МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ КАФЕДРА
ПРОГРАМУВАННЯ ТА МАТЕМАТИКИ

Пояснювальна записка

до дипломної роботи

бакалавр

(освітньо-кваліфікаційний рівень)

на тему «Розробка бази даних для аптек»

Виконала: студентка 4 курсу, групи ІТз-151
напряму підготовки 6.040302 „Інформатика”
_____ Шаповалова І.Ю.
(підпис)
Керівник,
доцент, д.т.н._____ Лифар В.О.
(підпис)

Рецензент,
доцент, к.т.н._____ Фесенко Т.М.
(підпис)

СЄВЄРОДОНЕЦЬК
2019 року

ЛИСТ ПОГОДЖЕННЯ І ОЦІНЮВАННЯ
дипломної роботи студентки гр. ІТз-151 Шаповалової І.Ю.

Науковий керівник
Професор, д.т.н. _____Лифар В.О.

Оцінка наукового керівника: _____

Рецензент Фесенко Т.М. доцент каф. ПМ СНУ ім. В. Даля____
ПІБ, місто роботи, посада
Оцінка рецензента: _____

Кінцева оцінка за результатами захисту:_____
_____

Голова ЕК,
Зав. Кафедри ПМ
д.т.н., доцент _____ Лифар В.О.
(підпис)

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет інформаційних технологій та електроніки

Кафедра програмування та математики

Освітньо-кваліфікаційний рівень бакалавр

Напрям підготовки 6.040302 „Інформатика”

ЗАТВЕРДЖУЮ
Завідувач кафедри ПМ,
д.т.н., доцент
_____ Лифар В.О
«___» _____ 2019 р.

З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ
Шаповалової Ірини Юріївни

1. Тема роботи: "Розробка бази даних для аптек"
керівник роботи доцент Лифар Володимир Олексійович

затверджені наказом вищого навчального закладу від “__”___2019 року №___

2. Строк подання студентом роботи «__»_____2019 р.

3. Вихідні дані до роботи: об'єктом даної роботи є розробка бази даних для аптек.

3.1 Літературні джерела:

Атре, Ш., Структурний підхід до організації баз даних. / [Текст] : Атре, Ш. - М.: Фінанси і статистика, 1983. - 320 с.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) .

4.1 Вступ

4.2 Аналіз баз даних та предметної галузі (огляд літератури), з висвітленням наступних питань: Тенденції та перспективи автоматизованої інформаційної системи обліку товару аптек. Аналіз ринку програмних продуктів.

4.3 Основна частина, в якої висвітлити: Інформаційна модель об’єкту. Опис процесу розробки комп'ютерної програми.

4.4 Висновки

4.5 Перелік використаних джерел

5. Перелік графічного матеріалу немає.

6. Дата видачі завдання 1 лютого 2018 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Одержання завдання на виконання роботи	01.02.19	
2	Укладання і погодження з керівником плану і етапів виконання роботи	20.02.19	
3	Узагальнення даних літературних джерел, укладання розділу «Аналіз предметної галузі»	1.03.19	
4	Аналіз шляхів виконання завдання. Вибір і погодження з керівником оптимального шляху	11.03.19	
5	Проектування інфологічної моделі задачі що реалізується.	01.04.19	
6	Укладання та тестування програмного продукту	19.04.19	
7	Укладання, оформлення та погодження пояснювальної записки з керівником	10.05.19	
8	Укладання, оформлення та погодження з консультантом розділу «Охорона праці»	15.05.19	
9	Здача готової пояснювальної записки на кафедру		
10	Укладання доповіді і презентації		

Студент _____
(підпис)

Керівник роботи _____
(підпис)

РЕФЕРАТ

Текст – 50, малюнків – 16, таблиць – 8, літературних джерел – 15

Метою роботи - є розробка бази даних для аптек.

За допомогою Microsoft Visual Studio на базі середовища Microsoft SQL Server, була розроблена комп'ютерна програма, яка дозволяє поліпшити роботу аптек з обліку і реалізації товару .

Комп'ютерна програма виконує наступні основні функції:

- інформація про наявність ліків в аптеці;

- редагувати інформацію про наявність ліків в аптеці;

- загальну вартість всіх ліків у даній аптеці;

- знаходить за запитом найдешевші ліки.

БАЗА ДАНИХ, АПТЕКА, Microsoft SQL Server

ЗМІСТ

Вступ 7
Розділ 1 Аналітичний огляд 8
1.1 База даних. Основні поняття та взначення 9
1.1.1 Типи баз даних 11
1.1.2 Система управління базами даних та її основні функції 13
1.2 Загальна схема процесу розробки БД 15
1.3 Стислий огляд сучасних типів СКБД та критерії вибору СКБД для реалізації проекту 17
1.4 Аналіз предметної області "Аптека" 18
Розділ 2 Реалізація бази даних "Аптека" 21
2.1 Проектування 21
2.2 Опис таблиць і логічної структури БД 21
2.4 Схема даних 29
2.5 Аналіз можливості застосування існуючих шаблонів 30
2.6 Реалізація за допомогою SQL Server 32
2.7 Реалізація запитів 33
2.7.1 Запити на вибірку. 34
2.7.2 Запити за параметрами. 35
2.7.3 Ітогові запити 36
2.7.4 Перехресні запити 37
2.8 Звіти 39
2.9 Розробка програмних модулів та форм 43
2.9.1 Розробка програмнх модлів 43
2.9.2 Створення форм 48
2.10 Інструкція користувача 48
Список використаних джерел інформації 50

Вступ

Однією з властивістю більшості сучасних інформаційних систем є використання баз данних. Бази даних, по своїй суті є тим, навколо чого і будується інформаційна система кожнго підприємства. Тому теорії створенню та використання баз даних приділяється велика увага протягом всього періоду функціонування інформаційних систем. Досить тривалий час основним типом були реляційні бази даних, вони на сьогодні вже вважаються класичними. Але розвиток інформаційних систем поставив перед сучасними базами даних завдання, вирішення яких неможливе використання тільки в межах реляційних баз даних.

Сьогоднішні бази даних, крім класичних завдань, мають забезпечувати зберігання величезних обсягів інформації та веику машинну обробку, швидко аналізувати дані, інтеграцію із мережею Інтернет, розмежувати доступу користувачів до зберігання інформації, та ри цьому захищати інформацію під час її передачі по мережі.На практиці і використовується чимало різноманітних баз даних, але при цьому, для більшості з них існує велика кількість спільних ознак, з погляду розробки та використання, що дає можливість вивчати сучасні бази даних і відповідно: прикладне та системне програмне забезпечення на прикладах, які вже стали класичними, незважаючи на свою сучасність. Як такі приклади вибрано загальні питання проектування, розробки та використання бази даних Microsoft SQL Server. Це пояснюється тим, що Microsoft SQL є однією із найпоширеніших і вдосконалених баз даних.

База даних має бути зручною при використанні, при цьому задовольняти всі стандартам розробки баз даних. Однією з найважливіших функцій бази даних є швидка обробка запитів від користувача та швидке видання інформації для нього.

Новостворена база даних повинна зберігати в собі усю повну інформацію про усі необхідні складові системи. При створенні бази даних, є дуже важливою частиною та стадія розробки було видання відповідей на запит для користувача. Тому в цій базі користувач має змогу отримати чіткі та, якщо це необхідно відсортовані дані.

Розділ 1 Аналітичний огляд

Однією із важливих областей застосування ПК є зберігання та переробка великих об'ємів даних у різних сферах діяльності людини: в банківській справі, транспорті, науці, торгівлі, медицині, економіці.

Існуючі сучасні інформаційні системи, окреслюються дуже великими об'ємами зберігання та обробленням даних, які мають необхідністю задовольняти різні вимоги користувачів та організації.

ІС це система, що реалізує автоматизований збір, маніпулювання і обробку даних, ПЗ і обслуговуючий персонал, та містить технічні засоби для обробки даних.

Мета ІС це оброблення інформації про об'єкти реального світу. Основа ІС це База Даних. В широкому сенсі слова БД є комплекс відомостей про певні об'єкти реального світу в будь-якій предметній області. Під предметною областю розуміють частину реального світу, що вивчається для організації керування його об'єктами й, у кінцевому рахунку, автоматизація, наприклад підприємство, вуз.

Створюючи БД користувач хоче впорядкувати інформацію за певними ознаками й швидко робити вибірку із довільним поєднанням ознак. Тож необхідно вибрати правильну модель даних. Моделлю даних називається формалізоване представлення головних категорій сприйняття реального світу, які представлені його зв'язками, властивостями, об'єктами, а також їх взаємодія.

БД є інформаційною моделлю, яка дає можливість впорядковано зберігати інформацію про групу об'єктів, які мають однакові властивості

Дані у БД зберігаються у впорядкованому вигляді.

Кожна база даних призначена, передусім, для відображення даних по якійсь конкретній предметній області. Предметна область відображає відомості про які-небудь взаємозв'язані об'єкти. Кожен об'єкт може бути охарактеризований цілком певними властивостями, які можуть мати складну структуру, ці властивості можуть відображатися кількісно, тобто у вигляді цифрового значення або якісно (характер зв'язку з іншими об'єктами цієї предметної області). Усі ці властивості зазвичай у базу даних відображаються визначенням атрибути об'єкту. Структура усіх даних що відображаються базою даних, має бути найголовнішим елементом Б.Д., оскільки вона відображає предметну область б.д. і усі логічні зв'язки між даними і усю структуру властивостей. Ця структура носить назву концептуальна схема, загальна схема або схема б. д.Усі дані входять у Б.Д., зберігаються у вигляді реальних фізичних даних про конкретні об'єкти (у

зовнішній пам'яті на малих дисках). Для ефективнішої обробки даних початкову схему розбивають на дещо дрібніших. Кожна частина охоплює деяку смислову область початкової схеми. Ці частини повинні обов'язково мати загальні поля (з однаковими іменами і даними), по яких здійснюється зв'язок між окремими підсхемами.

Предметна область є частиною реального світу, яка підлягає вивченню для організації управління та для її автоматизації. Предметна область представляється великою кількістю фрагментів, наприклад, підприємство − цехами, дирекцією, бухгалтерією. робітниками і так далі. Кожен фрагмент предметної області характеризується безліччю об'єктів і процесів, що використовують об'єкти, а також безліччю користувачів, що характеризуються різними поглядами на предметну область.

Предметна область "Аптека" - це найважливіший підрозділ лікарського забезпечення. Аптека готових лікарських пепаратів займаються продажем ліків, виготовлених на заводах різних фірм-виробників. Кожен лікарський засіб має своє призначення, весь асортимент ліків класифікований по групах: жарознижуючі, ті, що знеболюють, серцево-судинні і тому подібне. Предметну область "Аптека" можна описати у вигляді моделі даних. Модель даних відображає реальний світ в деякі зрозумілі людині концепції, повністю незалежні від параметрів середовища зберігання даних.

1.1 База даних. Основні поняття та взначення

База даних (БД) — впорядкований набір логічно взаємопов'язаних даних, що використовується спільно, та призначений для задоволення інформаційних потреб користувачів. У технічному розумінні включно й система керування БД.

Головним завданням БД є гарантоване збереження значних обсягів інформації (т.зв. записи даних) та надання доступу до неї користувачеві або ж прикладній програмі. Таким чином БД складається з двох частин : збереженої інформації та системи управління нею. З метою забезпечення ефективності доступу записи даних організовують як множину фактів (елемент даних).

Дані – це інформація, відомості, показники, необхідні для ознайомлення з ким-, чим-небудь, для характеристики когось, чогось або для прийняття певних висновків, рішень. В базах даних важливу роль відіграє інформація - Інформація — абстрактне поняття, що має різні значення залежно від контексту.

Кожна створена база даних має свою предметну область. Предметна область – це необхідний для розробки бази даних об’єкт, який має в собі дані, які будуть зберігатися в базі даних.

Модель даних — абстрактне представлення реального світу, що відображає тільки ті об'єкти, що безпосередньо стосуються програми. Це, як правило, визначає специфічну групу об'єктів, їх атрибутивне значення і відношення між ними.

Відомі два підходи до організації інформаційних масивів: файлова організація та організація у вигляді бази даних. Файлова організація передбачає спеціалізацію та збереження інформації, орієнтованої, як правило, на одну прикладну задачу, та забезпечується прикладним програмістом. Така організація дозволяє досягнути високої швидкості обробки інформації, але характеризується рядом недоліків. Характерна риса файлового підходу - вузька спеціалізація як обробних програм, так і файлів даних, що служить причиною великої надлишковості, тому що ті самі елементи даних зберігаються в різних системах. Оскільки керування здійснюється різними особами (групами осіб), відсутня можливість виявити порушення суперечливості збереженої інформації. Розроблені файли для спеціалізованих прикладних програм не можна використовувати для задоволення запитів користувачів, які перекривають дві і більше області. Крім того, файлова організація даних внаслідок відмінностей структури записів і форматів передання даних не забезпечує виконання багатьох інформаційних запитів навіть у тих випадках, коли всі необхідні елементи даних містяться в наявних файлах. Тому виникає необхідність відокремити дані від їхнього опису, визначити таку організацію збереження даних з обліком існуючих зв'язків між ними, яка б дозволила використовувати ці дані одночасно для багатьох застосувань. Вказані причини обумовили появу баз даних. База даних може бути визначена як структурна сукупність даних, що підтримуються в активному стані та відображає властивості об'єктів зовнішнього (реального) світу. В базі даних містяться не тільки дані, але й описи даних, і тому інформація про форму зберігання вже не схована в сполученні "файл-програма", вона явним чином декларується в базі.

База даних орієнтована на інтегровані запити, а не на одну програму, яку випадку файлового підходу, і використовується для інформаційних потреб багатьох користувачів. В зв'язку з цим бази даних дозволяють в значній мірі скоротити надлишковість інформації. Перехід від структури БД до потрібної структури в програмі користувача відбувається автоматично за допомогою систем управління базами даних (СУБД).

Системи управління базами даних – це програмні засоби, за допомогою яких можна створювати бази даних, заповнювати їх та працювати з ними. У світі існує багато різноманітних систем управління базами даних. Багато з них насправді є не закінченими продуктами, а спеціалізованими мовами програмування, за допомогою яких кожний, хто вивчить дану мову, може сам створювати такі структури, які йому потрібні, і вводити в них необхідні елементи управління. До таких мов відносяться Clipper, Paradox, FoxPro та інші

1.1.1 Типи баз даних

Групу пов'язаних між собою елементів даних називають зазвичай записом. Є три основні типи організації даних та зв'язків між ними: ієрархічний, мережевий та реляційний.

Ієрархічній базі даних характерне впорядкування елементів в записі, один елемент є головним, а інші - підлеглими. Дані у записі впорядковані в певну послідовність, як сходинки, і пошук даних може здійснюватися лише послідовним «спуском» зі сходинки на сходинку. Пошук елементу даних в такій системі є трудомістким, через необхідність послідовно проходити кілька попередніх ієрархічних рівнів. Ієрархічну базу даних утворюють каталоги файлів, збережені на диску, дерево каталогів, доступне для перегляду в Norton Commander, - наочна демонстрація структури такої БД та пошуку в ній потрібного елементу. Такою базою даних є родове генеалогічне дерево.

Мережева БД відрізняється більшою гнучкістю, оскільки в ній існує можливість встановлювати додатково до вертикальних ієрархічних зв'язків горизонтальні зв'язки. Це полегшує процес пошуку необхідних елементів даних, так як вже не вимагає обов'язкового проходження всіх попередніх щаблів.

Найбільш поширеним способом організації даних є третій, до якого можна звести, як ієрархічний, так і мережевий - реляційний. У реляційній БД під записом розуміється рядок прямокутної таблиці. Елементи записів утворюють стовпці цієї таблиці (поля). Всі елементи в стовпці мають однаковий тип, а кожен стовпець - містить ім'я, що не повторюється. В таблиці відсутні однакові рядки. Перевага таких БД - наочність і зрозумілість організації даних, швидкість пошуку потрібної інформації. Прикладом реляційної БД служить таблиця на сторінці класного журналу, в якій записом є рядок з даними про конкретного учня, а імена полів (стовпців) вказують, які дані про кожного учня повинні бути записані в елементах таблиці.

Сукупність БД та програми СУБД утворює інформаційно-пошукову систему, звану банком даних.

Він являє собою систему спеціально організованих даних - технічних, мовних, програмних, організаційно-методичних засобів, баз даних, що застосовуються для забезпечення колективного багатоцільового використання даних і централізованого накопичення.

Головні риси банку даних:

1 . БД утворюють групою користувачів для вирішення кількох пов'язаних завдань.

2 . Банк даних має спеціальні засоби, які покращують користувачам роботу із даними .

Вимоги, які пред'являються до банку даних:

- малий час на навчання, дружелюбність інтерфейсів;
- адекватне відображення предметної області;
- забезпечення для різних користувачів розмежування та секретності доступу до даних;
- висока ефективність доступу до даних, можливість взаємодії різних категорій користувачів;
- надійність захисту та збереження даних.

За технологією обробки даних БД розділяють на розподілені і централізовані. Централізована БД зберігається у пам'яті обчислювальної системи, що є компонентом мережі ПК, то можливий розподілений доступ до цієї БД. Цей варіант використання бази даних часто застосовують в локальних мережах ПК.

Розподілена база даних включає в себе декількох перетинаючих чи дублюючих один одного частин, що зберігаються у різних ПК мережі. Робота із цією БД здійснюється за допомогою СУРБД.

За способом доступу до даних БД поділяються на БД із локальним й віддаленим доступами. Системи централізованих БД із віддаленим доступом передбачають різні архітектури таких систем: файл-сервер, клієнт-сервер.

Файл-сервер

Архітектура систем баз даних із мережевим доступом передбачає виділення однієї із машин мережі, як центрального сервера файлів. На такій машині зберігається спільно використовувана централізована БД. Всі інші машини мережі реалізують функції робочих станцій, за рахунок яких підтримується доступ користувальницької системи до централізованої БД. Файли БД відповідно до користувацьких запитів передаються на робочі станції, де в основному й проводиться обробка. При великій інтенсивності доступу до одних й тих же даних продуктивність такої інформаційної системи знижується. Користувачі можуть створювати також на робочих станціях локальні бази даних, що використовуються ними монопольно.

Клієнт-сервер

На відміну від попередньої системи, центральна машина, окрім збереження централізованої БД, повинна забезпечувати виконання основного обсягу

обробки даних. Запит на використання даних, який видається клієнтом, призводить до пошуку і вилученню даних на сервері.

Витягнуті дані транспортуються по мережі від сервера до клієнта. Специфіка архітектури клієнт-сервер це використання мови SQL.

1.1.2 Система управління базами даних та її основні функції

СУБД є сукупністю програмних і мовних засобів, що призначені для введення, створення та сумісного використання БД багатьма користувачами.

СУБД виконують такі функції:

- управління зберіганням даних;

Ця функція надає користувачам можливості виконання таких операцій з даними, як збереження, оновлення і вилучення інформації. А також забезпечення потрібної структури зовнішньої пам'яті, як для службових цілей, так і для збереження даних, що належать до бази даних.

- керування буферами оперативної пам'яті;

Система управління БД співпрацюють із базами даних великого обсягу . Коли відбувається звернення до якогось елементу даних, здійснюється обмін із зовнішньою пам'яттю, уся система працює із швидкістю пристрою зовнішньої пам'яті. Буферизація є головним способом збільшення швидкості. Розвинені системи управління БД підтримують власний набір буферів із власною заміною буферів . При керуванні буферами потрібно застосовувати та розробляти узгоджені алгоритми журналізації, синхронізації та буферизації. Але є окремий напрямок СУБД , котрий орієнтований на постійну присутність всієї БД. Даний напрямок припускає, що в майбутньому пам'ять комп'ютера зможе бути настільки велика , що дасть можливість не турбуватися про буферизацію. Але поки, що дані роботи перебувають на стадії досліджень.

- керування словником даних;

Система управління БД використовує спеціальний системний каталог, що зветься словником даних, для пошуку потрібних структур даних та їх відносин, який допомагає уникнути кодування складних взаємозв'язків в кожній програмі, тому що будь-які програми отримують доступ до даних за рахунок системи управління БД.

Словником даних називається сховище інформації, що описує дані у БД. Передбачається, що каталог доступний як користувачам, так і функцій СУБД.

У словнику даних міститься така інформація: імена зв'язків; накладаються на дані обмеження підтримки цілісності;розміри, імена, типи елементів даних; імена користувачів, яким надано право доступу до даних; зовнішня, концептуальна і внутрішня схеми і відображення між ними; статистичні дані.

Система управління БД забезпечує абстракцію даних та завдяки цьому усуває в системі залежність за даними та структурну залежність.

- керування транзакціями;

Транзакцією називають послідовність операцій над базою даних, які розглядає система управління БД, як одне ціле. Транзакція чи успішно виконується, система управління БД фіксує зміни бази даних, зроблені нею, у зовнішній пам'яті, чи ні одне із даних змін зовсім не відбивається у стані БД. Транзакції потрібні для одержання логічної цілісності БД.

Простими транзакціями служать оновлення, додавання чи видалення у БД інформації про будь-який об'єкт. Складна транзакція створюється в тому випадку, коли в БД необхідно здійснити відразу кілька змінень. Викликається ініціалізація транзакції прикладною програмою чи окремим користувачем.

Поняття транзакції умова для однокористувацьких СУБД , але набагато істотніше для багатокористувацьких систем управління БД. Всі транзакції розпочинаються при цілісному стані БД та покидають даний стан цілісним після завершення , це робить зручним застосування поняття транзакція як одиниця активності користувача по відношенню до БД.

- журналізація;

Однією із головних вимог до систем управління БД є надійне збереження інформації у зовнішній пам'яті. Надійність збереження це коли система управління БД має можливість відновлення останнього узгодженого стану БД після програмного чи апаратного збою.

Зазвичай розглядають два типи апаратних збоїв: жорсткі збої, які характеризують втрату даних на носіях зовнішньої пам'яті та м'які збої це раптова зупинка роботи комп'ютера.

Підтримання надійного збереження даних у базі даних потребує надмірності збереження даних, але особливо надійно має зберігатися частина, яка застосовується для відновлення.

Найпоширенішим методом підтримання надмірної інформації це ведення журналу змінення бази даних.

Журнал є особливою частиною бази даних, яка недоступна користувачам системи управління БД та підтримується особливо ретельно, в неї надходять записи про зміни головної частини БД.

- керування безпекою;

Система управління БД утворює систему безпеки, що надає захист користувачам і конфіденційність інформації всередині бази даних.

Правила безпеки визначає, які операції над даними доступні користувачеві, а також до яких елементів даних користувач може отримати доступ,

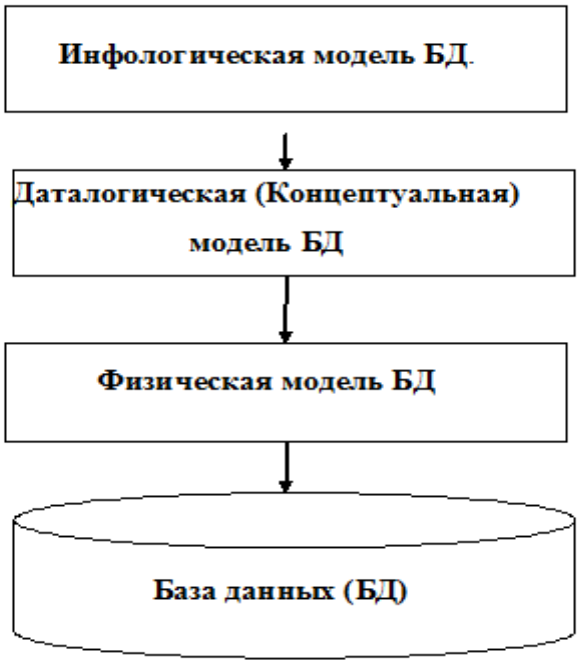
В багатокористувацьких системах ця функція дозволяє одночасно кільком користувачам отримати доступ до даних.

- підтримання мов БД.

Для роботи із БД застосовують спеціальні мови, що називають мовами БД. Сучасні системи управління БД підтримують єдину інтегровану мову, яка має всі потрібні засоби для роботи із базою даних та забезпечує базовий інтерфейс користувача із БД. У даний момент стандартною мовою є мова SQL. Вона містить два головні компоненти: мову маніпулювання даними і мову опису даних. SQL дає можливість маніпулювати даними та визначити схему реляційної БД.

1.2 Загальна схема процесу розробки БД

База даних (БД) - сукупність взаємопов'язаних і організованих за певними правилами даних, яке відображає стан об'єктів якоїсь конкретної предметної області і відносини між цими об'єктами Проектування БД здійснюється на основі трирівневої архітектури (мал.1).



малюнок 1.

1. Інфологічна модель БД- це модель, яка описує дані предметної області з використанням природної мови. Вихідними даними можуть бути стандартні довідки. Бланки. Документи. Інформаціоннологіческая модель-це модель, в якій визначена логіка відносин, також дана модель людиноорієнтованого.

Проектування інфологічної моделі БД

Проектування інфологічної моделі є основним завданням при створенні БД. Мета інфологічної модели- забезпечення найбільш природних для людини способів збору і представлення тієї або іншої інформації, яку передбачається зберігати в створюваній базі. Тому інфологічну модель даних намагаються будувати по аналогії з природною мовою (останній не може бути використаний в чистому вигляді із-за складності комп'ютерної обробки текстів і неоднозначності будь-якої природної мови). Основними конструктивними елементами інфологічних моделей є сутності, зв'язки між

ними і їх властивості. Тому необхідно чітко поставити мету цієї БД, а також встановити, яку інформацію отримує користувач в результаті роботи з програмою.

2. Даталогічна (Концептуальна) модель БД- це логічна організація даних і їх взаємозв'язок. Структурування даних виконується відповідно до обраної СУБД, яка будується на основі обраної моделі представлення даних: ієрархічної, мережевої або реляційної. На даному етапі проектувальник створює структуру даних і організовує зв'язок між об'єктами.

Проектування даталогічної моделі

Проектування даталогической модели- важливий етап в проектуванні БД. На цьому етапі важливо правильно виділити сутності і описати їх атрибути. Помилка на цьому етапі може обернутися розробникові значними втратами часу і сил надалі. Датологическая (концептуальна) модель- модель описує логіку організації даних. Датологическое проектування полягає в проектуванні логічної структури БД. Таким чином, головна відмінність даталогической моделі від інфологічної полягає в тому, що інфологічна модель зберігає в собі усю інформацію про предметну область, необхідну і достатню для проектування бази даних, але вона не прив'язана до певної СУБД.

Даталогічне проектування зводиться до наступних етапів:

1. Визначення таблиць.
2. Визначення полів таблиць.
3. Визначення типів даних відповідно до вибраної СУБД.
4. Визначення довжини кожного поля таблиць.
5. Визначення обов'язковості кожного поля.
6. Визначення індексації кожного поля.

3. Фізична модель БД- це структура БД, реалізована в середовищі обраної СУБД.

Також при створенні БД важливо враховувати наступні параметри: Цілісність БД- правильність даних в будь-який момент часу Надмірність БД- дублювання даних в декількох таблицях (надмірність тягне за собою суперечливість даних, її намагаються виключити введенням кодових полів).

Мета проектування БД- скорочення надлишкових даних. Ця мета досягається за допомогою теорії нормалізації баз даних. Нормалізація- це розбиття таблиць на два або більше, що володіють кращими властивостями при внесенні або зміні даних. В результаті виходить БД, в якій кожен факт з'являється лише в одному місці, що виключає надмірність інформації.

Проектування фізичної моделі

Фізична модель БД визначає спосіб розміщення даних на носіях (облаштуваннях зовнішньої пам'яті), а також спосіб і засоби організації ефективного доступу до них. Оскільки СУБД функціонує в складі і під управлінням операційної системи, то організація зберігання даних і доступу до них залежить від принципів і методів управління даними операційної системи. На відміну від ранніх СУБД, багато сучасних систем не надають розробникові якого-небудь вибору на цій стадії. Реально до питань проектування фізичної моделі можна віднести: вибір схеми розміщення даних (розділення по файлах або тип RAID- масиву); визначення числа і типу індексів (наприклад, кластеризований або некластеризований у разі MS SQL Server).

Спосіб зберігання БД визначається механізмами СУБД автоматично за умовчанням на основі специфікацій концептуальної схеми БД, і внутрішня схема в явному виді в таких системах не використовується. Зовнішні схеми БД зазвичай конструюються на стадії розробки додатків.

1.3 Стислий огляд сучасних типів СКБД та критерії вибору СКБД для реалізації проекту

Системи управління базами даних – це програмні засоби, за допомогою яких можна створювати бази даних, заповнювати їх та працювати з ними. У світі існує багато різноманітних систем управління базами даних. Багато з них насправді є не закінченими продуктами, а спеціалізованими мовами програмування, за допомогою яких кожний, хто вивчить дану мову, може сам створювати такі структури, які йому потрібні, і вводити в них необхідні елементи управління. До таких мов відносяться Clipper, Paradox, FoxPro та інші.

Необхідність програмувати завжди утримувала устаткування баз даних в малому бізнесі. Великі підприємства могли дозволити собі зробити наказ на програмування спеціальної системи „під себе”. Малим підприємствам

звичайно не по силам було не тільки вирішити, але й правильно сформулювати цю задачу своїми силами. Для цього треба володіти основами програмування на мові Basic.

EMS SQL Manager для MySQL Freeware це найпотужніший інструмент для адміністрування баз даних MySQL та розвитку. Вона працює з будь-якими версіями MySQL від 3,23 до новітніх і підтримує всі новітні функції MySQL, включаючи тригери, представлення, збережені процедури і функції, зовнішні ключі InnoDB, Unicode даних і так далі. SQL Manager для MySQL дозволяє створювати і редагувати всі об'єкти баз даних MySQL, проектування баз даних MySQL, запустіть SQL скрипти, керувати користувачами і їх привілеями, а також безліч інших корисних інструментів для ефективного адміністрування MySQL. SQL Manager для MySQL має стані арт-графічний користувальницький інтерфейс з добре описано майстер системи, настільки ясно у використанні, що навіть новачкові не слід плутати з ним.

Основні характеристики

* Підтримка даних UTF8
* Швидка навігація і управління базами
* Просте управління всіма об'єктами MySQL
* Потужні інструменти управління даними
* Ефективне управління параметрами безпеки
* Відмінні інструменти для побудови запитів
* Простота у використанні майстра для виконання завдань MySQL послуги
* Підключення до MySQL Server через HTTP

1.4 Аналіз предметної області "Аптека"

З розвитком інформаційного забезпечення систем автоматизованої обробки інформації, прагненням забезпечити виконання нових режимів обробки даних у реальному часі і з мультидоступом до схованих даних позначилась нова тенденція до складення інформаційного забезпечення розподілених баз даних. В умовах використання таких баз створюються комплексні масиви нелінійної структури, які мають усі дані про ту чи іншу предметну область або про керований об'єкт як постійного, так і перемінного характеру.

Основне завдання організації фармацевтичної допомоги - забезпечення населення, лікувально-профілактичних закладів та інших закладів охорони здоров’я лікарськими засобами і ви-робами медичного призначення.

Заклади охорони здоров'я, які можуть надавати послуги з реалізації лікарських засобів і виробів медичного призначен-ня, належать до аптечних закладів.

Згідно із затвердженим пе-реліком це:

- аптека;
- аптечна база (склад);
- база (склад) медичної техніки;
- база спеціального медичного постачання (центральна, рес-публіканська, обласна);
- контрольно-аналітична лабораторія;
- лабораторія з аналізу якості лікарських засобів;
- магазин (медичної техніки, медичної оптики).

Аптека - заклад охорони здоров'я, який функціонує з до-зволу і під контролем державних органів, основним завданням якого є забезпечення населення та закладів охорони здоров'я, підприємств, установ та організацій лікарськими засобами і ви-робами медичного призначення шляхом здійснення роздріб-ної торгівлі.

Роздрібна торгівля лікарськими засобами - діяльність із прид ання, зберігання та продажу готових лікарських засобів через аптеку та її структурні підрозділи (зокрема й ліків, ви-готовлених в умовах аптеки) безпосередньо громадянам для їх особистого споживання, закладам охорони здоров'я (крім аптечних закладів), а також підприємствам, установам, орга-нізаціям без права їх подальшого перепродажу.

Реалізація товарів аптечного асортименту може здійсню-ватись через відокремлені структурні підрозділи аптеки - ап-течні пункти та аптечні кіоски.

Аптечний пункт - структурний підрозділ аптеки, який створюється при лікувально-профілактичних закладах для тор-гівлі готовими лікарськими засобами.

Аптечний кіоск - структурний підрозділ аптеки, який ство-рюється на підприємствах, в установах та організаціях для роз-дрібної торгівлі готовими лікарськими засобами, що відпус-каються без рецепта лікаря.

Наказами Міністерства охорони здоров'я України затвер-джується перелік груп товарів (крім лікарських засобів), які мають право придбавати та продавати аптеки та їх структурні підрозділи.

До них належать:

- вироби медичного призначення;
- природні та штучні мінеральні води;
- засоби особистої гігієни (засоби з догляду за ротовою по-рожниною, мило туалетне, шампуні, засоби для гоління тощо);
- лікувальні косметичні засоби (креми, шампуні, солі, лось-йони, еліксири тощо);

- дитяче, дієтичне, спортивне, лікувальне харчування та інші спеціальні харчові продукти;
- дезінфекційні засоби;
- репеленти.*

Аптека має бути розміщена в окремому ізольованому при-міщенні, розташованому на перших поверхах будинків. Струк-турні підрозділи аптеки (аптечні пункти, аптечні кіоски) повинні займати відокремлені приміщення тільки у капітальних спорудах. На фасаді аптеки (структурного підрозділу) має бути вивіска із за-значенням назви господарюючого суб'єкта або власника. Біля входу до цього приміщення на видному місці розміщується інфор-мація про режим роботи, адреса чергової та найближчої аптеки.

Основні ***функції*** аптек:

- приймання рецептів від населення та вимог від лікувально-профілактичних закладів і відпуск за ними лікарських засобів;
- виготовлення лікарських засобів, контроль якості й оформ-лення їх до відпуску;
- продаж населенню лікарських засобів, дозволених до від-пуску без рецептів, а також предметів санітарії, гігієни, догля-ду за хворими, лікувальних мінеральних вод, засобів для пере-в'язування та інших товарів аптечного асортименту;
- надання першої медичної допомоги;
- санітарно-просвітницька робота серед населення;
- інформаційна робота серед лікарів та інших медичних пра-цівників про наявність лікарських засобів, нові препарати, їх застосування, протипоказання, зберігання тощо;
- фінансово-господарська діяльність.

Аптека складається з декількох приміщень: торговий зал, склад, бухгалтерія, включаючи кабінет директора. Аптека взаємодіє з постачальниками в особі менеджера, що здійснюють замовлення на постачання ліків, побутової хімії, медичного устаткування і косметичних засобів. Кожен препарат має свій склад, найменування, артикул і показання до застосування. Так само в аптеці є свій штат співробітників : фармацевти, бухгалтера, вантажники-водії, менеджери. Клієнт, приходячи в аптеку, взаємодіє з фармацевтом, який, у свою чергу, здійснює продаж.

Операції з грошовими коштами і відбиття чека здійснює так само фармацевт. При купівлі товару, клієнт отримує чек з вказівкою товару і його вартості. За відсутності товару на складі покупець може оформити бланк-заказа, де обмовляються терміни постачання, і вноситься передоплата за вибраний товар. У магазині ведеться документація: касова книга (сума проданого товару, яка віддається бухгалтерові, а потім передається в

податкову для звітності), прибуткова накладна, ведеться облік доходів і витрат аптеки.

Аптеки займаються продажем ліків, виготовлених на заводах фірм-виробників. Кожний лікарський засіб має своє призначення, весь асортимент ліків класифікований за групами та лікарських форм.

У базі даних необхідно зберігати різноманітну інформацію про замовлення, препаратах і постачальниках даних препаратів, щоб можна було оперативно визначати інформацію про приналежність того чи іншого препарату до певного виробнику, переглядати замовлення, виконані певним співробітником, вести облік прибутку від реалізації препаратів або збитків при списання прострочених препаратів.

Можуть існувати такі обмеження при роботі з базою даних:

1. один виробник може виробляти кілька препаратів;
2. один і той же препарат може бути виготовлений різними виробниками;
3. на складі можуть зберігатися однакові препарати від різних постачальників, мати різні ціни, різну лікарську форму і різний термін придатності;
4. препарати, термін придатності яких минув, не можна замовити. Також їх вартість враховується в розрахунку прибутку як збиток.

Розділ 2 Реалізація бази даних для Аптек

2.1 Проектування

БД аптеки повинна включати вісім таблиць, призначених для ефективності роботи підприємства і автоматизації обробки результатів його діяльності :

- Препарати;
- Призначення;
- Надходження;
- Постачальники;
- Клієнти;
- Співробітники;
- Лікування.

Таблиці пов'язані між собою, що забезпечує цілісність даних. База даних призначена, в першу чергу, для співробітників - повний облік товарів аптеки, контроль роботи аптеки в цілому; для клієнтів - зручний пошук потрібного препарату; для постачальників - контакти кожного із співробітників.

2.2 Опис таблиць і логічної структури БД

Логічне проектування виконується для певної моделі даних. Для реляційної моделі даних логічне проектування полягає у створенні реляційної схеми, визначенні числа і структури таблиць, формуванні запитів до БД, визначенні типів звітних документів, розробці алгоритмів обробки інформації, створенні форм для вводу і редагування даних в БД і рішенні цілого ряду інших задач. Концептуальні моделі за певними правилами перетворюються в логічні моделі даних. Коректність логічних моделей перевіряється за допомогою правил нормалізації, які дозволяють переконатися в структурній узгодженості, логічній цілісності і мінімальній збитковості прийнятої моделі даних.

Логічна структура БД в кінцевому вигляді повинна відповідати всім стандартам, а саме:

- не містити зв'язків «багато-до-багатьох»;
- не мати складних зв'язків;
- не повинно бути множинних атрибутів.

Всі зв'язки типу «багато-до-багатьох», видаляються за допомогою виділення проміжної сутності і замінюються на два зв'язки типу «один-до-одного» до новоствореної сутності.

Проаналізувавши структуру бази даних, було зроблено висновки про її відповідність вимогам. База даних не містить зв'язків «багато-до-багатьох», всі атрибути сутностей є простими а складні зв'язки відсутні. Першим кроком спрощення концептуальної моделі є попередні перетворення з метою усунення зв'язків, які є несумісними з реляційною моделлю.

Створюємо ER-діаграму проектованої системи, яка буде відповідати наступним правилам:

- кожна сутність, кожний атрибут і кожний зв'язок повинні мати ім'я (зв'язок супертипу або асоціативний зв'язок може не мати імені);
- ім'я сутності повинне бути унікальне в рамках моделі даних;
- ім'я атрибуту повинне бути унікальне в рамках сутності;
- ім'я зв'язку повинне бути унікальне, якщо для нього генерується таблиця БД;
- кожний атрибут повинен мати визначення типу даних;
- сутність в необов'язковому зв'язку повинна мати ключовий атрибут. Те ж саме відноситься до сильної сутності в слабкому зв'язку, супертипу в зв'язку «супертип-підтип» і необов'язкової сутності в обов'язковому (повному) зв'язку;
- підтип в зв'язку «супертип-підтип» не може мати ключовий атрибут;
- у асоціативному або слабкому зв'язку може бути тільки одна асоціативна (слабка) сутність;
- зв'язок не може бути одночасно обов'язковим, «супертип-підтип» або асоціативним.

СУБД, що розробляється, включає наступні підсистеми:

- Препарати;
- Призначення;

- Надходження;
- Постачальники;
- Клієнти;
- Співробітники;
- Лікування.

Підсистема Препарати включає усі медикаменти, які поступають на склад або замовляються клієнтами. Кожному препарату присвоєний унікальний код (первинний ключ). Характеристики препарату включають різні поля, у тому числі: одиницю виміру (наприклад, в ампулах по 25 мл), ціну закупівельну, ціну реалізації та ін.

Підсистема Призначення дає характеристику препаратів, визначає їх групу приналежності. Окрема група призначена для хворих з певним захворюванням якогось органу або організму в цілому. Кожній групі привласнюється унікальний код (первинний ключ). У цій підсистемі дається повний опис груп препаратів, їх дія і те, як може виглядати препарат цієї групи.

Підсистема Співробітники містить інформацію про співробітників аптеки. Полями таблиці є усі персональні дані співробітника : ПІБ, дата народження, дата найму, посада, адреса, телефон, фотографія, особисті дані. Співробітники працюють в аптеці, обслуговують клієнтів, відправляють замовлення. Кожному співробітникові привласнюється унікальний код (первинний ключ).

Підсистема Постачальники включає інформацію про постачальників аптеки. Кожному постачальникові привласнюється унікальний код (первинний ключ). Характеристики постачальника представлені полями таблиці, такими як: назва, представник, його посада, адреса, місто, країна і контактний телефон.

Підсистема Клієнти містить інформацію про клієнтів аптеки (назва, представник, контактна інформація і тому подібне). Кожному клієнтові привласнюється унікальний код (первинний код).

Підсистема Замовлення дає характеристику замовлень, які поступають в аптеку від клієнтів. Кожне замовлення обробляє співробітник аптеки. Кожному замовленню відповідає унікальний код (первинний ключ).

Підсистема Замовлено включає характеристику отриманих замовлень. Обчислюється ціна з урахуванням знижки.

Підсистема Лікування містить інформацію про хвороби, які необхідно лікувати за допомогою лікарських препаратів.

2.3 Детальний опис полів підсистем

Спершу розглянемо деякі властивості полів таблиці "Препарати".

У базі даних аптеки на кожну групу препаратів (антибіотики, антисептики і тому подібне) доводиться невелика кількість лікарських засобів, які знаходяться на складі аптеки. Одиниця виміру : або упаковка, або бульбашка. Залежно від фірми і якості доставки ціна закупівельна і ціна реалізації відрізняються один від одного.

таблиця 1"Препарати"

Назва поля	Приклад позначення	Характеристика
Код препарата	1	Співпадання в полі не допускаются; Ключове поле
Назва препарата	Арбідол	Тип даних - Текстовий Обовязковий ввод даних в поле
Код назначення	6676544A	Текстовий тип даних Обовязковий ввод даних в поле
код поставщика	123	Числовий тип даних Допускаются співпадання в полі
Одиниця вимірювання	50 упаковок по 10 таблеток	Текстовий тип даних
Ціна закупочна	145 грн	Тип даних - грошовий Условие на значение >=0
Ціна реалізації	200 грн	Тип даних - грошовий >=0

У підсистемі "Призначення" препаратів властивості полів наступні:
Таблиця Призначення препаратів. Препарати діляться на фармакологічні групи.

таблиця 2 "Призначення"

Назва поля	Приклад позначення	Характеристика
Код назначення	12	Збіги не допускаються Ключове поле
Группа	антибіотики	Обовязкове поле
Опис	речовини природного або напівсинтетич. походження, що пригнічують зростання живих клітин, найчастіше прокариотических або простих.	Текстовий тип даних Допускаються збіги

Характерні риси полів таблиці "Співробітники":

У штат аптеки входять:

Завідувач-провізор має права завідувача відділенням лікувально-профілактичної установи і контролює усю роботу аптеки. Завідувач-провізор зобов'язаний контролювати чітку роботу в аптеці по виготовленню, прийому і відпустці ліків по рецептах. Він контролює постачання медичним майном лікувально-діагностичні кабінети, лабораторії; забезпечує правильне зберігання медикаментів, медичного інструментарію; контролює витрату медикаментів у відділеннях лікувально-профілактичної установи, особливо контролюючи отруйні, наркотичні засоби. Завідувач контролює фармацевтичний порядок в аптеці, дотримання санітарно-гігієнічних правил, складає кошториси на отримання медичного майна.

Заступник завідувача-провізора підкоряється безпосередньо завідувачеві-провізорові і за відсутності останнього заміщає його. Він забезпечує контроль постійного виготовлення і відпустки ліків і іншого медичного матеріалу у відділення.Кілька разів в рік звіряє наявність і роботу (придатність) медичної апаратури і інструментарію в кабінетах відділення. Здійснює безпосередню участь в контролі виконання техніки безпеки і протипожежних заходів.

Провізор-аналітик повинен мати вищу фармацевтичну освіту. Його призначення і звільнення контролюється завідувачем-провізором аптеки. Він знаходиться під начальством завідувача-провізора і заступника провізора аптеки.

Провізор-технолог зобов'язаний реєструвати ліки і інші медичні засоби, які є в наявності або які відсутні в аптеці. Здійснює контроль і перевірку зберігання, використання медичних засобів у відділеннях, лабораторіях, кабінетах; повідомляє медичний (лікарям) персонал про наявність медичних засобів в аптеці.

Дефектар керує виготовленням в аптеці напівфабрикатів, концентратів і перефасовкой медикаментів.

Фармацевт повинен виготовляти ліки в міру необхідності для відділень і по рецептах лікарів, а також повинен правильно підготувати і оформити ліки перед їх відпусткою. Фармацевт повинен періодично підвищувати кваліфікацію.

Фасовщица займається розфасовкою і перефасовкой лікарських, дезинфікуючих засобів, аптечного і лабораторного посуду, видачею перев'язувальних засобів.

Санітар виконує різну роботу (миття і обробку посуду, прибирання приміщень та ін.).

Провізор-клініцист розробляє технологію виготовлення ліків, уніфікує її.

Бухгалтер-фармацевт контролюється завідувачкам аптекою. Бухгалтер-фармацевт забезпечує своєчасний, правильний облік усього медичного майна, предметно-кількісний облік отруйних і наркотичних засобів, етилового спирту та ін.

Інженер-технік по ремонту медичної техніки готує до експлуатації усю техніку, що поступає, здійснює технічне обслуговування і різний ремонт медичної і аптечної техніки, підвищує експлуатаційний надійність, робить профілактичний огляд апаратура, робить запис про проведений ремонт в паспорт апарат і реєструє вироблення вони в процесі експлуатація, стежить за робота (правильною) медичний і аптечний техніка. Періодично інструктує персонал по техніці безпеки.

таблиця 3 "Співробітники"

Назва поля	Приклад позначення	Характеристика
Код співробітника	12	Ключове поле Збігу не допускаються
Фамілія	Шевченко	Збіги допускаються Обов'язкове поле
Імя	Микола	Збіги допускаються Обов'язкове поле
По батькові	Олексійович	Текстовий тип даних Допускаються збіги
Должность	провизор	Текстовый тип данных
Дата народження	29.06.1956	Формат поля - короткий формат дати Умова на значення <Date ()
Дата найма	13.09.1999	Тип даних - Дата/Час
Адрес	Шосе Будівельників,17	Необовязкове поле
Місто	Київ	Текстовий тип даних Не допускаються порожні рядки
Країна	Україна	Текстовий тип даних Не допускаються порожні рядки
Дом. телефон	5645767	Необовязкове поле

Моб. телефон	0985646364	обовязкове поле Розмір поля - 30
Освіта	Фармацевтична Академія	Текстовий тип даних Допускаються порожні рядки
Зарплата	100 000,00 грн	Неіндексоване поле Тип даних - грошовий

Поля підсистеми "Постачальники" і їх характеристики:

таблиця 4 "Постачальники"

Назва поля	Приклад позначення	Характеристика
Код постачальника	3467	Ключеве поле Тип даних - числовий
Назва постаальника	"Видэн"	Размір поля - 50 Обовязкове поле
Представник	Валентинов І.А.	Обовязкове поле
Посада	завідуючий	Текстовий тип даних Допускаюється збіги
Адреса	Введенова,23	Необовязкове поле
Місто	Київ	Текстовий тип даних Не допускаются пусті строки
Країна	Україна	Необязательное поле Не допускаются пустые строки
Контактный телефон	(095) 543-8976	Используется маска (000) 000-0000

Таблиця "*Клієнти"* с характеристиками полів:

таблиця 5 "Клієнти"

Назва поля	Приклад позначення	Характеристика
Код клиента	584737	Ключове поле Длинное целое
Назва	"ВПром"	Размір поля - 50 Обовязкове поле
Предсиавник	Григорьев М.Ю.	Обязательное поле
Адрес	Подшипникова, 125	Необязательное поле
Місто	Вінниця	Текстовый тип данных Не допускаются пустые строки

Країна	Україна	Необязательное поле Не допускаются пустые строки
Телефон	(095) 677-8098	Используется маска (000) 000-0000

Характерні риси полів таблиці "Замовлення":

таблиця 6 "замовлення"

Назва поля	Приклад позначення	Характеристика
Код заказа	56752700	Ключевое поле Длинное целое
Код клієнта	6747473	Совпадает со значением поля "Код Клиента" в таблице "Клиенты".
Код сотрудника	3	Совпадает со значением поля "Код Сотрудника" в таблице "Сотрудники".
Дата размещения	23.12.2008	Обязательное поле
Дата назначения	01.02.2009	Тип данных - Дата/время
Дата исполнения	02.03.2009	Тип данных - Дата/время
Стоимость доставки	453, 00 р	Формат поля - Денежный
Получатель	Николаева О.Р.	Обязательное поле Тип данных - текстовый
Адрес получателя	42 Compyon street	Необязательное поле
Город получателя	New York	Текстовый тип данных Не допускаются пустые строки
Страна получателя	USA	Размер поля - 50 Обязательное поле

Деякі характерні властивості полів підсистеми "Замовлено":

таблиця 7 "замовлення"

Назва поля	Приклад позначення	Характеристика

Код заказа	657473	Совпадает со значением поля "Код Заказа" в таблице "Заказы".
Код препарата	17	Совпадает со значением поля "Код препарата" в таблице "Препараты".
Цена реализации	145, 00 р	Условие на значение >=0
Количество	18	Условие на значение >0
Скидка	15%	Сообщение об ошибке: "Необходимо ввести значение со знаком процентов"

У підсистемі "Лікування" властивості полів наступні.

таблиця 8 "лікування"

Название поля	Пример обозначения	Характеристика
Болезни	сердечные	Обязательное поле
Код назначения	453АР	Совпадает со значением поля "Код назначения" в таблице "Назначения препаратов".

2.4 Схема даних

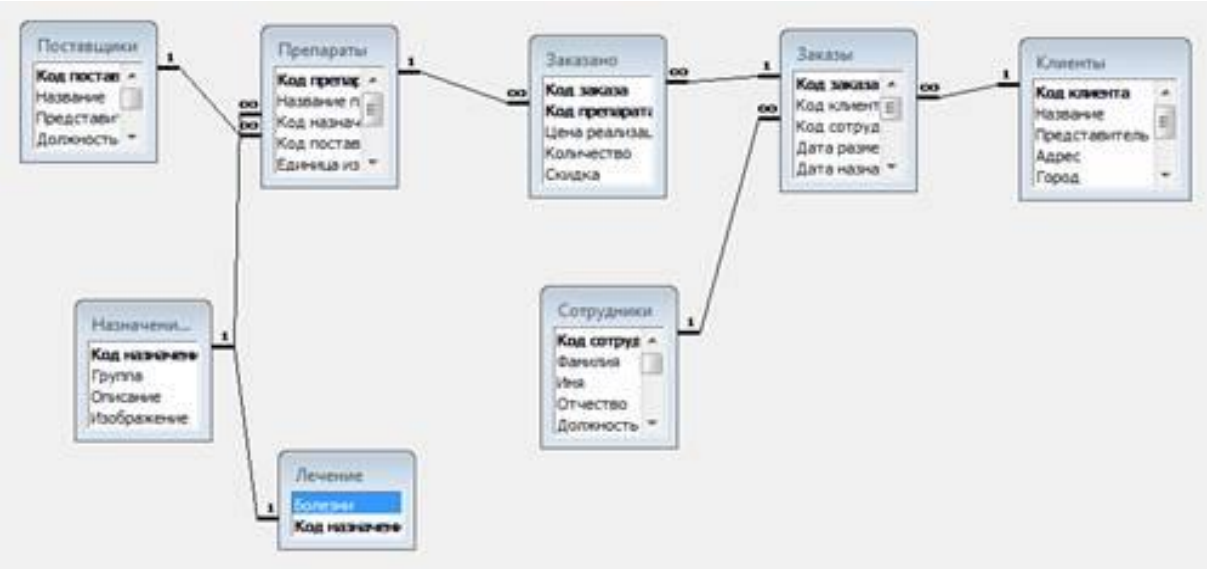
Схеми і підсхеми є діаграмами, що зображують типи елементів даних і зв'язку між ними. Існують різні способи зображення зв'язків. Зв'язки між двома елементами даних можуть бути двох типів.

Перший тип - зв'язок "один до одного", т. е. одному запису при цьому зв'язку в головній таблиці повинен відповідати один запис в підпорядкованій таблиці. Такі БД використовуються досить рідко. За допомогою таких зв'язків виділяють окремо рідко використовувану інформацію.

Другий тип - зв'язок "один до багатьом", найчастіше використовується це відношення. В даному випадку одному запису головній таблиці можуть відповідати декілька записів підпорядкованої таблиці. Розрізняють два різновиди зв'язку "один до багатьом". У першому випадку пред'являються жорсткі вимоги на обов'язкову наявність записів у вторинній таблиці. У другому випадку такі вимоги відсутні.

Третій тип - зв'язок "багато до багатьом". Багато реляційних СУБД цей зв'язок не підтримують. Для реалізації таких зв'язків таблиці пов'язані таким відношенням слід перетворити так, щоб в них були тільки зв'язки 1 :М, для цієї мети вводяться додаткові таблиці, які відображають зв'язки між таблицями, що відображаються, пов'язаними спочатку за типом М : М.

Взаємозв'язок між представленими вище підсистемами показаний на Мал..



2.5 Аналіз можливості застосування існуючих шаблонів

Шаблони проектування - це своєрідні групи вказівок спрямовані на полегшення вирішення задач об'єктно-орієнтованого програмування (в нашому випадку). Шаблон не можна вважати закінченим зразком, що не дозволяє проводити його трансляцію в код програми. Такі шаблони зазвичай служать прикладами для рішення певної задачі та ілюструють відношення між класами і об'єктами, проте не вказують як реалізувати саме відношення.

Існує 3 основних групи шаблонів:

1. шаблони, що породжують - це шаблони проектування, що абстрагують процес інстанціювання. Вони допоможуть зробити систему незалежною від способу створення, композиції та представлення її об'єктів. Шаблон, який породжує класи, використовує спадкування, щоб варіювати створюваний клас, а шаблон, що створює об'єкти, делегує інстанціювання іншому об'єктові. Ці шаблони важливі, коли система більше залежить від композиції об'єктів, ніж від спадкування класів. Таким чином, замість прямого кодування фіксованого набору поведінок, визначається невеликий набір фундаментальних, за допомогою композиції яких можна отримувати складніші;

2. структурні шаблони - шаблони проектування, у яких розглядається питання про те, як із класів та об'єктів утворюються більші за розмірами структури. Структурні шаблони рівня класу використовують спадковість для утворення композицій із інтерфейсів та реалізацій. Структурні шаблони рівня об'єкта компонують об'єкти для отримання нової функціональності. Додаткова гнучкість у цьому разі пов'язана з можливістю змінювати

композицію об'єктів під час виконання, що є неприпустимим для статичної композиції класів;

3. шаблони поведінки - шаблони проектування, що пов'язані з алгоритмами та розподілом обов'язків поміж об'єктів. Мова в них йде не тільки про самі об'єкти та класи, але й про типові способи їхньої взаємодії. Шаблони поведінки характеризують складний потік керування, котрий досить важко прослідкувати під час виконання програми. Увага акцентована не на потоці керування, а на зв'язках між об'єктами.

У даному проекті використовується технологія побудови бази даних з допомогою SQL Server.

Ключем до паттерну компонувальник є абстрактний клас, який є одночасно і примітивом, і контейнером(Component). У ньому оголошені методи, специфічні для кожного виду об'єкта (такі як Operation) і загальні для всіх складових об'єктів, наприклад операції для доступу і управління нащадками. Підкласи Leaf визначає примітивні об'єкти, які не є контейнерами. У них операція Operation реалізована відповідно до їх специфічних потреб. Оскільки у примітивних об'єктів немає нащадків, то жоден з цих підкласів не реалізує операції, пов'язані з управління нащадками (Add, Remove, GetChild). Клас Composite складається з інших примітивніших об'єктів Component. Реалізована в ньому операція Operation викликає однойменну функцію відтворення для кожного нащадка, а операції для роботи з нащадками вже не порожні. Оскільки інтерфейс класу Composite відповідає інтерфейсу Component, то до складу об'єкта Composite можуть входити і інші такі ж об'єкти.

Цей шаблон оголошує інтерфейс для компонованих об'єктів; Надає відповідну реалізацію операцій за замовчуванням, загальну для всіх класів; Оголошує єдиний інтерфейс для доступу до нащадків та управління ними; Визначає інтерфейс для доступу до батька компонента в рекурсивної структурі і при необхідності реалізує його (можливість необов'язкова).

Мета шаблону - гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонент програми. Крім того використання цього шаблону у великих системах призводить до певної впорядкованості їх структури і робить їх зрозумілішими завдяки зменшенню складності.

Зареєстровані події транслюються в різні запити, що спрямовуються компонентам Моделі або об'єктам, відповідальним за відображення даних. Відокремлення моделі від вигляду даних дозволяє незалежно використовувати різні компоненти для відображення інформації. Таким чином, якщо користувач через Контролер внесе зміни до Моделі даних, то інформація, подана одним або декількома візуальними компонентами, буде автоматично відкоригована відповідно до змін, що відбулися.

Також передбачається використовувати паттерн «Репозиторій» для доступу до даних.

Система зі складною моделлю області визначення може бути спрощена за допомогою додаткового рівня, наприклад Data Mapper, який би ізолював об'єкти від коду доступу до БД. У таких системах може бути корисним додавання ще одного шару абстракції поверх шару розподілу даних (Data Mapper), в якому б був зібраний код створення запитів. Це стає ще більш важливим, коли в області визначення безліч класів або при складних, важких запитах. У таких випадках додавання цього рівня особливо допомагає скоротити дублювання коду запитів.

Патерн «Репозиторій» - посередник між шаром області визначення і шаром розподілу даних, працюючий, як звичайна колекція об'єктів області визначення. Об'єкти-клієнти створюють опис запиту декларативно і направляють їх до об'єкта-репозиторію для обробки. Об'єкти можуть бути додані або видалені з репозиторію так, наче вони формують просту колекцію об'єктів. А код розподілу даних, прихований в об'єкті «Репозиторій», подбає про відповідні операції. Отже, паттерн «Репозиторій» інкапсулює об'єкти, представлені в сховищі даних і операції, зроблені над ними, надаючи більш об'єктно-орієнтоване уявлення реальних даних. «Репозиторій» також переслідує мету досягнення повного поділу і односторонньої залежності між рівнями області визначення і розподілу даних.

2.6 Реалізація за допомогою SQL Server

Відкриємо SQL Server Enterprise Manager, виберемо локальний сервер, з яким працюватимемо, відкриємо дерево баз даних і виберемо БД Apteka, в яку необхідно додати таблицю. Виділяємо компонент Tables і натискаємо значок New: Відкривається вікно визначення полів. Вводимо назви полів, вибираємо типи даних для кожного поля, зберігаємо таблицю Lekarstvennye sredstva. Так само поступаємо і відносно шести інших таблиць. Приступимо до заповнення таблиць. Для того, щоб заповнити таблицю, вибираємо її в списку таблиць нашої БД Аптека, в контекстному меню вибираємо команду Open Table - Return all rows. Для того, щоб створити діаграму БД Аптека, необхідно активувати компонент Diagrams - New Database Diagram. Тепер виберемо таблиці, які братимуть участь в діаграмі, натиснемо кнопку Add: Натискаємо Далі, у вікні, що відкрилося, вибираємо таблицю, яка стане основна : У вікні, що відкрилося, бачимо тільки що створену діаграму.

2.7 Реалізація запитів

Дані, що зберігаються у БД, зазвичай вимагають множинної обробки. Для цього застосовують запити, якими є спеціальним чином описані вимоги, що визначають склад вироблюваних над БД операцій по вибірці, видаленню, модифікації даних. Для підготовки запитів в різних СУБД найчастіше використовуються дві основні мови: Мова QBE (Query By Example) - мова запитів за зразком, Мова SQL (Structured Query Language) - структурована мова запитів. По можливостях маніпулювання даними в запитах вказані мови практично еквівалентні. Головна відмінність полягає в способі формування запитів : мова QBE припускає ручне або візуальне формування запиту, а SQL використовує програмування запиту. Теоретичною основою мови QBE є реляційне числення зі змінними - доменами. Мова дозволяє задавати складні запити до БД заповненням запитальної форми, що має вигляд таблиці, імена і назви полів якої співпадають з іменами і назвами полів відповідних початкових таблиць. Наочними є запитальні форми в Access. Мова SQL призначена для виконання операцій над таблицями (створення, видалення, зміна структури) і над даними таблиць (вибірка, зміна, додавання, видалення), і деяких супутніх операцій. SQL є непроцедурною мовою і не містить операторів управління, організації підпрограм, введення-виведення і тому подібне. У зв'язку з цим SQL зазвичай вбудовується в СУБД (наприклад, СУБД ACCESS, FoxPro СУБД.). У сучасних СУБД з інтерактивним інтерфейсом можна створювати запити, як було сказано, наприклад, за допомогою мови QBE. Проте застосування SQL часто дозволяє підвищити ефективність обробки даних. Так, при підготовці запиту в ACCESS можна перейти з вікна Конструктора запитів (формулювання запиту за зразком на мові QBE) у вікно з еквівалентним оператором SQL. Новий запит можна створити шляхом редагування вже наявного запиту або програмуванням нового. Мова SQL не має функцій повноцінної мови програмування, а орієнтований на доступ до даних, тому його включають до складу засобів розробки програм. В цьому випадку його називають вбудованим SQL. При побудові запиту у вікні конструктора система Access працює у фоновому режимі, записуючи еквівалентні інструкції SQL. Для перегляду програми SQL в меню Вид виберіть команду Режим SQL. Перехресний запит "К-ть ліків, що залишилися, з кол-ва що поставляються": Запит видає дані про розміри кредитів виданих в різних банках різним клієнтам. Інструкція SELECT витягає Код_клиента з таблиці "Банк _ клієнти" параметри FROM і GROUP BY є запитами на вибірку; вони вказують, які таблиці містять поля, приведені в інструкції SELECT. Операція GROUP BY робить угрупування усіх полів списку SELECT.

У рамках цієї роботи було поставлено завдання на основі об'єднаних таблиць створити наступні запити в режимі конструктора :

запити на вибірку (вибір символьних даних, числових, використання масок);

підсумкові запити;

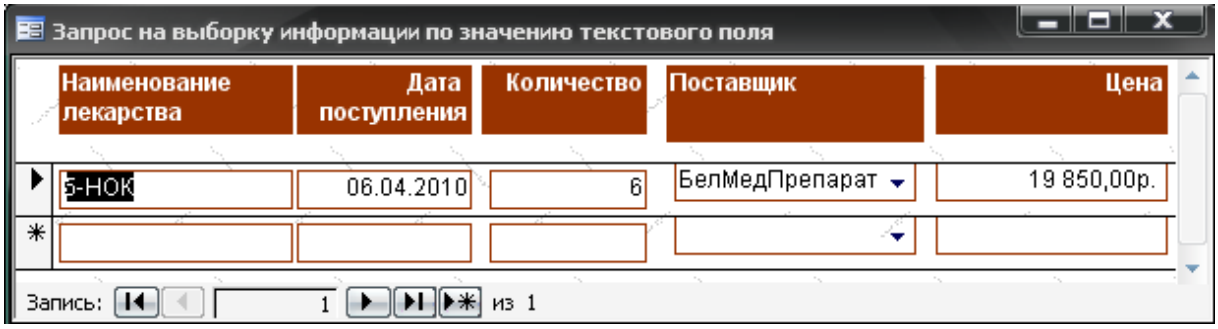
запит з параметром;

перехресні запити.

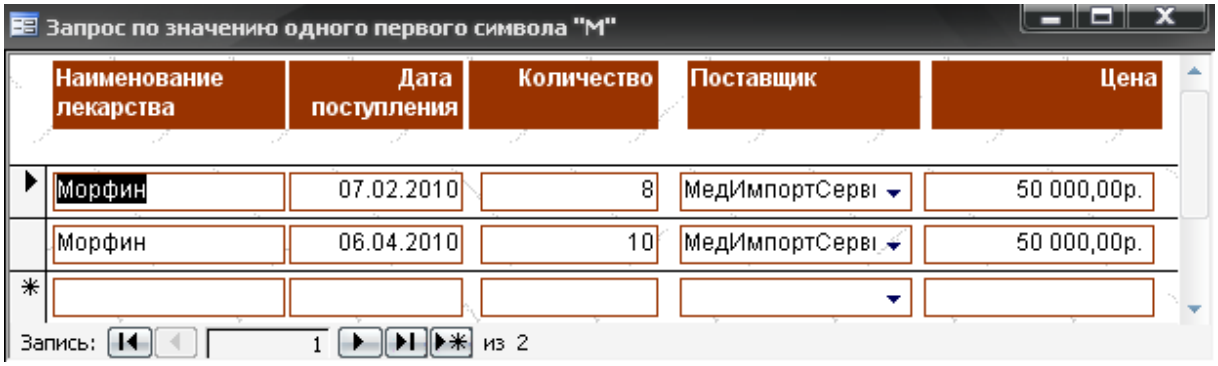
2.7.1 Запити на вибірку.

Запит на вибірку є найчастіше використовуваним типом запиту. Запити цього типу повертають дані з однієї або декількох таблиць і відображають їх у вигляді таблиці, запису в якій можна оновлювати (з деякими обмеженнями). Запити на вибірку можна також використати для угрупування записів і обчислення сум, середніх значень, підрахунку записів і знаходження інших типів підсумкових значень.

Запит на вибірку за значенням текстового поля, який відображає інформацію по приходу лікарського засобу "5-НОК". Результат запиту представлений на мал.



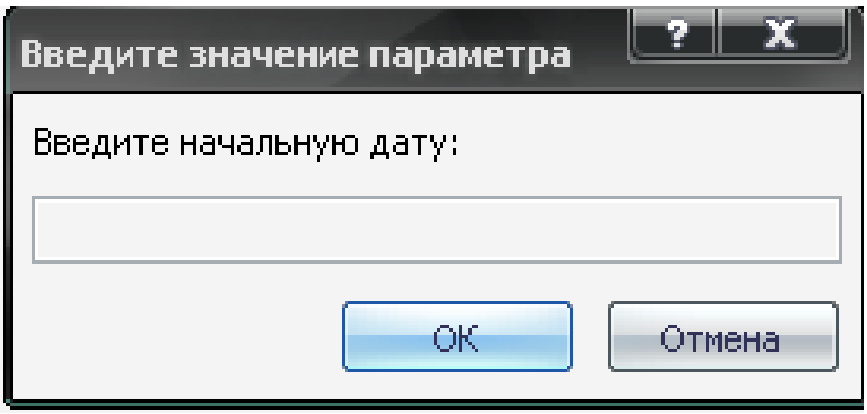
Запрос к БД по значению одного первого символа позволяет получить информацию по приходу всех лекарственных препаратов, название которых начинается с буквы «М»

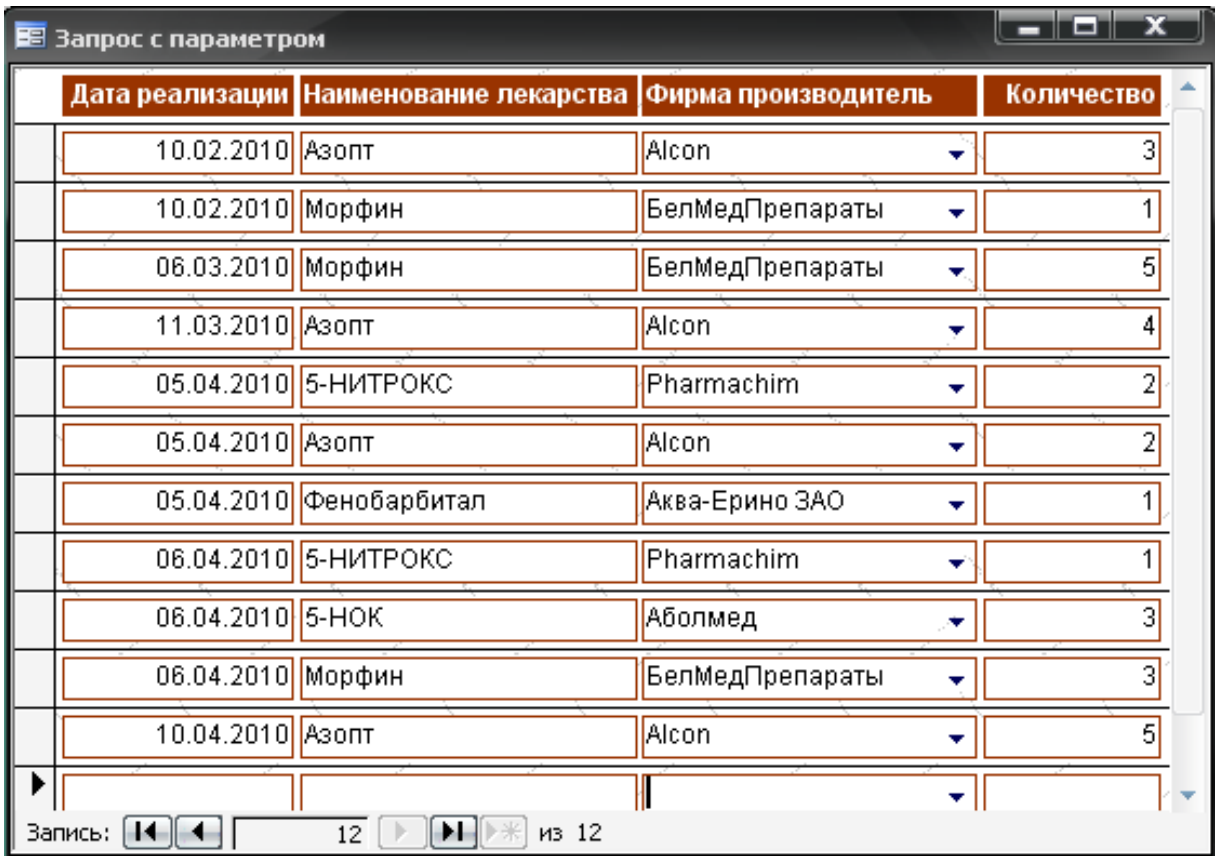


2.7.2 Запити за параметрами.

Запит з параметрами - це запит, при виконанні що відображає у власному діалоговому вікні запрошення ввести дані, напримерусловие (Умова. Обмеження, задане для відбору записів, що включаються в результуючий набір записів запиту або фільтру.) для повернення записів або значення, яке вимагається вставити в поле. Можна розробити запит, що виводить запрошення на введення декількох одиниць даних, наприклад двох дат.

Запит з параметром, організований до БД "Аптека", дозволяє отримати інформацію про приходи усіх ліків в заданий користувачем часовий інтервал. При виклику цього запиту на виконання з'являється діалогове вікно "Введіть значення параметра" (мал. 1.5), в якому послідовно необхідно ввести початкову, а потім кінцеву дату користувача тимчасового інтервалу, що цікавить. Результат роботи запиту з параметром, часовий інтервал для якого заданий 01.01.2010-01.05.2010.





2.7.3 Ітогові запити

Запит, що демонструє роботу операції групування даних з використанням функції sum () і використання будівника виразів . Цей запит дозволяє визначити суми від реалізації кожного лікарського засобу по датах реалізації. Для відображення вказаної суми було створено нове поле із заголовком "Сума", обчислюване як вираження "[Кількість реалізованого товару]·[Відпускна ціна за штуку]". У цьому запиті використано сортування даних (за збільшенням) по назві.

Запрос создать в запросе новое поле с исп-м построителя выр-ний : запрос на выборку

Дата реализации	Наименование	Отпускная цен	Количество реализ	Сумма
05.04.2010	5-НИТРОКС	25 000,00р.	2	50000
06.04.2010	5-НИТРОКС	25 000,00р.	1	25000
06.04.2010	5-НОК	23 500,00р.	3	70500
10.02.2010	Азопт	35 600,00р.	3	106800
11.03.2010	Азопт	35 900,00р.	4	143600
05.04.2010	Азопт	40 000,00р.	2	80000
10.04.2010	Азопт	40 000,00р.	5	200000
10.02.2010	Морфин	52 000,00р.	1	52000
06.03.2010	Морфин	52 000,00р.	5	260000
06.04.2010	Морфин	55 000,00р.	3	165000
12.05.2010	Морфин	15 000,00р.	3	45000
05.04.2010	Фенобарбитал	12 000,00р.	1	12000
11.05.2010	Фенобарбитал	40 000,00р.	2	80000

Запись: 5 из 13

2.7.4 Перехресні запити.

Перехресні запити використовують для розрахунків і представлення даних в структурі, що полегшує їх аналіз. Перехресний запит підраховує суму, середнє, число значень або виконує інші статистичні розрахунки, після чого результати групуються у вигляді таблиці по двох наборах даних, один з яких визначає заголовки стовпців, а інший заголовки рядків.

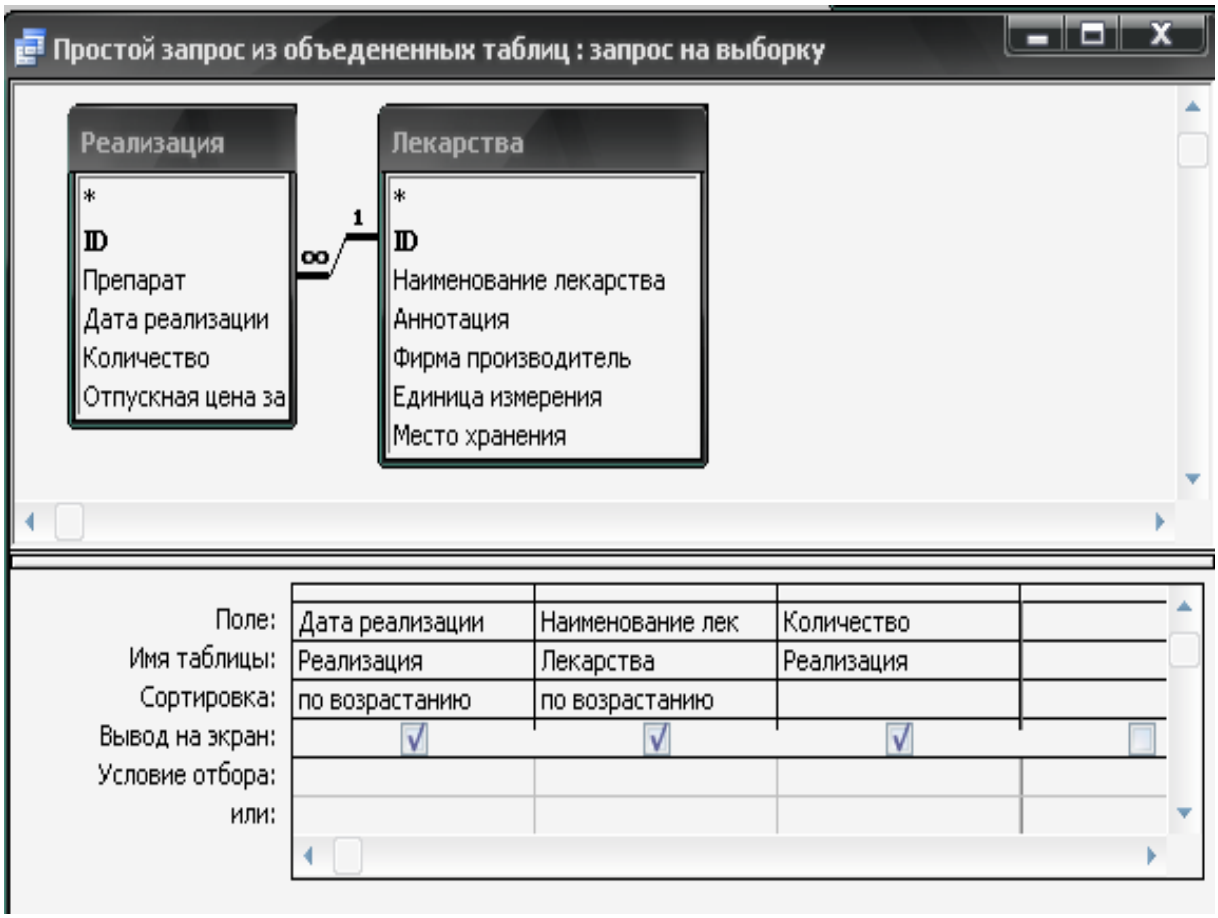
перехресний запит, який відбиває по датах кількість проданого в день лікарського засобу.

Лекарства Реализация Перекрестный Запрос : перекрестный запрос

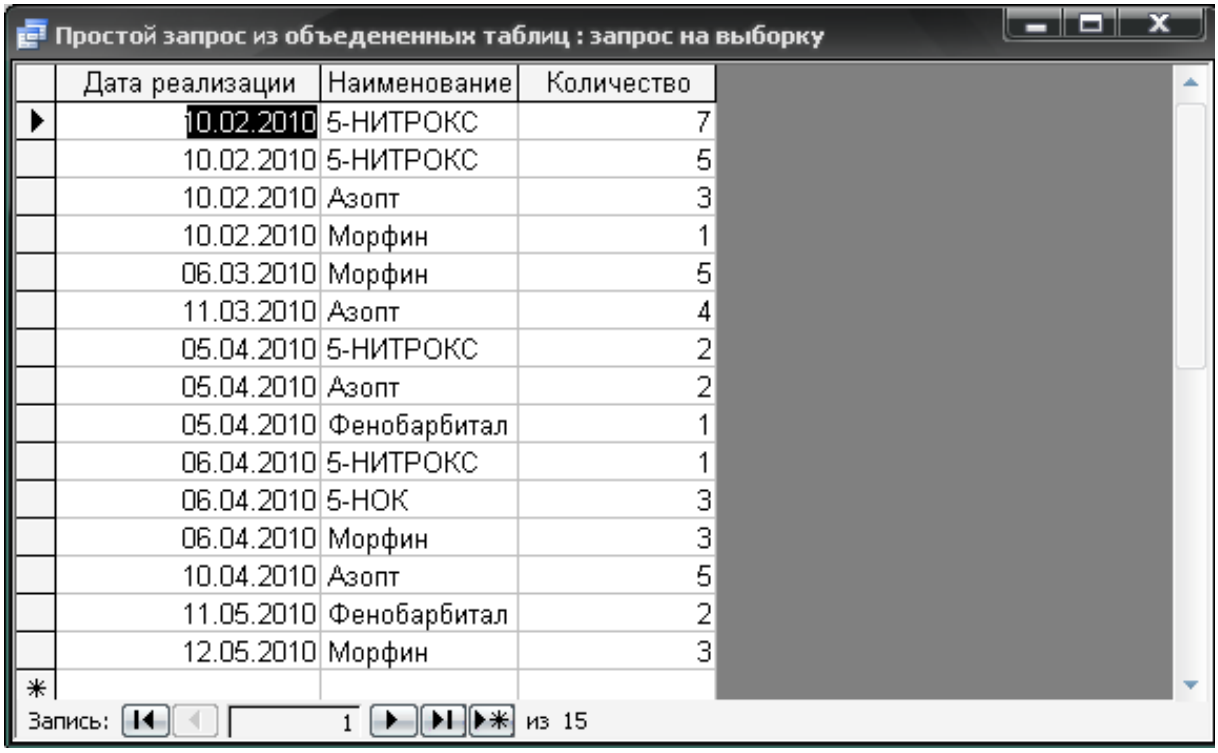
Наименование	10_02_2010	06_03_2010	11_03_2010	05_04_2010	06_04_2010	10_04_2010	11_05_2010	12_05_2010
5-НИТРОКС	12			2	1			
5-НОК					3			
Азопт	3		4	2		5		
Морфин	1	5			3			3
Фенобарбитал				1			2	

Запись: 1 из 5

Для роботи з база даних був створений запит з об'єднаних таблиць ("Ліки" і "Реалізація")



Результат роботи запиту з об'єднаних таблиць:


Простой запрос из объеденных таблиц : запрос на выборку

Дата реализации	Наименование	Количество
10.02.2010	5-НИТРОКС	7
10.02.2010	5-НИТРОКС	5
10.02.2010	Азопт	3
10.02.2010	Морфин	1
06.03.2010	Морфин	5
11.03.2010	Азопт	4
05.04.2010	5-НИТРОКС	2
05.04.2010	Азопт	2
05.04.2010	Фенобарбитал	1
06.04.2010	5-НИТРОКС	1
06.04.2010	5-НОК	3
06.04.2010	Морфин	3
10.04.2010	Азопт	5
11.05.2010	Фенобарбитал	2
12.05.2010	Морфин	3

Запись: 1 из 15

2.8 Звіти

Звіт - ефективний засіб для перегляду і роздруку підсумкової інформації.

Звіт є форматованим поданням даних, які виводиться на екран, в файл чи для друку. Звіти дають можливість витягти з бази необхідні відомості та представити їх у зручному вигляді для сприйняття та надають широкі можливості для аналізу даних і узагальнення. При друку запитів та таблиць інформація видається у вигляді, в якому вони були збережені. Звіти легко читаються, тому часто представляють дані у традиційному вигляді звітів. Детальний звіт містить всю інформацію із запиту чи таблиці, але містить заголовки та поділений на сторінки із зазначенням колонтитулів.

Звіти підтримують найрізноманітніші способи оформлення і дозволяють групувати дані, розбиваючи їх на логічно цілісні блоки.

В звіті відображаються дані із запиту або таблиці, додаючи до них текстові елементи, які спрощують його сприйняття. Таких елементів належать:

- Заголовок. Цей розділ друкується тільки у верхній частині першої сторінки звіту. Використовується для виведення даних, таких як текст

заголовка звіту, дата або констатуюча частина тексту документу, які слід надрукувати один раз на початку звіту.

- Верхній колонтитул. Використовується для виведення даних, таких як заголовки стовпців, дати або номери сторінок, що друкуються згори на кожній сторінці звіту. Для додавання або видалення верхнього колонтитулу необхідно вибрати в меню Вид команду Колонтитули. Microsoft Access додає верхній і нижній колонтитули одночасно. Щоб приховати один з колонтитулів, треба задати для його властивості Висота значення 0.

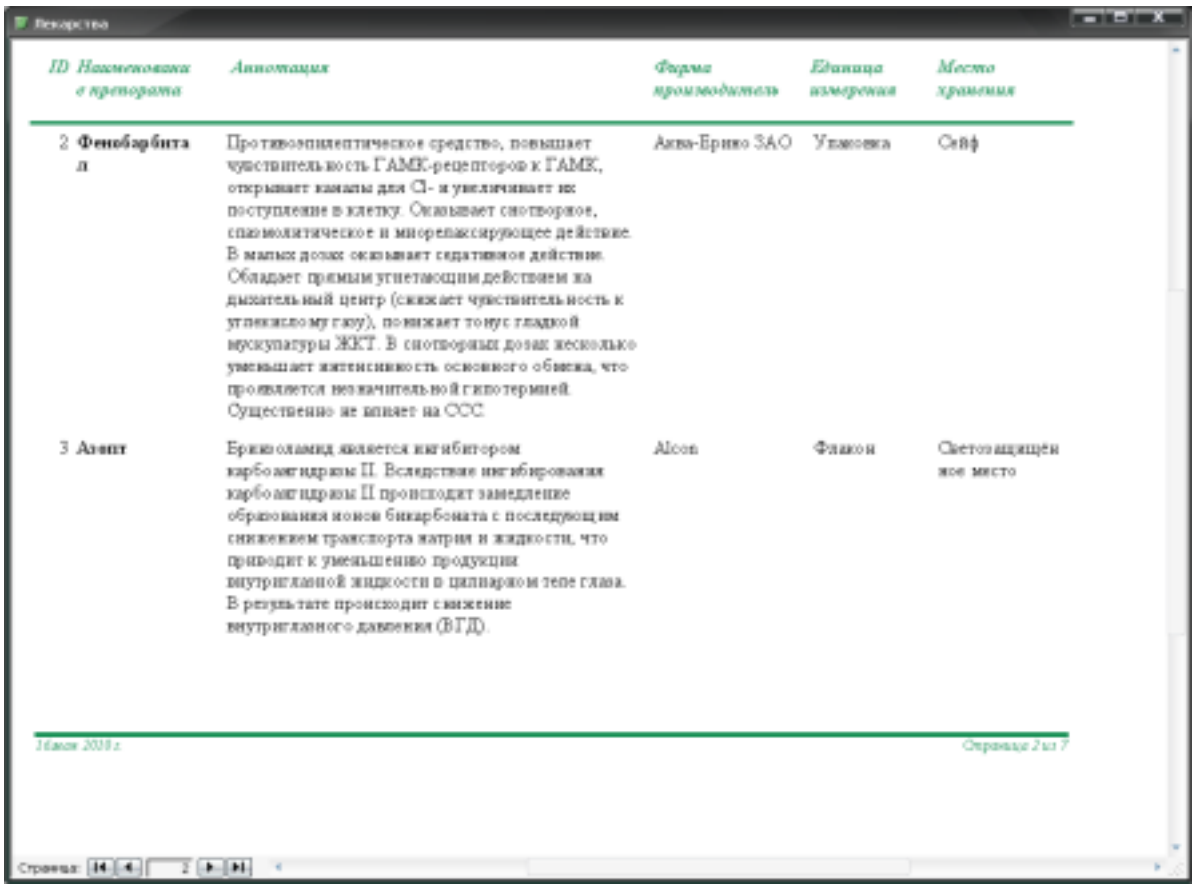
- Основной текст отчета − область данных, расположенная между верхним и нижним колонтитулами страницы. В этом разделе появляются данные, распечатываемые для каждой из тех записей в таблице или запросе, на которых основан отчет. Для размещения в области данных элементов управления используют список полей и панель элементов.

- Нижний колонтитул. Этот раздел появляется в нижней части каждой страницы. Используется для вывода данных, таких как итоговые значения, даты или номера страницы, печатающихся снизу на каждой странице отчета.

- Примітка. Використовується для виведення даних, таких як текст укладення, загальні підсумкові значення або підпис, які слід надрукувати один раз у кінці звіту. Попри те, що в режимі Конструктора розділ "Примітка" звіту знаходиться внизу звіту, він друкується над нижнім колонтитулом сторінки на останній сторінці звіту.

У рамках цієї роботи були створені три звіти:

1."Ліки"(Фрагмент)


Лекарства

ID	Наименование препарата	Аннотация	Форма производитель	Единица измерения	Место хранения
2	**Фенобарбитал**	Противоэпилептическое средство, повышает чувствительность ГАМК-рецепторов к ГАМК, открывает каналы для Cl- и увеличивает их поступление в клетку. Оказывает снотворное, спазмолитическое и миорелаксирующее действие. В малых дозах оказывает седативное действие. Обладает прямым угнетающим действием на дыхательный центр (снижает чувствительность к углекислому газу), понижает тонус гладкой мускулатуры ЖКТ. В снотворных дозах несколько уменьшает интенсивность основного обмена, что проявляется незначительной гипотермией. Существенно не влияет на ССС.	Аква-Бриио ЗАО	Упаковка	Сейф
3	**Азопт**	Бринзоламид является ингибитором карбоангидразы II. Вследствие ингибирования карбоангидразы II происходит замедление образования ионов бикарбоната с последующим снижением транспорта натрия и жидкости, что приводит к уменьшению продукции внутриглазной жидкости в цилиарном теле глаза. В результате происходит снижение внутриглазного давления (ВГД).	Alcon	Флакон	Светозащищённое место

Страница: 2

2. "Надходження лікарських засобів"

Поступления лекарств

Наименование лекарства	Дата поступления - по месяцам	Дата поступления	Количество	Цена закупки	Стоимость партии
Мор фин					
	Февраль 2010				
		07.02.2010	8	10 000,00р.	80 000,00р.
	Апрель 2010				
		06.04.2010	10	10 000,00р.	100 000,00р.
Фенобарбитал					
	Апрель 2010				
		01.04.2010	10	10 000,00р.	100 000,00р.
Азопт					
	Февраль 2010				
		02.02.2010	7	34 850,00р.	243 950,00р.
	Апрель 2010				

3. "Одиниці виміру"

Единицы измерения

ID	Наименование
1	Упаковка
2	Ампула
3	Флакон
4	Банка
5	Шт.
6	Пакет

Звіти доступні з форми "Звіти", кнопки "Звіти", що викликається по натисненню, з головної форми "Аптека". Звіти "Ліків" і "Одиниці виміру" створені по відповідних таблицях і без додаткових змін і виводять дані, що містяться в них. У звіті "Вступ лікарських засобів" створений додаткове обчислюване поле "Вартість партії", що розраховується як [Ціна закупівлі]*[Кількість].

При створенні звітів змінені розмір шрифту, колір полів і меж, доданий логотип аптеки.

2.9 Розробка програмних модулів та форм

2.9.1 Розробка програмнх модлів

Даний проект буде розроблятись засобами середовища програмування MS Visual Studio.

Microsoft Visual Studio - лінійка продуктів фірми Майкрософт, що включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів.

VisualStudio включає один або декілька з наступних компонентів:

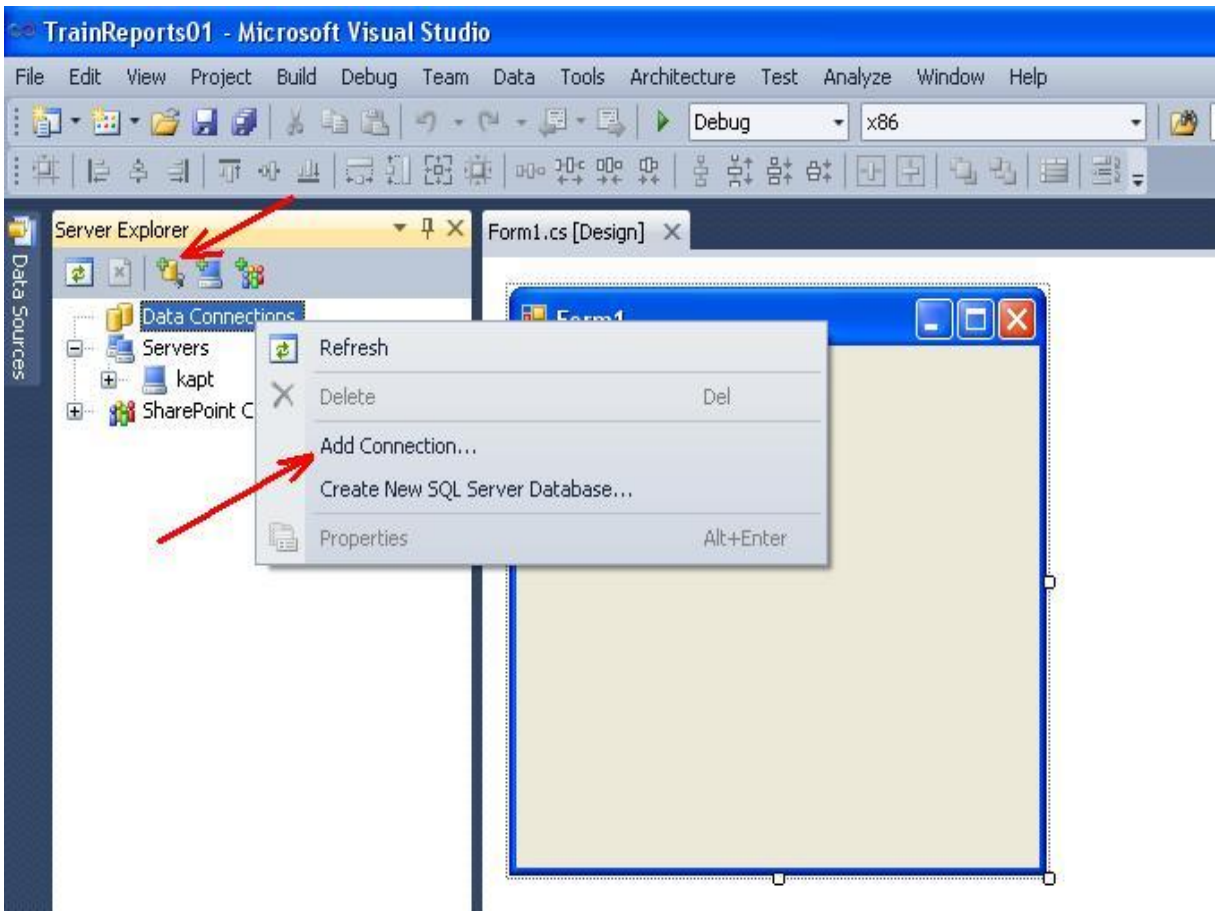
- Visual Basic.NET, а до його появи - VisualBasic;
- Visual C++;
- Visual C#;
- Visual J#.

Visual C# надає розробникам сучасну компоненто-орієнтовану мову, яка дозволяє швидко конструювати чудові рішення, керовані даними. Можливості C# дозволяють створювати рішення для широкого кола клієнтів, включаючи веб-програми, програми на основі Microsoft Windows Forms, додатки для «тонких» та інтелектуальних клієнтів. На відміну від Object Pascal, у C# підпрограми реалізовані тільки у вигляді функцій, проте спеціальний тип функції void не повертає ніякого значення. Саме тому деякі функції таким чином і були реалізовані.

На відміну від C++, C# більш структурована і типізована мова програмування. В неї немає глобальних змінних, тому що кожна змінна належить певному класу. Також в неї є безліч конструкторів по роботі з базами даних, які пришвидшать виконання і здешевлять проект. В результаті підключення до бази даних створюється конфігураційний файл, в якому можна змінювати шлях до бази даних без перекомпіляції програми. Підключення до бази даних здійснюється створенням копії даної бази, що дає можливість працювати з віддаленим сервером. В проекті буде використовуватись локальне підключення до бази даних, бо для створення хоча б мереженого підключення немає потрібних прав доступу і можливостей, а також підключення до бази даних через OLE за допомогою ADO.NET, яка підтримує мережене з'єднання, тому при необхідності можна звертатись до бази даних через мережу, прописавши попередньо шлях в конфігураційному файлі, який знаходиться в одній папці з виконуваним файлом. Також локальне підключення пришвидшить розробку і тестування проекту.

Базу даних додаток розроблено за допомогою ПЗ MS SQL Server 2008. Через технологію SQL підключаємо її вбудованими засобами MS Visual Studio.

Порядок підключення бази даних:
- на панелі інструментів обираємо Data->Add New Data Source;

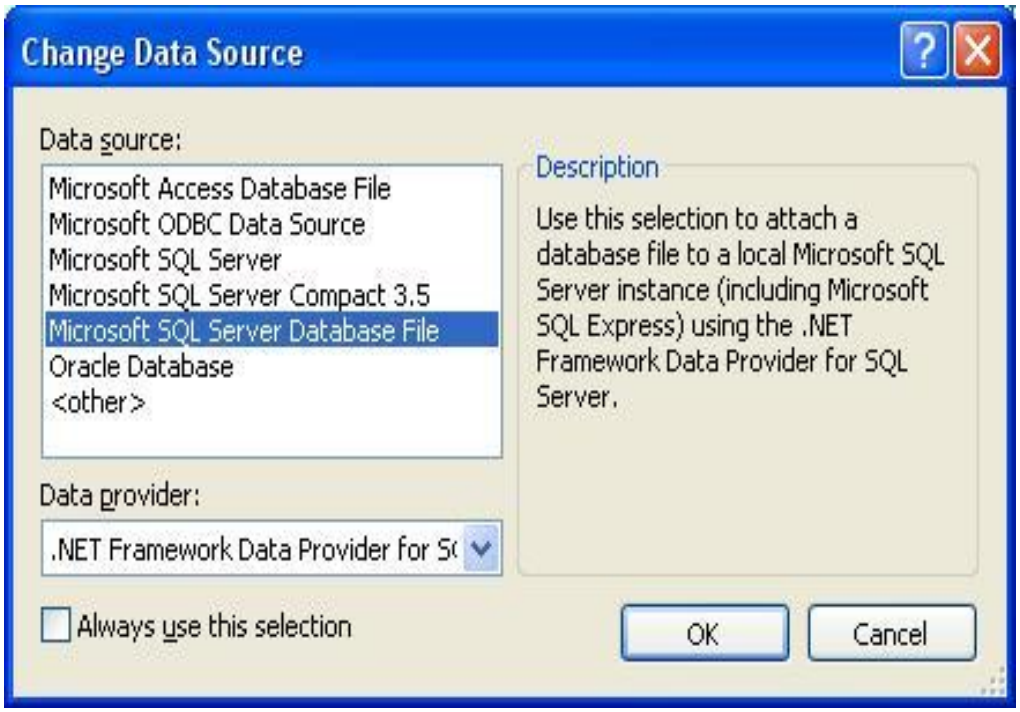


Створення нового підключення

- вибираємо тип підключення до БД та вказуємо шлях ;

Тип підключення

- створюємо нове підключення шляхом натискання кнопки «New connection» та обираємо MS SQL Server;



В результаті підключення відкривається вікно, яке призначене для редагування та перегляду параметрів БД в середовищі C#:

Отже, обрано засіб створення ПЗ VisualStudio та мову програмування C#(як одну з мов програмування, що розвивається найшвидше). Використання.NET Framework дає змогу використовувати дуже велику кількість вбудованих класів та виконувати відкомпільований код на будь - якій апаратній платформі (яка підтримується .NET).

Також серед переваг MS Visual Studio варто відзначити велику різноманітність засобів для зв'язку із базами даних (API). Для цього можна використовувати:

- OLEDB технологію - дозволяє звертатись до даних за допомогою уніфікованого доступу;
- ODBC (англ. Open DataBase Connectivity) - це відкритий інтерфейс доступу до баз даних, розроблений консорціумом X/Open. Даний інтерфейс надає можливості уніфікованого доступу, не турбуючись про технологію, яка застосовується для зберігання даних. Це надає можливості одному додатку зв'язуватись із джерелами, побудованими на різних СКБД.

Базу даних було розроблено за допомогою MS SQL Server. Використання двох технологій одного і того ж виробника полегшує завдання і дозволяє повною мірою реалізовувати можливості обох середовищ.

Програмування на Visual C# складається з двох процесів:

- розробка засобів діалогу користувача і програми, що мають візуальний вигляд за допомогою, як правило, вже існуючих компонентів;
- написання кодів, що додають функціональність діалоговим вікнам.

Для покращення роботи програми були створені власні процедури, які значно спрощують та полегшують обчислювальний процес та інші моменти у роботі програми.

Для безперебійної роботи програми, в результаті виникнення непередбачених ситуацій використовується спеціально створена процедура, яка викликається при завантаженні програми. Вона попереджає критичне завершення програми навіть при допущенні помилок у роботі:

```
Try // програму вже запущено
{DevExpress. UserSkins. BonusSkins. Register();
DevExpress. UserSkins. OfficeSkins. Register();
Application. EnableVisualStyles();
Application. SetCompatibleTextRenderingDefault (false
Application. Run (new MainF());
} // якщо виникає помилка: виводимо повідомлення
catch (Exception e)
{
MessageBox. Show («Message:» + e. Message + «Source:» + e. Source
+ «Data:» + e. Data + «InnerException:» + e. InnerException);}
```

Таким чином попереджаються «виліти» програми, і ПЗ буде працювати коректно незалежно від поведінки користувача.

2.9.2 Створення форм

Форми є типом об'єктів, який зазвичай використовується для відображення даних. Джерелом даних для форм є таблиці або запити, тому дані у формах є "пов'язаними" з відповідними об'єктами. Форми служать посередником між користувачами і базою даних, оскільки дозволяють відображати дані в потрібному виді.

У моїй СУБД представлені наступні форми.

Форма для введення даних - забезпечує: введення і редагування даних.

У формі "Препарати" можна змінювати назву препарату, ціну, зображення. Поміняти групу препарату і хвороби, які лікуються цим препаратом.

Аналогічна форма "Співробітники".

Кнопкова форма - використовується для відкриття інших форм або звітів.

Головна кнопкова форма "Аптека "Мед+" містить дві кнопки: "Препарати в наявності" і "Наші співробітники". Натисненням на ці кнопки автоматично переходимо на дві попередні форми.

Звідна таблиця - таблиця, узагальнювальна і аналізуюча дані з однієї або декількох таблиць.

2.10 Інструкція користувача

При запуску Бази Даних користувачеві пропонується зробити вибір між режимами входу в систему - режим клієнта, режим співробітника і режим постачальника. Для цього використовується форма, представлена нижче.

В першому режимі, відповідно, доступних функцій менше. Пропонується вибір між двома видами перегляду : Перегляд форми про препарати "Препарати в наявності", Ознайомлення із співробітниками аптеки "Наші співробітники". При цьому треба помітити, що режим переглядів таблиць саме перегляд, оскільки клієнт не має прав змінювати які-небудь дані в таблицях.. Вибір однієї з функцій в режимі клієнта.

У режимі співробітника доступними є вже чотири функції: Перегляд препаратів в наявності, перегляд звіту про співробітників, перегляд постачальників аптеки, а також перегляд ефективності роботи у вигляді звідної таблиці. Природно, що форми для менеджерів представлені в режимі зміни/додавання даних.

У режимі постачальника користувач може подивитися тільки інформацію про співробітників, з якими надалі він і зв'язується.

Щоб проглянути вибрану таблицю, форму або звіт, необхідно у формі, що з'явилася на екрані, кликнути на кнопку з потрібним поясненням. Наприклад, щоб в режимі звичайного користувача (клієнта) проглянути дані про співробітників, вимагається у формі входу в систему натиснути кнопку входу

для звичайного користувача, а потім із запропонованих варіантів у формі клієнта вибрати шуканий - "Наші співробітники".

ВИСНОВОК

Прийнято вважати, що використання концепції БД дозволяє :

- підвищити цілісність, надійність і збереження даних;
- закріпити витрати інтелектуальної праці;
- забезпечити легкість та простоту використання даних;
- задовольнити незалежність прикладних програм від даних;
- забезпечити достовірність даних;
- автоматизувати зміну даних;
- забезпечити потрібну швидкість доступу до даних;
- стандартизувати дані в межах однієї предметної області;
- забезпечити захист від спотворення і знищення даних;
- зменшити дублювання інформації за рахунок структурування даних;
- забезпечити оброблення незапланованих запитів до збереженої інформації;
- створити передумови для створення розподіленої обробки даних.

БД зручно застосовувати в різних організаціях, й аптека не є винятком. Аптека, як торгова установа характеризується високим ступенем звітності. Беручи до уваги значні темпи та обсяги товарообігу, специфічні вимоги до його реалізації та зберігання, то дуже важливо налагодити якісний та швидкий збір, аналіз і обробку інформації, яка належить до комерційної діяльності аптеки. Створення БД в аптеці дасть змогу зібрати всі дані в одному місці та забезпечити доступ до них всім співробітникам, а також надасть швидкий доступ до інформації та змогу вести чіткий структурований її облік, оптимізує роботу всієї аптеки та допоможе автоматизувати стандартні процеси.

Впровадження комп'ютерних систем в аптеках стало необхідним та в зв'язку із зростаючим потоком інформації, в якому всі співробітники зобов'язані орієнтуватися для того, щоб якісно виконувати свої обов'язки. Робити це стає все важче, тому що потрібно враховувати постійне розширення та оновлення асортименту лікарських засобів, не кажучи вже про наявність на вітчизняному фармринку величезної кількості фірм-посередників, ефективну взаємодію з якими неможливо здійснити без застосування комп'ютерних технологій.

У підсумку, на підставі завдання, поставлених в дипломному проекті, розроблена база даних «Аптека» готова до використання. Але за необхідності вона може бути доповнена і відредагована.

Список використаних джерел інформації

1. Атре, Ш., Структурний підхід до організації баз даних. / [Текст] : Атре, Ш. - М.: Фінанси і статистика, 1983. - 320 с.

2. Бойко, В.В., Проектування баз цих інформаційних систем. / [Текст] : Жваво, В.В., Савинков, В.М., - М.: Фінанси і статистика, 1989. - 351 с.

3. Гарсия-Молина, Гектор, Ульман, Джеффри, Д., Уидом, Дженнифер Системи баз даних. Повний курс. : Пер. з англ. - М. : Видавничий дім "Вільямс", 2003. - 188 с. : мул.

4. Дейт, К., Керівництво по реляційній СУБД DB2. / [Текст] : Дейт, К. - М.: Фінанси і статистика, 1988. - 320 с.

5. Джексон, Г., Проектування реляційних баз даних для використання з МІКРОЕОМ. / [Текст] : Джексон, Г., -М.: Світ, 1991. - 252 с.

6. Кирилов, В.В., Структуризованный мова запитів (SQL). / [Текст] : Кирилов, В.В., - Спб.: ИТМО, 1994. - 80 с.

7. Мартін, Джоуль., Планування розвитку автоматизованих систем. / [Текст] : Мартін, Джоуль., - М.: Фінанси і статистика, 1984. - 196 с.

8. Жарков В.А. Компьютерная графика, мультимедиа и игры на Visual C# [Текст] / В.А. Жарков - М.: ЮНИТИ, 2008. - 355 с.

9. Лабор В. Visual C# Создание приложений для Windows [Текст] /
В. Лабор - М.: Издательский дом «Вильямс», 2007. - 385 с.

10. Агуров П.В. C#. Сборник рецептов [Текст] / П.В. Агуров - М.: Издательский дом «Вильямс», 2007. - 429 с.

11. Форкун Ю.В. Бази даних: методичні вказівки до курсового проектування для студентів напряму підготовки «Програмна інженерія» [Текст] / Ю.В Форкун. - Хмельницький: ХНУ, 2012. - 35 с.

12. Библиотека MSDN - библиотека официальной технической документации для разработчиков под ОС Microsoft Windows. Архитектура ADO.NET [Електронна публікація] URL: http://msdn.microsoft.com/ru-ru/library/27y4ybxw (v=vs.100).aspx (дата звернення 12.12.2012)

13. Форум программистов и сисадминов CyberForum.ru. FAQ по ADO.NET [Електронна публікація] URL: http://www.cyberforum.ru/ado-net/thread342184.html (дата звернення 12.12.2012)

14. Форкун Ю.В., Кисіль Т.М. Проектний практикум: методичні вказівки до курсового проектування для студентів напряму підготовки «Програмна інженерія» [Текст] / Ю.В. Форкун, Т.М. Кисіль. - Хмельницький: ХНУ, 2011. - 48 с.

15. Сандерсон С. ASP.NET MVC для профессионалов [Текст] / С. Сандерсон - М.: Издательский дом «Вильямс», 2009. - 557 с.