коментарі

Сторінка 16 з 41

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723

### 3.1 Проектування й реалізація таблиць і їх зв'язків

Задача дипломного проєкту полягає в створенні бази даних медичного закладу в середовищі MS Access, однією з основних переваг котрого є відносна простота і практичність для недосвідченого користувача СУБД. Я вирішив розробляти БД саме поліклініки оскільки в лікарні потрібно зберігати набагато більше даних як про персонал так і про пацієнтів, а також лікарня може дозволити собі спеціалістів, задачею котрих було б оперування складнішою системою управління базою даних, тож тут Access уступав би складнішій в користуванні та більш потужній програмі, як то наприклад Oracle. А якщо основні дані, котрими повинна оперувати поліклініка, такими як контактна інформація про лікарів і їх графік роботи, історія хворіб конкретного пацієнта, реєстрація людей прив'язаних до їхнього сімейного лікаря, журнал видачі лікарняних й подібна інформація то в даному випадку вибране середовище розробки цілком доречне.

Отже в цій БД я хочу реалізувати можливість швидкого введення, редагування, пошуку й виведення такої інформації:

- Пацієнти – їх ПІБ, стать, вік, паспортні дані, контактна інформація, відомості про наявність страхування чи виходу на лікарняний, а також коротка історія хвороби в електронному вигляді й номер амбулаторної карти.

- Персонал поліклініки, на даному етапі лише лікарів – їхня спеціалізація, загальна інформація, контактні дані, стаж і категорія працівника, відомість про те чи не перебуває зараз лікар у відпустці.

- Інформація про розклад лікаря і кабінети поліклініки – в які дні у який час і в якому кабінеті можна скерувати пацієнта до того чи іншого спеціаліста, який інвентар мусить бути в цих кабінетах.

- Журнал прийомів – який пацієнт звернувся до котрого лікаря, в який час і з якою причиною і з якими скаргами, яка послуга була надана і який діагноз встановлено.



Схожість Цитати Посилання  Вилучений

текст  Підміна символів Коментарі

Сторінка 17 з 41

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723

■ Обстеження – на якому прийомі людину було скеровано на той чи інший аналіз, який діагноз це допомогло встановити і скільки коштувала ця послуга.

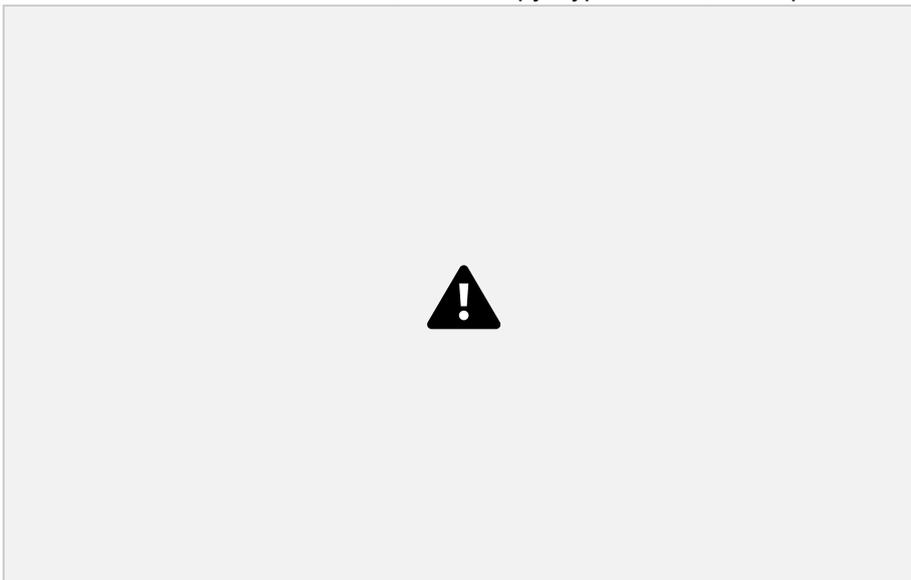
■ Видача листів непрацездатності (лікарняні) – так само, на якому прийомі і яким лікарем конкретному пацієнту був виданий, статус (закритий чи відкритий) та інформація про дату видачі й закриття.

Для зручності оперування вихідною інформацією бази даних потрібно передбачити створення форм і запитів, що будуть виводитись на основну форму, наприклад список пацієнтів приділених до певного сімейного лікаря, або перелік лікарів із конкретною спеціальністю й визначеною категорією чи стажем.

Та спочатку потрібно ввести в базу даних вхідну інформацію, а для цього треба створити таблиці котрі міститимуть необхідні нам дані й реалізувати зв'язки між ними.

Планується вище згадану інформацію розподілити задля сортування по дев'ятьом таблицям. Структура й тип збережених даних у трьох основних таблицях наводяться у табл.3.1 – табл.3.3.

Таблиця 3.1 – Структура таблиці «Лікар»



Ім'я поля	Ключове поле	Тип даних	Довжина поля
ID Лікаря + Автонумерація			
ПІБ Лікаря		Короткий текст	50
Спеціальність		Короткий текст	20
Фото		Об'єкт OLE	
Дата народження		Дата й час	
Категорія		Короткий текст	5
1. Стаж		Число	2.
Е-пошта		Короткий текст	319
3. Телефон		Короткий текст	10
		Короткий текст	80

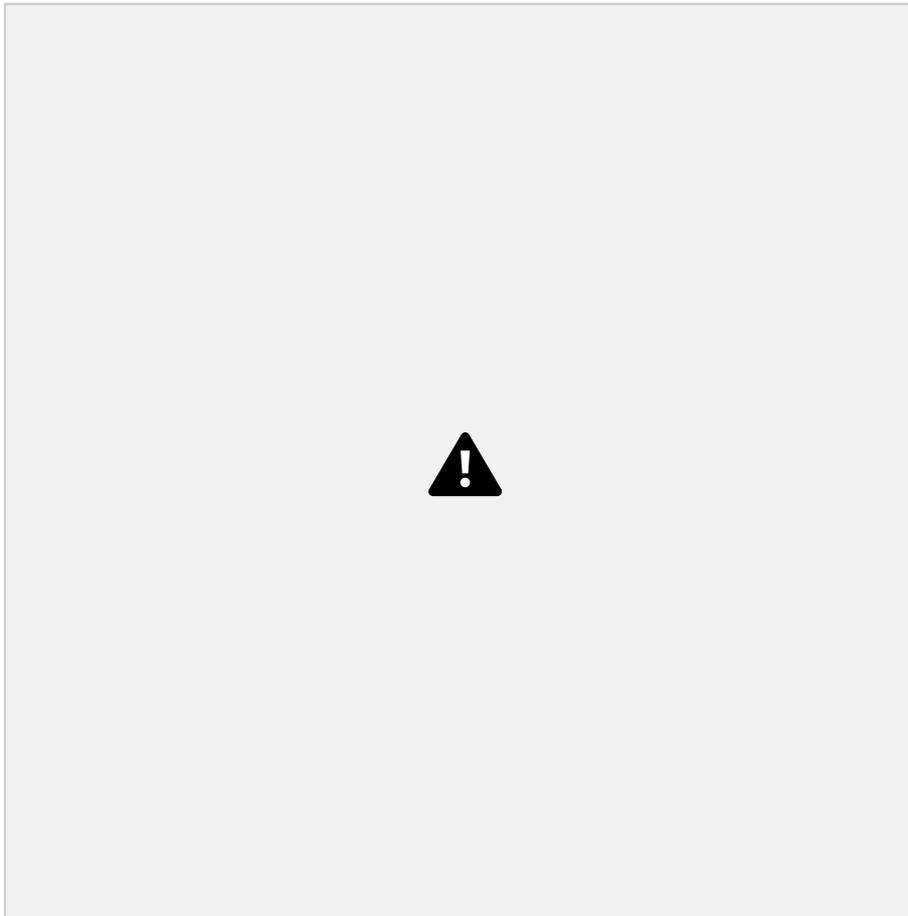
4. Адреса проживання

5. Відпустка Так/Ні

Таблиця 3.2 – Структура таблиці «Пацієнт»

Ім'я поля Ключове поле Тип даних Довжина поля





23

кий текст 50

он

7. Лікарняний Так/Ні

8. Страхівка Так/Ні

9. Історія хвороби

Довгий текст 10000

Сімейний лікар

Короткий текст 50

Таблиця 3.3 – Структура таблиці «Прийом»

Ім'я поля

Ключове поле

Тип даних

Довжина поля ID Прийому + Автонумерація

Амбулаторна карта Число Ціле число Пацієнт Короткий текст 50 Лікар

Короткий текст 50 Кабінет Число 255 Дата Дата й час

Причина звернення Короткий текст 12 Скарги Короткий текст 255

Код діагнозу Число Ціле число

Решта даних планується записати у інші шість менших таблиць, котрі будуть продемонстровані на рисунках нижче. Задля уніфікації усі імена таблиць записано в одиничному числі, а не наприклад «Лікарі» чи «Аналізи», аби не плутатись в подальших запитах і не згадувати як же таблиця була записана.

Правильний вибір типу даних і обмеження їхньої довжини дає економію місця яке БД займатиме на жорсткому диску, що позитивно вплине на її оптимізацію. Стабільна й швидка робота таблиць – одна з пріоритетних задач. Використані типи полів мають такі параметри:

 Автономерація – використовується для задання певному полю свого унікального номеру – Access автоматично збільшує на одиницю ідентифікатор попереднього запису для наступного.

 Число - зберігає числові дані, представляється в одному із трьох основних форматів – «Байт» - займає один байт місця на диску й може містити додатні значення від 0 до 255, «Ціле число» - в діапазоні (від -32768 до 32767) і займає два байти, й «Довге ціле число» (від -2147483648 до 2147483647), що займає чотири байти.

 Короткий текст – дозволяє вводити до 255 символів за замовчуванням, відповідно займатиме до 255 байтів пам'яті в залежності від використовуваної довжини поля.

 Дата й час – дає можливість вводити дні/місяці/роки, або вибрати формат «час» і зберігати години/хвилини/секунди й займає 3 байти пам'яті.  Об'єкт OLE – дозволяє вкладати у БД медіа-файли, як картинка або відео й об'єм місця котре займатиме таке поле залежатиме від розміру вкладеного файлу.

 Так/Ні – логічний елемент, в якому можна вибирати між двома типами значень, за замовчуванням виглядає як чекбокс.

Було б неправильно використати наприклад формат «Довге ціле число», що дозволяє вписувати максимальне додатнє число 4294967295 для поля «Стаж», в котрому планується зберігати числа, що складаються не більше як із двох цифр, оскільки люди рідко доживають до ста років і тим паче не працюють стільки часу лікарем. Тож доречно вибрати мінімальний формат «Байт», якого достатньо щоб вводити двозначні числа й витратити на такі поля один байт пам'яті замість чотирьох. Також я виставив довжину поля «Е-пошта» в таблиці «Лікар» у 319-ть символів, оскільки за стандартом можна створити адресу з іменем максимум в 64-и символи, один символ займе рівлик й домен може займати 254-и символи. Таким чином ім'я скриньки не може складатися більше ніж із 319-ти символів і немає потреби у більшій довжині поля.

Джерела на цій сторінці: 26

Сторінка 20 з 41

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723

25

о входить до пакету MS Office, ми  
.1.



Рисунок 3.1 – Знімок екрану MS Access

Зверху маємо трєю із вкладками «Файл», «Основне», «Створення» й так далі. Під цією трєю розміщені інструменти вибраної вкладки. Зліва розміщене поле, в котре будуть поміщатись посилання на створені об'єкти, як то таблиці, форми, запити й звіти. Основне поле це – робоча зона.

Перший крок – створення бази

даних – натискаю кнопку «Файл» – «Створити», вибираю плитку «Пуста база даних», вводжу ім'я своєї БД і її розташування на диску, після чого натискаю кнопку «Створити». У жовтому рядку повідомлень натискаю «Увімкнути зміст» для того щоб могли почати працювати із новоствореною базою.

Коли БД створена – починаю будувати таблиці, для цього використовую кнопку конструктор таблиць, що у вкладці «Створення» й пуста таблиця виглядає таким чином, як на рис. 3.2.

Рисунок 3.2 – Майбутня таблиця «Лікар»



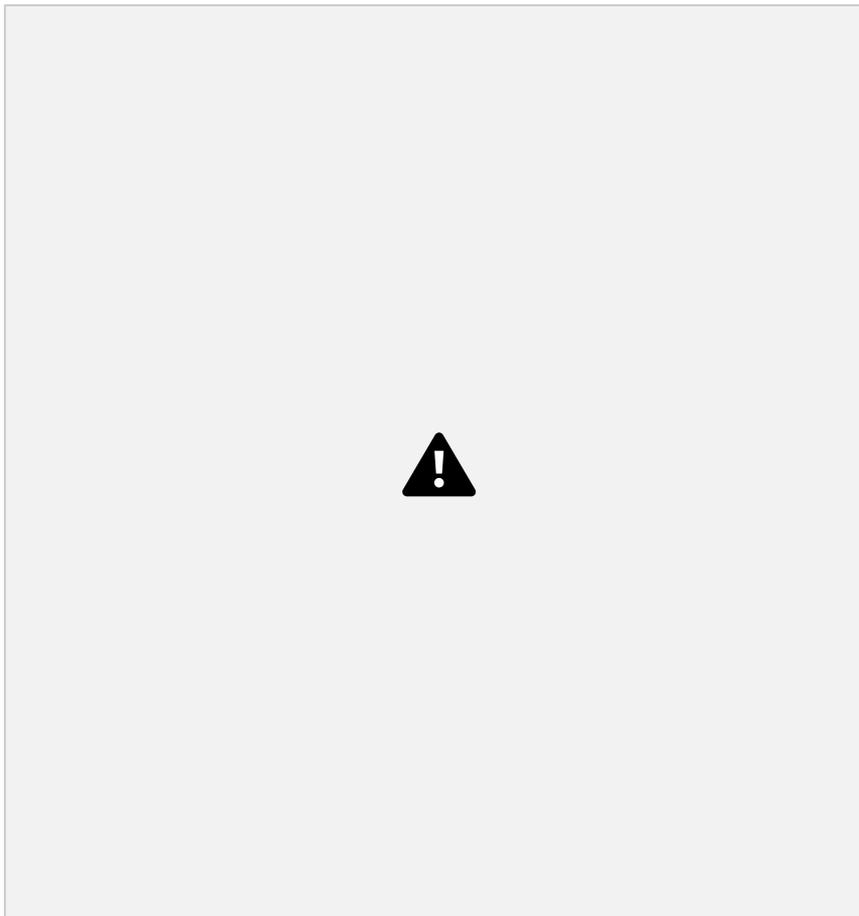
Схожість Цитати Посилання  Вилучений

Підміна символів Коментарі  
текст

Джерела на цій сторінці: 27

Сторінка 21 з 41

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723



26

тип їхніх даних, крім  
», як до прикладу розмір

Рисунок 3.3 – Зміна  
розміру поля  
числових даних, що  
за замовчуванням  
встановлюється як  
«Довге ціле число»

Після того як усі поля  
створено й

налаштовано, а також задане ключове поле, таблиця «Лікар» у режимі конструктора має такий вигляд як на рисунку 3.4.

### Рисунок 3.4 – Таблиця «Лікар» у режимі конструктор

Таким самим чином я створюю і налаштовую й інші вісім таблиць, у більшості з яких задля зручності і швидкості вводу інформації, в типі даних вибираю майстер підстановок, де програма пропонує мені або створити вручну значення для підстановки, або взяти їх у іншій таблиці, в процесі наповнення таблиць я звертався до обох методів. Після чого на базі зв'язку підстановки даних



Схожість Цитати Посилання  Вилучений

 Підміна символів Коментарі  
текст

Сторінка 22 з 41

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723

27

між таблицями у вкладці «Зв'язки» я створював повноцінні відношення між таблицями.

Для з'єднання таблиць через майстер підстановки, після вибору відповідного варіанту переносимось у вікно вибору батьківської таблиці з якої відбуватиметься підстановка, після чого в цій таблиці вибираю потрібні поля, як на рисунку 3.5, наведеному нижче. Далі майстер підстановок пропонує вибрати поле за яким буде відбуватись сортування даних. Після чого потрібно задати ширину стовбців, або залишити її такою ж. Далі я натискаю кнопку «Збереження цілісності даних» й за потреби в окремих полях ставлю галочку навпроти «Дозволити кілька значень» й нарешті кнопка «Готово»

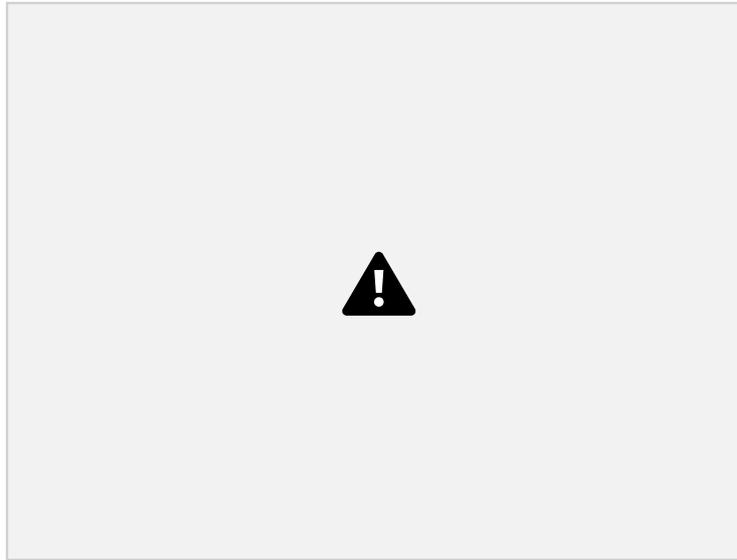


Рисунок 3.5 – Одне з

діалогових вікон майстра підстановок

Коли усі таблиці створено й налаштовано у конструкторі, а також проставлено підстановку даних, я починаю заповнювати таблиці, перейшовши у відповідний режим «Подання таблиці», декілька полів у певних таблицях залишаю порожніми, щоб потім продемонструвати їх заповнення через інструмент форми. На цьому етапі таблиці виглядають, як на рисунках 3.6 – 3.14.



Схожість Цитати Посилання Вилучений

текст Підміна символів Коментарі

062723

28



Рисунок 3.6 – Вигляд заповненої таблиці «Лікар»

Рисунок 3.7 – Вигляд заповненої таблиці «Пацієнт»

Рисунок 3.8 – Вигляд заповненої таблиці «Прийом»

Рисунок 3.9 – Вигляд заповненої таблиці «Кабінет»



Схожість Цитати Посилання  Вилучений

текст  Підміна символів Коментарі

Сторінка 24 з 41

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723



Рисунок 3.10 – Вигляд заповненої таблиці «Розклад»



Рисунок 3.11 – Вигляд заповненої таблиці «Лікарняний»

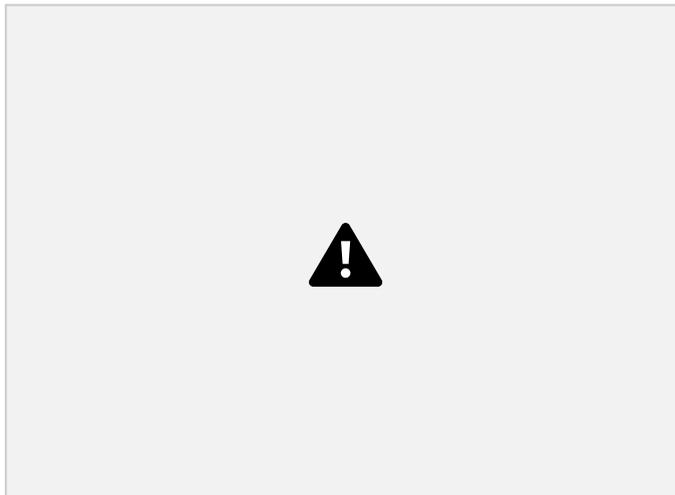


Рисунок 3.12 – Вигляд заповненої таблиці «Діагноз»

Рисунок 3.13 – Вигляд таблиці «Обстеження»

Рисунок 3.14 – Вигляд структури таблиці «Аналіз»

Тепер заповнені таблиці потрібно зв'язати у схемі даних, для цього вибираю будь-яку таблицю, вибираю вкладку «Таблиця» й натискаю кнопку «Зв'язки». Відкрився конструктор зв'язків. В ньому відображено створені таблиці і наявні зв'язки підстановок. Ці зв'язки вибираю правою кнопкою миші, вибираю «Змінити зв'язок» - відкривається вікно редагування зв'язків. В цьому вікні бачимо які два поля зв'язуємо і видно автоматично встановлений тип зв'язку. Окрім кнопки забезпечення цілісності даних бачимо ще дві. Каскадне оновлення пов'язаних полів означає, що при редагуванні даних відповідного поля в одній таблиці, інформація в іншій оновиться. А якщо поставити галочку навпроти каскадного видалення пов'язаних полів, то видалення запису в одній таблиці призведе до видалення його й в іншій. Проставляю потрібні мені галочки для кожного зв'язку окремо. Крім того, за допомогою кнопки «Тип об'єднання» можна відредагувати попередні налаштування майстра підстаночки. Окрім налаштування зв'язків створених через майстер підстаночки створюю нові – для цього просто затискаю курсор на потрібному полі й тягну його до того поля, з котрим хочу з'єднати. Після цієї дії відкривається таке ж вікно редагування зв'язків, воно представлене на рисунку 3.15.





Рисунок 3.15 – Вікно редагування зв'язків

Зв'язую таблиці переважно одним найефективнішим типом зв'язків – один до багатьох (наприклад в одного пацієнта

може бути декілька діагнозів, діагноз не може бути поставлений без пацієнта), але також використовую і зв'язок один до одного (в одного лікаря може бути лише один код працівника). Усі зв'язки багато до багатьох в моїй таблиці реалізовані через проміжкові таблиці. Один лікар може провести безліч прийомів, як і один пацієнт може пройти безліч прийомів. Ключові поля (первинний ключ), які я створював при конструюванні таблиць допомагають встановити батьківську таблицю при зв'язку один до багатьох – таблиця із первинним ключем є головною, а таблиця із вторинним (поле що не є ключем) – дочірня. Наприклад для зв'язку кабінет-прийом дочірньою є таблиця «Прийом», оскільки для підстановки номера кабінету в якій проводиться прийом у ній використовується первинний ключ таблиці «Кабінети». На рисунку 3.16 зображено схему даних моєї БД.



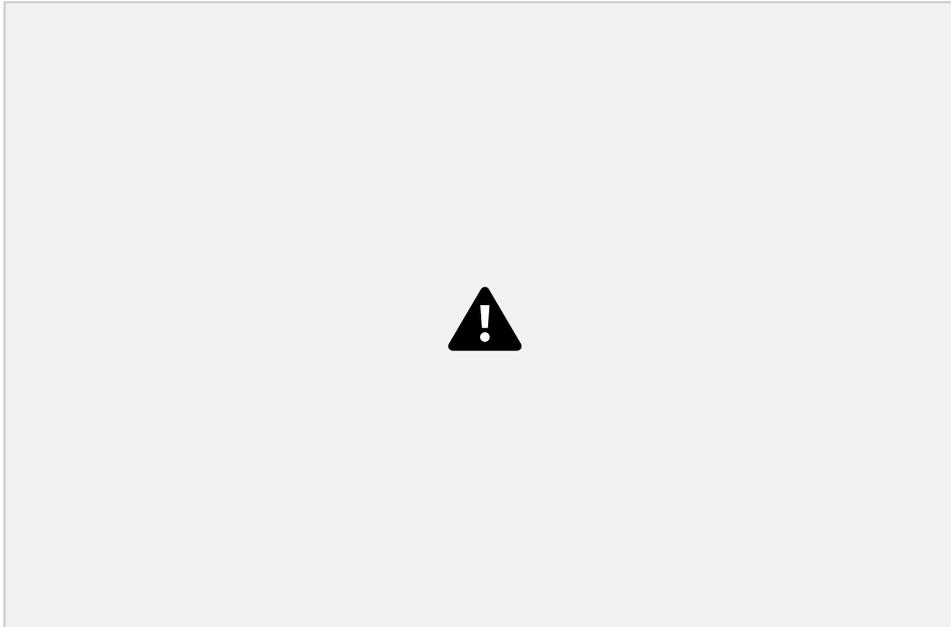


Рисунок 3.16 – Схема бази даних поліклініки 3.2 Створення форм для заповнення таблиць

Вводити дані в таблиці напряму не надто зручно – працівнику медичного закладу, не надто добре володіючому комп'ютером потрібен інтуїтивно зрозумілий спрощений інтерфейс в якому легше працювати. Для цього в Access існують форми – потрібно зробити форми через які можна буде згруповано ввести дані у всі використовувані в організації медичного процесу поля таблиць БД поліклініки. Для таблиць «Аналіз» і «Діагноз» форми можна не створювати оскільки вони зберігають лише ідентифікатор наперед визначених обстежень і хворіб, тож не планується часте редагування інформації в них.

Також не доведеться регулярно вводити нову інформацію про контактні дані персоналу поліклініки, тож задля зручності можна об'єднати інформацію з таблиць «Лікар» і «Розклад» в одній формі. Ця форма зображена на рис. 3.17.



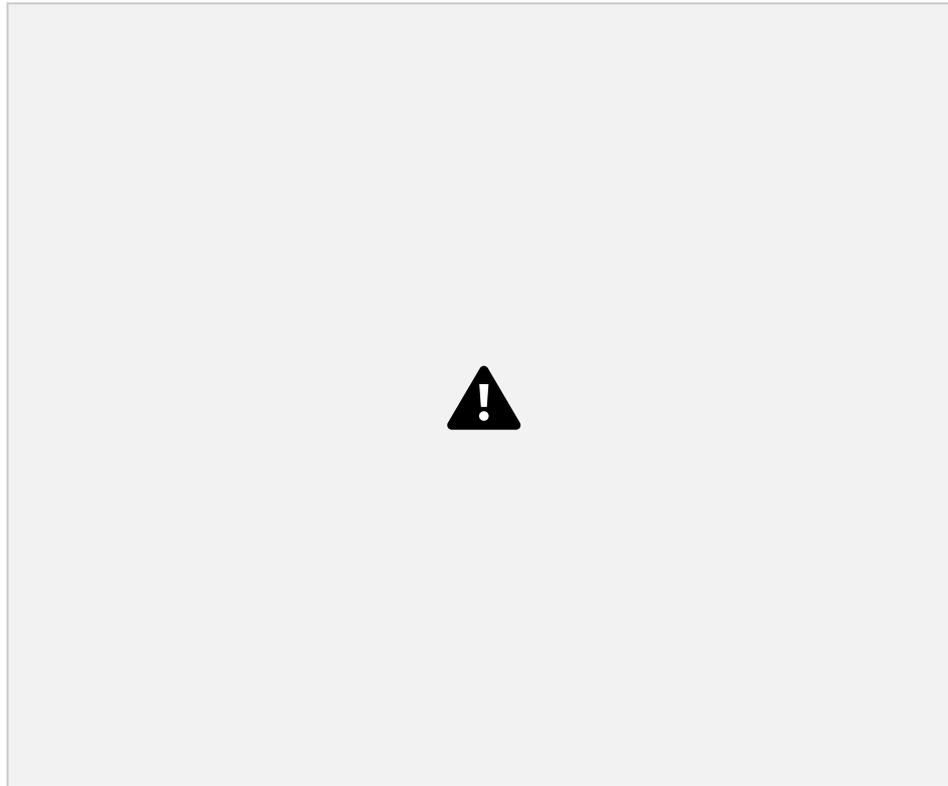


Рисунок 3.17 –

Форма «Лікарі і їх графік роботи»

Для того щоб створити наступну форму переходжу у вкладку створення й вибираю конструктор форм, там із списку полів перенешу потрібні мені поля із вибраної таблиці, наразі це усі поля таблиці «Пацієнти» для форми «Інформація про пацієнтів». У конструкторі виставляю на сітці поля форми в зручному й ергономічному порядку, що зображено на рисунку 3.18





Рисунок 3.18 – Конструктор форм

Наступним кроком я копіюю кнопки функціоналу з попередньої форми «Лікарі і їх графік роботи», а саме кнопки навігації – перехід до попереднього і наступного запису, кнопки роботи із записами – додання, збереження й скасування запису, а також кнопку виходу з форми. Після цього за допомогою вкладки формат і аркушу властивостей трохи стилізую форму, змінивши шрифти, задавши сині рамки, вирівнявши текст в полях по центру й прибравши непотрібні елементи й після цього форма «Інформація про пацієнтів» стала виглядати таким чином, як на рис. 3.19.



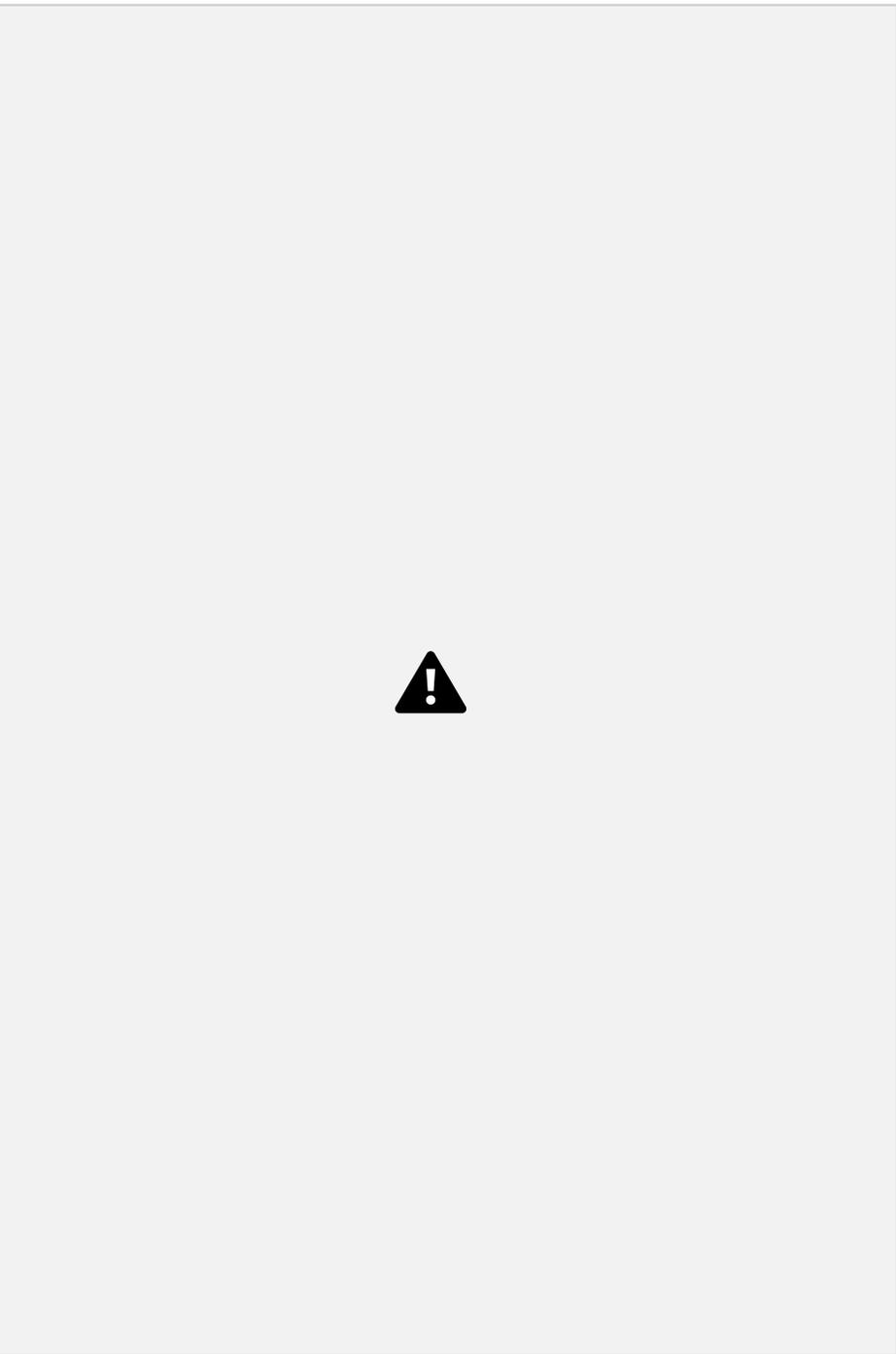


Рисунок 3.19 – Форма «Інформація про пацієнтів»

Тепер у готовій формі заповнюю поле «Історія хвороби» й створюю наступні форми

аналогічно тому як і дві попередні, надаючи їм схожий стиль і так само вводжу інформацію в пусті поля. У вікні властивостей вмикаю вивід форми як окремого вікна. Решту готових форм зображено на рисунках 3.20 – 3.22

Рисунок 3.20 – Форма «Кабінет»



текст Підміна символів Коментарі

41

**1015062723**

36

Рисунок 3.21 – Форма «Лікарняний»

Рисунок 3.22 – Форма «Прийом і обстеження»



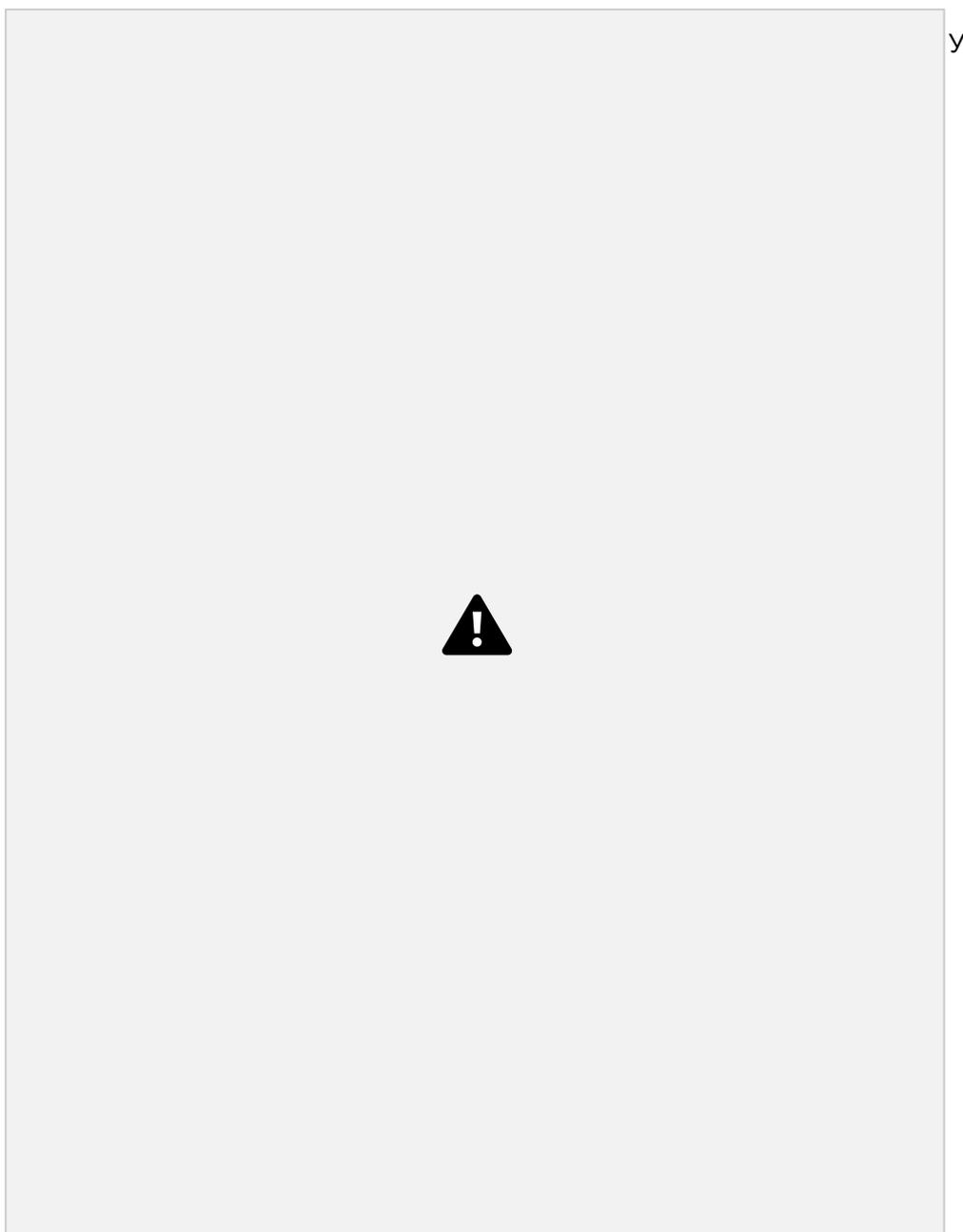
Схожість Цитати Посилання  Вилучений

текст  Підміна символів Коментарі

Сторінка 32 з 41

Назва документа: Ковальчук\_Роман\_OK\_41 ID файлу: 1015062723

37



вищенаведеній формі «Прийом і обстеження» я об'єднав інформацію з двох таблиць використавши майстер форм і функцію підформи, для того щоб можна

було переглядати всі записи з таблиці «Прийом», а не лише ті прийоми, на яких пацієнта було скеровано проходити аналіз, довелось змінити зв'язок між цими двома таблицями, аби таблиця «Обстеження» стала дочірньою. Тепер, маючи усі ці форми можна зручно вносити записи в базу даних поліклініки.

### 3.3 Розробка запитів для вибірки даних

Тепер, коли у БД поліклініки є інструмент для введення інформації, потрібно реалізувати здатність виводити потрібні дані. Для забезпечення такої функції доцільно застосувати запити, які дозволяють виконувати пошук і обробку даних у потрібному форматі й зручному для користувача вигляді.

У Microsoft Access є вбудований майстер для конструювання запитів, та він аналогічний майстру побудови форм й немає сенсу окремого розглядати його. Замість цього можна побудувати запити вручну на мові SQL, такі запити можна налаштувати більш гнучко й добитися більшого функціоналу, а ще складніші виводи інформації планується реалізувати за допомогою інструменту звіти. Крім перегляду інформації звіти також дозволяють модифікувати дані у таблицях, та для цього в моїй базі уже створено форми. Тож є сенс побудувати декілька простих звітів для аналізування конкретних полів з вибіркою за певними критеріями, для цього знадобляться основні ключові слова із таблиці 3.4.

Таблиця 3.4 – Команди виводу в мові SQL

Синтаксис Функція

SELECT Перелічує поля, які містять потрібні дані.

FROM Перелічує таблиці, у яких містяться поля, зазначені в реченні SELECT.

WHERE Визначає умови для поля, яким мають відповідати всі записи, щоб бути включеними до результатів.

ORDER BY Визначає спосіб сортування результатів.



Схожість Цитати Посилання Вилучений

Підміна символів Коментарі  
текст

Джерела на цій сторінці: 14

Сторінка 33 з 41

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723

Ці команди правильно писати великими літерами й з нового рядка. Назви полів ж можна писати з малої оскільки SQL не розрізняє їх. Назви полів, що

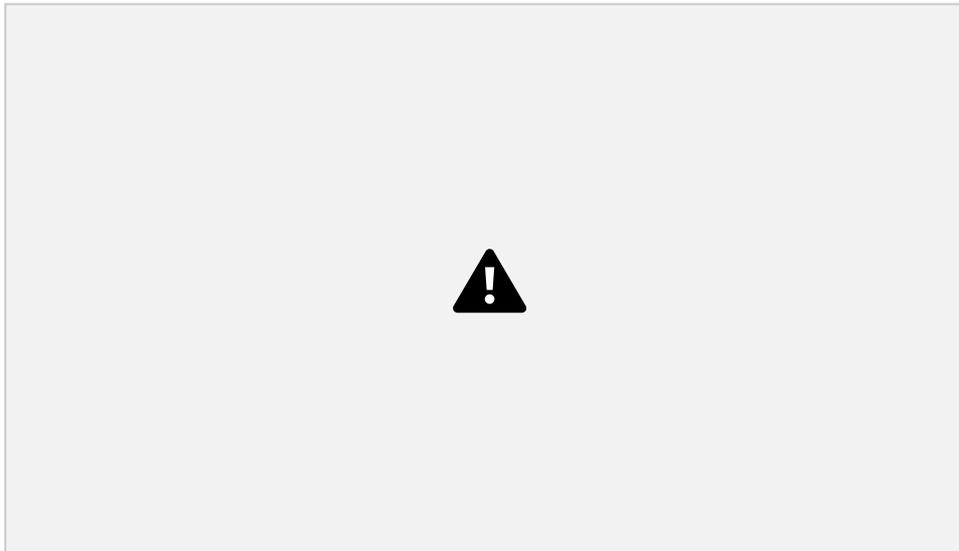
містять пробіли потрібно писати у квадратних дужках.

В базі даних поліклініки можна використати наступні запити:

- Виведення інформації про лікарів певної спеціальності.
- Перелік повнолітніх пацієнтів, закріплених за вибраним сімейним лікарем. ■ Список відкритих листів непрацездатності.
- Вибірка пацієнтів, що пройшли медогляд у конкретну дату.

Спочатку створюю запит на виведення всієї інформації про лікарів алергологів із таблиці «Лікар», для цього набираю такий код:

```
SELECT *  
FROM лікар  
WHERE спеціальність
```



о на панелі швидкого  
нформації наступного разу.  
23.

Рисунок 3.23 – Збереження запиту «Лікарі-алергологи»

Наступний запит – перелік

повнолітніх пацієнтів, закріплених за сімейним лікарем Черногуз Оленою Денисівною:

```
SELECT [ПІБ пацієнта],[Дата народження],[Сімейний лікар]  
FROM пацієнт  
WHERE [дата народження] BETWEEN #01/01/1900# AND  
#29/05/2005# AND [сімейний лікар] LIKE 11;
```



Схожість Цитати Посилання Вилучений

текст Підміна символів Коментарі

Результат виконання на рисунку 3.24.

Рисунок 3.24 – Запит «Пацієнти лікаря Черногуз»

Третій запит – список листів непрацездатності (лікарняних) із статусом «Відкритий»:

```
SELECT *  
FROM лікарняний  
WHERE статус  
LIKE "Відкритий"  
ORDER BY [Дата_видачі];
```

Результат виконання на рисунку 3.25.

Рисунок 3.25 – Запит «Відкриті лікарняні»

Ще один запит –

вибірка пацієнтів, чиїм приводом запису на прийом було проходження медогляду, і запис прийому датується 20-им травня 2023-го року: SELECT [Пацієнт\*], [ID прийому], Дата, [Причина\_звернення]

```
FROM прийом
```

```
WHERE Дата
```

```
LIKE #5/20/2023#;
```

Результат виконання представлено на рисунку 3.26.

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723

40

Рисунок 3.26 – Запит «Медогляди за 20/05/2023»

3.4 Створення звітів для відображення результатів. Створення головної кнопкової форми

Звіти в програмі MS Access потрібні для того, щоб виводити на екран, або друкувати вибрану інформацію із бази даних. Звіт необов'язково будувати вручну, його автоматично можна створити на базі таблиці, форми чи звіту.

Один такий створюю на базі таблиці «Аналіз» для виведення інформації про вартість аналізів. Для цього виділяю дану таблицю, переходжу на вкладку створення й натискаю кнопку звіт. Тепер потрібно лише розставити на потрібні місця поля й за бажання змінити дизайн, бодай зміню колір тексту на більш контрастний і заберу непотрібні поля у режимі конструктора. Також



встановлюю сортування за кодом аналізу в порядку зростання. Звіт на основі таблиці представлено на рисунку 3.27.

Рисунок 3.27 – Звіт «Аналізи і ціни»

Наступним запрошується звіт на базі форми «Лікарі і їх графік роботи» - чому б не роздрукувати таку корисну схему. Для цього переходжу в конструктор макетів звітів й першим ділом у вкладці параметри сторінки міняю книжкову



Схожість Цитати Посилання  Вилучений

 Підміна символів Коментарі

Сторінка 36 з 41

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723

41

сторінку на альбомну (розворот листа А4 на дев'яносто градусів) задля того щоб помістити якомога більше інформації на



одному аркуші. Після цього задаю звіт стиль й розставляю поля на свої місця.  
Рисунок 3.28 демонструє цей звіт.

### Рисунок 3.28 – Звіт «Графік роботи лікарів»

Головна кнопкова форма – це своєрідне меню, що запускається при відкритті бази даних. На цій формі розміщують посилання на всі попередньо створені основні елементи БД (таблиці форми запити і звіти). Для того щоб створити її, потрібно спочатку ввімкнути відповідну кнопку на панелі швидкого доступу в налаштуваннях Access, як на рисунку 3.29.



Схожість Цитати Посилання  Вилучений

текст  Підміна символів Коментарі

Сторінка 37 з 41

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723

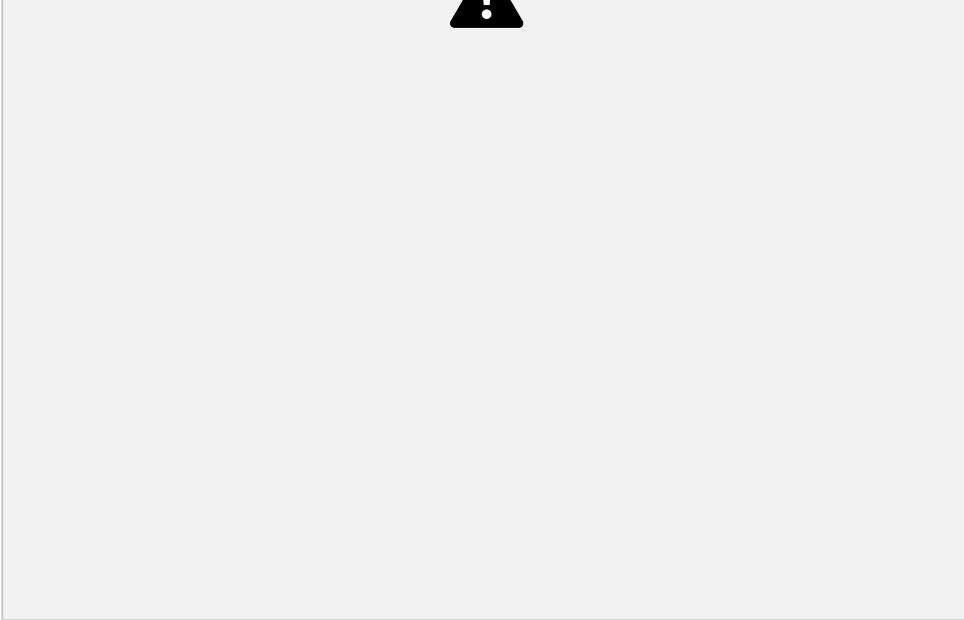


Рисунок 3.29 – Налаштування панелі швидкого доступу

Тепер натиснувши цю кнопку програма запропонує створити кнопку форму – натискаю окей і додаю на макет форми чотири сторінки. На цих чотирьох сторінках і планується розмістити посилання, тобто кнопки, для відкриття таблиць, форм, звітів і запитів, крім того потрібно передбачити кнопки навігації по формі. Нажаль на головну кнопку форму неможна помістити посилання напряму на форму чи звіт, та можна посилатись на макрос, тож доведеться створювати макроси для всіх таблиць і звітів, котрі я хочу розмістити на формі. Після того як функціонал головної форми налаштовано потрібно довести до ладу її зовнішній вигляд у режимі конструктор. Кінцевий результат вигляду форми зображено на рисунку 3.30.



Схожість Цитати Посилання  Вилучений

текст  Підміна символів Коментарі

Сторінка 38 з 41

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723



Рисунок 3.30 – Головна кнопкова форма

І останнє, що хочу зробити – налаштувати автозавантаження цієї форми при відкритті бази даних. Для цього у вкладці файл, пункт параметри, поточна база даних, вікно форма перегляду – кнопкова форма.

На даний момент база даних поліклініки має дев'ять зв'язаних між собою таблиць, що дозволяють зберігати багато необхідних для організації медичного процесу даних. Форми в якості графічного їх представлення для зручного оперування інформацією. Сконструйовано запити для швидкого знаходження потрібної інформації у базі даних. Розроблено два звіти з інформацією що може бути корисною для друку, а також є головна кнопкова форма, що автоматично відкривається при запуску програми й покращує навігацію по елементам бази даних.

## Схожість



Схожість Цитати Посилання  Вилучений

символів Коментарі

Сторінка 39 з 41

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723

## Схожість

- 1 Студентська робота ID файлу: 1003995049 Навчальний заклад: National Aviation University 2.6% [30 Джерело](#)
- 2 Студентська робота ID файлу: 3629069 Навчальний заклад: Yuriy Fedkovych Chernivtsi National University 1.89% [3 Джерело](#)
- 3 Студентська робота ID файлу: 1003902798 Навчальний заклад: National Aviation University 1.87% [9 Джерело](#)
- 4 Студентська робота ID файлу: 3557942 Навчальний заклад: Lviv Polytechnic National University 1.85% [22 Джерело](#)
- 5 Студентська робота ID файлу: 1005742463 Навчальний заклад: National Aviation University 1.77% [4 Джерело](#)
- 6 Студентська робота ID файлу: 962031 Навчальний заклад: Yuriy Fedkovych Chernivtsi National University 1.37% [67 Джерело](#)
- 7 Студентська робота ID файлу: 1000091188 Навчальний заклад: Lviv Polytechnic National University 0.98% 8 Студентська робота ID файлу: 1000808764 Навчальний заклад: V.I. Vernadsky Taurida National University 0.96% 9 Студентська робота ID файлу: 1011519581 Навчальний заклад: Lviv Polytechnic National University 0.89% 10 Студентська робота ID файлу: 1008296055 Навчальний заклад: Cherkasy State Technological University 0.83% [2 Джерело](#)
- 11 Студентська робота ID файлу: 1003921647 Навчальний заклад: National Aviation University 0.73% [2 Джерело](#)
- 12 Студентська робота ID файлу: 1000096690 Навчальний заклад: National University of Water Management and an... 0.62% [2 Джерело](#)
- 13 Студентська робота ID файлу: 1025438 Навчальний заклад: Lviv Polytechnic National University 0.47% 14 Студентська робота ID файлу: 1015034156 Навчальний заклад: Taras Shevchenko National University of Kyiv 0.42% [3 Джерело](#)
- 15 Студентська робота ID файлу: 9804713 Навчальний заклад: National University of Water Management and N... 0.41% [2 Джерело](#)
- 16 Студентська робота ID файлу: 4646309 Навчальний заклад: Yuriy Fedkovych Chernivtsi National University 0.24% 17 Студентська робота ID файлу: 1004204528 Навчальний заклад: National University of Life and Environmenta ... 0.24% 18 Студентська робота ID файлу: 5898790 Навчальний заклад: Lviv Polytechnic National University 0.21% [3 Джерело](#)
- 19 Студентська робота ID файлу: 1008360781 Навчальний заклад: National Aviation University 0.18% 20 Студентська робота ID файлу: 1013153756 Навчальний заклад: National Technical University of Ukraine "Kyi ... 0.16%

Сторінка 40 з 41

Назва документа: Ковальчук\_Роман\_ОК\_41 ID файлу: 1015062723

- 21 Студентська робота ID файлу: 1008146824 Навчальний заклад: Yuriy Fedkovych Chernivtsi National University 0.15% [2 Джерело](#)
- 22 Студентська робота ID файлу: 1011426611 Навчальний заклад: Lviv Polytechnic National University 0.15% [2 Джерело](#)



23 Студентська робота ID файлу: 1015047736 Навчальний заклад: Taras Shevchenko National University of Kyiv 0.13%

[2 Джерело](#)



24 Студентська робота ID файлу: 106398 Навчальний заклад: National University of Life and Environmental Scie... 0.13% 25 Студентська

[2 Джерело](#)

робота ID файлу: 1014382396 Навчальний заклад: National Aviation University  0.13% 26 Студентська робота ID файлу: 1011514513

Навчальний заклад: National Technical University of Ukraine "Kyj  ... 0.13% 27 Студентська робота ID файлу: 1011441471

Навчальний заклад: Donetsk National Technical University  0.13%